
トランザクショナル分散オブジェクト基盤
TPBroker Object Transaction Monitor
ユーザーズガイド

解説・手引・操作書

3000-3-908

マニュアルの購入方法

このマニュアル，および関連するマニュアルをご購入の際は，
巻末の用紙をご利用ください。

HITACHI

対象製品

・適用 OS : AIX 5L

P-1M64-E151 TPBroker Object Transaction Monitor 05-00

P-1M64-E251 TPBroker Object Transaction Monitor - Client 05-00

P-1M64-E551 TPBroker Object Transaction Monitor - Connector for Object Request Broker 02-00

・適用 OS : HP-UX 11i , 11.0

P-1B64-E151 TPBroker Object Transaction Monitor 05-00

P-1B64-E251 TPBroker Object Transaction Monitor - Client 05-00

P-1B64-E551 TPBroker Object Transaction Monitor - Connector for Object Request Broker 02-00

・適用 OS : HP-UX 11i V2.0 (IPF)

P-1J64-E151 TPBroker Object Transaction Monitor 05-00

P-1J64-E251 TPBroker Object Transaction Monitor - Client 05-00

P-1J64-E551 TPBroker Object Transaction Monitor - Connector for Object Request Broker 02-00

・適用 OS : Red Hat Enterprise Linux AS 3.0

P-9S64-E151 TPBroker Object Transaction Monitor 05-00

P-9S64-E251 TPBroker Object Transaction Monitor - Client 05-00

P-9S64-E551 TPBroker Object Transaction Monitor - Connector for Object Request Broker 02-00

・適用 OS : Red Hat Enterprise Linux AS 3.0 (IPF)

P-9V64-E151 TPBroker Object Transaction Monitor 05-00

P-9V64-E251 TPBroker Object Transaction Monitor - Client 05-00

P-9V64-E551 TPBroker Object Transaction Monitor - Connector for Object Request Broker 02-00

・適用 OS : Solaris

P-9D64-E151 TPBroker Object Transaction Monitor 05-00

P-9D64-E251 TPBroker Object Transaction Monitor - Client 05-00

P-9D64-E551 TPBroker Object Transaction Monitor - Connector for Object Request Broker 02-00

・適用 OS : Windows Server 2003 , Windows XP , Windows 2000

P-2464-E154 TPBroker Object Transaction Monitor 05-00

P-2464-E254 TPBroker Object Transaction Monitor - Client 05-00

P-2464-E554 TPBroker Object Transaction Monitor - Connector for Object Request Broker 02-00

・適用 OS : Windows Server 2003 (IPF)

P-2864-E154 TPBroker Object Transaction Monitor 05-00

P-2864-E254 TPBroker Object Transaction Monitor - Client 05-00

P-2864-E554 TPBroker Object Transaction Monitor - Connector for Object Request Broker 02-00

印のプログラムプロダクトについては、発行時期をご確認ください。

これらのプログラムプロダクトのほかにも、このマニュアルをご利用になれる場合があります。詳細は「ソフトウェア添付資料」または「Readme ファイル」でご確認ください。

輸出時の注意

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

なお、ご不明な場合は、弊社担当営業にお問い合わせください。

商標類

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

Borland のブランド名および製品名はすべて、米国 Borland Software Corporation の米国およびその他の国における商標または登録商標です。

CORBA は、Object Management Group が提唱する分散処理環境アーキテクチャの名称です。

HP-UX は、米国 Hewlett-Packard Company のオペレーティングシステムの名称です。

Itanium は、アメリカ合衆国および他の国におけるインテル コーポレーションまたはその子会社の登録商標です。

Java 及びすべての Java 関連の商標及びロゴは、米国及びその他の国における米国 Sun Microsystems, Inc. の商標または登録商標です。

Linux は、Linus Torvalds の米国およびその他の国における登録商標あるいは商標です。

Microsoft は、米国およびその他の国における米国 Microsoft Corp. の登録商標です。

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

Solaris は、米国 Sun Microsystems, Inc. の米国およびその他の国における商標または登録商標です。

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

Windows は、米国およびその他の国における米国 Microsoft Corp. の登録商標です。

Windows Server は、米国およびその他の国における米国 Microsoft Corp. の商標です。

プログラムプロダクト「P-9D64-E151, P-9D64-E251, P-9D64-E551」には、米国 Sun Microsystems, Inc. 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-E151, P-9D64-E251, P-9D64-E551」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。

発行

2004年9月(第1版) 3000-3-908

著作権

All Rights Reserved. Copyright (C) 2004, Hitachi, Ltd.

はじめに

このマニュアルは、TPBroker Object Transaction Monitor、TPBroker Object Transaction Monitor - Client および TPBroker Object Transaction Monitor - Connector for Object Request Broker の機能、環境設定、および運用について説明したものです。

なお、このマニュアルでは、TPBroker Object Transaction Monitor を OTM と表記し、TPBroker Object Transaction Monitor - Client を OTM - Client と表記しています。また、TPBroker Object Transaction Monitor - Connector for Object Request Broker を OTM - Connector for ORB と表記しています。これらすべてを総称するときは、TPBroker OTM と表記します。

OTM - Connector for ORB は、TPBroker OTM 05-00 ではサポートされていません。サポート時期をご確認の上、ご使用ください。

対象読者

TPBroker OTM を使用するシステム管理者およびアプリケーションプログラマを対象としています。

マニュアルの構成

このマニュアルは、次に示す章と付録から構成されています。

第 1 章 概要

TPBroker OTM の特長、およびシステム構成について説明しています。

第 2 章 機能

TPBroker OTM の機能について説明しています。

第 3 章 OTM - Client の運用

OTM - Client の環境設定、開始と終了、および障害対策について説明しています。

第 4 章 OTM の運用

OTM の環境設定、開始と終了、OTM - Connector for ORB の運用方法、TPBroker との連携、および障害対策について説明しています。

第 5 章 コマンドリファレンス

アプリケーションプログラムの作成時に使用するコマンドを除いた、運用コマンドおよびアプリケーションプログラム開始時のオプションについて説明しています。

第 6 章 メッセージ

OTM が出力するメッセージについて説明しています。

付録 A アポートコード一覧

OTM のメッセージが出力するアポートコードについて説明しています。

はじめに

付録 B tscgetref コマンドの出力例

tscgetref コマンドで接続先情報を出力する例を示しています。

付録 C tscstsed コマンドの出力例

tscstsed コマンドで稼働統計情報を出力する例を示しています。

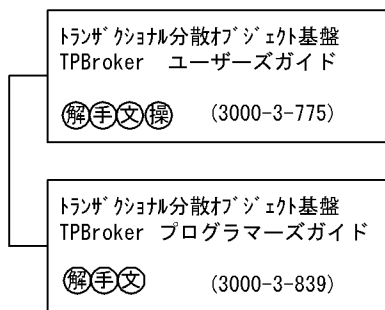
付録 D 例外変換テーブルファイルの作成方法

OTM のシステム例外を CORBA 例外に変換するときを使用される、例外変換テーブルファイルの作成方法について説明しています。

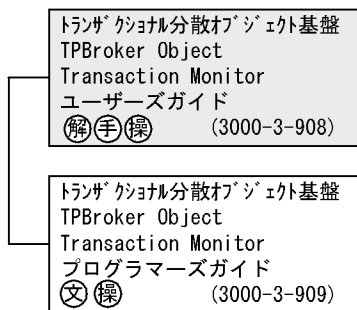
関連マニュアル

このマニュアルの関連マニュアルを次に示します。必要に応じてお読みください。

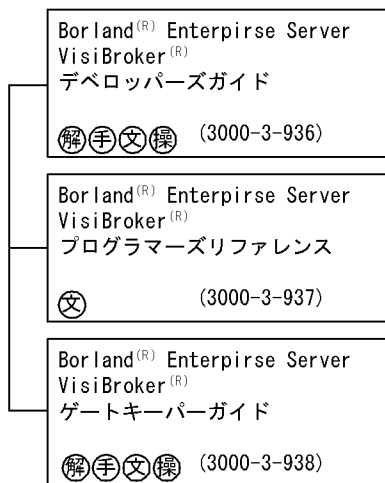
・ TPBroker Version 5



・ TPBroker OTM Version 5



・ VisiBroker Version 5

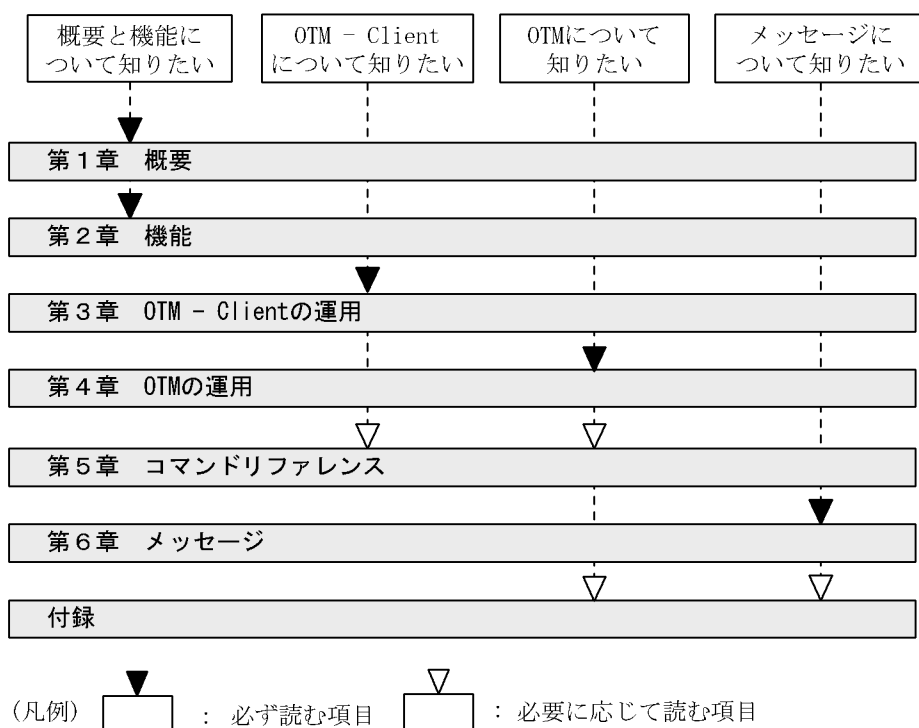


<記号>

- 解 : 解説書
- 手 : 手引書
- 文 : 文法書
- 操 : 操作書

読書手順

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



このマニュアルでの表記

このマニュアルで使用する製品名称の略称を次に示します。

製品名称	略称
AIX 5L	AIX
Borland(R) Enterprise Server VisiBroker(R)	Borland Enterprise Server VisiBroker
HP-UX 11.0	HP-UX
HP-UX 11i	
HP-UX 11i V2.0 (IPF)	
Itanium(R) Processor Family	IPF
Java(TM)	Java
JavaServer(TM)	JavaServer
Red Hat Enterprise Linux AS 3.0	Linux
Red Hat Enterprise Linux AS 3.0 (IPF)	
TPBroker for C++	TPBroker
TPBroker for Java	

はじめに

製品名称	略称	
TPBroker Object Transaction Monitor	OTM	TPBroker OTM
TPBroker Object Transaction Monitor - Client	OTM - Client	
TPBroker Object Transaction Monitor - Connector for Object Request Broker	OTM - Connector for ORB	
Microsoft(R) Windows(R) 2000 Advanced Server Operating System	Windows 2000	
Microsoft(R) Windows(R) 2000 Datacenter Server Operating System		
Microsoft(R) Windows(R) 2000 Professional Operating System		
Microsoft(R) Windows(R) 2000 Server Operating System		
Microsoft(R) Windows Server(TM) 2003, Standard Edition	Windows Server 2003	
Microsoft(R) Windows Server(TM) 2003, Enterprise Edition		
64 ビットバージョン Microsoft(R) Windows Server(TM) 2003, Enterprise Edition (IPF)	Windows Server 2003 (IPF)	
Microsoft(R) Windows(R) XP Professional Operating System	Windows XP	

- Windows 2000 , Windows Server 2003 , Windows Server 2003 (IPF) および Windows XP で機能差がない場合 , Windows と表記しています。
- AIX , HP-UX , Linux および Solaris を合わせて UNIX と表記しています。

このマニュアルで使用する英略語の一覧を示します。

英略語	英字での表記
API	<u>A</u> pplication <u>P</u> rogramming <u>I</u> nterface
CORBA	<u>C</u> ommon <u>O</u> bject <u>R</u> equest <u>B</u> roker <u>A</u> rchitecture
CSV	<u>C</u> omma <u>S</u> eparated <u>V</u> alue
FTP	<u>F</u> ile <u>T</u> ransfer <u>P</u> rotocol
IDL	<u>I</u> nterface <u>D</u> efinition <u>L</u> anguage
IOR	<u>I</u> nteroperable <u>O</u> bject <u>R</u> eference
LAN	<u>L</u> ocal <u>A</u> rea <u>N</u> etwork
ORB	<u>O</u> bject <u>R</u> equest <u>B</u> roker
OS	<u>O</u> perating <u>S</u> ystem
PP	<u>P</u> rogram <u>P</u> roduct
TSC	<u>T</u> PBroker <u>S</u> chedule <u>C</u> ontrol
UDP	<u>U</u> ser <u>D</u> atagram <u>P</u> rotocol
URL	<u>U</u> niform <u>R</u> esource <u>L</u> ocator

図中で使用する記号

このマニュアルの図中で使用する記号を、次のように定義します。

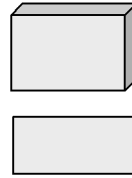
●クライアント



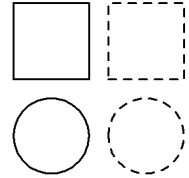
●サーバ



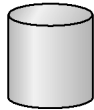
●プログラム



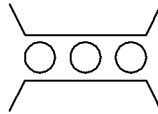
●クラス, または
インスタンス



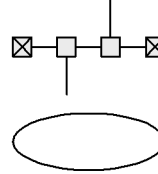
●ファイル



●キュー



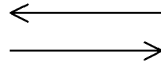
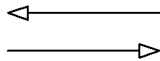
●LAN



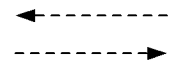
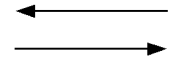
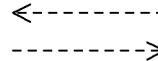
●作業項目の流れ



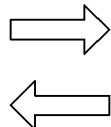
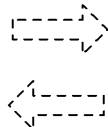
●クラスの継承の流れ ●制御の流れ



●接続の方向 ●スレッドの流れ



●内部参照 (アクセス) ●公開メソッドの
呼び出し



文法の記号

このマニュアルで使用する記号の意味を示します。

文法記述記号	意味
	この記号で区切られた項目を選択できることを示します。 (例) <code>-TSCRequestWay 0 1</code> これは、 <code>-TSCRequestWay</code> オプションに 0 か 1 のどちらかを指定できることを示します。
[]	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) <code>tscdmstart [-TSCForceStart]</code> これは、 <code>tscdmstart</code> と指定するか、または <code>tscdmstart -TSCForceStart</code> と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複数の項目のうちから一つを選択することを示しています。 (例) {-TSCID TSC 識別子 -TSCIDALL} これは、-TSCID TSC 識別子と -TSCIDALL のうち、どちらかを指定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) -TSCID TSC 識別子 [,TSC 識別子 ...] これは、-TSCID オプションに、TSC 識別子を繰り返し指定できることを示します。
~	この記号のあとにユーザ指定値の属性を示します。
《 》	ユーザが指定を省略したときの省略値を示します。
< >	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。
< 英字 >	アルファベット (A ~ Z, a ~ z) と _ (アンダースコア)
< 英数字 >	英字と数字 (0 ~ 9)
< 文字列 >	任意の文字の配列
< 符号なし整数 >	数字 (0 ~ 9)
< 2進数字 >	数字 (0 または 1)

注

すべて1バイトコードで指定してください。

注

数字は10進数表現で指定してください。

常用漢字以外の漢字の使用について

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

個所 (かしょ) 閉塞 (へいそく)

KB (キロバイト) などの単位表記について

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

目次

1	概要	1
1.1	TPBroker OTM の概要	2
1.1.1	キューによるリクエストのスケジューリング	2
1.1.2	サーバオブジェクトの常駐化	3
1.1.3	スケーラブルな負荷分散	4
1.1.4	コネクションの集約	4
1.1.5	サーバの動的なメンテナンス	5
1.1.6	素早い異常検知と障害の局所化	5
1.1.7	多様なプログラミング言語への対応	5
1.1.8	CORBA に準拠するクライアントとの接続	5
1.2	システム構成	7
1.2.1	クライアントの構成	9
1.2.2	サーバの構成	9
1.2.3	プロセス数の範囲	12
2	機能	13
2.1	クライアントアプリケーションの機能	14
2.1.1	リクエストの形態	14
2.1.2	TSC ノードとの接続	15
2.1.3	TSC ノードとのリバインド	16
2.1.4	セッション呼び出し機能	19
2.1.5	クライアントアプリケーションの状態の検出	21
2.2	接続先情報ファイルを使用する接続	22
2.2.1	マルチノードリトライ接続の概要	22
2.2.2	マルチノードリトライ接続と IOR 文字列を使用する接続の併用	22
2.2.3	マルチノードリトライ接続での接続方式の種類	23
2.2.4	マルチノードリトライ接続の接続対象	24
2.2.5	接続先情報ファイル	27
2.2.6	接続および再接続の実行	28
2.2.7	TSCClient オブジェクトでの接続の保持	29
2.3	TSC レギュレータの機能	31
2.3.1	コネクションのレギュレート	31
2.3.2	TSC レギュレータと TSC デーモンの対応	32

2.3.3	TSC レギュレータの動作モード	32
2.4	TSC デーモンの機能	34
2.4.1	リクエストの処理	34
2.4.2	リクエストの負荷分散	35
2.4.3	リクエストの流量制御	37
2.4.4	リクエストの閉塞制御	38
2.4.5	プロセスの配置	40
2.4.6	プロセスの管理	41
2.4.7	アプリケーションプログラムの管理	42
2.4.8	セッション呼び出しインターバル時間監視	44
2.5	TSC ドメインマネージャの機能	46
2.5.1	TSC ドメインマネージャの開始形態	46
2.6	複数のネットワークセグメントで構成する TSC ドメイン	48
2.6.1	TSC ノード情報の配布	48
2.6.2	複数のネットワークセグメントで TSC ドメインを構成する場合の注意事項	48
2.6.3	複数のネットワークセグメントでの TSC ドメインの構成例	49
2.7	サーバアプリケーションの機能	54
2.7.1	サーバアプリケーションの開始と終了	54
2.7.2	サーバアプリケーションからのリクエストの要求	55
2.7.3	スレッドと TSC ユーザオブジェクト	55
2.7.4	リクエストのスケジュール	59
2.7.5	リクエストの振り分け	62
2.7.6	サーバアプリケーションの時間監視	66
2.7.7	サーバアプリケーションの状態の検出	67
2.8	TSCORB コネクタの機能	69
2.8.1	TSCORB コネクタ経由の接続の仕組み	69
2.8.2	TSCORB コネクタと TSC デーモンの対応	71
2.8.3	ORB クライアントへの例外の通知	71
2.9	稼働統計情報の取得	72
2.9.1	稼働統計情報の概要	72
2.9.2	稼働統計情報のファイル運用	73
2.9.3	稼働統計出力の出力形式	74
2.9.4	稼働統計出力の出力情報	75
2.9.5	稼働統計情報取得時のオペレーション名の取り扱い	81

3	OTM - Client の運用	83
3.1	OTM - Client の環境設定	84
3.1.1	OTM - Client の環境設定の流れ	84
3.1.2	OTM - Client の環境変数	84
3.2	OTM - Client の開始と終了	86
3.2.1	OTM - Client の開始	86
3.2.2	OTM - Client の終了	86
3.3	OTM - Client の障害対策	88
3.3.1	OTM - Client で発生する障害の種類	88
3.3.2	OTM - Client で取得できる障害情報	88
4	OTM の運用	91
4.1	OTM の環境設定	92
4.1.1	OTM の環境設定の流れ	92
4.1.2	OTM の環境変数	92
4.1.3	環境変数による TSC ドメインの管理	93
4.1.4	ユーザ環境変数定義ファイル	94
4.2	OTM の開始と終了	97
4.2.1	OTM の開始	97
4.2.2	OTM の終了	98
4.3	OTM - Connector for ORB の運用	100
4.3.1	TSCORB コネクタの開始と終了	100
4.3.2	TSCORB コネクタの状態の検出	101
4.4	OTM と TPBroker との連携	102
4.5	OTM の障害対策	104
4.5.1	OTM で発生する障害の種類	104
4.5.2	OTM で取得できる障害情報	104
5	コマンドリファレンス	105
	運用コマンドの一覧	106
	tscdminfo (TSC ドメイン情報の表示と削除)	108
	tscdmstart (TSC ドメインマネージャの開始)	110
	tscdmstop (TSC ドメインマネージャの終了)	115
	tscgetior (IOR 文字列の取得)	117

tscgetpid (プロセス ID の取得)	121
tscgetref (接続先情報の取得と編集)	123
tsholdra (TSC ルートアクセプタの閉塞)	130
tscjver (jar ファイルのバージョン情報の出力)	133
tsclogcat (メッセージの編集と出力)	135
tscscon (コネクション情報の出力)	137
tscsra (TSC ルートアクセプタ情報の出力)	144
tscprcls (プロセス情報の出力)	151
tscrasget (トラブルシュート情報の取得と出力)	155
tscregltd (TSC レギュレータの開始)	157
tscridinfo (TSC ドメイン情報の出力)	160
tscrlsra (TSC ルートアクセプタの閉塞解除)	162
tscstart (TSC デーモンの開始)	164
tscstartgw (TSCORB コネクタの開始)	172
tscstartprc (サーバアプリケーションの開始)	179
tscstop (TSC デーモンの終了)	183
tscstopgw (TSCORB コネクタの終了)	185
tscstopprc (サーバアプリケーションの終了)	187
tscstsed (稼働統計情報の編集と出力)	189
tscststart (稼働統計情報の取得の開始)	201
tscststop (稼働統計情報の取得の終了)	203
tscver (実行形式ファイルおよびライブラリのバージョン情報の出力)	205
クライアントアプリケーションの開始	207
サーバアプリケーションの開始	213

6

メッセージ	221
6.1 メッセージの形式	222
6.1.1 メッセージの出力形式	222
6.1.2 メッセージの記述形式	222
6.2 メッセージ一覧	224
6.2.1 KFOT26000 ~ KFOT26999	224
6.2.2 KFOT27000 ~ KFOT27999	228
6.2.3 KFOT70000 ~ KFOT70999	231
6.2.4 KFOT71000 ~ KFOT71999	288
6.2.5 KFOT73000 ~ KFOT73999	299

6.2.6	KFOT74000 ~ KFOT74999	319
6.2.7	KFOT75000 ~ KFOT75999	326
6.2.8	KFOT76000 ~ KFOT76999	339
6.2.9	KFOT77000 ~ KFOT77999	342

付録 347

付録 A	アボートコード一覧	348
付録 B	tscgetref コマンドの出力例	350
付録 B.1	一つの TSC ノードの接続先情報を出力する例	352
付録 B.2	複数の TSC ドメインと TSC ノードの接続先情報をソートして出力する例	352
付録 B.3	すべての TSC ドメインから TSC ノードを選択して接続先情報を出力する例	353
付録 B.4	複数の接続先情報ファイルをマージして出力する例	353
付録 C	tscstsed コマンドの出力例	356
付録 C.1	TSC ノード全体でまとめた情報の出力例	357
付録 C.2	TSC ノード全体でまとめた情報の CSV 形式での出力例	358
付録 C.3	TSC ルートアクセプタ単位の情報の出力例	358
付録 C.4	TSC ルートアクセプタ単位の情報の CSV 形式での出力例	359
付録 C.5	サーバアプリケーションのオペレーション単位の出力例	359
付録 C.6	サーバアプリケーションのオペレーション単位の CSV 形式での出力例	359
付録 C.7	稼働統計情報ファイルに対象レコードがない場合の出力例	360
付録 C.8	稼働統計情報の出力を何度か繰り返した場合の出力例	361
付録 D	例外変換テーブルファイルの作成方法	363

索引 367

目次

図 1-1	TPBroker OTM を使用した分散処理システムの仕組み	2
図 1-2	システムの構成	8
図 2-1	セッション呼び出し機能の動作	20
図 2-2	クライアントアプリケーションの状態遷移	21
図 2-3	マルチノードリトライ接続の動作	22
図 2-4	TSCClient オブジェクトに保持される接続	29
図 2-5	リクエストが集中する仕組み	31
図 2-6	コネクションのレギュレートの仕組み	32
図 2-7	クライアントアプリケーションからサーバアプリケーションを呼び出す流れと負荷分散のタイミング	36
図 2-8	流量制御の概要	37
図 2-9	サーバアプリケーションの閉塞制御の概要	38
図 2-10	スケジュール用キューの閉塞制御の概要	39
図 2-11	サーバアプリケーション異常終了時のリクエスト保持の概要	40
図 2-12	TSC デーモンを中心とする各プロセスの配置	41
図 2-13	TSC デーモンによるサーバアプリケーションの登録と管理の流れ	43
図 2-14	TSC ノード間での通信の概要	46
図 2-15	複数のネットワークセグメントで TSC ドメインを構成する例 1	49
図 2-16	複数のネットワークセグメントで TSC ドメインを構成する例 2	50
図 2-17	複数のネットワークセグメントで TSC ドメインを構成する例 3	51
図 2-18	複数のネットワークセグメントで TSC ドメインを構成する例 4	52
図 2-19	複数のネットワークセグメントで TSC ドメインを構成する例 5	53
図 2-20	サーバアプリケーションからのリクエストの要求	55
図 2-21	スレッドと TSC ユーザオブジェクトの構成	56
図 2-22	マルチスレッド環境での処理の仕組み	58
図 2-23	スケジュール用キューを共有するスケジュール方法	59
図 2-24	スケジュール用キューを共有しないスケジュール方法	61
図 2-25	TSC アクセプタ名称を指定しない場合のリクエストの振り分け	62
図 2-26	TSC アクセプタ名称を指定する場合のリクエストの振り分け	64
図 2-27	セッション呼び出し機能を使用する場合のリクエストの振り分け	66
図 2-28	サーバアプリケーションの状態遷移	67
図 2-29	ORB クライアントから OTM に接続する仕組み	70
図 2-30	稼働統計情報の概要	73

図 2-31 稼働統計情報ファイルの世代切り替え	74
図 3-1 OTM - Client の環境設定の流れ	84
図 3-2 OTM - Client の開始の流れ	86
図 3-3 OTM - Client の終了の流れ	87
図 4-1 OTM の環境設定の流れ	92
図 4-2 複数の TSC ドメインと環境変数	94
図 4-3 OTM の開始の流れ	97
図 4-4 OTM の終了の流れ	99
図 4-5 TSCORB コネクタの開始の流れ	100
図 B-1 tscgetref コマンドを使用する環境	351

表目次

表 1-1	プロセス数の範囲	12
表 2-1	オートリバインド機能が動作する例外（TSC デモンへ直結するクライアントアプリケーションの場合）	16
表 2-2	オートリバインド機能が動作する例外（TSC レギュレータを経由するクライアントアプリケーションの場合）	17
表 2-3	-TSCTimeOutRebind オプション指定時にオートリバインド機能が動作する例外	17
表 2-4	各接続でのオートリバインド機能の動作の有無	17
表 2-5	マルチノードリトライ接続での接続方式の組み合わせ	23
表 2-6	ファイル検索方式の場合の接続対象	25
表 2-7	API 優先方式の場合の最初の接続対象	25
表 2-8	マルチノードリトライ接続での接続方式の指定方法	26
表 2-9	稼働統計情報ファイルの格納ディレクトリ	73
表 2-10	稼働統計出力ファイルの出力内容の指定	74
表 2-11	各出力形式の稼働統計出力ファイル名の一覧	74
表 2-12	稼働統計出力ファイルへの出力情報	75
表 2-13	TSC ノード全体でまとめた稼働統計出力情報	78
表 2-14	TSC ルートアクセプタ単位の稼働統計出力情報	80
表 2-15	サーバアプリケーションのオペレーション単位の稼働統計出力情報	80
表 3-1	ライブラリ格納ディレクトリを指定する環境変数（OTM - Client）	85
表 3-2	OTM - Client の障害	88
表 4-1	ライブラリ格納ディレクトリを指定する環境変数（OTM）	93
表 4-2	プロセス監視定義に記述する項目	102
表 4-3	OTM の障害	104
表 5-1	運用コマンドの一覧	106
表 5-2	tscdmstart コマンドのオプション指定値と TSC ノード情報の送信の関係	113
表 5-3	tscstsed コマンドのコマンドオプションの組み合わせ	191
表 6-1	標準出力に出力される付加情報とメッセージ	222
表 6-2	メッセージ ID 記号の意味	223
表 A-1	アボートコードの一覧	348
表 B-1	tscgetref コマンドの処理	350
表 C-1	tscstsed コマンドのオプション指定	356
表 D-1	各例外に対応する例外コード	363

1

概要

この章では、TPBroker OTM の特長、およびシステム構成について説明します。

1.1 TPBroker OTM の概要

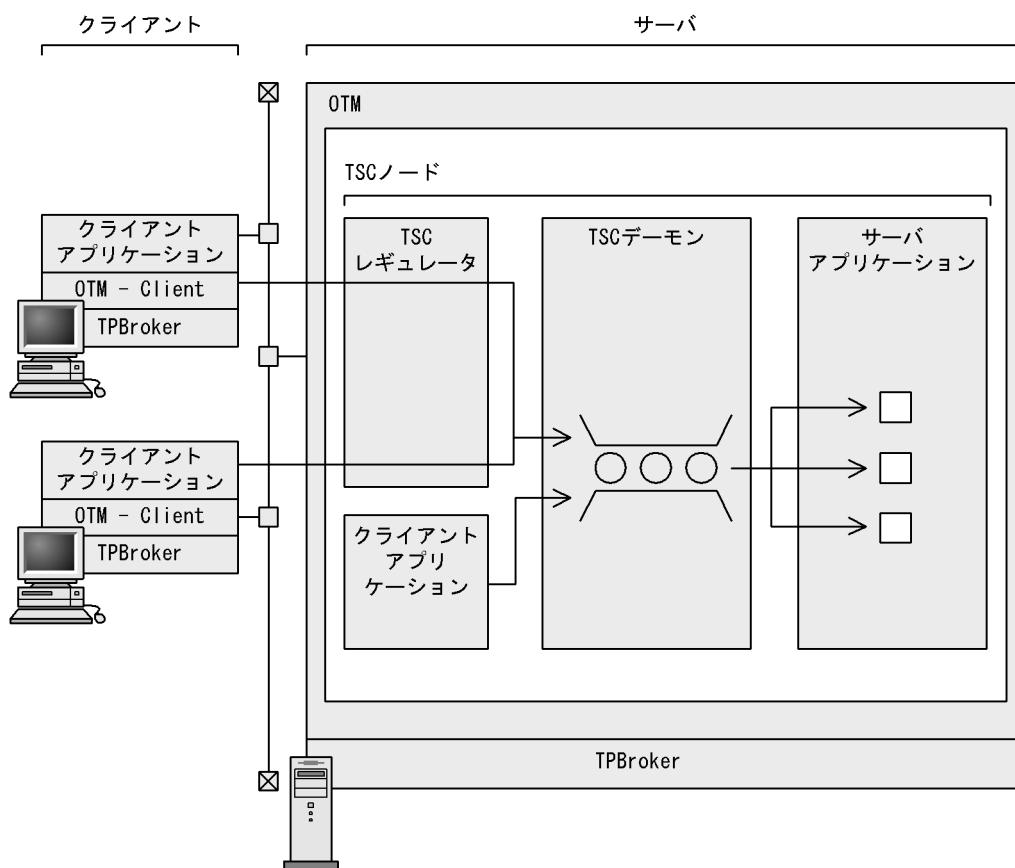
1.2 システム構成


1.1 TPBroker OTM の概要


TPBroker OTM は、TPBroker を使用した分散処理システムの構築を支援するプログラムです。クライアントアプリケーションからのリクエストをよりきめ細かく制御するとともに、オンラインプログラムとしての時間監視機能や運用機能を提供します。

TPBroker OTM を使用した分散処理システムの仕組みを次の図に示します。

図 1-1 TPBroker OTM を使用した分散処理システムの仕組み



(凡例)  : スケジュール用キュー

 : サーバオブジェクト

1.1.1 キューによるリクエストのスケジューリング

TPBroker OTM は、サーバアプリケーションに対するリクエストをサーバ側でスケ

ジュール用キューに滞留します。スケジュール用キューは、リクエストを処理するオブジェクトを次の単位で多重化します。

- サーバアプリケーションプロセス数
- スレッド数

そのため、複数のクライアントアプリケーションからのリクエストを処理する場合でも、サーバアプリケーションに関連するリソース（データベースシステムへのコネクション数など）の最大数は一定となります。また、CPU 資源の競合および排他競合の多発を回避できます。

さらに、次に示すようにスケジュール用キューを制御することで、効率の良い処理を実現します。

プライオリティによる制御

リクエストにプライオリティ（優先順位）を設定できます。この設定によって、プライオリティの高いリクエストから優先的に処理されます。

流量の制御

サーバアプリケーションで一度に実行される処理数の最大値を制限することで、サーバアプリケーションの負荷を一定に抑え、安定した高いスループットを実現できます。

閉塞の制御

サーバアプリケーションやスケジュール用キューを閉塞させて、リクエストの受け付けを停止したり、リクエストを滞留させたりすることができます。これによって、システムを停止することなくサーバアプリケーションの入れ替えなどができます。

タイムアウトの監視

指定した応答タイムアウト時間を過ぎているリクエストをスケジュール用キューから削除します。これによって、不要な処理を省けます。

1.1.2 サーバオブジェクトの常駐化

サーバアプリケーションプロセスの内部には、多重化されたスレッドが生成されます。ユーザはスレッドを多重化する数を指定できます。スレッド上には、オブジェクトのインスタンスが常駐します。一つのスレッド上に、複数の種類のオブジェクトのインスタンスを常駐させることができます。

オブジェクトを常駐化することで、次のようにスレッドを制御できます。

スレッド単位でのリソースの割り当て

サーバアプリケーションプロセス内で同時に実行されるインスタンスの数は、スレッドの数と同じです。そのため、必要なリソースは、スレッドの数と同じになります。サーバアプリケーションではスレッド単位の初期化処理ができます。また、そのスレッドで必要なリソースを準備できます。準備したリソースは、そのスレッド上に常駐するオブジェクトのインスタンスで共用して使用できます。リソースがスレッド単位の準備できるため、リソース使用時の排他処理が不要になります。

オブジェクトインスタンスが動作するスレッドの固定

1. 概要

一つのオブジェクトインスタンスに関するリクエストの処理は、必ず同じスレッド上で動作するように制御されます。また、オブジェクトインスタンスの初期化処理をするためのコンストラクタや終了処理をするためのデストラクタも同じスレッド上で呼び出されます。これによって、スレッドに関連する処理が容易になります。なお、コネクション切断などの障害時には、デストラクタは別のスレッド上で呼び出されることもあります。

1.1.3 スケーラブルな負荷分散

TPBroker OTM はコンピュータ間、スレッド間、およびプロセス間の負荷情報を交換し負荷分散します。また、同じ種類のオブジェクトに対応するスケジュール用キューの間で、負荷情報を交換して負荷分散します。

負荷分散をすることで、次のように制御できます。

ロードバランスの調整

負荷情報の交換の間隔や負荷バランスの度合いを調整することによって、システム構成やネットワーク環境に適した負荷分散を実現できます。

ローカルの TSC ノードの優先

TSC ノードはシステムプロセスである TSC デーモンやサーバアプリケーションなどで構成するプロセスの集合です。複数の TSC ノードに同じ種類のオブジェクトがある場合、通常、各 TSC ノードに同量のリクエストが割り当てられます。しかし、リクエストを受け付けた TSC ノードのオブジェクトのインスタンスに優先的に割り当てるように指定できます。

異常終了時の縮退

同じオブジェクトを複数のサーバアプリケーションプロセスに割り当てることができます。その場合、一つのサーバアプリケーションプロセスが異常終了しても、残りのサーバアプリケーションプロセスで縮退してリクエストを処理できます。

また、同じオブジェクトを複数の TSC ノードに配置できます。その場合、一つの TSC ノードが異常終了しても、残りの TSC ノードで縮退してリクエストの処理を続行できます。

1.1.4 コネクションの集約

クライアントアプリケーションからの複数のオブジェクトに対するリクエストを、一つのコネクションを通して要求します。

また、複数のクライアントアプリケーションからのコネクションを専用のプロセス (TSC レギュレータ) で集約してリクエストできます。TSC レギュレータは、複数のプロセスを開始できます。なお、TSC レギュレータを経由することなく、スケジュール用キューのプロセス (TSC デーモン) に直接、リクエストすることもできます。

1.1.5 サーバの動的なメンテナンス

サーバアプリケーションは、動的に追加および削除できます。また、スケジュール用キューを閉塞させることによって、システムを停止することなくサーバアプリケーションの入れ替えもできます。

さらに、システムの動作中に新しいTSC ノードを追加することもできます。追加したあとは、そのTSC ノードに対応するサーバアプリケーションにもリクエストが負荷分散されるようになります。

1.1.6 素早い異常検知と障害の局所化

TPBroker OTM では、次に示す時間を監視することで、アプリケーションプログラムの異常を素早く検知できます。このため、部分的な障害がシステム全体に影響するのを抑えられます。また、TPBroker の運用管理機能と連携して、異常終了したアプリケーションプログラムを自動再起動することもできます。

ユーザメソッドの実行時間の監視

サーバアプリケーションのユーザメソッドを開始してから終了するまでの実行時間を監視します。指定した時間を超えると、サーバアプリケーションプロセスが強制終了します。この監視機能は、アプリケーションプログラムがデッドロックしてしまった場合などに有効です。

クライアントでの応答時間の監視

クライアントからサーバアプリケーションにリクエストを送信してから応答が返るまでの時間を監視します。指定した時間を超えると、リクエストを送信したメソッドに例外が返ります。

1.1.7 多様なプログラミング言語への対応

アプリケーションプログラムの作成には、C++ および Java 言語を使用できます。クライアントアプリケーションは、Java アプリケーションだけでなく Java アプレットでも作成できます。また、クライアントアプリケーションとサーバアプリケーションとを異なる言語で作成することもできます。

1.1.8 CORBA に準拠するクライアントとの接続

OTM - Connector for ORB は、CORBA 2.5 準拠するクライアント（ORB クライアント）からのリクエストを TPBroker OTM で処理するためのゲートウェイ機能を提供します。これによって、TPBroker で構築した既存のシステムや、ORB 機能を持つさまざまなベンダのクライアントと接続して、スケジュール用キューを使用したリクエストのスケジューリング、スケーラブルな負荷分散といった TPBroker OTM の特長を生かした分散処理システムを構築できます。なお、このとき、ORB クライアントの既存のアプリケーションプログラムを修正したりコンパイルし直したりする必要はなく、そのまま使用で

1. 概要

きます。

注意事項

OTM - Connector for ORB については、サポート時期をご確認ください。

1.2 システム構成

この節では、TPBroker OTM を使用するシステムを構成する各プロセスについて説明します。さらに、TSC ドメインおよび TSC ノード当たりのプロセス数の範囲を示します。

TPBroker OTM を使用するシステムは、OTM、OTM - Client、OTM - Connector for ORB、および TPBroker で構成されます。TPBroker OTM を使用したシステムでは、OTM と TPBroker があるホストでサーバアプリケーションおよびクライアントアプリケーションが動作します。また、OTM - Client と TPBroker があるホストでクライアントアプリケーションが動作します。OTM - Connector for ORB は、ORB 機能を持つクライアントから OTM のサーバアプリケーションに対してリクエストする場合に必要なプログラムです。

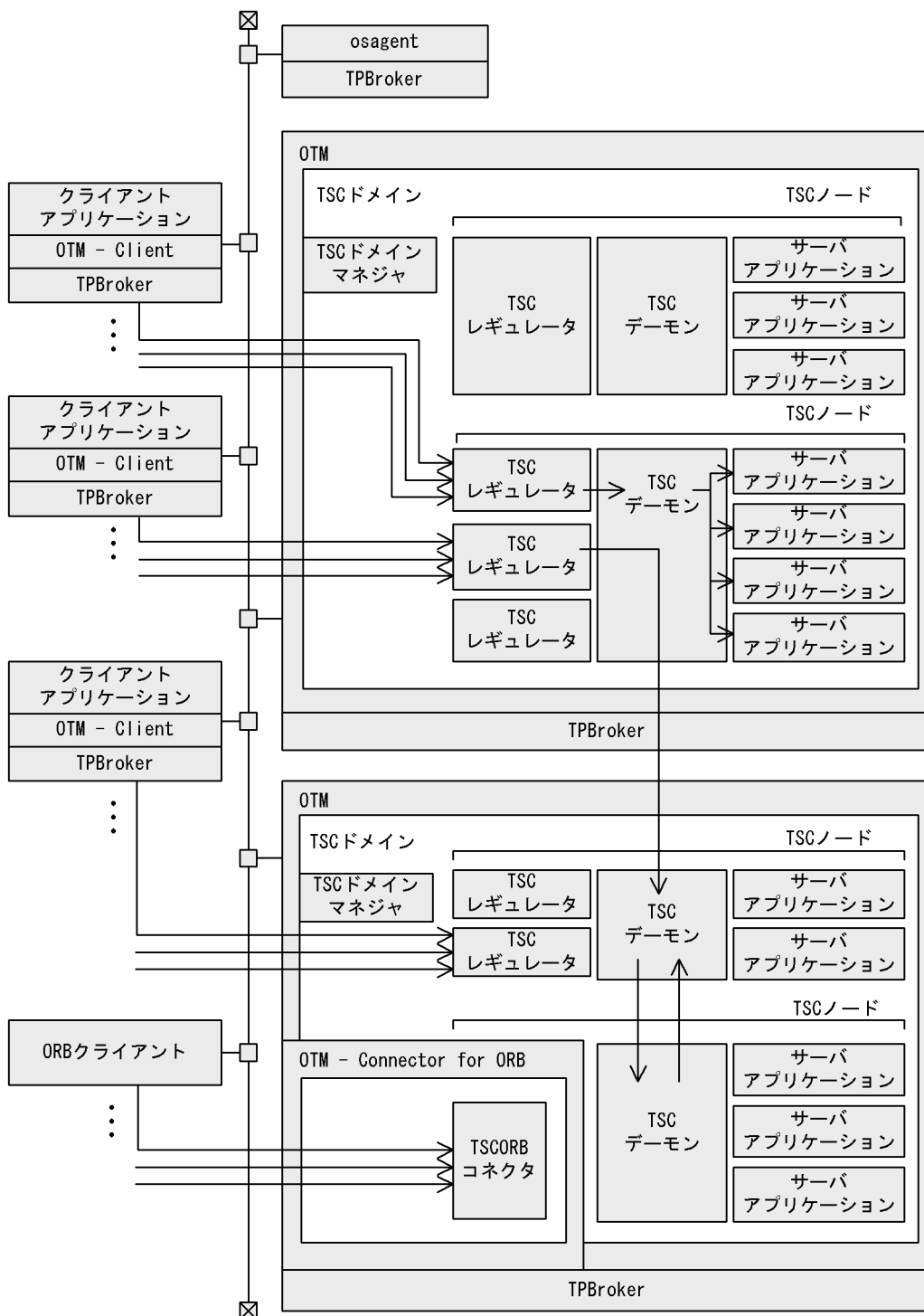
! 注意事項

OTM - Connector for ORB については、サポート時期をご確認ください。

TPBroker OTM を使用したシステムの構成を次の図に示します。

1. 概要

図 1-2 システムの構成



以降、図中のプロセスなどについて説明します。

1.2.1 クライアントの構成

OTM - Client

TPBroker OTM を使用するシステムでは、OTM - Client の動作するコンピュータがクライアントとなります。OTM - Client は TPBroker 上で動作します。

クライアントアプリケーション

ユーザが業務に合わせて作成するアプリケーションプログラムです。作成時には、C++ または Java 言語を使用できます。Java アプリケーションだけでなく Java アプレットでも作成できます。また、クライアントアプリケーションとサーバアプリケーションとを異なる言語で作成できます。

クライアントアプリケーションは、サーバアプリケーションに対してリクエストします。クライアントアプリケーションの作成時には、IDL ファイルに定義したユーザ定義 IDL インタフェースから生成されたプログラミングコードに基づいて記述してください。

クライアントアプリケーションの詳細については、「2.1 クライアントアプリケーションの機能」を参照してください。

TPBroker

TPBroker は、VisiBroker を ORB とするトランザクショナル分散オブジェクト基盤です。

OTM - Client は、Cosminexus TPBroker Version 5 に対応します。ORB の詳細については、マニュアル「Borland Enterprise Server VisiBroker デベロッパーズガイド」を参照してください。

ORB クライアント

ORB クライアントとは、TPBroker クライアント、および ORB 機能を持つさまざまなベンダのクライアントの総称です。ORB クライアントには、CORBA 2.5 に準拠するクライアントアプリケーションを配置します。TSCORB コネクタを経由することで、通常の CORBA 呼び出しと同様の手順で OTM のサーバアプリケーションに対するリクエストができます。

1.2.2 サーバの構成

osagent

TPBroker のスマートエージェントです。OTM は、TSC デーモンおよび TSC ドメインマネージャのオブジェクトリファレンスを osagent に登録します。OTM のサーバアプリケーション内のオブジェクトは、OTM が管理するため、osagent には登録されません。

osagent の詳細については、マニュアル「Borland Enterprise Server VisiBroker デベロッパーズガイド」を参照してください。

OTM

OTM は TPBroker 上で動作し、次に示すプロセスから構成されます。

1. 概要

- TSC ドメインマネージャ
- TSC レギュレータ
- TSC デーモン
- サーバアプリケーション

TSC ドメイン

TSC ドメインは、サーバアプリケーションの存在や負荷情報を交換する TSC ノードの集合で、ネットワーク内にある TSC デーモンの情報などを管理します。一つのコンピュータに複数の TSC ドメインを配置できます。また、同一の TSC ドメイン内にあるサーバアプリケーションであれば、位置を意識しないで呼び出せます。

TSC ドメインは TSC ドメイン名称を持ちます。この TSC ドメイン名称は TSC ドメインマネージャの開始時に実行する `tscdmstart` コマンドの `-TSCDomain` オプションで指定します。

TSC ドメインマネージャ

TSC ドメインマネージャは、TSC ドメイン内で TSC ノードの情報を管理するプロセスです。TSC ノードの情報には、TSC デーモンの追加などの情報があります。TSC ドメインマネージャは、各コンピュータにある TSC ドメインの一つずつ配置します。

TSC ドメインマネージャを開始するには、`tscdmstart` コマンドを実行します。その後、`tsctestart` コマンドによって TSC デーモンが開始されると、TSC ドメインマネージャは、ほかの TSC ドメインマネージャに TSC ドメインの構成が変化したことを連絡します。TSC ドメインマネージャの詳細については、「2.5 TSC ドメインマネージャの機能」を参照してください。

TSC ノード

TSC ノードは、TSC レギュレータ、TSC デーモン、およびサーバアプリケーションから構成されます。

TSC ノードは、TSC ドメイン内でユニークな TSC 識別子を持ちます。この TSC 識別子は TSC デーモンの開始時に `tsctestart` コマンドの `-TSCID` オプションで指定します。

TSC レギュレータ

TSC レギュレータは、クライアントアプリケーションからのリクエストを受け付けるコネクションを集約し、管理するプロセスです。コネクションを集約して TSC デーモンの資源を分散することによって、システムを安定して動作させます。一つの TSC デーモンに対して複数の TSC レギュレータを関連づけられます。

TSC レギュレータを開始するには、`tscregltd` コマンドを実行します。なお、TSC レギュレータは省略することもできます。TSC レギュレータを省略した場合は、クライアントアプリケーションは TSC デーモンに直結してリクエストします。

TSC レギュレータの詳細については、「2.3 TSC レギュレータの機能」を参照してください。

TSC デーモン

OTM の機能の中心となるデーモンで、TSC レギュレータおよびサーバアプリケー

ションとともに TSC ノードを構成します。TSC デーモンは、クライアントアプリケーションまたは TSC レギュレータからのリクエストをスケジュール用キューに滞留します。また、負荷情報を基にサーバアプリケーションを呼び出します。TSC デーモンを開始するには、tscstart コマンドを実行します。TSC デーモンの詳細については、「2.4 TSC デーモンの機能」を参照してください。

サーバアプリケーション

ユーザが業務に合わせて作成するアプリケーションプログラムです。作成時には、C++ または Java 言語を使用できます。また、クライアントアプリケーションとサーバアプリケーションとを異なる言語で作成できます。サーバアプリケーションの作成時には、IDL ファイルに定義したユーザ定義 IDL インタフェースから生成されたプログラミングコードに基づいて記述してください。サーバアプリケーションでは、サービスを提供する複数のオブジェクトが動作します。これを TSC ユーザオブジェクトといいます。また、一つの TSC デーモンに対して複数のサーバアプリケーションが関連づけられます。サーバアプリケーションの詳細については、「2.7 サーバアプリケーションの機能」を参照してください。

TPBroker

TPBroker は、VisiBroker を ORB とするトランザクショナル分散オブジェクト基盤です。

OTM は、Cosminexus TPBroker Version 5 に対応します。ORB の詳細については、マニュアル「Borland Enterprise Server VisiBroker デベロッパーズガイド」を参照してください。

OTM - Connector for ORB

OTM - Connector for ORB は、TSCORB コネクタを提供するプログラムです。

! 注意事項

OTM - Connector for ORB については、サポート時期をご確認ください。

TSCORB コネクタ

TSCORB コネクタは、ORB クライアントからのリクエストを受け付けて、OTM のサーバアプリケーションで処理できるようにするゲートウェイプロセスです。TSC レギュレータと同様に、ORB クライアントからのリクエストを受け付けるコネク션을集約し、管理します。なお、TSCORB コネクタを使用する場合は OTM - Connector for ORB が必要です。

! 注意事項

OTM - Connector for ORB については、サポート時期をご確認ください。

1. 概要

1.2.3 プロセス数の範囲

コンピュータ内にある TSC ドメイン当たりの各プロセス数の範囲と、TSC ノード当たりの各プロセス数の範囲を次の表に示します。

表 1-1 プロセス数の範囲

プロセスの名称	TSC ドメイン当たりの プロセス数	TSC ノード当たりの プロセス数
クライアントアプリケーション	0 以上	-
TSC ドメインマネージャ	1	-
TSC レギュレータ	0 以上	0 以上
TSCORB コネクタ	0 以上	0 以上
TSC デーモン	1 以上	1
サーバアプリケーション	0 以上	0 以上

(凡例)

- : 該当しません。

注

TSCORB コネクタを使用するには、OTM - Connector for ORB が必要です。OTM - Connector for ORB については、サポート時期をご確認ください。

2

機能

この章では、TPBroker OTM の機能について説明します。

2.1 クライアントアプリケーションの機能

2.2 接続先情報ファイルを使用する接続

2.3 TSC レギュレータの機能

2.4 TSC デーモンの機能

2.5 TSC ドメインマネージャの機能

2.6 複数のネットワークセグメントで構成する TSC ドメイン

2.7 サーバアプリケーションの機能

2.8 TSCORB コネクタの機能

2.9 稼働統計情報の取得

2.1 クライアントアプリケーションの機能

クライアントアプリケーションはユーザが業務に合わせて作成するアプリケーションプログラムです。クライアントアプリケーションは、サーバアプリケーションに対してリクエストします。

この節では、クライアントアプリケーションで実行できるリクエスト形態、およびサーバアプリケーションにリクエストするために必要な TSC ノードとの接続について説明します。また、TSC ノードとのリバインド、およびクライアントアプリケーションの状態の検出について説明します。

2.1.1 リクエストの形態

OTM や OTM - Client で実行できるリクエストの形態には、同期型呼び出し、非応答型呼び出し、およびセッション型呼び出しがあります。

(1) 同期型呼び出し

クライアントアプリケーションが、サーバアプリケーションへリクエストを送信してから、サーバアプリケーションからの応答を受信するまで、クライアントアプリケーションのリクエスト送信スレッドが待機する形態です。

(2) 非応答型呼び出し

クライアントアプリケーションが、サーバアプリケーションへリクエストを送信するだけで、サーバアプリケーションからの応答を受信しない形態です。クライアントアプリケーションのリクエスト送信スレッドは、サーバアプリケーションからの処理結果を待たないため、すぐに次の処理を実行できます。ただし、サーバアプリケーションでリクエストが実行される保証はありません。

非応答型呼び出しを使用するアプリケーションを作成するには、IDL の operation に oneway を記述してトランザクションフレームを出力する必要があります。また、応答メッセージがないため、operation に out, inout 引数や return 値は定義できません。詳細については、マニュアル「TPBroker Object Transaction Monitor プログラマーズガイド」を参照してください。

(3) セッション型呼び出し

同期型、非応答型の通常のステートレスな OTM の呼び出しでは、TSC ユーザオブジェクトの負荷状況に応じてリクエストが振り分けられるため、連続したリクエストがサーバアプリケーションの同じ TSC ユーザオブジェクトで実行されるとは限りません。

セッション型呼び出しを使用すると、サーバアプリケーションの同じ TSC ユーザオブジェクトを連続して同期型呼び出しすることができます。

詳細については、「2.1.4 セッション呼び出し機能」を参照してください。

2.1.2 TSC ノードとの接続

クライアントアプリケーションは、サーバアプリケーションの TSC ユーザオブジェクトにリクエストするために TSC ノードと接続します。このとき、クライアントアプリケーションを TSC デモンおよび TSC レギュレータと同じホストに配置する必要はありません。

TSC ノードとの接続では、接続経路、接続方法、および接続する TSC ノードを指定できません。

(1) 接続経路の指定

クライアントアプリケーションから TSC ノードに接続するときに、リクエストの経路を指定できます。指定方法には、TSCAdm クラスおよび TSCDomain クラスからプログラム上で実行する方法、クライアントアプリケーションのコマンドオプション引数で運用時に実行する方法、または両者を組み合わせる方法があります。

TSC デモンに直結する方法

TSC レギュレータを経由しないで、クライアントアプリケーションと TSC デモンを直結する方法です。オーバーヘッドが少ない接続経路です。

TSC レギュレータを経由して TSC デモンに接続する方法

クライアントアプリケーションから TSC デモンに接続するときに、TSC レギュレータを経由する方法です。オーバーヘッドが発生しますが、コネクションを集約して TSC デモンの資源を分散することによって、システムを安定して動作させます。

TSC レギュレータの詳細については、「2.3 TSC レギュレータの機能」を参照してください。

(2) 接続方法の指定

接続先情報ファイルで、TSC ノードへの接続を確立する方法を指定できます。接続先情報ファイルを使用しない場合は、TPBroker のスマートエージェントを使用した接続が選択されます。接続先情報ファイルについては、「2.2 接続先情報ファイルを使用する接続」を参照してください。

TPBroker のスマートエージェントの使用

通常は、TPBroker のスマートエージェント (osagent) を使用して接続を確立します。

IOR 文字列の使用

IOR 文字列を使用して接続を確立します。IOR 文字列を使用する接続では、TPBroker のスマートエージェント (osagent) を使用しないため、問い合わせ処理が不要です。さらに、ORB ドメインを意識しないため、複数の ORB ドメインにわたった接続もできます。したがって、多数のクライアントアプリケーションから同時に接続要求があるときに、osagent および TSC デモンの負荷を軽減させ、コネクションの不足を防止するために有効です。

2. 機能

なお、TSC レギュレータを経由する接続で IOR 文字列を使用する場合は、TSC レギュレータの開始時に、`tscregld` コマンドの `-TSCIORConnect` オプションを指定する必要があります。

(3) 接続する TSC ノードの指定

TSC ノードの動作するコンピュータの能力と負荷を考慮して、接続する TSC ノードを指定できます。次に示す 3 とおりの指定方法があります。

クライアントアプリケーションで TSC ドメイン名称と TSC 識別子を指定する場合、ユーザの指定した TSC ノードが選択されます。

クライアントアプリケーションで TSC ドメイン名称だけを指定する場合、TSC ドメイン内の任意の TSC ノードが選択されます。

接続先情報ファイルによって指定する場合、接続先情報ファイル中の情報に従って、一つ以上の TSC ノードが選択されます。接続の対象となるための条件については、「2.2.4 マルチノードリトライ接続の接続対象」を参照してください。

2.1.3 TSC ノードとのリバインド

TSC ノード (TSC デモンまたは TSC レギュレータ) との接続が障害などによって切断された場合、クライアントアプリケーションからメソッドを呼び出すと例外が返ります。その場合に再度メソッドを呼び出すには、クライアントアプリケーションで `getTSCClient()` メソッドを発行して TSC ノードに接続してから TSC ユーザプロキシを生成するなどの複雑な処理が必要です。

このような場合には、オートリバインド機能を使用することで、TSC ノードに自動的にリバインドできます。クライアントアプリケーションは、例外が発生する前に生成していた TSC ユーザプロキシを使用して、再度、メソッドを呼び出せます。

(1) オートリバインド機能の設定

オートリバインド機能を使用する場合は、クライアントアプリケーションの開始時に `-TSCRebindTimes` オプションにリバインド回数を指定し、`-TSCRebindInterval` オプションにリバインド間隔を秒単位で指定します。

オートリバインド機能は、次の表に示す例外が発生したときに動作します。

表 2-1 オートリバインド機能が動作する例外 (TSC デモンへ直結するクライアントアプリケーションの場合)

内容コード	完了状態	例外通知
SEND_CLNT_FAILURE	COMPLETED_NO	×
	COMPLETED_MAYBE	○

内容コード	完了状態	例外通知
CALL_IN_END	COMPLETED_NO	×
RECEIVE_CONN_DOWN	COMPLETED_NO	×

表 2-2 オートリバインド機能が動作する例外（TSC レギュレータを経由するクライアントアプリケーションの場合）

内容コード	完了状態	例外通知
SEND_THIN_CLNT_FAILURE	COMPLETED_NO	×
	COMPLETED_MAYBE	○
CALL_IN_END	COMPLETED_NO	×

（凡例）

- ：例外を受け取ったあと、リクエストを再送してリバインドします。
- ×

-TSCTimeOutRebind オプションを指定すると、次の表に示すタイムアウト例外が発生した場合にもオートリバインド機能が動作します。

表 2-3 -TSCTimeOutRebind オプション指定時にオートリバインド機能が動作する例外

内容コード	完了状態	例外通知
TIMED_OUT	COMPLETED_MAYBE	

（凡例）

- ：例外を受け取ったあと、リクエストを再送してリバインドします。

-TSCTimeOutRebind オプションを指定すると、サーバのハード障害やネットワーク障害などの、TCP/IP ですぐに検知できない障害が発生した場合に、オートリバインド機能を動作させることができます。ただし、サーバアプリケーションからの応答が監視時間を超えた場合にもオートリバインド機能が動作するので、注意が必要です。

これらの例外が発生した場合には、コネクション切断などの障害時ではなくても、オートリバインド機能が動作します。

さらに、オートリバインド機能の動作の有無は、接続元とその接続対象によっても異なります。各接続でオートリバインド機能が動作するかどうかを次の表に示します。

表 2-4 各接続でのオートリバインド機能の動作の有無

接続元	接続対象	動作の有無
クライアントアプリケーション	TSC デーモン	
クライアントアプリケーション	TSC レギュレータ	
TSC レギュレータ	TSC デーモン	×

2. 機能

接続元	接続対象	動作の有無
サーバアプリケーション	TSC デーモン	×
TSC デーモン	TSC デーモン	×

(凡例)

○ : オートリバインド機能が動作します。

× : オートリバインド機能は動作しません。

注

TSC デーモンはリクエストを受信後、目的の TSC ユーザオブジェクトがあるほかの TSC デーモンにリクエストを振り分けます。したがって、ほかの TSC デーモンにリクエストを転送するときにオートリバインド機能に対応する例外が発生しても、リバインドしません。それは、クライアントアプリケーションと TSC ノードとの接続は確立されているためです。この場合は、オートリバインド機能に対応する例外がそのままメソッド呼び出しの例外として、クライアントアプリケーションに返されます。

このとき、クライアントアプリケーションは、例外が発生する前に生成していた TSC ユーザプロキシを使用して、再度、メソッドを呼び出せます。

(2) 注意事項

オートリバインド機能を使用する場合の注意事項を次に示します。

-TSCRebindTimes オプションに指定したリバインド回数が超過しても、再度メソッドを呼び出せば、リバインドできます。

マルチスレッド環境下で複数のメソッドを同時に呼び出す場合、メソッドの呼び出しごとにリバインドできます。

メソッド呼び出しの時間監視は、次に示す時間 (秒) だけ長くなることがあります。

(リバインド回数 - 1) × (リバインド間隔)

リバインドするときに使用する TSC ノードの情報は、TSC ユーザプロキシに関連する TSCDomain クラスから取得されます。そのため、TSCDomain クラスを生成するときに一つの引数を取るコンストラクタを使用した場合で、リバインド前に getTSCDomain() メソッドまたは getTSCID() メソッドで取得した情報を使用するときは、必要な TSC ノードの情報がリバインド後に変更されることがあります。この場合は、リバインド後に getTSCDomain() メソッドまたは getTSCID() メソッドを再発行して、情報を取得し直してください。

接続先情報ファイルに複数の接続対象を指定した場合、リバインド回数 1 回ごとに、すべての接続対象に対して接続を実行します。詳細については、「2.2.6 接続および再接続の実行」を参照してください。

複数のスレッドがメソッドを呼び出す場合、内容コードが SEND_CLNT_FAILURE

で、完了状態が COMPLETED_MAYBE の例外を、複数回クライアントアプリケーションに返す場合があります。

2.1.4 セッション呼び出し機能

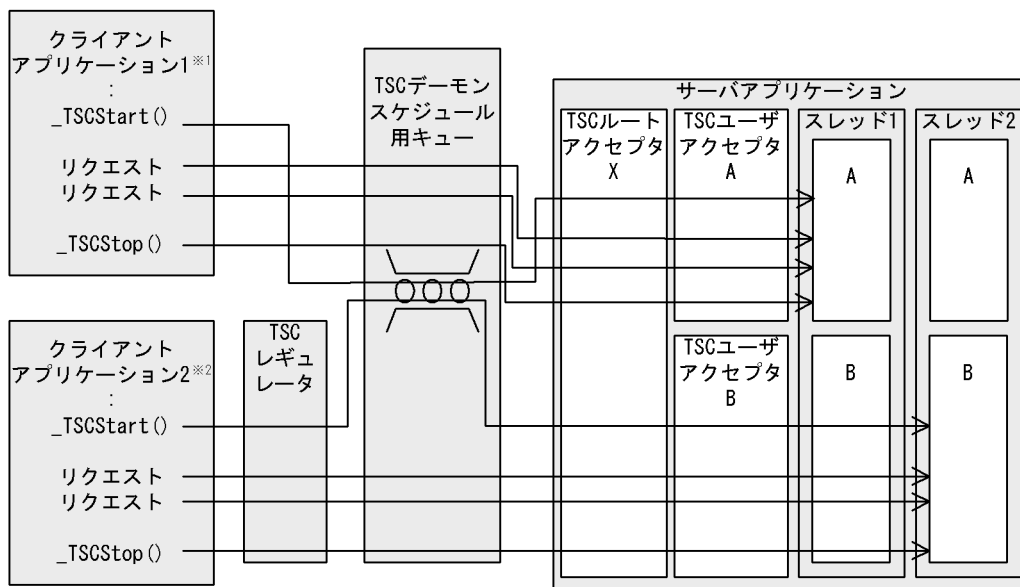
セッション呼び出し機能を使用すると、クライアントアプリケーション内の TSC ユーザプロキシとサーバアプリケーション内の TSC ユーザオブジェクトとを 1 対 1 で対応づけます。セッションを確立した TSC ユーザプロキシからのリクエストはすべて、対応づけられた同じ TSC ユーザオブジェクトで実行できます。セッションごとにサーバアプリケーションプロセスのスレッドが割り当てられ、ほかのリクエストからは呼び出されないように占有するため、スレッド多重度が最大セッション数になります。

セッション呼び出し機能を使用するには、TSC セッションプロキシ (TSCSessionProxy) クラスを継承した TSC ユーザプロキシが必要です。

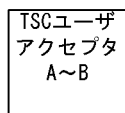
セッションを確立するには、TSC セッションプロキシからセッション開始 (_TSCStart() メソッド) を要求します。確立したセッションを解放するには、TSC セッションプロキシからセッション解放 (_TSCStop() メソッド) を要求します。なお、セッション開始要求は、スケジュール用キューに登録されますが、セッション確立後のリクエストやセッション解放要求は、スケジュール用キューに登録されません。

セッション呼び出し機能を使用した場合の動作を次の図に示します。

図 2-1 セッション呼び出し機能の動作



(凡例)



: A~Bの各インターフェースに対応するTSCユーザアクセプタ



: A~Bの各インターフェースに対応するTSCユーザオブジェクト

注 1

セッション呼び出し用 TSC ユーザプロキシを継承したプロキシ A を持つクライアントアプリケーションです。

注 2

セッション呼び出し用 TSC ユーザプロキシを継承したプロキシ B を持つクライアントアプリケーションです。

(1) セッション呼び出しインターバル時間監視

セッション確立中に、セッションに割り当てられたサーバの資源が、クライアントの障害などで次のリクエストを受け取れないまま占有され続けることを防止するため、リクエストの時間間隔を監視します。セッションを確立して対応づけられた TSC ユーザオブジェクトが、この監視時間を経過しても次のリクエストを受信しなかった場合、セッションは強制的に解放されます。詳細については、「2.4.8 セッション呼び出しインターバル時間監視」を参照してください。

(2) セッション呼び出しアプリケーションの作成方法

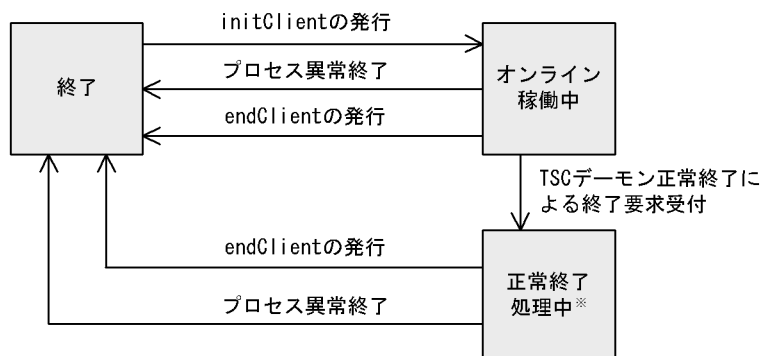
セッション呼び出しを使用するアプリケーションを作成するには、トランザクションフレームジェネレータに `-TSCspxy` オプションを指定して、TSC セッションプロキシを継承した TSC ユーザプロキシを出力する必要があります。詳細については、マニュアル「TPBroker Object Transaction Monitor プログラマーズガイド」を参照してください。

2.1.5 クライアントアプリケーションの状態の検出

クライアントアプリケーションで発行する TSCAdm クラスの `get_status()` メソッドの戻り値によって、オンライン稼働中かどうかといったクライアントアプリケーションの状態を検出できます。

クライアントアプリケーションの状態遷移を次の図に示します。

図 2-2 クライアントアプリケーションの状態遷移



注

クライアントアプリケーションを `kill` コマンドで強制終了した場合、または `tscstop` コマンドに `-TSCForce` オプションを指定して強制終了した場合は、プロセス異常終了と同じ遷移になります。

注

OTM のクライアントアプリケーションは、TSC デーモンの正常終了時にクライアントアプリケーションの状態を変更してプロセスの終了を待ち合わせます。OTM - Client のクライアントアプリケーション、および `-TSCWithSystem` オプションに "0" を指定して開始したクライアントアプリケーションは、TSC デーモンからの監視および終了を実行しないため、正常終了処理中にはなりません。

2.2 接続先情報ファイルを使用する接続

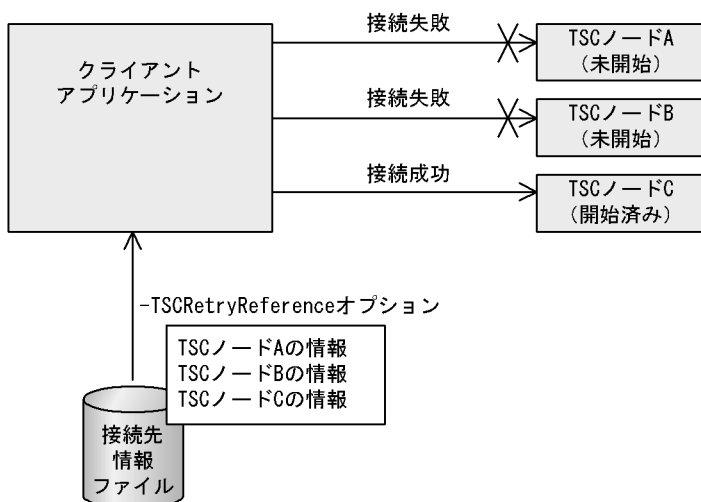
接続先情報ファイルを使用することで、クライアントアプリケーションからの接続要求に失敗した場合やリクエストに障害が発生した場合、指定した複数の TSC ノードを対象として接続対象を切り替えることができます。この機能をマルチノードリトライ接続といいます。また、接続先情報ファイルを使用することで、TPBroker のスマートエージェントを使用することなく、IOR 文字列を使用した接続もできます。

2.2.1 マルチノードリトライ接続の概要

クライアントアプリケーションの開始時に、`-TSCRetryReference` オプションに接続先情報ファイルを指定することによって、複数の TSC ノードを対象とした再接続（マルチノードリトライ接続）を実行できます。TSC ノードの二重化や複数の TSC ノードによる大規模クラスタを構成している場合に、系切り替えが発生したり、一つの TSC ノードに障害が発生したりしても、クライアントアプリケーションの接続対象を容易に切り替えられます。

マルチノードリトライ接続の動作を次の図に示します。

図 2-3 マルチノードリトライ接続の動作



2.2.2 マルチノードリトライ接続と IOR 文字列を使用する接続の併用

接続先情報ファイルを使用することで、マルチノードリトライ接続と同時に IOR 文字列を使用する接続を選択できます。

TSC ノードへの接続方法は、クライアントアプリケーションの開始時に

-TSCRetryReference オプションに指定する接続先情報ファイルの内容によって決定されます。-TSCRetryReference オプションを指定しない場合は、TPBroker のスマートエージェントを使用する接続方法が選択されます。また、TSC レギュレータを経由する接続で IOR 文字列を使用する場合は、TSC レギュレータの開始時に -TSCIORConnect オプションを指定する必要があります。

2.2.3 マルチノードリトライ接続での接続方式の種類

マルチノードリトライ接続では、マルチノードリトライ順序とマルチノードリトライ方式の 2 種類を組み合わせて指定できます。

マルチノードリトライ順序

接続先情報ファイルの先頭から順に接続するか、ランダムに接続対象を選択して接続するかを指定できます。

- 順次方式 (デフォルト)

接続先情報ファイルの先頭から順に接続する方式です。この方式を採用すると、接続先情報ファイルのレコードの記述順序を編集することによって、接続および再接続の対象に優先順位を付けられます。

- ランダム方式

接続先情報ファイルからランダムに接続対象を選択して接続する方式です。この方式を採用すると、複数のクライアントアプリケーションで同一の接続先情報ファイルを使用する場合や、一つのクライアントアプリケーションで複数の TSCClient オブジェクトを使用する場合でも、TSC ノードへのコネクションを分散させられます。

マルチノードリトライ方式

接続対象を選択する方法を指定できます。

- ファイル検索方式 (デフォルト)

接続先情報ファイルから接続対象を選択して接続する方式です。TSCAdm クラスの getTSCClient() メソッドに指定した条件に従って接続対象が選択されます。

- API 優先方式

最初の接続では、TSCDomain オブジェクトに指定した TSC ドメイン名称と TSC 識別子を持つ TSC ノードに対して、TPBroker のスマートエージェントを使用します。接続に失敗した場合は、接続先情報ファイルから接続対象が選択されます。

上記のマルチノードリトライ順序とマルチノードリトライ方式を組み合わせることで、次の表に示すように接続対象が選択されます。

表 2-5 マルチノードリトライ接続での接続方式の組み合わせ

マルチノードリトライ順序	マルチノードリトライ方式	接続対象の選択方法
順次方式	ファイル検索方式	接続先情報ファイルの先頭から順に接続します。
ランダム方式	ファイル検索方式	接続先情報ファイルからランダムに接続対象が選択されます。

マルチノード リトライ順序	マルチノードリト ライ方式	接続対象の選択方法
順次方式	API 優先方式	最初の接続では、TSCAdm クラスの getTSCClient() メソッドで指定された TSC ノードに対して TPBroker のスマートエージェントを使用します。以降は、接続先情報ファイルの先頭から順に接続します。
ランダム方式	API 優先方式	最初の接続では、TSCAdm クラスの getTSCClient() メソッドで指定された TSC ノードに対して TPBroker のスマートエージェントを使用します。以降は、接続先情報ファイルからランダムに接続対象が選択されます。

2.2.4 マルチノードリトライ接続の接続対象

マルチノードリトライ接続の対象となるための条件は、接続方式がファイル検索方式なのか、API 優先方式なのかによって異なります。マルチノードリトライ接続では、条件を満たす接続対象の TSC ノードに対して接続および再接続を実行します。なお、マルチノードリトライ接続の接続方式は、クライアントアプリケーションの開始時に `-TSCRetryWay` オプションで指定します。

ここでは、接続の対象となるための条件を接続方式ごとに示します。さらに、接続方式を指定する方法を示します。

(1) ファイル検索方式の場合の接続対象

ファイル検索方式は、`-TSCRetryWay` オプションの指定値が "0000" または "0001" の場合、または `-TSCRetryWay` オプションの指定を省略して "0000" が仮定された場合に採用される接続方式です。この場合は、次に示すすべての条件を満たす接続先情報ファイル中の TSC ノードが接続対象となります。

- TSCAdm クラスの getTSCClient() メソッドに指定した TSCDomain オブジェクトのコンストラクタに指定した TSC ドメイン名称 (NULL を指定した場合は接続先情報ファイル中にあるすべての TSC ドメイン名称)、および TSC 識別子 (省略または NULL を指定した場合は接続先情報ファイル中にあるすべての TSC 識別子) を持つこと。
- TSCAdm クラスの getTSCClient() メソッドに指定した (指定がない場合は `-TSCRequestWay` オプションの指定値) 接続経路であること。

ファイル検索方式の場合の接続対象を次の表に示します。

表 2-6 ファイル検索方式の場合の接続対象

TSCDomain コンストラクタの引数		接続対象
TSC ドメイン名称	TSC 識別子	
NULL	NULL または 指定なし	接続経路 が等しいすべての TSC ノード
NULL	NULL 以外	TSC 識別子および接続経路 が等しいすべての TSC ノード
NULL 以外	NULL または 指定なし	TSC ドメイン名称および接続経路 が等しいすべての TSC ノード
NULL 以外	NULL 以外	TSC ドメイン名称, TSC 識別子, および接続経路 が等しいすべての TSC ノード

注

接続先情報ファイルに設定される、クライアントアプリケーションと TSC デーモン間の接続経路です。TSC デーモンに直結する方法と、TSC レギュレータを経由して TSC デーモンに接続する方法の 2 種類があります。

(2) API 優先方式の場合の接続対象

API 優先方式は、`-TSCRetryWay` オプションの指定値が "0010" または "0011" の場合に採用される接続方式です。

この場合、最初の接続には TPBroker のスマートエージェントを使用します。このときの接続対象は、TSCAdm クラスの `getTSCClient()` メソッドに指定した TSC ノードです。最初の接続での接続対象を表 2-7 に示します。

以降の接続では、接続先情報ファイル中の TSCAdm クラスの `getTSCClient()` メソッドに指定した接続経路を持つ TSC ノードが接続対象となります。`getTSCClient()` メソッドに指定がない場合は、`-TSCRequestWay` オプションの指定値に従います。

表 2-7 API 優先方式の場合の最初の接続対象

TSCDomain コンストラクタの引数		接続対象
TSC ドメイン名称	TSC 識別子	
NULL	指定なし	<code>-TSCDomain</code> オプションに指定した TSC ドメイン名称を持つ任意の TSC ノード
NULL	NULL	<code>-TSCDomain</code> オプションに指定した TSC ドメイン名称、および <code>-TSCID</code> オプションに指定した TSC 識別子を持つ TSC ノード
NULL	NULL 以外	<code>-TSCDomain</code> オプションに指定した TSC ドメイン名称、および TSCDomain コンストラクタに指定した TSC 識別子を持つ TSC ノード

2. 機能

TSCDomain コンストラクタの引数		接続対象
TSC ドメイン名称	TSC 識別子	
NULL 以外	指定なし	TSCDomain コンストラクタに指定した TSC ドメイン名称を持つ任意の TSC ノード
NULL 以外	NULL	TSCDomain コンストラクタに指定した TSC ドメイン名称、および -TSCID オプションに指定した TSC 識別子を持つ TSC ノード
NULL 以外	NULL 以外	TSCDomain コンストラクタに指定した TSC ドメイン名称、および TSC 識別子を持つ TSC ノード

注

TSC ノードへの接続経路は、TSCAdm クラスの `getTSCClient()` メソッドの指定に従います。`getTSCClient()` メソッドに指定がない場合は、`-TSCRequestWay` オプションの指定に従います。

(3) マルチノードリトライ接続の接続方式を指定する方法

マルチノードリトライ接続の接続方式を指定するには、クライアントアプリケーションの開始時に `-TSCRetryWay` オプションで、4 けたの符号なし整数を指定します。

`-TSCRetryWay` オプションは、必ず `-TSCRetryReference` オプションと同時に指定しなければなりません。

接続方式の指定方法を次の表に示します。

表 2-8 マルチノードリトライ接続での接続方式の指定方法

クライアントアプリケーションの開始時の指定オプション	マルチノードリトライ接続の有効性	マルチノードリトライ順序	マルチノードリトライ方式
なし	無効	-	-
<code>-TSCRetryWay 00xx</code>	無効	-	-
<code>-TSCRetryReference filename</code>	有効	順次方式	ファイル検索方式
<code>-TSCRetryReference filename -TSCRetryWay 0000</code>	有効	順次方式	ファイル検索方式
<code>-TSCRetryReference filename -TSCRetryWay 0001</code>	有効	ランダム方式	ファイル検索方式
<code>-TSCRetryReference filename -TSCRetryWay 0010</code>	有効	順次方式	API 優先方式
<code>-TSCRetryReference filename -TSCRetryWay 0011</code>	有効	ランダム方式	API 優先方式

(凡例)

x : 0 または 1

filename : 接続先情報ファイルの名称

- : 該当しません。

2.2.5 接続先情報ファイル

ユーザは、tscgetref コマンドを使用して接続先情報ファイルを作成します。接続先情報ファイルは、tscgetref コマンドのオプションに指定された条件（TSC ノード、接続経路、および接続方法など）を満たす、一つ以上のプロセスの情報を含むテキストファイルです。

接続先情報ファイルの形式を次に示します。

```

ファイル管理情報
レコード1
レコード2
:

```

先頭の 1 行は OTM が使用するファイル管理情報で、2 行目以降の各行はプロセスを表すレコードです。一つのレコードはシャープ (#) をデリミタ（区切り文字）とする、次に示す形式です。

TSCドメイン名称#TSC識別子#接続経路#接続方法#IPアドレス#ポート番号##IOR文字列

（凡例）

項目	意味
TSC ドメイン名称	TSC ノードが属する TSC ドメインの TSC ドメイン名称です。
TSC 識別子	TSC ノードの TSC 識別子です。
接続経路	TSC ノードへの接続経路です。TSC デーモンに直結する場合は "0"、TSC レギュレータを経由して TSC デーモンに接続する場合は "1" が設定されます。
接続方法	TSC ノードへの接続方法です。TPBroker のスマートエージェントを使用する接続は "0"、IOR 文字列を使用する接続は "1" が設定されます。
IP アドレス	TSC ノードの IP アドレスです。マルチホームドホスト環境では、TSC デーモンまたは TSC レギュレータの開始時に -TSCMyHost オプションに指定した IP アドレスまたはホスト名称に対する IP アドレスです。
ポート番号	TSC デーモンまたは TSC レギュレータのポート番号です。接続経路が "1" かつ接続方法が "0" の場合、"0" が設定されます。
IOR 文字列	TSC デーモンまたは TSC レギュレータの IOR 文字列です。接続方法が "0" の場合、何も設定されません。

作成した接続先情報ファイルの保存、削除、移動、および複製はユーザが任意に実行できます。また、次に示す規則に従ってテキストエディタなどで編集できます。

- 各レコードの順序を並び替えることができます。

- 各行を削除したり、ほかの接続先情報ファイルの行を追加したりできます。ただし、空行や空白行は追加しないでください。
- 1行目のファイル管理情報は削除、追加、および移動できません。
- 各行（1行目のファイル管理情報を含む）の内容は変更できません。
- 改行コードは、接続先情報ファイルを配置する OS の表現に従ってください。

接続先情報ファイルはテキスト形式のファイルなので、FTP コマンドなどで転送する場合はテキストモードで実行してください。

2.2.6 接続および再接続の実行

クライアントアプリケーションの開始時に `-TSCRetryReference` オプションに接続先情報ファイルを指定する場合に、接続および再接続が実行されるメソッドを次に示します。

TSCAdm クラスの `getTSCClient()` メソッド

`getTSCClient()` メソッドでは、接続対象となる各レコードに対して 1 回の接続が実行されます。このときの接続は、クライアントアプリケーション開始時に `-TSCRetryWay` オプションに指定したマルチノードリトライ接続の接続方式に従います。接続が成功した時点で `getTSCClient()` メソッドは正常にリターンします。すべてのレコードについて失敗した場合は、`TSCTransientException` 例外が発生します。

なお、TSCAdm クラスの `getTSCClient()` メソッド、および TSCDomain クラスのコンストラクタの動作は、`-TSCRetryReference` オプションの指定の有無と、`-TSCRetryWay` オプションの指定内容によって異なります。

TSC ユーザオブジェクトのメソッド

TSC ユーザオブジェクトのメソッド呼び出し時に通信障害が発生した場合、接続対象となる各レコードに対して 1 回の再接続が実行されます。このときの再接続は、クライアントアプリケーション開始時に `-TSCRetryWay` オプションに指定したマルチノードリトライ接続の接続方式に従います。接続が成功した時点でリクエストが送信されます。

すべてのレコードについて再接続に失敗した場合は、`-TSCRebindInterval` オプションで指定された間隔（秒）だけ待ったあと、再度、`-TSCRetryWay` オプションに指定したマルチノードリトライ接続の接続方式に従って再接続が実行されます。これを `-TSCRebindTimes` オプションで指定した回数分繰り返します。`-TSCRebindTimes` オプションで指定された回数分がすべて失敗すると、`TSCTransientException` 例外が発生します。

なお、一つの接続対象に対する接続が失敗したあと、次の接続対象に対する接続を実行するまでの間隔はありません。

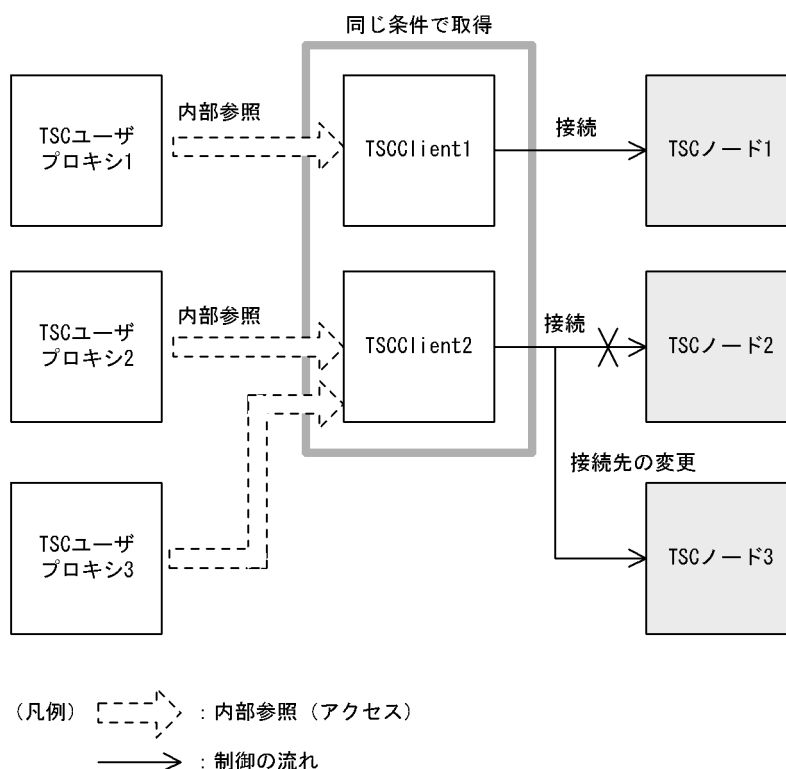
再接続が実行される条件は、オートリバインド機能と同じです。詳細については、「2.1.3(1) オートリバインド機能の設定」を参照してください。ただし、クライアントアプリケーションの開始時に `-TSCRebindTimes` オプションを指定しなかった場合、または "0" を指定した場合、再接続は実行されません。

2.2.7 TSCClient オブジェクトでの接続の保持

クライアントアプリケーションの開始時に、`-TSCRetryReference` オプションで接続先情報ファイルを指定する場合、`TSCAdm` クラスの `getTSCClient()` メソッドで取得される一つの `TSCClient` オブジェクトに対して、複数の接続対象を対応づけられます。複数の接続対象を持つ `TSCClient` オブジェクトを使用すると、一つの `TSC` ノードへの接続が失敗した場合、および確立済みの接続に通信障害が発生した場合に、異なる接続対象に対して再接続が実行されます。

接続および再接続の結果は、`TSCClient` オブジェクトに保持されます。`TSCClient` オブジェクトに保持される接続を次の図に示します。

図 2-4 TSCClient オブジェクトに保持される接続



複数の `TSC` ユーザプロキシオブジェクトが異なる `TSCClient` オブジェクトを内部参照する場合は、それぞれの `TSCClient` オブジェクト取得時の条件 (`TSCAdm` クラスの `getTSCClient()` メソッドに対する指定) が同じであっても、接続対象は異なることがあります。また、複数の `TSC` ユーザプロキシオブジェクトが同じ `TSCClient` オブジェクトを内部参照する場合は、一つの `TSC` ユーザプロキシオブジェクトで再接続が発生して接続対象が変わると、その結果が同じ `TSCClient` オブジェクトを内部参照する、ほかの `TSC` ユーザプロキシオブジェクトにも反映され、その `TSC` ユーザプロキシオブジェクト

2. 機能

の接続対象も変わります。

2.3 TSC レギュレータの機能

TSC レギュレータは、リクエストの集中による問題をコネクションのレギュレート（集約）によって解決するためのプロセスです。

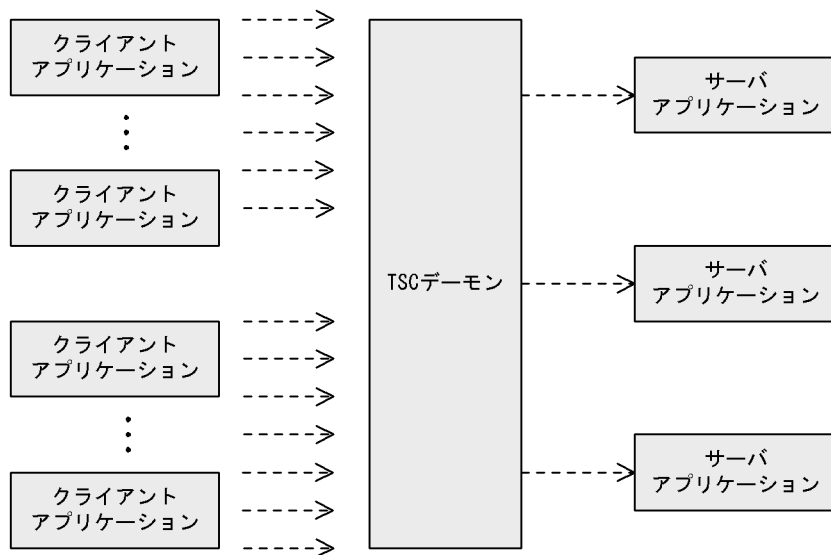
この節では、TSC レギュレータがリクエストの集中による問題を解決する仕組みと、TSC レギュレータと TSC デーモンとの対応について説明します。

2.3.1 コネクションのレギュレート

大規模なシステムでクライアントアプリケーションの数が増大した場合、システムの動作状態が不安定になったり、システムで管理する資源が不足して正常な処理ができなくなったりします。それは、TSC デーモンにリクエストが集中して、コネクション数が増大し、ファイルやソケットのオープン数などのプロセス使用資源が増大するためです。

リクエストが集中する仕組みを次の図に示します。

図 2-5 リクエストが集中する仕組み



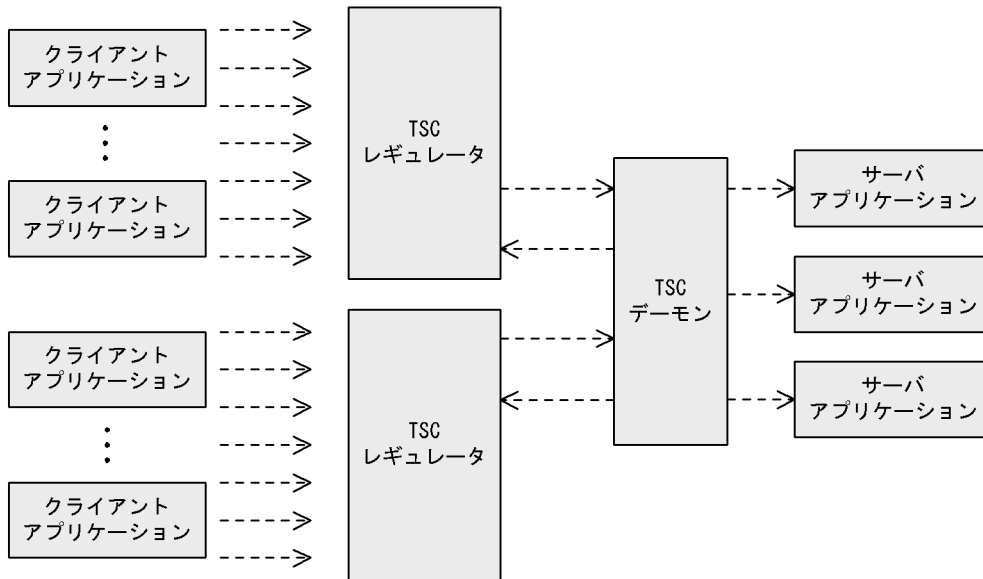
（凡例） -.-.-> : コネクションの方向

TSC レギュレータは、リクエストの集中による問題を解決するための専用のプロセスです。TSC レギュレータはクライアントアプリケーションからの幾つかのコネクションを集約し、TSC デーモン当たりのコネクションの上限を管理します。これをコネクションのレギュレートといいます。TSC レギュレータがコネクションをレギュレートすることによって、使用資源を複数プロセスに分散し、大規模なシステムをより安定して動作さ

せることができます。

コネクションのレギュレートの仕組みを次の図に示します。

図 2-6 コネクションのレギュレートの仕組み



(凡例) -----> : コネクションの方向

2.3.2 TSC レギュレータと TSC デモンの対応

OTM では、一つの TSC デモンに対して、必要に応じて複数の TSC レギュレータを開始できます。TSC レギュレータは、対応する TSC デモンと同じコンピュータに配置する必要があります。

TSC レギュレータがクライアントアプリケーションからのリクエストを受け付けると、対応する TSC デモンにリクエストを転送してリクエストの応答を待ちます。応答が返ってくると、応答をクライアントアプリケーションに返します。そのときの動作はマルチスレッドで並行処理されます。

2.3.3 TSC レギュレータの動作モード

TSC レギュレータは、通常クライアントアプリケーションからのリクエストをスレッドパーセションモデルで処理し、サーバアプリケーションからのリクエストの応答を TSC レギュレータ内でスレッドごとに待ち合わせます。

TSCAdm クラスの `getTSCClient()` メソッドで接続した TSC レギュレータへのコネクションを複数のスレッドで共有した場合は、リクエストを同時に処理できません。

マルチスレッド環境のクライアントアプリケーションからの TSC レギュレータ経由のリクエストを同時に処理させるには、リクエストスレッドごとに TSCAdm クラスの `getTSCClient()` メソッドを実行してコネクションを割り当てるか、または `tscregltd` コマンドに `-TSCTPool` オプションを指定して TSC レギュレータをスレッドプーリングモデルで動作させてください。

2.4 TSC デーモンの機能

TSC デーモンは、OTM の機能の中心となるデーモンです。

この節では、TSC デーモンによるリクエストの処理の手順や制御方法と、TSC デーモンを中心とするプロセスの配置について説明します。また、TSC デーモンによる、プロセスとアプリケーションプログラムの管理について説明します。

2.4.1 リクエストの処理

クライアントアプリケーションまたは TSC レギュレータからのリクエストを受信すると、TSC デーモンは次の手順でリクエストを処理します。

1. リクエストを受け付けた TSC デーモンに受信したリクエストを格納するか、ほかの TSC デーモンに転送するかを振り分けます。
2. 振り分けたリクエストをスケジュール用キューに登録します。
3. スケジュール用キューに登録されたリクエストは、サーバアプリケーションの TSC ユーザオブジェクトを呼び出します。その後、TSC デーモンは、TSC ユーザオブジェクトからの応答をリクエストの要求元に返します。

(1) リクエストの振り分け

TSC デーモンは、リクエストの受信後、目的の TSC ユーザオブジェクトがある TSC デーモンにリクエストを振り分け、スケジュール用キューに登録します。そのとき、負荷情報をほかの TSC デーモンと連絡し合い、リクエストを受け付けた TSC デーモンに格納するか、ほかの TSC デーモンに転送するかを振り分けます。ほかの TSC デーモンが異なるコンピュータにある場合でも TSC ドメインが同じであれば、リクエストを転送できます。

リクエストを振り分ける方法（スケジュールポリシー）には、次の 2 種類があります。

- TSC ルートアクセプタの負荷状況に応じてリクエストを振り分けます。
- リクエストを受け付けた TSC デーモンに登録されている TSC ルートアクセプタに、優先的にリクエストを振り分けます。ただし、その TSC デーモンに登録されている TSC ルートアクセプタが高負荷状態、および閉塞状態の場合は、ほかの TSC デーモンに登録している TSC ルートアクセプタにリクエストを振り分けます。

なお、スケジュールポリシーは、`tscstart` コマンドの `-TSCDispatchPolicy` オプションで指定できます。

(2) リクエストの登録

スケジュール用キューに同時に登録できるリクエスト数は、`tscstart` コマンドの `-TSCMaxRequestCount` オプションで指定できます。このオプションを省略した場合、

"50" が設定されます。これを超えてリクエストが発生した場合は例外が返ります。

リクエストには、プライオリティ（処理の優先順位）を設定できます。プライオリティ値が 1 の場合、最もプライオリティが高く、プライオリティ値が大きくなるほどプライオリティは低くなります。また、リクエストに設定されたプライオリティが高いかどうかは、`tscstart` コマンドの `-TSCHighPriorBorder` オプションの指定値で判断します。さらに、スケジュール用キューに登録できるリクエスト数のうち、プライオリティの高いリクエストだけを登録する数は、`tscstart` コマンドの `-TSCHighPriorRequestCount` オプションに指定できます。

これらのコマンドオプションの指定値を利用して、プライオリティの高いリクエストを優先してスケジュール用キューに登録できます。具体的には、プライオリティの低いリクエストの発生件数が、`-TSCMaxRequestCount` オプション指定値から `-TSCHighPriorRequestCount` オプション指定値を減算した値を超えた場合は例外が返ります。プライオリティの高いリクエストは、`-TSCMaxRequestCount` オプション指定値を超えて発生した場合にだけ例外が返ります。

(3) TSC ユーザオブジェクトの呼び出しと要求元への応答

スケジュール用キューに登録されたリクエストは、サーバアプリケーションの使用中でないスレッド上に配置された TSC ユーザオブジェクトを呼び出します。どのサーバアプリケーションの、どのスレッド上の TSC ユーザオブジェクトが呼び出されるかは、クライアントアプリケーションからはわかりません。

その後、TSC デーモンは、TSC ユーザオブジェクトからの応答をリクエストの要求元に返します。ほかの TSC デーモンから転送されてきたリクエストについては、該当する転送元を経由して、リクエストの要求元に応答を返します。

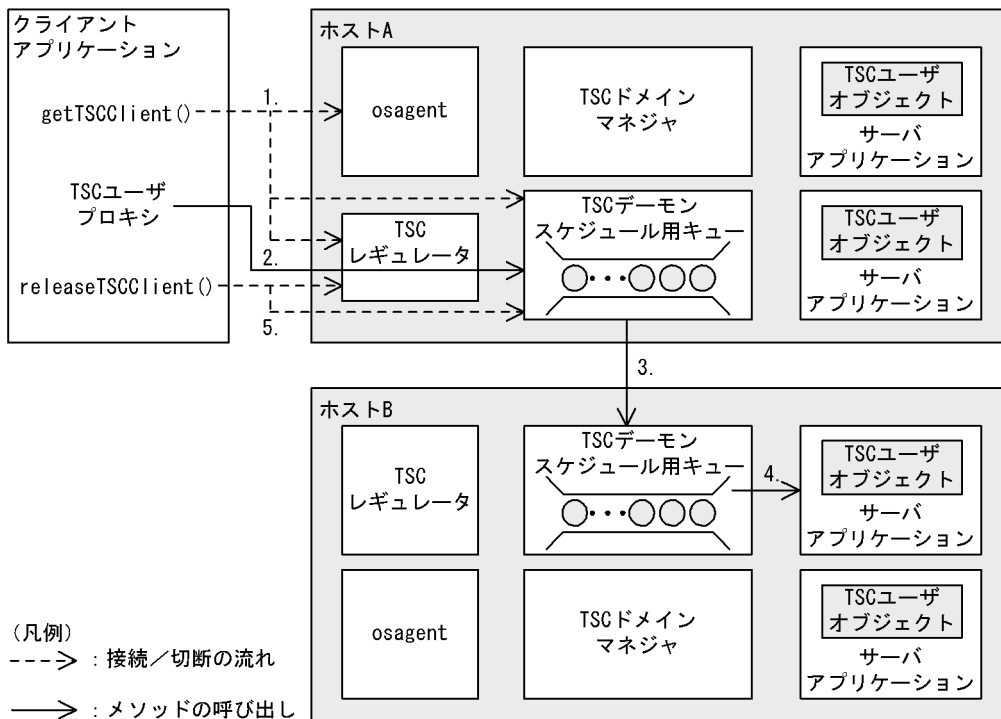
なお、リクエストがスケジュール用キューに登録されてから取り出されるまでの時間がリクエストのタイムアウト値を超えている場合は、非応答型呼び出しを除きリクエストを破棄します。

2.4.2 リクエストの負荷分散

負荷分散は、クラスタ構成などで並列に運用しているサーバアプリケーション間で、負荷が均等になるように処理を分散して割り当て、システム全体の可用性を高める機能です。クライアントアプリケーションからの要求をサーバ間、プロセス間およびスレッド間で負荷分散できます。負荷分散は、スケジュール用キューを共有しているサーバアプリケーション間で実行できるほか、複数の TSC デーモン間で負荷情報を交換することで、異なるスケジュール用キューで制御されているサーバアプリケーションに含まれる TSC ユーザオブジェクトに対しても実行できます。

クライアントアプリケーションからサーバアプリケーションを呼び出す流れと負荷分散のタイミングを、次の図に示します。

図 2-7 クライアントアプリケーションからサーバアプリケーションを呼び出す流れと負荷分散のタイミング



図について説明します。

1. クライアントアプリケーションは、TSCAdm クラスの getTSCClient() メソッドで TSC レギュレータまたは TSC デーモンに接続します。
2. 取得したリファレンスから TSC ユーザプロキシを作成し、TSC ユーザオブジェクトを呼び出します。
3. リクエストを受け付けた TSC デーモンは、スケジュールポリシーに従って TSC ドメイン内の各 TSC デーモンの負荷情報を参照して、負荷の少ない TSC デーモンにリクエストを転送します。
 図の場合は、ホスト B の TSC デーモンに振り分けられました。振り分けられたリクエストはスケジュール用キューに登録されます。
4. TSC デーモンは、使用されていないスレッド上に配置された TSC ユーザオブジェクトの業務処理プログラムを呼び出します。このとき、異常終了したサーバアプリケーションやハングアップしてタイムアウトした TSC ユーザオブジェクトを呼び出すことはありません。
5. TSCAdm クラスの releaseTSCClient() メソッドで TSC レギュレータまたは TSC デーモンとの接続を切断します。

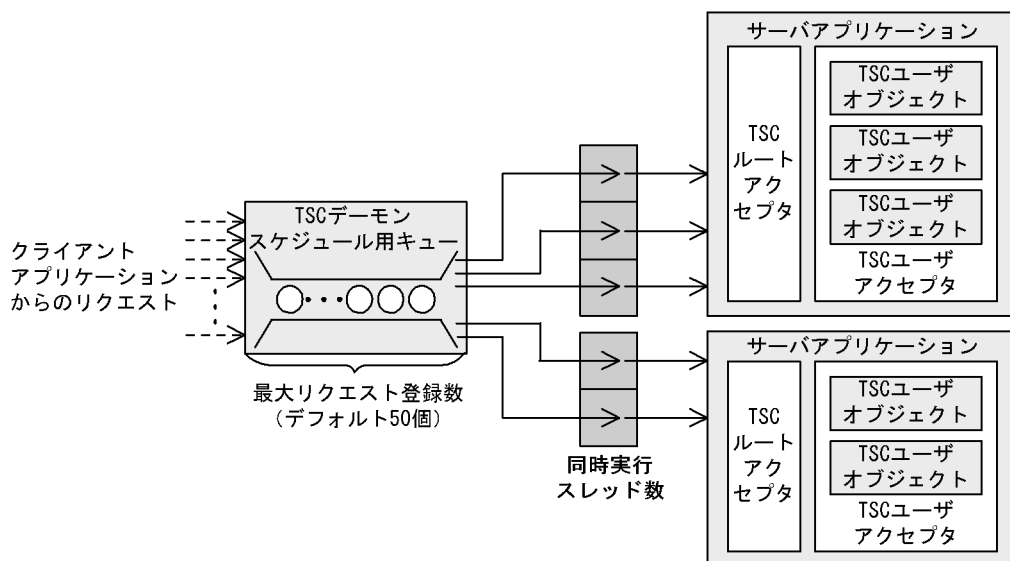
サーバアプリケーションからの応答は、リクエストを受け付けた TSC デーモンを経由して、クライアントアプリケーションに返されます。

2.4.3 リクエストの流量制御

流量制御は、サーバアプリケーションで一度に実行される処理数の最大値を設定して、リクエストの同時実行数を制限する機能です。これによって、サーバアプリケーションの負荷を一定に抑え、安定した高いスループットを実現します。CPU や排他資源の競合も抑止できます。流量制御は、TSC デーモンおよび TSC デーモンが管理しているスケジュール用キューを使用して実現します。

流量制御の概要を次の図に示します。

図 2-8 流量制御の概要



TSC デーモンでは、クライアントアプリケーションから受け付けたリクエストをスケジュール用キューに登録して、TSC ルートアクセプタ単位に設定された同時実行スレッド数分ずつ実行します。クライアントアプリケーションからのリクエストが瞬間的に増加した場合でも、TSC デーモンによって流量が制御されるため、サーバアプリケーションで実行されるリクエストは同時実行スレッド数以上には増加しません。また、複数のサーバアプリケーションの TSC ルートアクセプタで同じスケジュール用キューを共有している場合は、その TSC ルートアクセプタ数および各 TSC ルートアクセプタの同時実行スレッド数の設定で、一度に処理できる TSC ユーザオブジェクトを多重化できます。リクエストは、スケジュール用キューの最大リクエスト登録数分まで受け付けられます。最大リクエスト登録数は、TSC デーモン単位で設定できます。これを超えると、エラーが返されます。

2.4.4 リクエストの閉塞制御

閉塞制御（サービス閉塞）は、特定の TSC ルートアクセプタに対するリクエストの受け付けを停止したり、リクエストを滞留させたりする機能です。これによって、システム全体を停止させないで TSC ユーザオブジェクトの入れ替えや再起動を可能にして、システムの可用性を高められます。

閉塞制御でできることは、次のとおりです。

サーバアプリケーションの閉塞制御

スケジュール用キューに滞留したリクエストを破棄して、サーバアプリケーションを終了させます。

スケジュール用キューの閉塞制御

スケジュール用キューをすぐに閉塞させます。キューに登録済みのリクエストを破棄するかどうかを選択できます。

サーバアプリケーション異常終了時のリクエスト保持

サーバアプリケーション異常終了時にスケジュール用キューのリクエストを一定時間保持します。

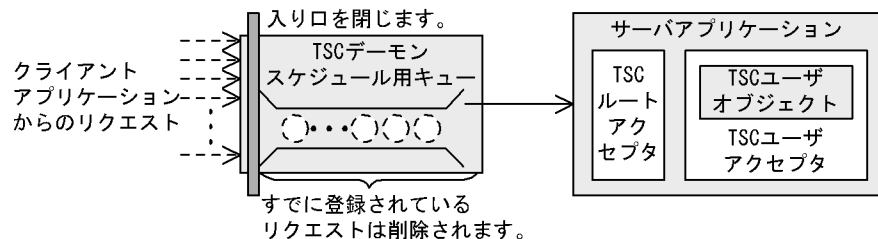
(1) サーバアプリケーションの閉塞制御

TSC ルートアクセプタを非活性化するとき、スケジュール用キューに登録されたリクエストを破棄して、TSC ユーザオブジェクトを停止できます。キューを共有する最後の TSC ルートアクセプタが停止する場合、TSC デモンは、スケジュール用キューの入口を閉じてサービスを停止して、それ以上リクエストを受け付けないようにします。

サーバアプリケーションの閉塞制御は、`tscstopprc` コマンドで指定できます。

サーバアプリケーションの閉塞制御の概要を次の図に示します。

図 2-9 サーバアプリケーションの閉塞制御の概要



(2) スケジュール用キューの閉塞制御

スケジュール用キューに対して、直接閉塞を実行することもできます。これによって、複数の TSC ルートアクセプタでスケジュール用キューが共有されている場合に、一度に複数の TSC ルートアクセプタを停止できます。スケジュール用キューに登録されている

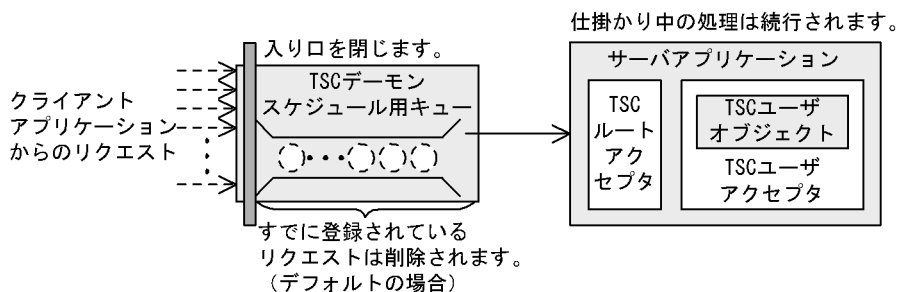
リクエストについては、破棄するか、一定の時間処理を続けるかを選択できます。処理を続ける場合は、一定の時間内に処理ができなければ強制的に破棄するように、タイムアウト時間が指定できます。また、仕掛かり中のリクエストについては、処理が続行されます。

スケジュール用キューの閉塞が指示されると、TSC デーモンは、スケジュール用キューの入り口を閉じてサービスを停止して、それ以上リクエストを受け付けないようにします。また、すでにスケジュール用キューに登録されたリクエストは、設定に従って、破棄するか、または処理を実行してからスケジュール用キューの閉塞を完了します。リクエストを破棄する場合は、キューに登録されていたリクエストの処理はすべてエラーとしてクライアントアプリケーションに返されます。処理を実行してから閉塞する場合は、一定時間処理を継続して、時間内に終了しなかった処理がエラーとして返されます。

スケジュール用キューの閉塞制御は、`tscholdra` コマンドで閉塞を、`tsclresra` コマンドで閉塞解除を指定できます。

スケジュール用キューの閉塞制御の概要を次の図に示します。

図 2-10 スケジュール用キューの閉塞制御の概要



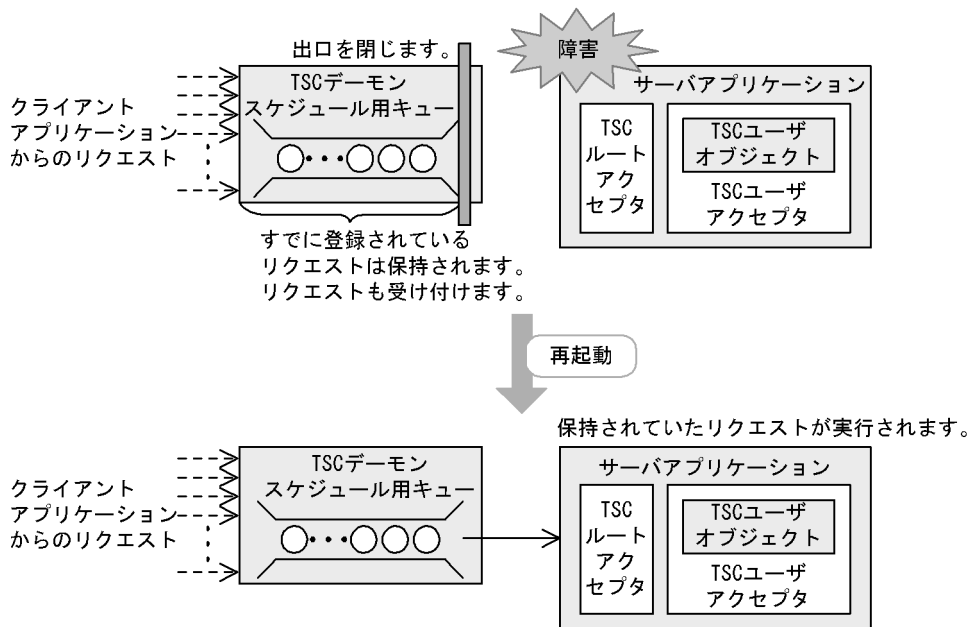
(3) サーバアプリケーション異常終了時のリクエスト保持

サーバアプリケーション異常終了時に、スケジュール用キューのリクエストを一定時間保持します。これによって、サーバアプリケーションが異常終了した場合でも、すぐにユーザにエラーは返されません。さらに、サーバアプリケーションが再起動するまでの間、クライアントアプリケーションからのリクエストは受け付け続けます。リクエストは、スケジュール用キューの最大リクエスト登録数分まで受け付けられます。このため、サーバアプリケーションに障害が発生した場合でも、すぐに再起動すれば、クライアントアプリケーションに障害を気づかせないで運用を続けられます。ただし、リクエストがスケジュール用キューの最大リクエスト登録数を超えた場合は、クライアントアプリケーションにエラーが返されます。

サーバアプリケーション異常終了時のリクエスト保持は、`tsctest` コマンドの `-TSCQueueDeleteWait` オプションで指定できます。

サーバアプリケーション異常終了時のリクエスト保持の概要を次の図に示します。

図 2-11 サーバアプリケーション異常終了時のリクエスト保持の概要



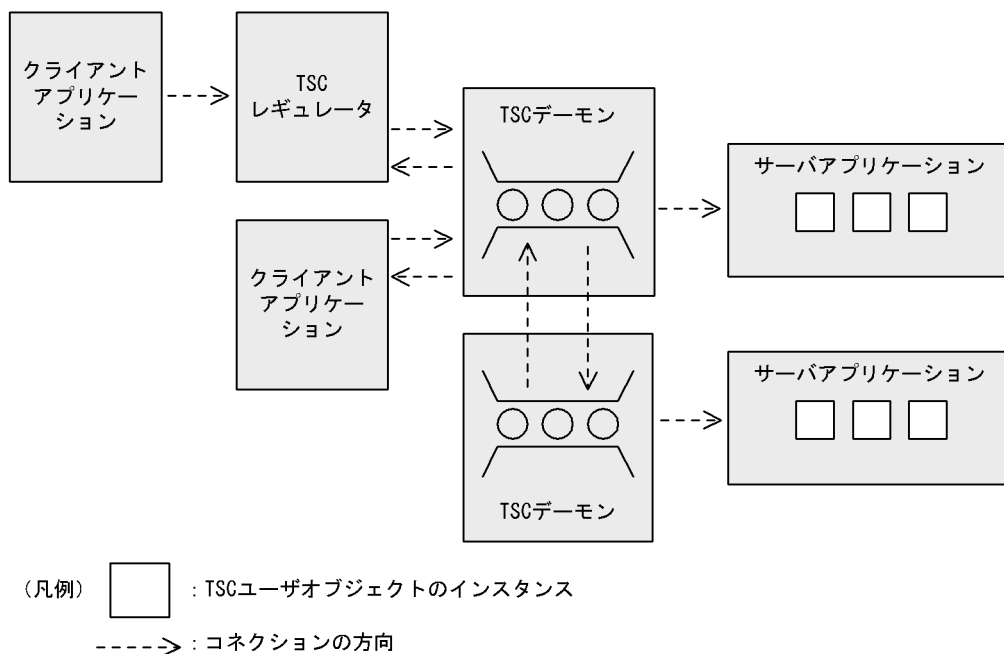
2.4.5 プロセスの配置

サーバアプリケーションと TSC レギュレータは、TSC デーモンと同じコンピュータに配置してください。ほかのコンピュータにサーバアプリケーションや TSC レギュレータを配置する場合は、該当するコンピュータにも TSC デーモンを配置してください。クライアントアプリケーションは、TSC デーモンとは別のコンピュータに配置できます。ただし、OTM - Client のクライアントアプリケーションでない場合、同じコンピュータに TSC デーモンが必要です。

また、複数の TSC デーモンを一つの TSC ドメイン内に配置できます。

TSC デーモンを中心とする各プロセスの配置を次の図に示します。

図 2-12 TSC デーモンを中心とする各プロセスの配置



2.4.6 プロセスの管理

(1) TSC デーモンが管理するプロセス

各プロセスの開始時に `-TSCDomain` オプションおよび `-TSCID` オプションによって指定された TSC デーモンは、各プロセスが正常終了または異常終了するまで、各プロセスを登録および管理します。

TSC デーモンが管理するプロセスを次に示します。

- サーバアプリケーション
- クライアントアプリケーション（OTM・Clientのもの、および `-TSCWithSystem` オプションに "0" を指定して開始したものは除く）
- TSC レギュレータ
- TSCORB コネクタ
- TSC システムプロセス（TSC デーモン開始時に自動開始されるプロセス）

これらのプロセスは、管理元の TSC デーモンと同じホストになければいけません。

(2) プロセスを管理する内容

TSC デーモンは、次のように各プロセスを管理します。

- 各プロセスのステータス、プロセス ID、およびアプリケーション識別子などプロセス

2. 機能

についての情報を保持します。

TSC デーモンが管理する各プロセスの情報を参照する場合は、`tscprcls` コマンドを使用してください。

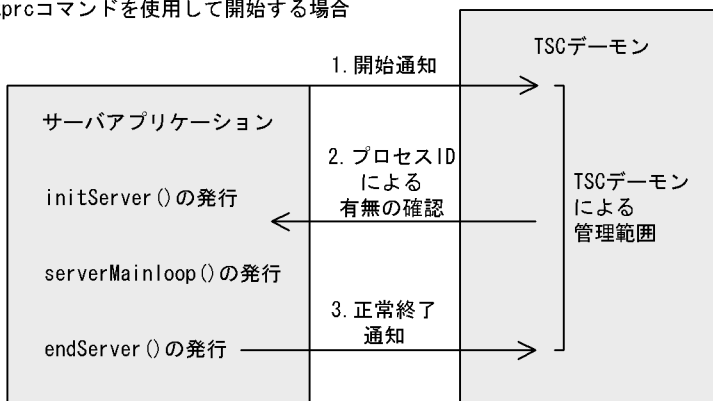
- 各プロセスの有無を確認し、異常終了を検出します。
- TSC システムプロセスの異常終了時に再開始します。
- TSC デーモンの正常終了または強制終了時に登録済みのプロセスがある場合、それらを TSC デーモンと同期を取りながら終了します。
- TSC デーモンの異常終了時に登録済みのプロセスがある場合、それらを次回の TSC デーモン開始時に強制終了します。

2.4.7 アプリケーションプログラムの管理

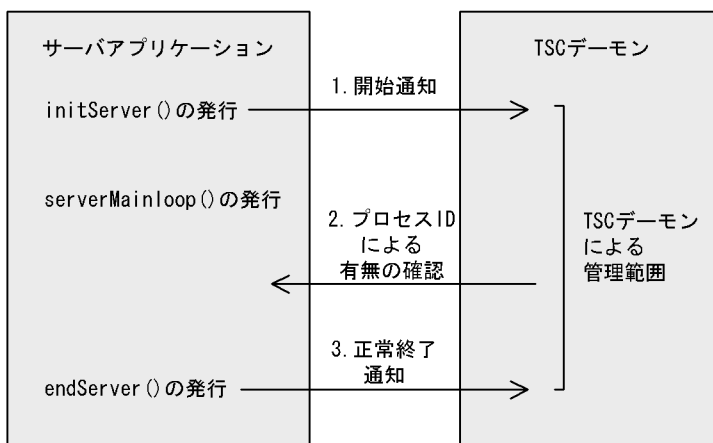
TSC デーモンによるサーバアプリケーションの登録と管理の流れを次の図に示します。

図 2-13 TSC デモンによるサーバアプリケーションの登録と管理の流れ

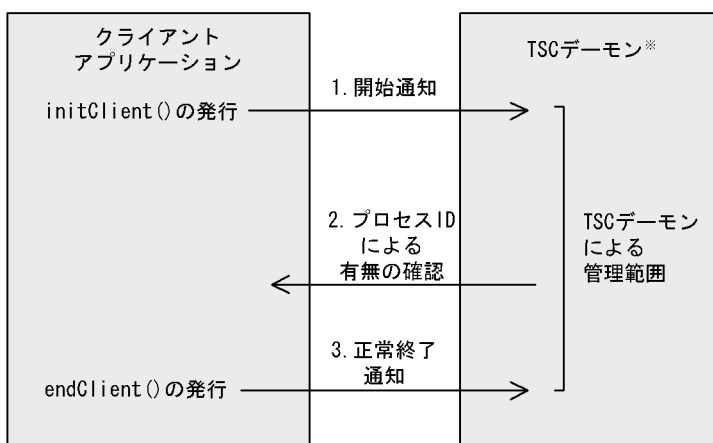
- tscstartprcコマンドを使用して開始する場合



- tscstartprcコマンドを使用しないで開始する場合



- クライアントアプリケーションの場合



注

クライアントアプリケーションおよびサーバアプリケーションには、管理元の TSC デーモンに加えて、接続対象の TSC デーモンがあります。接続対象の TSC デーモンは、リクエストを送受信するための TSC デーモンです。

接続対象の TSC デーモンとして、管理元の TSC デーモンと同じ TSC デーモンを選択することも、管理元の TSC デーモンとは異なる TSC デーモンを選択することもできます。

図中の番号に沿って、TSC デーモンの処理を説明します。

1. サーバアプリケーションからの開始通知を受けて、TSC デーモンはサーバアプリケーションを登録し、管理を開始します。開始通知が発生するのは、`tscstartprc` コマンドを使用する場合はサーバアプリケーションプロセスの生成時です。`tscstartprc` コマンドを使用しない場合は `initServer()` の発行時です。
2. 管理中、サーバアプリケーションプロセスがあることを TSC デーモンはプロセス ID によって確認します。
3. サーバアプリケーションからの終了通知を受けると、TSC デーモンはサーバアプリケーションが正常終了したもとして登録を削除し、管理を終了します。

2.4.8 セッション呼び出しインターバル時間監視

セッション呼び出し処理中に、クライアントの障害などで次のリクエストを受け取れない場合、TSC ユーザプロキシと対応づけられている TSC ユーザオブジェクトがセッション呼び出し待ち状態のままになってしまうことがあります。このような状況を防ぐため、セッション呼び出しインターバル時間監視機能は、セッション呼び出し待ちの TSC ユーザオブジェクトが応答を返してから、次のリクエストまたはセッション呼び出しの終了要求を受け取るまでの時間（最大時間間隔）を監視します。

セッション呼び出しインターバル時間監視の対象になる時間を次に示します。

- TSC ユーザオブジェクトが、セッション呼び出し開始要求に対する応答を返してから次のリクエストを受け取るまでの時間
- TSC ユーザオブジェクトが、リクエストに応答を返してから次のリクエストを受け取るまでの時間
- TSC ユーザオブジェクトが、リクエストに応答を返してからセッション呼び出し終了要求を受け取るまでの時間

(1) セッション呼び出しインターバル監視時間がタイムアウトになった場合の動作

セッション呼び出しインターバル監視時間がタイムアウトになった場合は、`TSCRootAcceptor` クラスの `deactivate()` メソッドと同じ処理をして、セッション呼び出し待ち状態の TSC ユーザオブジェクトが使用されないようにします。

(2) セッション呼び出しインターバル監視時間の指定方法

セッション呼び出しインターバル監視時間の指定には、次の二つの方法があります。

1. クライアントアプリケーションの開始時に `-TSCSessionInterval` オプションにセッション呼び出しインターバル監視時間を指定する方法
2. `TSCSessionProxy` クラスの API でセッション呼び出しインターバル監視時間を指定する方法

1 と 2 の両方に指定している場合は、2 の指定値が設定されます。

(3) セッション情報の削除

セッション呼び出し中のアプリケーションおよび OTM プロセスが異常終了した場合、または接続が切断された場合は、TSC デモンにあるセッション情報を削除します。削除対象になるのは、異常終了したプロセスと関連するセッション情報です。

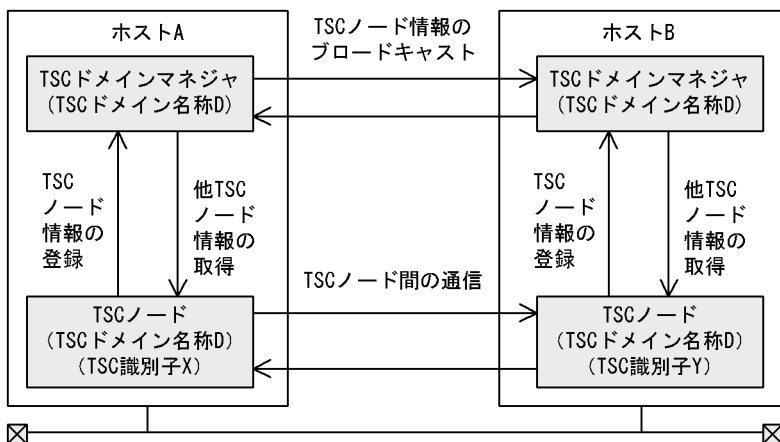
2.5 TSC ドメインマネージャの機能

TSC ドメインマネージャは、同じ TSC ドメイン内の TSC ノードの情報を管理するデーモンプロセスです。

TSC ノードは互いの負荷情報を連絡し、同じ TSC ドメイン内の負荷の低い TSC ノードに、クライアントアプリケーションのリクエストを振り分けます。そのために、TSC ドメインマネージャは、同一ホストの TSC ドメインに属する TSC ノードについての情報を管理し、他ホストの TSC ドメインマネージャにブロードキャストしています。この情報を TSC ノード情報といいます。TSC ノードは、TSC ドメインマネージャから他 TSC ドメインの TSC ノード情報を取得し、TSC ノード間の通信に使用します。

TSC ノード間での通信の概要を次の図に示します。

図 2-14 TSC ノード間での通信の概要



TSC ノード情報のブロードキャスト間隔は、`tscdmstart` コマンドの `-TSCSendInterval` オプションで変更できます。

また、複数のホストで同じ TSC ドメイン名称を持つ TSC ドメインマネージャを開始する場合は、`tscdmstart` コマンドの `-TSCPort` オプションに同じポート番号を指定する必要があります。

さらに、一つのホスト上で複数の TSC ドメインマネージャを開始するためには、`TSCPOOL` 環境変数を TSC ドメインごとに設定する必要があります。詳細については、「4.1.3 環境変数による TSC ドメインの管理」と 5 章の「`tscdminfo` (TSC ドメイン情報の表示と削除)」を参照してください。

2.5.1 TSC ドメインマネージャの開始形態

TSC ドメインマネージャが正常終了しなかった場合の再開には、管理情報を初期化して

再開する方法が、または終了前の管理情報を引き継いで再開する方法があります。TSC ドメインマネージャの開始形態を次に示します。

正常開始

新しく TSC ドメインマネージャを開始する場合、または前回に TSC ドメインマネージャが正常終了している場合は、正常開始します。

前回に TSC ドメインマネージャが強制終了または異常終了している場合は、前回の管理情報を引き継いで TSC ドメインマネージャの再開を試みます。前回の管理情報を正常に引き継げたときは、終了前のシステム状態を復元できます。前回の管理情報が破壊され、正常に引き継げなかったときは、TSC ドメインマネージャを再開できません。

強制正常開始

前回に TSC ドメインマネージャが強制終了または異常終了している場合に、前回に管理していた TSC デモンを強制終了し、管理情報を初期化して TSC ドメインマネージャを再開します。このとき、TSC デモンは再開されません。前回に TSC ドメインマネージャが強制終了または異常終了している場合に、TSC ドメインマネージャを強制正常開始するには、`tscdmstart` コマンドに `-TSCForceStart` オプションを指定します。

自動強制開始

前回に TSC ドメインマネージャが強制終了または異常終了している場合に、前回の管理情報を引き継いで TSC ドメインマネージャを再開します。前回の管理情報を正常に引き継げたときは、終了前のシステム状態を復元できます。前回の管理情報が破壊され、正常に引き継げなかったときは、管理情報を初期化して強制正常開始します。前回に TSC ドメインマネージャが強制終了または異常終了している場合に、TSC ドメインマネージャを自動強制開始するには、`tscdmstart` コマンドに `-TSCAutoForce` オプションを指定します。

2.6 複数のネットワークセグメントで構成する TSC ドメイン

同じ TSC ドメイン名称の TSC ドメインマネージャを異なるネットワークセグメントに配置して、複数のネットワークセグメントで TSC ドメインを構成し、TSC ノード間でリクエストを転送できます。この節では、その仕組みとシステム構成例について説明します。

OTM のネットワークセグメントは、`tscdmstart` コマンドの `-TSCSubnetMask` オプションによって決まります。

2.6.1 TSC ノード情報の配布

TSC ノードによるリクエストの転送がネットワークセグメントを越えるためには、TSC ドメインマネージャの TSC ノード情報がネットワークセグメントを越えることが前提となります。ブロードキャストはネットワークセグメントを越えられませんが、異なるネットワークセグメントのホストにある TSC ドメインマネージャに TSC ノード情報を配布することができます。

TSC ノード情報の配布に必要な設定を次に示します。

相手ホストの名称または IP アドレスの指定

TSC ドメインマネージャの開始時に `tscdmstart` コマンドの `-TSCSendHost` オプションで、相手ホストの名称または IP アドレスを指定します。この指定によって、相手ホストの TSC ドメインマネージャは、自 TSC ドメインで管理する TSC ノード情報を自動的に返します。ただし、相手ホストの TSC ドメインマネージャは OTM 02-00 以降でなければいけません。

TPBroker のスマートエージェントの設定

TSC ドメインマネージャは、`tscdmstart` コマンドの `-TSCSendHost` オプションで指定されたホストにある TSC ドメインマネージャを検索するために、TPBroker のスマートエージェント (`osagent`) を使用します。異なるローカルネットワーク上のスマートエージェントを相互に接続するよう設定してください。この設定の詳細については、マニュアル「[Borland Enterprise Server VisiBroker デベロッパーズガイド](#)」を参照してください。

2.6.2 複数のネットワークセグメントで TSC ドメインを構成する場合の注意事項

TSC ノード情報を基に各ネットワークセグメントの TSC ドメインマネージャ間で確立されるコネクションは、TSC ノード情報の配布先または配布元の TSC ドメインマネージャが終了するまで切断されません。TSC ノード情報の送信間隔は、`tscdmstart` コマンドの `-TSCSendHostInterval` オプションで指定します。

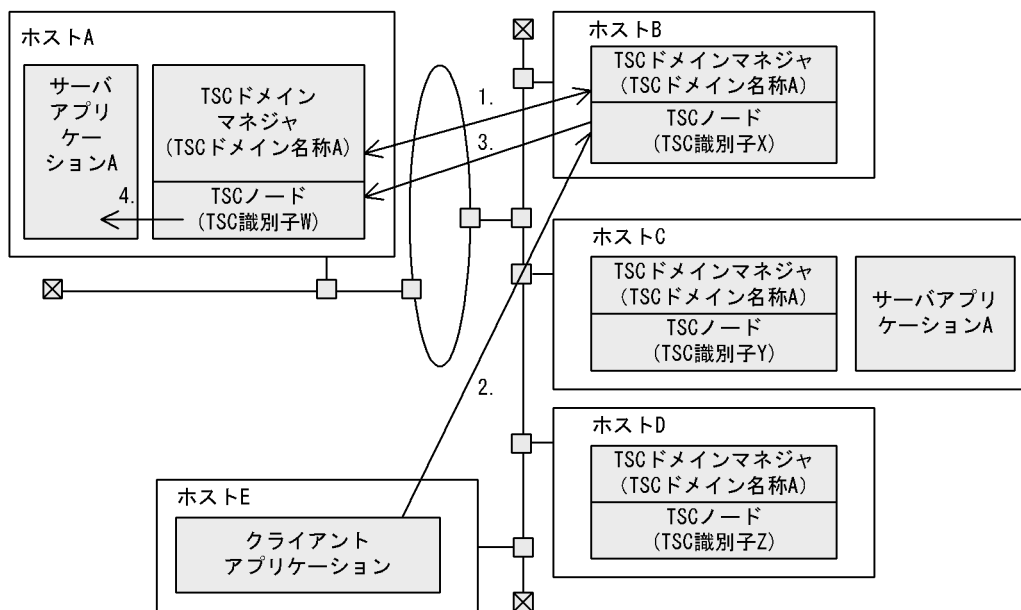
なお、ブロードキャストと同様に、TSC ドメインマネージャ開始直後から TSC ノード情報は配布されます。そのため、ネットワークセグメント間を接続している回線使用料が有料の場合は注意が必要です。

TSC ドメインマネージャは、自ホストと同じネットワークセグメントにある同じ TSC ドメイン名称の TSC ドメインマネージャと、TSC ノード情報を互いにブロードキャストします。そのため、TSC ドメインマネージャは同じネットワークセグメント内の同じ TSC ドメイン名称の TSC ノード情報をすべて保持します。一方、TSC ドメインマネージャは、`tscdmstart` コマンドの `-TSCSendHost` オプションに指定したホストに対しては、すべての TSC ノード情報を転送しません。`-TSCSendHost` オプションに指定したホストに対しては、自 TSC ドメインマネージャが管理している TSC ノード情報だけを配布します。そのため、ネットワークセグメントを越えてリクエストが転送されるようにシステムを構成するには、注意が必要です。

2.6.3 複数のネットワークセグメントでの TSC ドメインの構成例

`tscdmstart` コマンドの `-TSCSendHost` オプションを指定した場合のシステム構成例を以降に示します。本文中の番号は図中の番号と対応しています。

図 2-15 複数のネットワークセグメントで TSC ドメインを構成する例 1

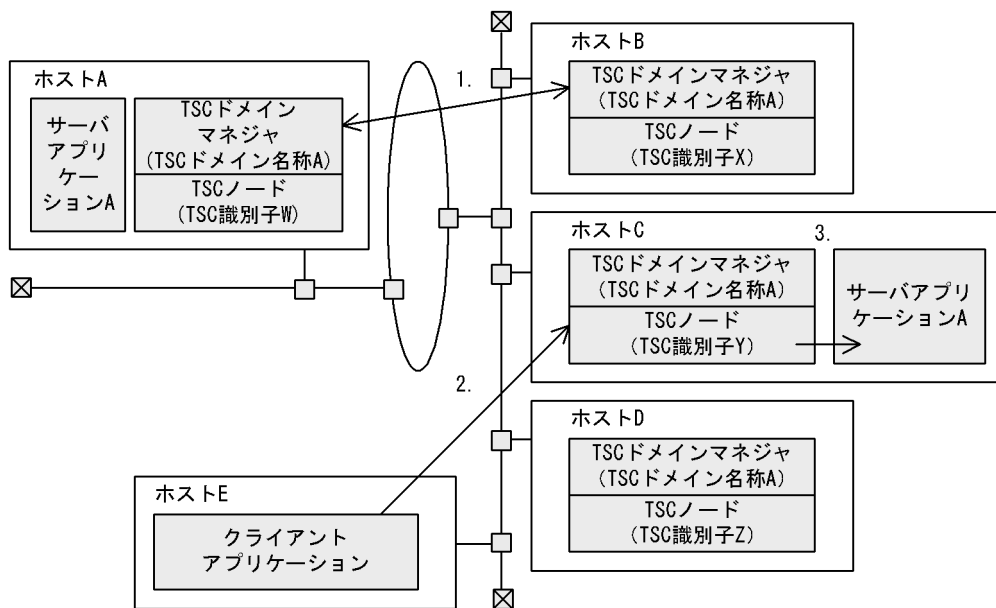


1. `tscdmstart` コマンドの `-TSCSendHost` オプションに Host B のホスト名称を指定して、Host A の TSC ドメインマネージャを開始します。
Host A の TSC ドメインマネージャの TSC ノード情報が配布されます。このとき、ホ

2. 機能

- スト B からは、ホスト B の TSC ドメインマネージャの TSC ノード情報が返されます。
2. ホスト B にある TSC ノードの TSC 識別子を指定して、ホスト E にあるクライアントアプリケーションを開始します。
 3. ホスト E のクライアントアプリケーションが、ホスト B にある TSC ノードに接続した場合、リクエストはホスト A に転送されます。
ホスト B にはサーバアプリケーション A がいないため、TSC ノード間の通信によって、リクエストはホスト A に転送されます。ホスト B からの転送先は負荷状況に応じて選択されるので、ホスト C に転送されることもあります。
 4. ホスト A でサーバアプリケーション A がリクエストを処理します。

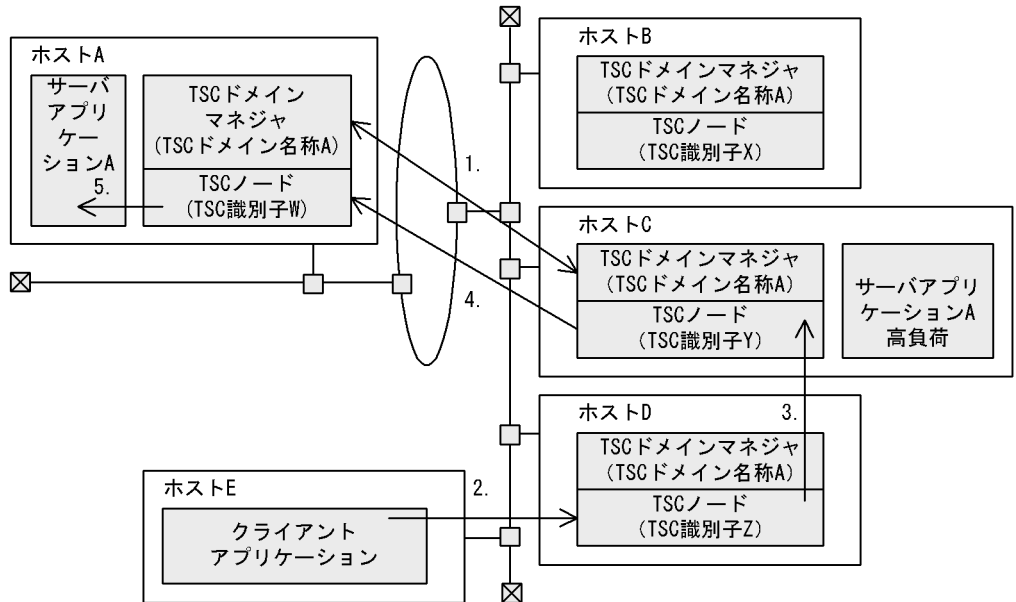
図 2-16 複数のネットワークセグメントで TSC ドメインを構成する例 2



1. `tscdmstart` コマンドの `-TSCSendHost` オプションにホスト B のホスト名称を指定して、ホスト A の TSC ドメインマネージャを開始します。
ホスト A の TSC ドメインマネージャの TSC ノード情報が配布されます。このとき、ホスト B からは、ホスト B の TSC ドメインマネージャの TSC ノード情報が返されます。
2. ホスト C にある TSC ノードの TSC 識別子を指定して、ホスト E にあるクライアントアプリケーションを開始します。
3. ホスト E のクライアントアプリケーションが、ホスト C にある TSC ノードに接続した場合、サーバアプリケーション A はホスト C でリクエストを処理します。
ホスト C にはサーバアプリケーション A があるため、サーバアプリケーション A は

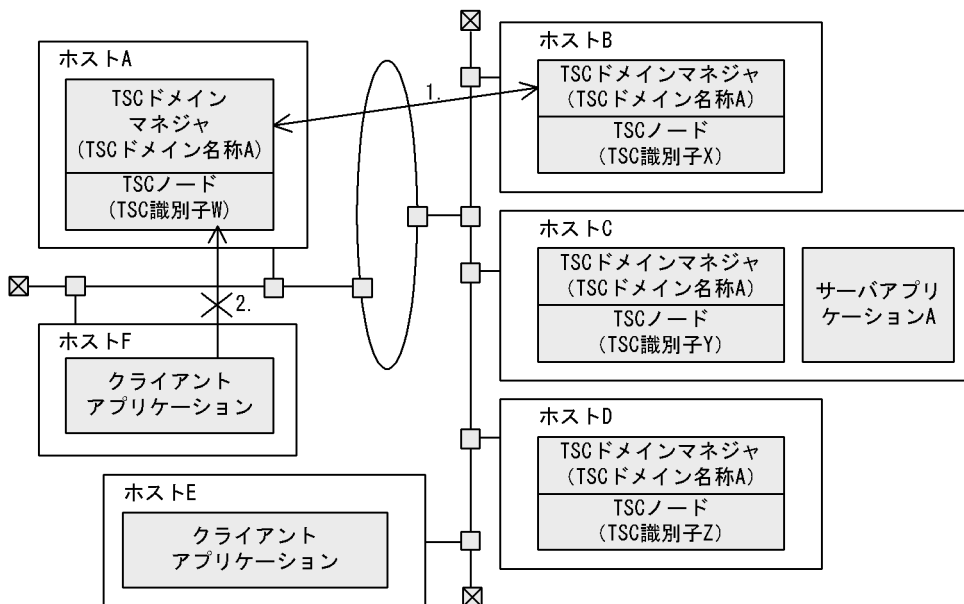
ホストCでリクエストを処理します。ホストCのサーバアプリケーションAの負荷が高い場合でも、ホストAには転送されません。それは、ホストCには、ホストAのTSCノード情報がないためです。

図 2-17 複数のネットワークセグメントで TSC ドメインを構成する例 3



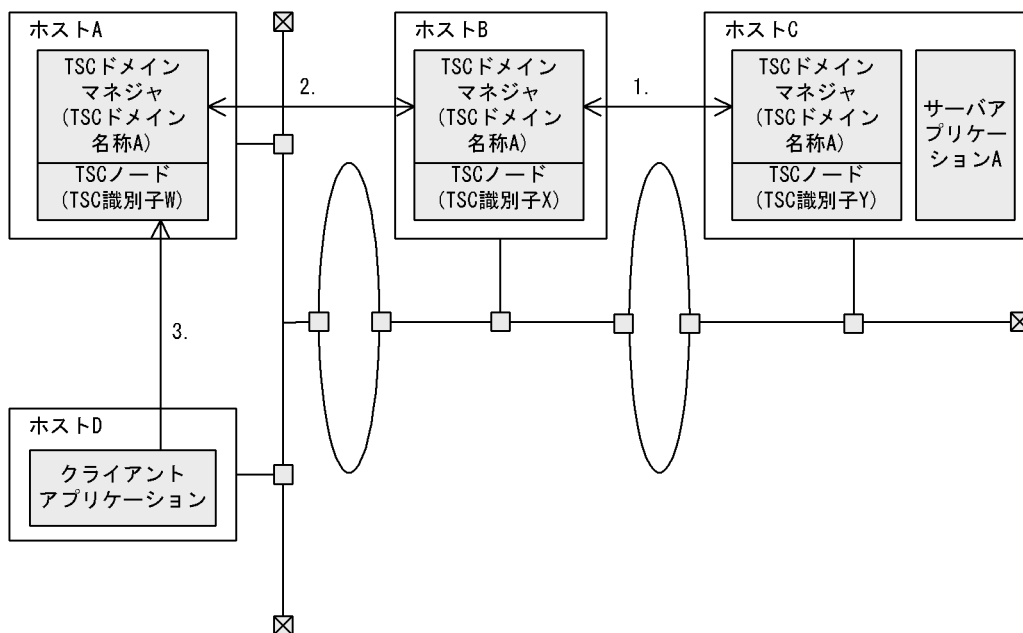
1. `tscdmstart` コマンドの `-TSCSendHost` オプションにホスト C のホスト名称を指定して、ホスト A の TSC ドメインマネージャを開始します。
ホスト A の TSC ドメインマネージャの TSC ノード情報が配布されます。このとき、ホスト C からは、ホスト C の TSC ドメインマネージャの TSC ノード情報が返されます。
2. ホスト D にある TSC ノードの TSC 識別子を指定して、ホスト E にあるクライアントアプリケーションを開始します。
3. ホスト E のクライアントアプリケーションが、ホスト D にある TSC ノードに接続した場合、リクエストはホスト C に転送されます。
ホスト D にはサーバアプリケーション A がないため、TSC ノード間の通信によって、クライアントアプリケーションのリクエストはホスト C に転送されます。
4. リクエストはホスト A に転送されます。
ホスト C のサーバアプリケーション A は負荷が高い状態なので、TSC ノード間の通信によって、クライアントアプリケーションのリクエストはホスト A に転送されます。それは、ホスト C には、1. で配布されたホスト A の TSC ノード情報があるためです。
5. ホスト A でサーバアプリケーション A がリクエストを処理します。

図 2-18 複数のネットワークセグメントで TSC ドメインを構成する例 4



1. `tscdmstart` コマンドの `-TSCSendHost` オプションに Host B の Host 名称を指定して、Host A の TSC ドメインマネージャを開始します。
Host A の TSC ドメインマネージャの TSC ノード情報が配布されます。このとき、Host B からは、Host B の TSC ドメインマネージャの TSC ノード情報が返されます。
2. Host A にある TSC ノードの TSC 識別子を指定して、Host E または Host F にあるクライアントアプリケーションを開始します。
Host E または Host F のクライアントアプリケーションが Host A にある TSC ノードに接続した場合、Host A にはサーバアプリケーション A がないため、リクエストはエラーになります。また、Host A が Host B から取得した TSC ノード情報には、Host C の情報は含まれないので、リクエストは転送されません。

図 2-19 複数のネットワークセグメントで TSC ドメインを構成する例 5



1. `tscdmstart` コマンドの `-TSCSendHost` オプションに Host C の Host 名称を指定して、Host B の TSC ドメインマネージャを開始します。
Host B の TSC ドメインマネージャの TSC ノード情報が配布されます。このとき、Host C からは、Host C の TSC ドメインマネージャの TSC ノード情報が返されます。
2. `tscdmstart` コマンドの `-TSCSendHost` オプションに Host B の Host 名称を指定して、Host A の TSC ドメインマネージャを開始します。
Host A の TSC ドメインマネージャの TSC ノード情報が配布されます。このとき、Host B からは、Host B の TSC ドメインマネージャの TSC ノード情報が返されます。
3. Host A にある TSC ノードの TSC 識別子を指定して、Host D にあるクライアントアプリケーションを開始します。
4. Host D のクライアントアプリケーションが Host A にある TSC ノードに接続した場合、リクエストはエラーになります。
Host B にはサーバアプリケーション A についての情報があります。しかし、Host A が Host B から取得した TSC ノード情報には、Host C の情報は含まれないので、リクエストは転送されません。

2.7 サーバアプリケーションの機能

サーバアプリケーションは、クライアントアプリケーションからのリクエストを処理します。

サーバアプリケーションプロセスは、接続する TSC デーモンと同じコンピュータに配置してください。複数の種類のサーバアプリケーションプロセスを一つの TSC デーモンに接続できます。また、同じサーバアプリケーションを複数プロセス開始して、一つの TSC デーモンに接続できます。

この節では、サーバアプリケーションの開始と終了、サーバアプリケーションからのリクエストの要求、およびサーバアプリケーションを構成するスレッドと TSC ユーザオブジェクトについて説明します。また、サーバアプリケーションが実行するリクエストの処理として、リクエストのスケジュールと振り分けについて説明します。さらに、サーバアプリケーションの時間監視、および状態の検出について説明します。

2.7.1 サーバアプリケーションの開始と終了

サーバアプリケーションの開始方法と、`tscstopprc` コマンドによる終了について説明します。

(1) サーバアプリケーションの開始

サーバアプリケーションプロセスの開始には、シェルなどを使用してください。サーバアプリケーションプロセスを開始するには、次に示す方法があります。

- `tscstartprc` コマンドの `-TSCAPName` オプションにサーバアプリケーションの実行形式ファイル名称を指定して開始する方法
- `tscstartprc` コマンドを使用しないで開始する方法

`tscstartprc` コマンドは、`TSCAdm` クラスの `serverMainloop()` メソッドの発行を確認してリターンします。そのため、複数のプログラムを開始する際に処理の順序性があっても、開始の待ち合わせをして、初期化エラーの発生を防止できます。

通常、`tscstartprc` コマンドを使用してサーバアプリケーションを開始することをお勧めします。

(2) サーバアプリケーションの終了

次に示す場合には、`tscstopprc` コマンドでサーバアプリケーションを終了します。

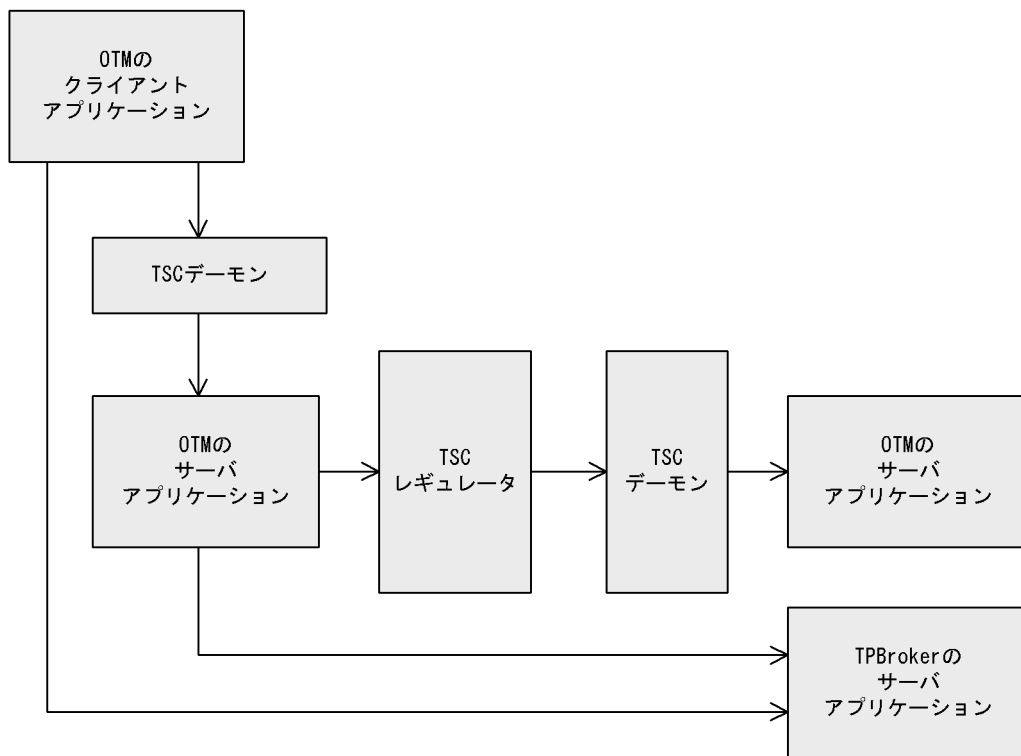
- `tscstartprc` コマンドの `-TSCAPName` オプションにサーバアプリケーションの実行形式ファイル名称を指定して開始した場合
- `-TSCAPID` オプションを指定して `tscstartprc` コマンドを使用しないで開始した場合

2.7.2 サーバアプリケーションからのリクエストの要求

サーバアプリケーションは、クライアントアプリケーションからのリクエストを処理しますが、サーバアプリケーションからもリクエストできます。また、TPBrokerのリクエスト呼び出しもできます。さらに、サーバアプリケーションプロセス内にTPBrokerのサーバオブジェクトを配置して、ほかのプロセスからTPBrokerのリクエスト呼び出しで呼び出すこともできます。

サーバアプリケーションからのリクエストの要求を次の図に示します。

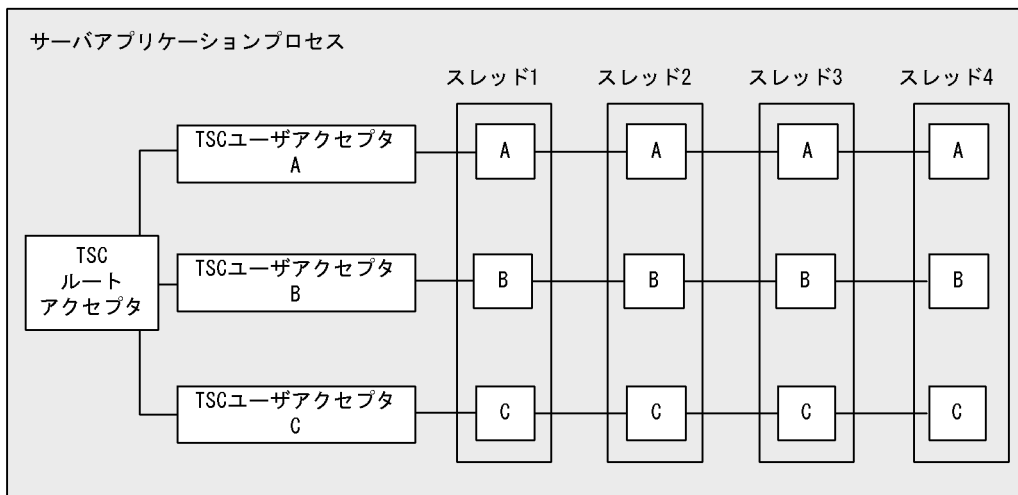
図 2-20 サーバアプリケーションからのリクエストの要求



2.7.3 スレッドと TSC ユーザオブジェクト

サーバアプリケーションプロセスは、スレッドと TSC ユーザオブジェクトとで構成されます。その構成の例を次の図に示します。

図 2-21 スレッドと TSC ユーザオブジェクトの構成



- (凡例)
- TSC ユーザ アクセプタ
A~C : A~Cの各インタフェースに対応するTSCユーザアクセプタ
 - A~C : A~Cの各インタフェースに対応するTSCユーザオブジェクト

サーバアプリケーションプロセス内では、サービスを提供する複数のオブジェクトが動作します。これらの、アプリケーションとして動作するオブジェクトを TSC ユーザオブジェクトといいます。また、サーバアプリケーションごとに、TSC ルートアクセプタのオブジェクトを持ちます。TSC ルートアクセプタは、TSC ユーザアクセプタを管理するオブジェクトです。TSC ユーザアクセプタは、TSC ユーザオブジェクトにリクエストを割り振るオブジェクトです。

(1) ユーザ定義 IDL インタフェースとコード中の登録処理部分の例

サーバアプリケーションのユーザ定義 IDL インタフェースの例、および対応するコード中の登録処理部分の例を次に示します。

ユーザ定義 IDL インタフェースの例

```
interface A{
    void func();
};
interface B {
    void func();
};
interface C {
    void func();
};
```

対応するコード中の登録処理部分

```

//ABC_TSCfactimplの生成
TSCObjectFactory_ptr A_my_obj_fact = new A_TSCfactimpl();
TSCObjectFactory_ptr B_my_obj_fact = new B_TSCfactimpl();
TSCObjectFactory_ptr C_my_obj_fact = new C_TSCfactimpl();

//TSCAcceptorの生成
TSCAcceptor_ptr A_my_acpt= new A_TSCacpt(A_my_obj_fact);
TSCAcceptor_ptr B_my_acpt= new B_TSCacpt(B_my_obj_fact);
TSCAcceptor_ptr C_my_acpt= new C_TSCacpt(C_my_obj_fact);
:

//TSCRootAcceptorの生成
TSCRootAcceptor_ptr my_rt_acpt = TSCRootAcceptor::create(tsc_s);
:

//TSCRootAcceptorへの登録
my_rt_acpt->registerAcceptor(A_my_acpt);
my_rt_acpt->registerAcceptor(B_my_acpt);
my_rt_acpt->registerAcceptor(C_my_acpt);

//TSCRootAcceptorの平行カウント（常駐するスレッド数）指定
my_rt_acpt->setParallelCount(4);

//オブジェクトの活性化
my_rt_acpt->activate("serviceX");
:

```

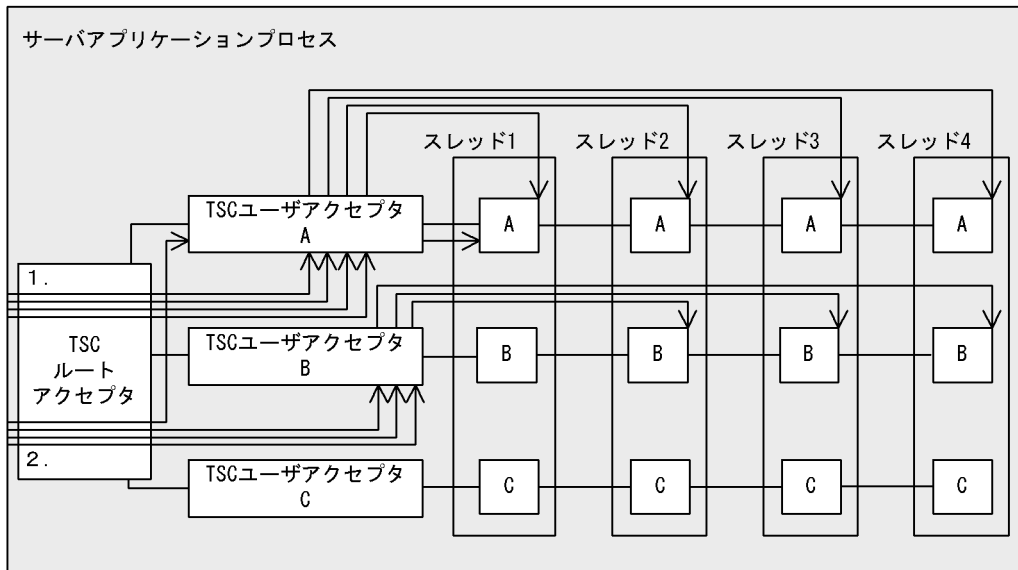
TSCRootAcceptor クラスの activate() メソッドを発行して TSC ルートアクセプタを active 状態にするときに、TSCAcceptor クラスは、TSC ユーザオブジェクトのインスタンスを生成します。このとき、登録された TSCFactoryObject を基に、ユーザ定義 IDL インタフェースの interface 句ごとに生成します。

OTM は、インスタンスとスレッドを対応づけて制御するために、TSCRootAcceptor クラスの setParallelCount() メソッド、またはサーバアプリケーションの開始時に指定する -TSCParallelCount オプションで指定する平行カウントを使用します。また、TSC ユーザオブジェクトのインスタンスは作成されたスレッド上で常に動作するよう制御されます。

(2) マルチスレッド環境での処理

マルチスレッド環境でサーバアプリケーションを運用する場合のプロセスの処理（スレッドの並行実行）の仕組みを次の図に示します。

図 2-22 マルチスレッド環境での処理の仕組み



- (凡例)
- TSC ユーザ アクセプタ
A~C : A~Cの各インターフェースに対応するTSC ユーザ アクセプタ
 - A~C : A~Cの各インターフェースに対応するTSC ユーザ オブジェクト

図 2-22 の場合，TSCRootAcceptor クラスの setParallelCount() メソッド，またはサーバアプリケーションの開始時に指定する -TSCParallelCount オプションには "4" が設定されています。

1. で示す処理では，A のインターフェースに対して同時にリクエストがある場合，四つのスレッド上で同時に TSC ユーザオブジェクトのインスタンスが動作します。リクエスト処理中には，ほかのインターフェースへのリクエストがあっても，それ以上のリクエストは処理されません。
2. で示す処理では，A のインターフェースが一つのスレッド上で動作し，同時に B のインターフェースが三つのスレッド上で動作している場合，C のインターフェースに対するリクエストがあるとき，どれかのスレッドで実行されている処理が終了するまで，C のインターフェースに対するリクエストは処理されません。

つまり，TSC ルートアクセプタに登録されているインターフェースが幾つあっても，1 スレッド上で同時に動作するインスタンスは，各インターフェースの TSC ユーザオブジェクトの一つだけです。

2.7.4 リクエストのスケジュール

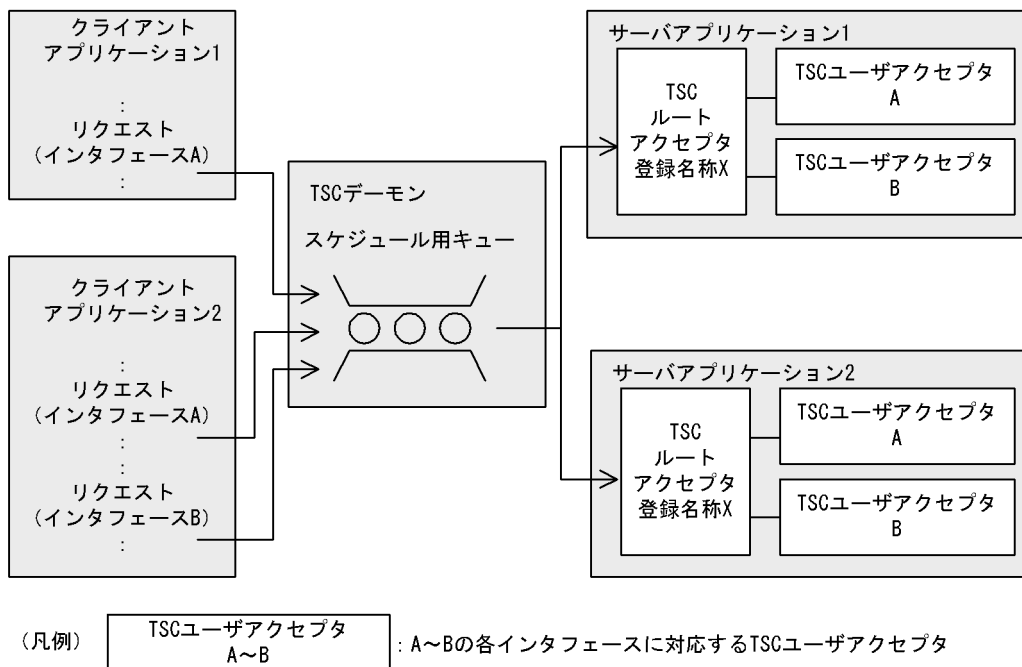
サーバアプリケーションは、リクエストのスケジュールを管理します。スケジュールの管理方法には、スケジュール用キューを共有する方法と共有しない方法があります。

(1) スケジュール用キューを共有する方法

同じ TSC デーモンに対応づけられた複数のサーバアプリケーションで、同じ TSC ルートアクセプタ登録名称を指定すると、スケジュール用キューを共有できます。

スケジュール用キューを共有するスケジュール方法を次の図に示します。

図 2-23 スケジュール用キューを共有するスケジュール方法



TSC ルートアクセプタ登録名称をサーバアプリケーション 1 でもサーバアプリケーション 2 でも "X" になるよう構成すると、複数のインタフェースをまとめて一つのスケジュール用キューでスケジュールしてサーバアプリケーションに振り分けます。ただし、同じ TSC デーモンに対して `getTSCServer()` を発行するサーバアプリケーションを対応づける必要があります。

(2) スケジュール用キューを共有しない方法

それぞれ異なる TSC デーモンに対してサーバアプリケーションを対応づけると、同じ TSC ルートアクセプタ登録名称を指定しても、異なるスケジュール用キューにスケジュールされます。また、同じ TSC デーモンに対してサーバアプリケーションを対応付けていても、異なる TSC ルートアクセプタ登録名称を指定すると、異なるスケジュール

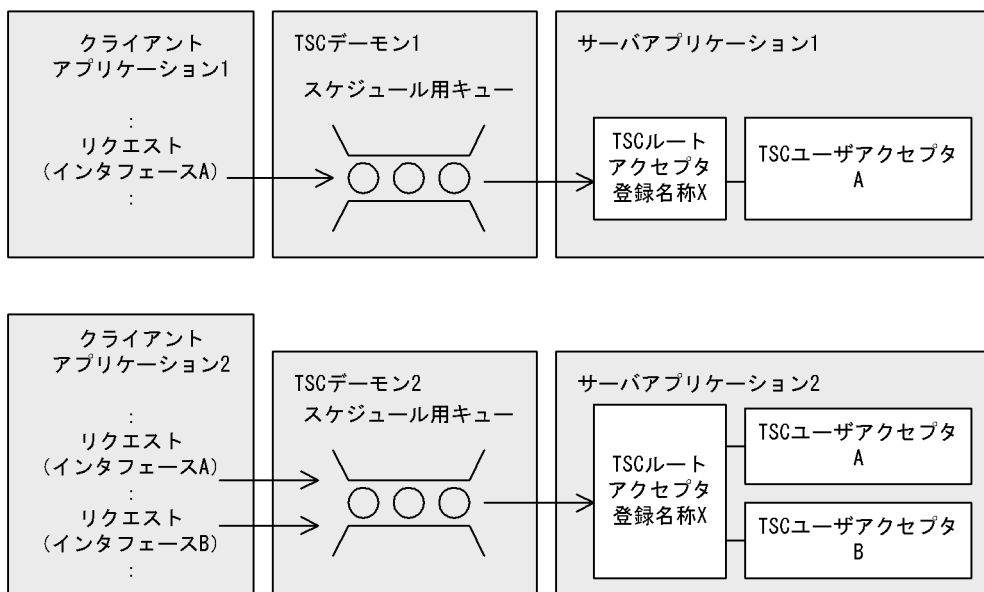
2. 機能

用キューにスケジュールされます。

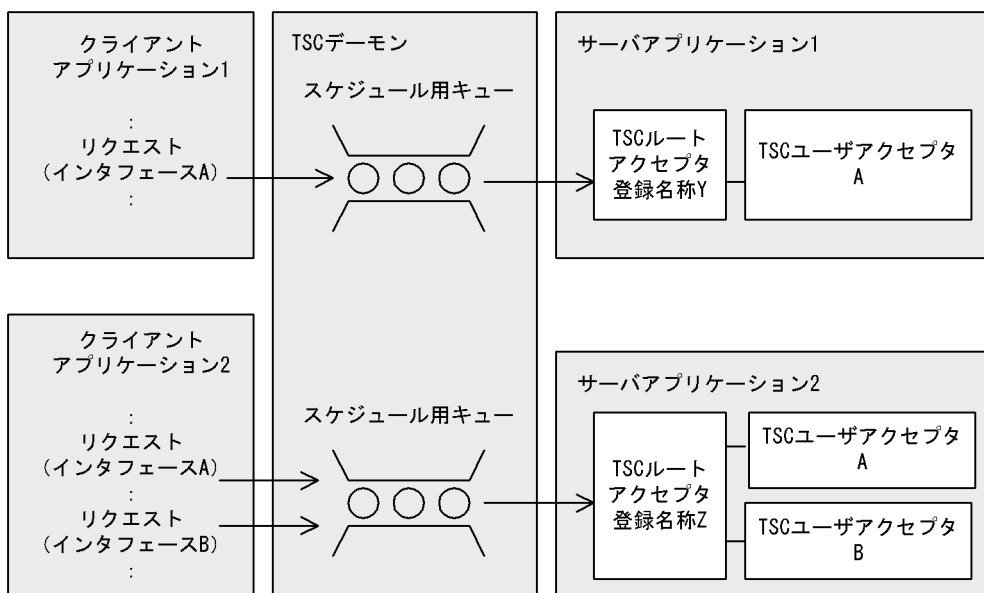
スケジュール用キューを共有しないスケジュール方法を次の図に示します。

図 2-24 スケジュール用キューを共有しないスケジュール方法

- 異なるTSCデーモンに対してサーバアプリケーションを対応付ける場合



- 異なるTSCルートアクセプタ登録名称を指定する場合



(凡例) TSCユーザアクセプタ A~B : A~Bの各インタフェースに対応するTSCユーザアクセプタ

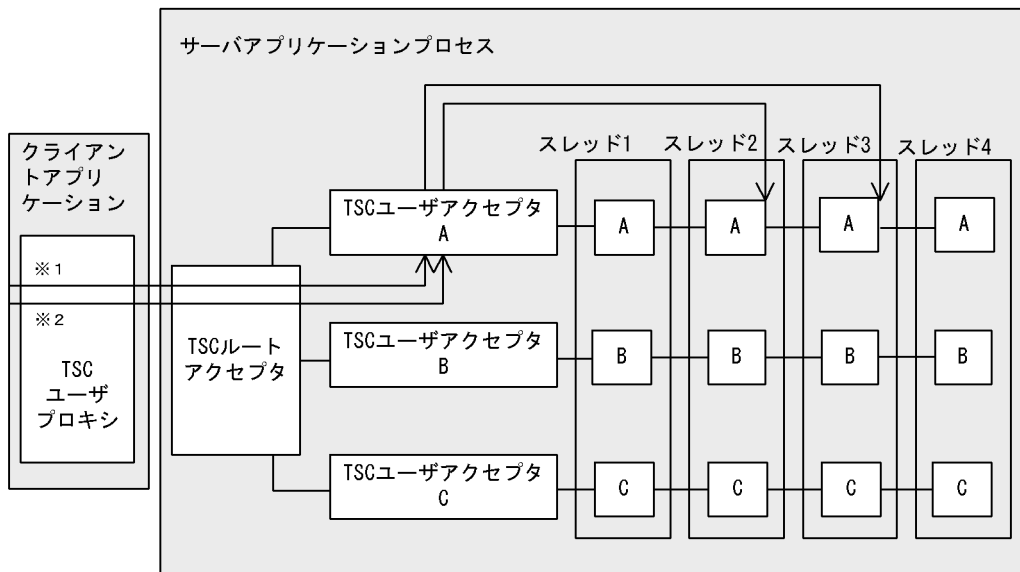
2.7.5 リクエストの振り分け

サーバアプリケーションは、リクエストの振り分けをします。リクエストを振り分ける方法には、TSC アクセプタ名称を指定する方法と指定しない方法があります。

(1) TSC アクセプタ名称を指定しないリクエストの振り分け

TSC アクセプタ名称を指定しない場合のリクエストの振り分けを次の図に示します。

図 2-25 TSC アクセプタ名称を指定しない場合のリクエストの振り分け



(凡例) TSC ユーザアクセプタ
A~C : A~Cの各インターフェースに対応するTSC ユーザアクセプタ

A~C : A~Cの各インターフェースに対応するTSC ユーザオブジェクト

注 1

A のインターフェースを持つインスタンスに対する 1 回目のリクエストです。

注 2

A のインターフェースを持つインスタンスに対する 2 回目のリクエストです。

TSC ユーザプロキシは、クライアントアプリケーションが TSC ユーザオブジェクトを呼び出すための代理オブジェクトです。

使用する TSC ユーザプロキシは、1 回目、2 回目とも同じオブジェクトを使用します。1 回目のリクエストでは、スレッド 3 上で動作するインスタンスを実行してクライアントアプリケーションへ返します。続いて A のインターフェースに対して 2 回目のリクエスト

が実行された場合、スレッド2が実行されます。つまり、同じインタフェースに対してリクエストした場合でも、同じスレッド上で動作するという保証はありません。また、同じインタフェースを持つほかのサーバアプリケーションプロセスがある場合、複数のリクエストをすると、OTMの負荷状況に応じて処理するサーバアプリケーションプロセスが振り分けられます。そのため、リクエストを処理するサーバアプリケーションプロセスおよびスレッドが常に同じになるという保証はありません。

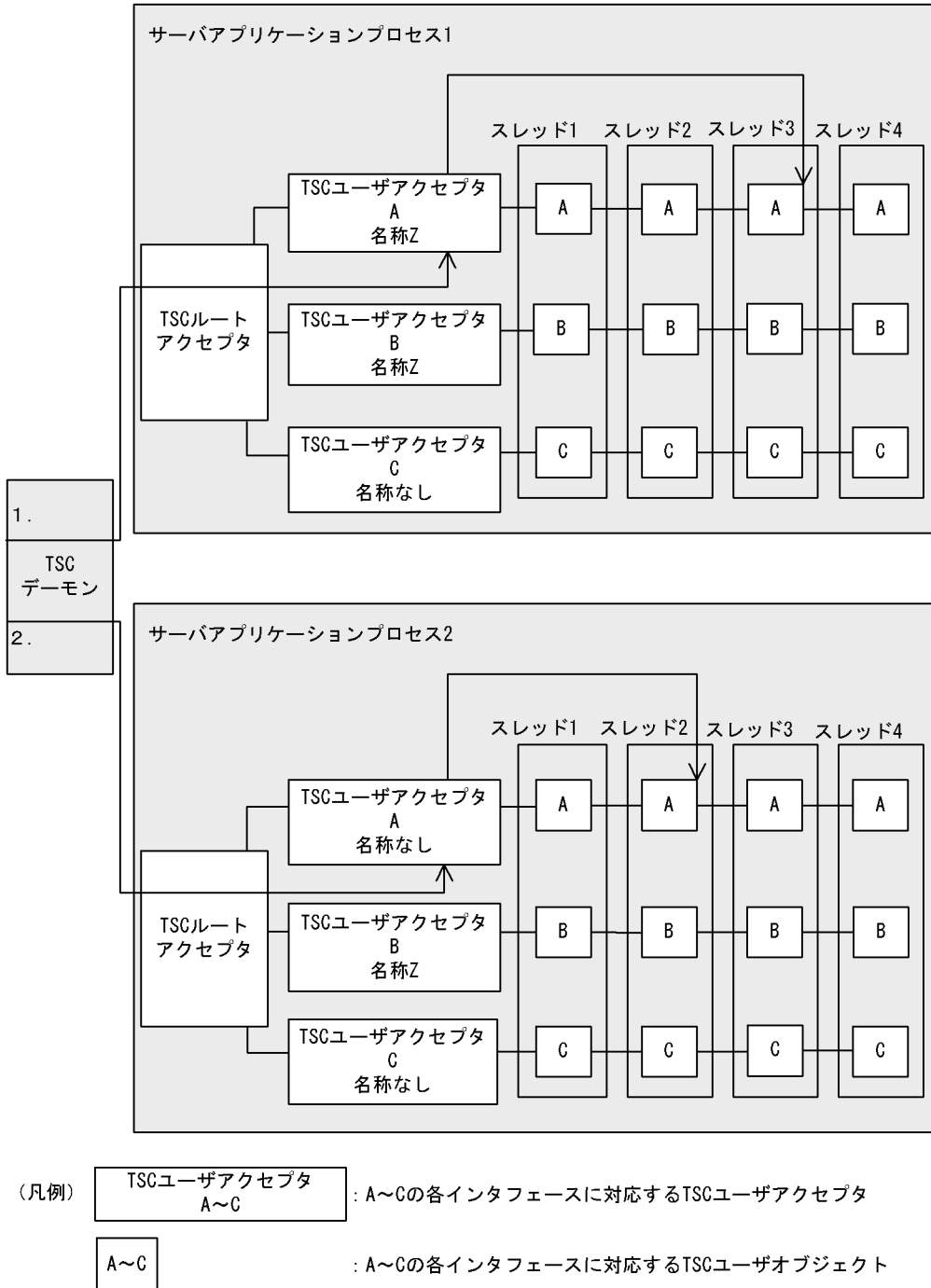
(2) TSC アクセプタ名称を指定するリクエストの振り分け

TSC ユーザアクセプタの生成時に TSC アクセプタ名称を指定できます。同じインタフェースでも TSC アクセプタ名称を使用することによって、異なるインタフェースとして扱うことができます。

TSC アクセプタ名称を指定する場合のリクエストの振り分けを次の図に示します。

2. 機能

図 2-26 TSC アクセプタ名称を指定する場合のリクエストの振り分け



この例では、サーバアプリケーションプロセス 1 にある A のインタフェースを担当する TSC アクセプタ名称を "Z" とし、サーバアプリケーションプロセス 2 にある A のインタ

フェースを担当する TSC ユーザアクセプタの名称はないものとします。

1. で示す処理では、クライアントアプリケーションから A のインタフェースの TSC アクセプタ名称 Z でリクエストするとき、サーバアプリケーションプロセス 1 内の A のインタフェースを担当する TSC アクセプタ名称 Z に対してリクエストします。TSC アクセプタ名称を指定する場合、名称がない TSC ユーザアクセプタ、および TSC アクセプタ名称が異なる TSC ユーザアクセプタにはリクエストは振り分けられません。

この場合でもリクエスト単位に実行されるインタフェースについて、毎回同じ TSC ユーザオブジェクトが実行される保証はありません。

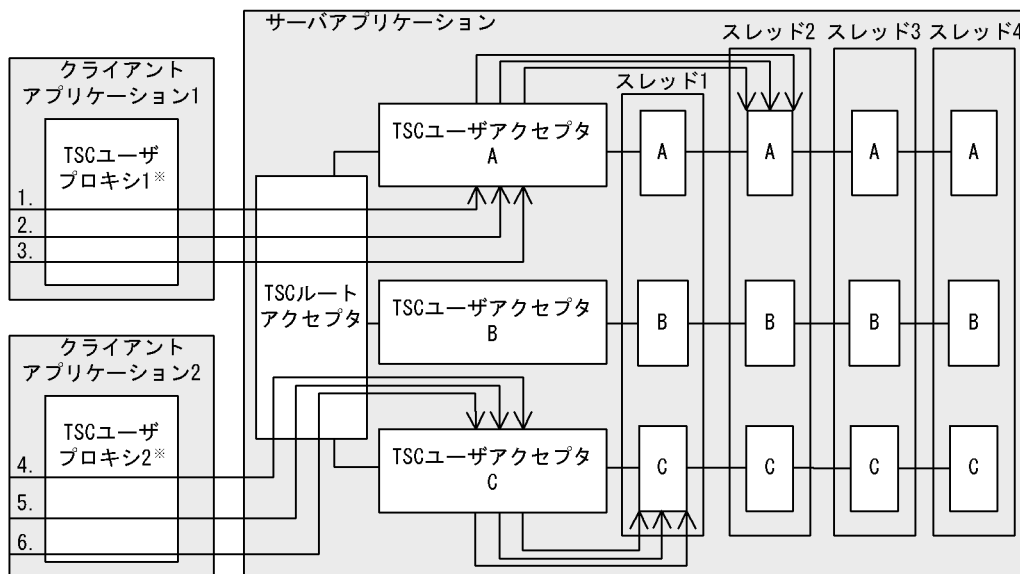
2. で示す処理では、クライアントアプリケーションから A のインタフェースで TSC アクセプタ名称を指定しないでリクエストするとき、リクエストが処理されるサーバアプリケーションプロセス、TSC ユーザアクセプタ、およびスレッドが常に同じになるという保証はありません。したがって、名称がある TSC ユーザアクセプタが呼び出されることがあります。

サーバアプリケーションプロセス 1 およびサーバアプリケーションプロセス 2 に同じ TSC アクセプタ名称がある B のインタフェースの場合、複数のリクエストをするとき、リクエストを処理するサーバアプリケーションプロセスおよびスレッドが常に同じになるという保証はありません。

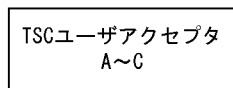
(3) セッション呼び出し時のリクエストの振り分け

セッション呼び出し機能を使用する場合のリクエストの振り分けを次の図に示します。

図 2-27 セッション呼び出し機能を使用する場合のリクエストの振り分け



(凡例)



: A~Cの各インターフェースに対応するTSCユーザアクセプタ



: A~Cの各インターフェースに対応するTSCユーザオブジェクト

1. A のインターフェースを持つインスタンスのセッション呼び出し開始要求。
2. A のインターフェースを持つインスタンスに対するリクエスト。
クライアントアプリケーション 1 のすべてのリクエストを同じインスタンスに送信します。
3. A のインターフェースを持つインスタンスのセッション呼び出し終了要求。
4. C のインターフェースを持つインスタンスのセッション呼び出し開始要求。
5. C のインターフェースを持つインスタンスに対するリクエスト。
クライアントアプリケーション 2 のすべてのリクエストを同じインスタンスに送信します。
6. C のインターフェースを持つインスタンスのセッション呼び出し終了要求。

注

セッション呼び出し用プロキシを継承した TSC ユーザプロキシです。

2.7.6 サーバアプリケーションの時間監視

サーバアプリケーションでは、時間監視の機能を提供しています。監視を終了する前に、監視時間が経過するとプロセスを異常終了させます。

ユーザメソッドの呼び出しからリターンまでの実行時間の監視
IDL ファイルに記述したユーザメソッドの呼び出しからリターンまでの実行時間を監視する場合は、`tscstartprc` コマンドの実行時、またはサーバアプリケーションの開始時に `-TSCWatchMethod` オプションで監視時間を指定してください。

ユーザメソッド以外の実行時間の監視

ユーザメソッド以外の実行時間を監視する場合は、`TSCWatchTime` クラスを使用してください。`TSCWatchTime` クラスは、サーバアプリケーションで `TSCWatchTime` クラスの `start()` が発行されてから `stop()` が発行されるまでの時間を監視します。

`initServer()` 発行後から `endServer()` を発行するまでの間であれば、いつでも使用できます。

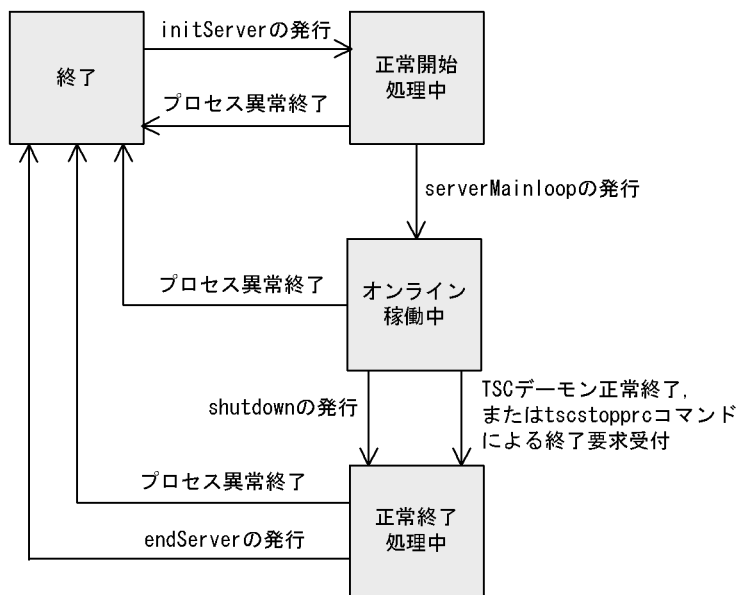
`TSCWatchTime` クラスのコンストラクタに監視時間を指定するか、`tscstartprc` コマンドの実行時またはサーバアプリケーションの開始時の `-TSCWatchTime` オプションに監視時間を指定します。

2.7.7 サーバアプリケーションの状態の検出

サーバアプリケーションで発行する `TSCAdm` クラスの `get_status()` メソッドの戻り値によって、オンライン稼働中かどうかといったサーバアプリケーションの状態を検出できます。

サーバアプリケーションの状態遷移を次の図に示します。

図 2-28 サーバアプリケーションの状態遷移



注

自プロセス内での `TSCAdm` クラスの `shutdown()` メソッドの発行、`tscstopprc` コマ

2. 機能

ンドでの正常終了，または TSC デーモンの正常終了によって，サーバアプリケーションは終了します。なお，次に示す場合は，プロセスの異常終了と同じ遷移になります。

- サーバアプリケーションを kill コマンドで強制終了した場合
- tscstopprc コマンドに -TSCForce オプションを指定して強制終了した場合
- tscstop コマンドに -TSCForce オプションを指定して強制終了した場合

2.8 TSCORB コネクタの機能

TSCORB コネクタは、ORB クライアント (TPBroker クライアントを始めとする、CORBA 2.5 に準拠するクライアント) からのリクエストを受け付けて、OTM のサーバアプリケーションで処理できるようにするゲートウェイプロセスです。さらに、TSC レギュレータと同様に、ORB クライアントからのリクエストを受け付ける接続を集約することで、リクエストの集中による問題を解決します。なお、このゲートウェイ機能を使用するには、OTM - Connector for ORB が必要です。

! 注意事項

OTM - Connector for ORB については、サポート時期をご確認ください。

この節では、TSCORB コネクタを経由する場合の接続の仕組み、TSCORB コネクタと TSC デーモンの対応、および ORB クライアントへの例外の通知について説明します。

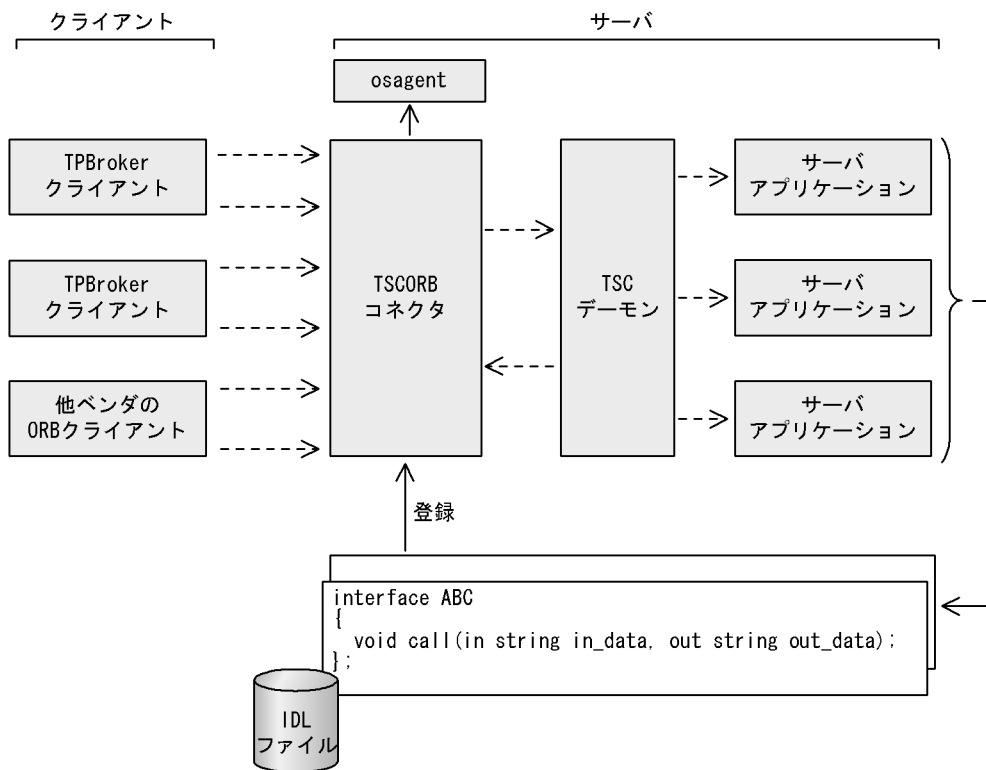
2.8.1 TSCORB コネクタ経由の接続の仕組み

TPBroker では、クライアントはサーバオブジェクトと直接接続します。一方、ORB クライアントから OTM に接続する場合は、TSCORB コネクタ経由で TSC ノードに接続します。そのため、TSCORB コネクタの開始時に、サーバオブジェクトの情報を登録する必要があります。この登録には、アプリケーションプログラムの開発で定義したインタフェースやオペレーションについて記述した IDL ファイルを使用します。具体的には、TSCORB コネクタの開始時に `tscstartgw` コマンドの `-TSCIDLFile` オプションで IDL ファイルを指定します。なお、一つの TSCORB コネクタの開始時に、複数の IDL ファイルを指定することもできます。

ORB クライアントは、IDL ファイルの記述内容に従って、該当するインタフェースを持つ TSCORB コネクタに接続します。なお、接続に成功した ORB クライアントからのリクエストでも、登録されていないオペレーションから送信された場合は、TSCORB コネクタはそのリクエストを受け付けられません。

IDL ファイルの記述内容に従って、ORB クライアントから OTM に接続する仕組みを次の図に示します。

図 2-29 ORB クライアントから OTM に接続する仕組み



(凡例) ———> : 制御の流れ
 -----> : コネクションの方向

ORB クライアントから OTM への接続方法には、TPBroker のスマートエージェントを使用する方法と IOR 文字列を使用する方法の 2 とおりがあります。それぞれの接続方法について次に説明します。

TPBroker のスマートエージェントの使用

通常は、TPBroker のスマートエージェント (osagent) を使用して接続を確立します。このとき、IDL ファイルの記述内容に従って、オブジェクトリファレンスが osagent に登録されます。これは、クライアントが TPBroker クライアントの場合に有効な接続方法です。

IOR 文字列の使用

IOR 文字列を使用して接続を確立します。IOR 文字列を使用する接続は、他ベンダの ORB クライアントから接続する場合などで、osagent を使用した接続ができない場合や、osagent の負荷を軽減させる場合に有効な接続方法です。また、この接続では、osagent へのオブジェクトリファレンスの登録の抑止もできます。

この場合の接続の手順を次に示します。

1. TSCORB コネクタの開始時に、tscstartgw コマンドに -TSCIORConnect オプ

ションを指定します。このとき、同時に `-TSCPort` オプションも指定します。

2. `tscgetior` コマンドで、IOR 文字列が記述されたテキストファイルを取得します。
3. 2. で取得したテキストファイルを、接続する ORB クライアントに配布します。
4. ORB クライアントは、起動時に IOR 文字列のファイルを読み込んで TSC ノードに接続します。このとき、CORBA の ORB クラスで提供される `string_to_object()` を使用します。

なお、IOR 文字列が記述されたファイルは、TSCORB コネクタに登録した IDL ファイルのインタフェース、サーバの IP アドレス、または TSCORB コネクタの受信ポート番号を変更しないかぎり、繰り返し使用できます。そのため、2. と 3. の手順は、システムの構築時に一度実施すれば、以降の接続では省略できます。

2.8.2 TSCORB コネクタと TSC デーモンの対応

TSC レギュレータと同様に、一つの TSC デーモンに対して、必要に応じて複数の TSCORB コネクタを開始できます。TSCORB コネクタは、対応する TSC デーモンと同じコンピュータに配置する必要があります。

TSCORB コネクタが ORB クライアントからのリクエストを受け付けると、対応する TSC デーモンにリクエストを転送してリクエストの応答を待ちます。応答が返ってくると、その応答をクライアントアプリケーションに返します。そのときの動作はマルチスレッドで並行処理されます。

2.8.3 ORB クライアントへの例外の通知

サーバアプリケーションのユーザオブジェクト内で発生した CORBA 例外やユーザ例外は、リクエストを送信した ORB クライアントにそのまま通知されます。

一方、OTM のシステム例外が発生した場合は、CORBA 例外に変換して ORB クライアントに通知されます。このとき、マイナーコードには `0x48542400 ~ 0x485424ff` の値が設定されます。デフォルトの例外変換以外の方法で変換して通知する場合は、例外変換テーブルファイルを作成して使用します。例外変換テーブルファイルについては、「付録 D 例外変換テーブルファイルの作成方法」を参照してください。

2.9 稼働統計情報の取得

OTM は、サーバ側のシステムチューニングの目安となる稼働統計情報を提供します。稼働統計情報は、排他制御をしないで取得されます。そのため、高負荷の環境では情報が欠落することがあることに注意してご使用ください。

2.9.1 稼働統計情報の概要

稼働統計情報は、処理の状態、数量、および時間についての情報です。これらの情報は、TSC デーモン、TSC レギュレータ、クライアントアプリケーション、およびサーバアプリケーションを対象として取得できます。稼働統計情報は、TSC ノードを単位として稼働統計情報ファイルに出力されます。稼働統計情報を使用する場合は、`tscstart` コマンドの実行時に `-TSCStatsUse` オプションで "Y" を指定してください。

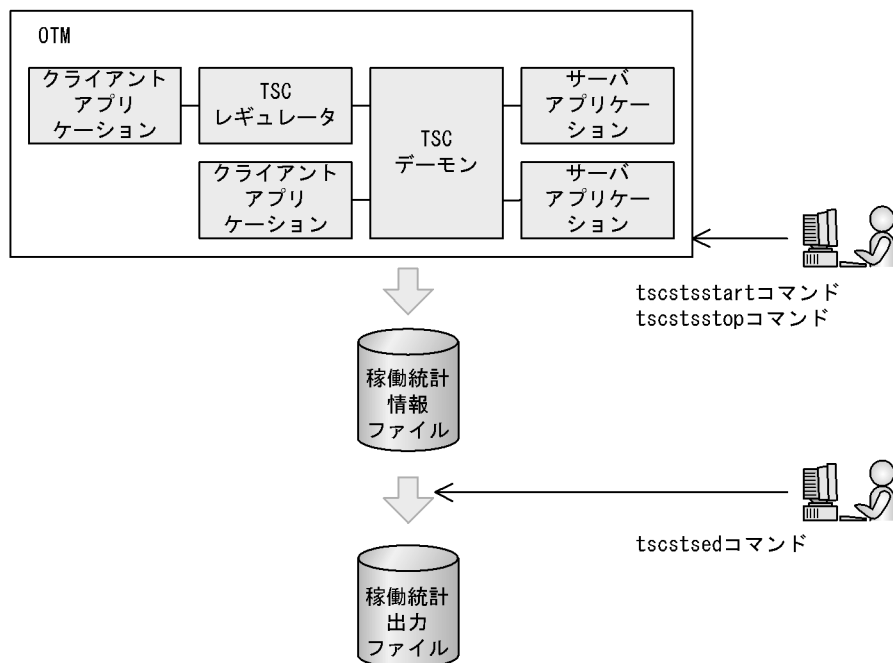
オンラインの開始後、稼働統計情報ファイルへの出力を開始するには、`tscstatsstart` コマンドを実行します。出力を終了するには `tscstatsstop` コマンドを実行します。その後、`tscstatsd` コマンドを実行して稼働統計情報ファイルを基にカレントディレクトリに稼働統計出力ファイルを出力します。`tscstatsd` コマンドのオプション指定によって、出力先を次の項目から選択できます。

- テキストファイル (デフォルト)
- CSV 形式のテキストファイル (`-TSCCsv` オプション指定時)
- 標準出力 (`-TSCStdout` オプション指定時)

ユーザはテキストエディタなどを使用して稼働統計出力ファイルを参照します。

稼働統計情報の概要を次の図に示します。

図 2-30 稼働統計情報の概要



2.9.2 稼働統計情報のファイル運用

稼働統計情報ファイルの格納ディレクトリを次の表に示します。

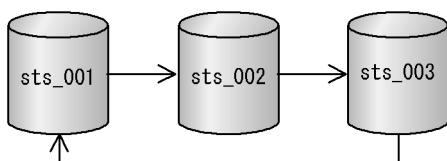
表 2-9 稼働統計情報ファイルの格納ディレクトリ

ファイルの種類	使用する OS	格納ディレクトリ
稼働統計情報ファイル	UNIX	<code>\$TSCSPOOL/utt/sts/TSC 識別子 /dcopltrc</code>
	Windows	<code>%TSCSPOOL%\%utt%\sts%\TSC 識別子 %dcopltrc</code>
稼働統計情報ファイルのバックアップファイル	UNIX	<code>\$TSCSPOOL/utt/sts/TSC 識別子 /spool/save</code>
	Windows	<code>%TSCSPOOL%\%utt%\sts%\TSC 識別子 %spool%save</code>

稼働統計情報ファイルのファイル名は、`sts_nnn` (`nnn` : 001 ~ 256) です。`sts_001` から開始し、ファイルがいっぱいになったとき、またはオンラインの再開時にメッセージを出力して世代を切り替えます。世代数は、`tscstart` コマンドの `-TSCStatsFileCount` オプションで指定できます。デフォルトでは3世代のファイルを使用し、世代はラウンドロビン方式で切り替えられます。

稼働統計情報ファイルの世代切り替えを次の図に示します。

図 2-31 稼働統計情報ファイルの世代切り替え



稼働統計情報ファイルは TSC デーモンの開始時に作成され、TSC デーモンの終了時には、バックアップファイル格納ディレクトリにコピーが作成されます。稼働統計情報ファイルのバックアップファイル名は sts_nnn.bk1 および sts_nnn.bk2 (nnn : 001 ~ 256) です。

2.9.3 稼働統計出力の出力形式

tscstsed コマンドの -TSCEdit オプションのオペランド指定とほかのオプションを組み合わせることで、次の表に示す稼働統計出力ファイルの出力内容を指定できます。また、-TSCCsv オプションを同時に指定すると、CSV 形式で出力できます。

表 2-10 稼働統計出力ファイルの出力内容の指定

-TSCEdit のオペランド	出力内容
nod	TSC ノード全体でまとめた情報の稼働統計出力ファイルを出力します。
rac	TSC ルートアクセプタ単位のスケジュール情報の稼働統計出力ファイルを出力します。 -TSCRootAcceptor オプションを同時に指定して、特定の TSC ルートアクセプタに関する情報を出力することもできます。
opr	サーバアプリケーションのオペレーション単位のリクエスト情報の稼働統計出力ファイルを出力します。 -TSCOperation オプションを同時に指定して、特定のオペレーションに関する情報を出力することもできます。

各稼働統計出力ファイルは、tscstsed コマンドの実行時のカレントディレクトリに同時に作成されます。前回実行したときのファイルがある場合は、メッセージを出力してファイルを上書きしてよいかどうかユーザに問い合わせます。

各出力形式の稼働統計出力ファイル名の一覧を次の表に示します。

表 2-11 各出力形式の稼働統計出力ファイル名の一覧

-TSCEdit の オペランド	ファイル名	
	-TSCCsv オプションの指定なし	-TSCCsv オプションの指定あり
nod	nodata.txt	nodata.csv

-TSCEdit の オペランド	ファイル名	
	-TSCCsv オプションの指定なし	-TSCCsv オプションの指定あり
rac	racdata.txt	racdata.csv
opr	oprdata.txt	oprdata.csv

-TSCEdit オプションを省略する場合は、すべての出力形式について稼働統計出力ファイルが出力されます。つまり、三つの稼働統計出力ファイルが出力されます。

2.9.4 稼働統計出力の出力情報

稼働統計出力ファイルへの出力情報を示します。さらに、TSC ノード全体でまとめた出力情報、TSC ルートアクセプタ単位の出力情報、およびサーバアプリケーションのオペレーション単位の出力情報を示します。

表 2-12 稼働統計出力ファイルへの出力情報

情報種別	項目	説明	オペランド		
			nod	rac	opr
リクエスト 情報	リクエスト種別	リクエストが「同期型」か「非応答型」かまたは「セッション型」かの種別です。この情報は、リクエストを処理するサーバアプリケーションで取得します。なお、リクエストの要求元が ORB クライアント (CORBA 2.5 に準拠するクライアント) の場合、そのリクエストが非応答型であっても、同期型として情報を取得します。また、リクエストの要求元が OTM クライアントなのか、ORB クライアントなのかの区別はしません。	×	×	
	レスポンス時間	サーバアプリケーションへリクエストを送信した時間と、その応答を受信した時間の差です。この情報は、リクエスト送信元であるクライアントアプリケーションで取得します。ただし、OTM - Client のクライアントアプリケーションおよび WWW Application Developer's Kit の OTM ゲートウェイからの呼び出しの場合は取得できません。		×	
	リクエスト実行時間	ユーザが実装したメソッドが、クライアントアプリケーションからリクエストを受信した時間と、その応答を送信した時間の差です。この情報はサーバアプリケーションで取得します。		×	

2. 機能

情報種別	項目	説明	オペランド		
			nod	rac	opr
	リクエストタイムアウト	リクエストを送信してからその応答が返されるまでの間に発生したタイムアウト障害です。クライアントアプリケーションで取得します。ただし、OTM・ClientのクライアントアプリケーションおよびWWW Application Developer's KitのOTMゲートウェイからの呼び出しの場合は取得できません。		×	
	リクエスト障害	リクエストを送信してからその応答が返されるまでの処理で発生した障害です。クライアントアプリケーションで取得します。ただし、OTM・ClientのクライアントアプリケーションおよびWWW Application Developer's KitのOTMゲートウェイからの呼び出しの場合は取得できません。		×	
	TSCORB コネクタのレスポンス時間	TSCORB コネクタがORBクライアントの同期型リクエストを受信した時間と、その応答をサーバから受信した時間の差です。リクエスト種別が「非応答型」の場合は取得できません。		×	
	TSCORB コネクタのリクエストタイムアウト	TSCORB コネクタがリクエストを送信してからその応答が返されるまでの間に発生したタイムアウト障害です。リクエスト種別が「非応答型」の場合は取得できません。		×	
	TSCORB コネクタのリクエスト障害	TSCORB コネクタがリクエストを送信してからその応答が返されるまでの処理で発生した障害です。リクエスト種別が「非応答型」の場合は取得できません。		×	
セッション情報	セッション開始メソッド発行数	セッション開始メソッドを発行し、TSCデーモンでセッション呼び出しを受け付けられる状態になった回数です。	○	×	×
	セッション数	情報を取得したときに、TSCデーモンとクライアントアプリケーションの間で確立しているセッションの数です。この情報は、tscstsstart コマンドの-TSCCheckInterval オプションに指定した時間間隔で取得します。	○	×	×
スケジュール情報	スケジュール待ち件数	スケジュール用キューに滞留したリクエストの数です。キューにリクエストをキューイングするときに、この情報を取得します。キューに滞留しているリクエストの数をプライオリティの高低に関係なく取得します。			×

情報種別	項目	説明	オペランド		
			nod	rac	opr
	スケジュール待ち件数 (高プライオリティ)	プライオリティの高いリクエストだけが使用できるキューの部分 (-TSCHighPriorRequestCount オプションの指定値) に滞留しているリクエスト数です。プライオリティの高いリクエストをキューイングするときに、この情報を取得します。-TSCHighPriorRequestCount オプションに値を指定していない場合はこの情報は取得されません。			×
	スケジュール待ち時間	スケジュール用キューにリクエストを登録してから取り出すまでの、リクエストの滞留時間です。			×
	スケジュールできなかったリクエスト数	キューに同時に登録できる数 (tscstart コマンドの -TSCMaxRequestCount オプションで指定) を超え、スケジュールできなかったリクエストの数です。リクエストのスケジューリング失敗のたびにこの情報を取得します。 キューに同時に登録できる数を超えている場合、リクエストのプライオリティの高低に関係なく、この情報を取得します。			×
	スケジュールできなかったリクエスト数 (低プライオリティ)	キューに滞留可能な値 (-TSCMaxRequestCount オプション指定値) を超えない範囲で、プライオリティの低いリクエストをスケジューリングできなかった件数です。つまり、-TSCMaxRequestCount オプションの指定値から -TSCHighPriorRequestCount オプションの指定値を減算した件数を、プライオリティの低いリクエストの件数が超えている状態です。この状態で、プライオリティの低いリクエストのスケジューリングが失敗するたびに、この情報を取得します。 -TSCHighPriorRequestCount オプションに値を指定していない場合はこの情報は取得されません。			×
プロセス情報	アプリケーションプログラム異常終了の件数	TSC デーモンが管理するアプリケーションプログラムの異常終了の発生件数です。異常終了が発生するたびに、この情報を取得します。なお、TSC デーモンが管理するプロセスについては、「2.4.6 プロセスの管理」を参照してください。		×	×
	TSC レギュレータ異常終了の件数	TSC レギュレータの異常終了の発生件数です。異常終了が発生するたびに、この情報を取得します。		×	×
	TSCORB コネクタ異常終了の件数	TSCORB コネクタの異常終了の発生件数です。異常終了が発生するたびに、この情報を取得します。		×	×

2. 機能

情報種別	項目	説明	オペランド		
			nod	rac	opr
コネクション情報	TSC レギュレータとクライアントアプリケーション間	TSC レギュレータとクライアントアプリケーション間のコネクション情報です。tscstsstart コマンドの -TSCCheckInterval オプションに指定した時間間隔でこの情報を取得します。		×	×
	TSC デーモンとクライアントアプリケーション間	TSC デーモンとクライアントアプリケーション間のコネクション情報です。tscstsstart コマンドの -TSCCheckInterval オプションに指定した時間間隔でこの情報を取得します。		×	×
	TSC デーモンとサーバアプリケーション間（制御用）	TSC デーモンとサーバアプリケーション間のコネクション情報です。tscstsstart コマンドの -TSCCheckInterval オプションに指定した時間間隔でこの情報を取得します。		×	×
	TSC デーモンとサーバアプリケーション間（リクエスト用）	TSC デーモンとサーバアプリケーション間のコネクション情報です。tscstsstart コマンドの -TSCCheckInterval オプションに指定した時間間隔でこの情報を取得します。			×
	TSC デーモンと TSC レギュレータ間	TSC デーモンと TSC レギュレータ間のコネクション情報です。tscstsstart コマンドの -TSCCheckInterval オプションに指定した時間間隔でこの情報を取得します。		×	×
	TSC デーモンと TSCORB コネクタ間	TSC デーモンと TSCORB コネクタ間のコネクション数です。tscstsstart コマンドの -TSCCheckInterval オプションに指定した時間間隔でこの情報を取得します。		×	×

（凡例）

- ：該当する出力形式で出力します。
 - ×
- ×：該当する出力形式で出力しません。

表 2-13 TSC ノード全体でまとめた稼働統計出力情報

情報種別	項目	出力する情報				
		発生件数	最大数	最小数	平均数	最大, 最小, 平均数の単位
リクエスト情報	レスポンス時間					マイクロ秒
	リクエスト実行時間					マイクロ秒
	リクエストタイムアウト		-	-	-	
	リクエスト障害		-	-	-	

情報種別	項目	出力する情報				
		発生 件数	最大 数	最小 数	平均 数	最大, 最小, 平均 数の単位
セッション情報	セッション開始メソッド発行数		-	-	-	
	セッション数	-				件数
スケジュール情報	スケジュール待ち件数					件数
	スケジュール待ち件数 (高プライオリティ)					件数
	スケジュール待ち時間					マイクロ秒
	スケジュールできなかったリクエスト数		-	-	-	
	スケジュールできなかったリクエスト数 (低プライオリティ)		-	-	-	
	TSCORB コネクタのレスポンス時間					マイクロ秒
	TSCORB コネクタのリクエストアウト		-	-	-	
	TSCORB コネクタのリクエスト障害		-	-	-	
プロセス情報	アプリケーションプログラム異常終了		-	-	-	
	TSC レギュレータ異常終了		-	-	-	
	TSCORB コネクタ異常終了		-	-	-	
コネクション情報	TSC レギュレータとクライアントアプリケーション間	-				件数
	TSC デーモンとクライアントアプリケーション間	-				件数
	TSC デーモンとサーバアプリケーション間 (制御用)	-				件数
	TSC デーモンとサーバアプリケーション間 (リクエスト用)	-				件数
	TSC デーモンと TSC レギュレータ間	-				件数
	TSC デーモンと TSCORB コネクタ間	-				件数

(凡例)

: 出力します。

- : "-" を出力します。

2. 機能

表 2-14 TSC ルートアクセプタ単位の稼働統計出力情報

情報種別	項目	出力する情報				
		発生 件数	最 大 数	最 小 数	平 均 数	最大, 最小, 平 均数の単位
スケジュール情報	スケジュール待ち件数					件数
	スケジュール待ち件数 (高プライオリティ)					件数
	スケジュール待ち時間					マイクロ秒
	スケジュールできなかったリクエスト数		-	-	-	
	スケジュールできなかったリクエスト数 (低プライオリティ)		-	-	-	
コネクション情報	TSC デーモンとサーバアプリケーション間 (リクエスト用)	-				件数

(凡例)

- : 出力します。
- : "-" を出力します。

表 2-15 サーバアプリケーションのオペレーション単位の稼働統計出力情報

情報種別	項目	出力する情報				
		発生 件数	最 大 数	最 小 数	平 均 数	最大, 最小, 平 均数の単位
リクエスト情報	リクエスト種別	同期型	-	-	-	
		非応答型	-	-	-	
		セッション型	-	-	-	
	レスポンス時間					マイクロ秒
	リクエスト実行時間					マイクロ秒
	リクエストタイムアウト		-	-	-	件数
	リクエスト障害		-	-	-	件数
	TSCORB コネクタのレスポンス時間					マイクロ秒
	TSCORB コネクタのリクエストアウト		-	-	-	
TSCORB コネクタのリクエスト障害		-	-	-		

(凡例)

: 出力します。

- : "-" を出力します。

2.9.5 稼働統計情報取得時のオペレーション名の取り扱い

稼働統計情報では、データ量を小さくするため 32 文字以上のオペレーション名を次の形式で取り扱います。

"先頭から15文字" + "*" + "最後から15文字"

例えば, "callOnly0000000aa0000000callOnly" というオペレーション名と
"callOnly0000000bb0000000callOnly" というオペレーション名はともに
"callOnly0000000*0000000callOnly" という同じオペレーション名として稼働統計情報が取得されます。

3

OTM - Client の運用

この章では、OTM - Client の環境設定、開始と終了、および障害対策について説明します。

3.1 OTM - Client の環境設定

3.2 OTM - Client の開始と終了

3.3 OTM - Client の障害対策

3.1 OTM - Client の環境設定

この節では、OTM - Client の環境設定の流れと環境変数について説明します。

3.1.1 OTM - Client の環境設定の流れ

OTM - Client の環境設定の流れを次の図に示します。

図 3-1 OTM - Client の環境設定の流れ



1. OTM 実行環境ディレクトリの作成

UNIX を使用する場合、OTM 実行環境ディレクトリを作成し、アクセス権限に "777" を指定してください。

OTM 実行環境ディレクトリは TSCSPool 環境変数に指定します。

2. 環境変数の設定

ユーザは、OS の環境変数を設定する必要があります。

3. クライアントアプリケーションの作成

OTM - Client で使用するクライアントアプリケーションを作成します。

クライアントアプリケーションの作成方法とアプリケーションプログラミングインタフェースについては、マニュアル「TPBroker Object Transaction Monitor プログラマーズガイド」を参照してください。

3.1.2 OTM - Client の環境変数

TPBroker OTM は、「TSC」で始まる環境変数を使用します。そのため、実行環境ではここで説明する以外の「TSC」で始まる環境変数を使用しないでください。

また、ユーザは TPBroker で使用する環境変数について設定する必要があります。詳細については、ご使用の環境に応じて、マニュアル「TPBroker ユーザーズガイド」または「Borland Enterprise Server VisiBroker デベロッパーズガイド」を参照してください。

TSCDIR

インストール先ディレクトリを指定します。

TSCSPOOL

OTM 実行環境ディレクトリを指定します。

OTM - Client は該当するディレクトリに障害情報を出力します。TSCSPOOL 環境変数の指定を省略した場合、障害情報は出力されません。

ライブラリパス

C++ 言語を使用する場合、OS の環境変数にライブラリ格納ディレクトリを指定してください。

ライブラリ格納ディレクトリを指定する環境変数は OS によって異なります。OTM - Client のライブラリ格納ディレクトリを指定する環境変数を、使用する OS ごとに次の表に示します。

表 3-1 ライブラリ格納ディレクトリを指定する環境変数 (OTM - Client)

使用する OS	HP-UX	Solaris	AIX	Windows
環境変数	SHLIB_PATH	LD_LIBRARY_PATH	LIBPATH	PATH

詳細については OS のマニュアルを参照してください。OS が仕様変更した場合、環境変数も変更されることがあります。

CLASSPATH (Java2 言語の場合)

UNIX を使用する場合は、" 格納ディレクトリ /tscj2cl.jar" を追加します。Windows を使用する場合は、" 格納ディレクトリ ¥tscj2cl.jar" を追加します。

PATH

トランザクションフレームジェネレータで使用する環境変数です。

TPBroker の idl2cpp または idl2java , osagent , idl2ir , および irep が格納されているディレクトリを設定してください。

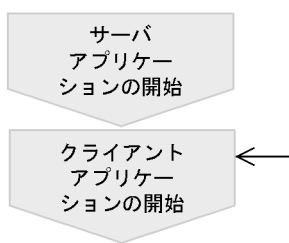
3.2 OTM - Client の開始と終了

この節では、OTM - Client の開始と終了の流れについて説明します。クライアントアプリケーションの開始時に指定するオプションについては、「5. コマンドリファレンス」を参照してください。

3.2.1 OTM - Client の開始

OTM - Client の開始の流れを次の図に示します。

図 3-2 OTM - Client の開始の流れ



1. サーバアプリケーションの開始

クライアントアプリケーションがリクエストを送信するときには、サーバアプリケーションが動作している必要があります。サーバアプリケーションの開始方法の詳細については、「4.2 OTM の開始と終了」を参照してください。

2. クライアントアプリケーションの開始

クライアントアプリケーションの開始時に、次に示す OTM - Client のオプションを指定できます。

- -TSCDomain オプションに、TSC ドメイン名称を指定します。
- -TSCID オプションに、対応する TSC デーモンの TSC 識別子を指定します。

クライアントアプリケーションの開始の詳細については、「5. コマンドリファレンス」を参照してください。また、マニュアル「TPBroker ユーザーズガイド」または「Borland Enterprise Server VisiBroker デベロッパーズガイド」をあわせて参照してください。

一つの TSC デーモンに複数のクライアントアプリケーションを対応させる場合は、繰り返しクライアントアプリケーションを開始します。

3.2.2 OTM - Client の終了

OTM - Client の終了の流れを次の図に示します。

図 3-3 OTM - Client の終了の流れ



1. クライアントアプリケーションの終了

OTM - Client は `tscstop` コマンドの処理の延長では終了しません。そのため、ユーザがクライアントアプリケーションを終了する必要があります。一つの TSC デーモンに複数のクライアントアプリケーションが対応する場合、繰り返しクライアントアプリケーションを終了します。

3.3 OTM - Client の障害対策

OTM - Client の障害と取得する情報について説明します。

3.3.1 OTM - Client で発生する障害の種類

OTM - Client で発生する障害の種類を次の表に示します。

表 3-2 OTM - Client の障害

障害内容	OTM - Client の処理	ユーザの処理
クライアントアプリケーション異常終了	ありません。	クライアントアプリケーションを再開始します。

3.3.2 OTM - Client で取得できる障害情報

TSCSPOOL 環境変数が指定されている場合、OTM - Client は該当するディレクトリに障害情報を出力します。出力メッセージの詳細については、「6. メッセージ」を参照してください。TSCSPOOL 環境変数の指定を省略した場合は、障害情報は出力されません。

Java アプリケーション（または Java アプレット）を使用する場合は、クライアントアプリケーションの開始時に `-TSCDebugInfo` オプションを指定してください。省略した場合は、障害情報は標準出力に出力されます。`-TSCDebugInfo` オプションを指定する場合は、次に示す条件を満たす必要があります。

- OTM - Client がインストールされていること。
- 日立共通クラスライブラリがインストールされていること。
日立共通クラスライブラリは、TPBroker のインストール時にインストールされます。
- OTM - Client が提供する C++ ライブラリのパス（UNIX を使用する場合は `$TSCDIR/lib`、Windows を使用する場合は `%TSCDIR%\bin`）、および日立共通クラスライブラリのパスが、ライブラリパス環境変数に指定されていること。
各 OS のライブラリパス環境変数については、「3.1.2 OTM - Client の環境変数」を参照してください。
- Java アプレットでこのオプションを指定する場合は、OTM - Client が提供する C++ ライブラリの読み込みが、Java アプレットの署名によって許可されていること。

TSCAdm クラスの `initClient()` メソッド呼び出しに成功する前のログメッセージおよびトレース情報は、標準出力に出力されたり失われたりすることがあります。ライブラリのロードに失敗した場合は、TSCAdm クラスの `initClient()` メソッド呼び出しで `TSCInitializeException` 例外が発生します。

また、C++ 言語を使用して作成したクライアントアプリケーション、または OTM の

Java アプリケーション（または Java アプレット）にこのオプションを指定した場合は無視されます。

4

OTM の運用

この章では、OTM の環境設定、開始と終了、OTM - Connector for ORB の運用方法、TPBroker との連携、および障害対策について説明します。

4.1 OTM の環境設定

4.2 OTM の開始と終了

4.3 OTM - Connector for ORB の運用

4.4 OTM と TPBroker との連携

4.5 OTM の障害対策

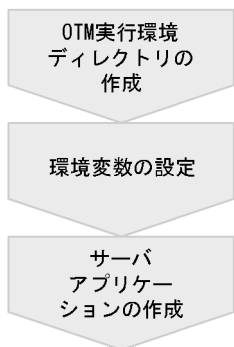
4.1 OTM の環境設定

この節では、OTM の環境設定の流れと環境変数について説明します。

4.1.1 OTM の環境設定の流れ

OTM の環境設定の流れを次の図に示します。

図 4-1 OTM の環境設定の流れ



1. OTM 実行環境ディレクトリの作成

UNIX を使用する場合、OTM 実行環境ディレクトリを作成し、アクセス権限に "777" を指定してください。

OTM 実行環境ディレクトリは TSCSPOOL 環境変数に指定します。

2. 環境変数の設定

ユーザは、OS の環境変数を設定する必要があります。

3. サーバアプリケーションの作成

OTM で使用するサーバアプリケーションを作成します。OTM が提供するライブラリファイルでクライアントアプリケーションを作成することもできます。

IDL 文法、アプリケーションプログラムの作成方法、およびアプリケーションプログラミングインタフェースについては、マニュアル「TPBroker Object Transaction Monitor プログラマーズガイド」を参照してください。

4.1.2 OTM の環境変数

TPBroker OTM は、"TSC" で始まる環境変数を使用します。そのため、実行環境ではここで説明する以外の "TSC" で始まる環境変数を使用しないでください。

また、ユーザは TPBroker で使用する環境変数について設定する必要があります。詳細については、ご使用の環境に応じて、マニュアル「TPBroker ユーザーズガイド」または「Borland Enterprise Server VisiBroker デベロッパーズガイド」を参照してくださ

い。

TSCDIR

インストール先ディレクトリを指定します。

TSCSPOOL

OTM 実行環境ディレクトリを指定します。

TSCSPOOL 環境変数は、TSC ドメインの管理に使用されます。該当するディレクトリには、OTM によって障害情報が出力されます。

TSCSPOOL 環境変数の指定を省略した場合、UNIX の場合は "\$TSCDIR/spool" が設定され、Windows の場合は "%TSCDIR%\spool" が設定されます。

ライブラリパス

C++ および Java 言語を使用する場合、OS の環境変数にライブラリ格納ディレクトリを指定してください。Java 言語を使用する場合も、C++ のライブラリを使用するため指定する必要があります。

ライブラリ格納ディレクトリを指定する環境変数は OS によって異なります。OTM のライブラリ格納ディレクトリを指定する環境変数を、使用する OS ごとに次の表に示します。

表 4-1 ライブラリ格納ディレクトリを指定する環境変数 (OTM)

使用する OS	HP-UX	Solaris	AIX	Windows
環境変数	SHLIB_PATH	LD_LIBRARY_PATH	LIBPATH	PATH

詳細については OS のマニュアルを参照してください。OS が仕様変更した場合、環境変数も変更されることがあります。

CLASSPATH (Java2 言語の場合)

UNIX を使用する場合は、"格納ディレクトリ /tscj2sv.jar" を追加します。Windows を使用する場合は、"格納ディレクトリ %tscj2sv.jar" を追加します。

PATH

トランザクションフレームジェネレータで使用する環境変数です。

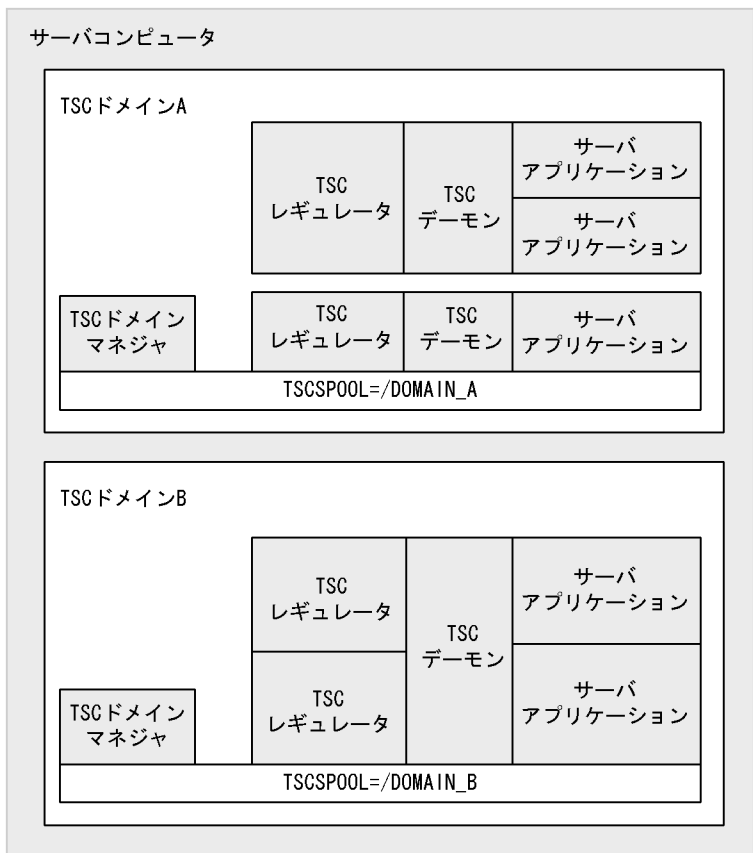
TPBroker の idl2cpp または idl2java、osagent、idl2ir、および irep が格納されているディレクトリを設定してください。

4.1.3 環境変数による TSC ドメインの管理

コンピュータ内の TSC ドメインの管理には、TSCSPOOL 環境変数が参照されます。一つのコンピュータ内で複数の TSC デーモンを開始して TSC ドメインを構成する場合、TSC ドメイン内では、TSCSPOOL 環境変数の指定値を同じ環境にしてください。

複数の TSC ドメインと環境変数の関係を次の図に示します。

図 4-2 複数の TSC ドメインと環境変数



TSC ドメイン A は、"TSCSPool=/DOMAIN_A" という環境変数を参照し、TSC ドメイン B は、"TSCSPool=/DOMAIN_B" という環境変数を参照します。

4.1.4 ユーザ環境変数定義ファイル

`tscstart` コマンドおよび `tscstartprc` コマンドの `-TSCEnviron` オプションには、ユーザ環境変数定義ファイルを指定できます。ユーザ環境変数定義ファイルには、開始するプロセスが使用する環境変数を記述します。記述した環境変数は、それぞれ、次のプロセスに渡されます。

`tscstart` コマンドの場合

TSC デーモン、および TSC デーモンが開始する、次のプロセスに渡されます。

- `tscstartprc` コマンドで開始するサーバアプリケーション
- `tscstart` コマンドの `-TSCRegStart` オプションによって自動開始される TSC レギュレータ

`tscstartprc` コマンドの場合

tscstartpre コマンドで開始するサーバアプリケーションに渡されます。-TSCEnviron オプションを使用すると、例えば、特定のサーバアプリケーションだけに環境変数から値を渡すことができます。

(1) ユーザ環境変数定義ファイルの記述形式

ユーザ環境変数定義ファイルの記述形式を次に示します。

環境変数名称=指定値

環境変数名称、イコール(=)、および環境変数値のどれも省略できません。行の先頭からその行の最初のイコール(=)の直前までが環境変数名称とみなされます。また、最初のイコール(=)の直後から改行またはファイルの終端の直前までが環境変数値とみなされます。

シャープ(#)で始まる行は、その行全体がコメントとみなされて無視されます。また、行の途中からシャープ(#)の場合はそれ以降から行の最後までがコメントとみなされて無視されます。このとき、シャープ(#)の手前の空白およびタブも無視されます。したがって、シャープ(#)は有効な環境変数値としては指定できません。

一つの定義を複数行に分けて記述する場合は、改行の直前に継続符号(¥)を記述してください。継続符号(¥)を記述した場合、直前の空白およびタブは無視されます。ただし、行の途中に記述された継続符号(¥)は文字として扱われます。また、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。シャープ(#)を記述すると、その行の終わりまでがコメントとみなされるため、継続符号(¥)を記述しても行の継続はできなくなります。

指定された環境変数は、すでに同一の環境変数名称があれば上書きされ、なければ追加されます。tscstart コマンドでは、tscstart コマンド実行時の環境変数が基となります。tscstartpre コマンドでは、TSC デーモンの実行環境の環境変数、つまり、tscstart コマンド実行時の環境変数に tscstart コマンドの -TSCEnviron オプションで指定された環境変数を上書きまたは追加した環境変数が基となります。

(2) ユーザ環境変数定義ファイルの記述例

ユーザ環境変数定義ファイルの記述例を次に示します。

```
# コメント行
LANG=ja_JP.SJIS                                # 以降コメント
CLASSPATH=/opt/TPBrokerOTM_S/lib/tscj2sv.jar:   ¥
/opt/java/lib/classes.zip                       # 継続行
```

(3) ユーザ環境変数定義ファイル使用時の注意事項

ユーザ環境変数定義ファイルを使用する場合の注意事項を次に示します。

4. OTM の運用

先頭が "TSC" または "tsc" で始まる環境変数名称は使用できません。

環境変数名称には、英数字およびアンダースコア (_) で構成される文字列を指定してください。

行の先頭から環境変数名称までは空白およびタブは記述できません。行の先頭から環境変数名称を記述してください。

環境変数名称とイコール (=) の間、およびイコール (=) と環境変数定義の間に空白およびタブを記述しないでください。

環境変数定義ファイル中に \$AAA (UNIX の場合) または %AAA% (Windows の場合) の記述がある場合、これらはそのままの文字列として認識されます。環境変数 AAA として展開されません。

空行 (環境変数定義もコメントもない行) を記述する場合は、空白およびタブを含めないで改行コードだけにしてください。

次に示す場合は定義内容不正となります。

- 環境変数定義内にイコール (=) がない場合
- 行の先頭または最後にイコール (=) がある場合
- 環境変数名称の先頭が "TSC" または "tsc" で始まる場合
- 空白およびタブだけの行がある場合

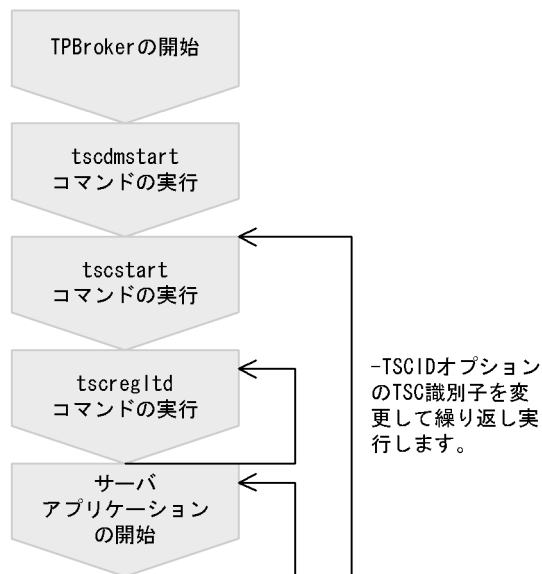
4.2 OTM の開始と終了

この節では、OTM の開始と終了の流れについて説明します。サーバアプリケーションの開始時に指定するオプションについては、「5. コマンドリファレンス」を参照してください。

4.2.1 OTM の開始

OTM の開始の流れを次の図に示します。なお、クライアントアプリケーションの開始は OTM の開始後になります。

図 4-3 OTM の開始の流れ



1. TPBroker の開始

サーバアプリケーションは、TPBroker のアプリケーションプログラムです。クライアントアプリケーションからのリクエストを受け付けるためには、TPBroker の osagent が動作している必要があります。osagent の開始については、マニュアル「[Borland Enterprise Server VisiBroker デベロッパーズガイド](#)」を参照してください。

2. tscdmstart コマンドの実行

TSC ドメインマネージャを開始します。

3. tscstart コマンドの実行

TSC デーモンを開始します。

-TSCDomain オプションに TSC ドメイン名称を指定し、-TSCID オプションに TSC

4. OTM の運用

識別子を指定します。このとき、TSC 識別子は TSC ドメイン内でユニークである必要があります。

4. tscregld コマンドの実行

TSC レギュレータを開始します。TSC レギュレータは必要に応じて開始します。

TSC レギュレータを経由したリクエストを getTSCClient によって要求する場合だけ開始してください。

TSC レギュレータを開始する必要がある場合は、-TSCDomain オプションに TSC ドメイン名称を指定し、対応する TSC デーモンの TSC 識別子を -TSCID オプションに指定します。また、一つの TSC デーモンに複数の TSC レギュレータを対応させる場合、繰り返し tscregld コマンドを実行します。

5. サーバアプリケーションの開始

サーバアプリケーションの開始時に OTM のオプションを指定できます。

-TSCDomain オプションに TSC ドメイン名称を指定し、対応する TSC デーモンの TSC 識別子を -TSCID オプションに指定します。サーバアプリケーションの開始の詳細については、「5. コマンドリファレンス」を参照してください。また、マニュアル「TPBroker ユーザーズガイド」または「Borland Enterprise Server VisiBroker デベロッパーズガイド」をあわせて参照してください。

一つの TSC デーモンに複数のサーバアプリケーションを対応させる場合は、繰り返しサーバアプリケーションを開始します。

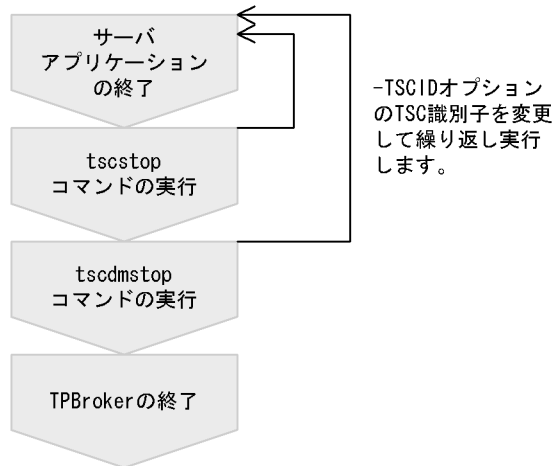
注

一つの TSC ドメインで複数の TSC デーモンを開始する場合、-TSCID オプションに指定する TSC 識別子を変更して、3. ~ 5. の手順を繰り返します。

4.2.2 OTM の終了

OTM の終了の流れを次の図に示します。通常、クライアントアプリケーションを事前に終了する必要があります。詳細については、「3.2.2 OTM - Client の終了」を参照してください。

図 4-4 OTM の終了の流れ



1. サーバアプリケーションの終了
サーバアプリケーションを終了します。一つの TSC デーモンに複数のサーバアプリケーションが対応する場合、繰り返し終了します。
2. tscstop コマンドの実行
TSC デーモンを終了します。このとき、TSC レギュレータが tscstop コマンドの延長で終了します。
3. tscdmstop コマンドの実行
TSC ドメインマネージャを終了します。
4. TPBroker の終了
TPBroker の終了については、マニュアル「TPBroker ユーザーズガイド」を参照してください。

注

一つの TSC ドメインにある複数の TSC デーモンを終了する場合、-TSCID オプションを変更して、1. ~ 2. の手順を繰り返す必要があります。

4.3 OTM - Connector for ORB の運用

OTM - Connector for ORB は、TSCORB コネクタを提供するプログラムです。

この節では、OTM - Connector for ORB の運用方法として、TSCORB コネクタの開始と終了、および TSCORB コネクタの状態の検出について説明します。

! 注意事項

OTM - Connector for ORB については、サポート時期をご確認ください。

4.3.1 TSCORB コネクタの開始と終了

TSCORB コネクタの開始と終了の方法について説明します。

(1) TSCORB コネクタの開始

TSCORB コネクタの開始の流れを次の図に示します。

図 4-5 TSCORB コネクタの開始の流れ



1. tscdmstart コマンドの実行
TSC ドメインマネージャを開始します。
2. tscstart コマンドの実行
TSC デーモンを開始します。
3. サーバアプリケーションの開始
サーバアプリケーションが開始されていない場合は、リクエストがエラーになるため、サーバアプリケーションの開始後に TSCORB コネクタを開始してください。

4. tscstartgw コマンドの実行

TSCORB コネクタを開始します。

各コマンドの詳細については、「5. コマンドリファレンス」を参照してください。

(2) TSCORB コネクタの終了

TSCORB コネクタを終了するときは、tscstopgw コマンドを実行します。tscstopgw コマンドを実行すると、処理中のリクエストの終了を待って TSCORB コネクタを終了します。なお、tscstopgw コマンドの実行以降に ORB クライアントから送信されたリクエストはエラーとなります。

(3) TSCORB コネクタの異常終了時の再起動

TSCORB コネクタの開始後は、TSC デーモンによって管理されますが、TSCORB コネクタが異常終了しても TSC デーモンによる再起動はされません。再起動が必要な場合は、TPBroker の運用支援機能を使用してプロセス監視をしてください。

4.3.2 TSCORB コネクタの状態の検出

TSCORB コネクタを開始すると、tscstartgw コマンドの -TSCDomain オプションおよび -TSCID オプションで指定した TSC デーモンによって管理されます。したがって、TSCORB コネクタの状態を検出するには、tscprels コマンドを実行して TSC デーモンが管理する各プロセスの情報を出力してください。

4.4 OTM と TPBroker との連携

OTM と TPBroker の運用支援機能（プロセス監視機能）とを連携して、OTM が提供するシステムプロセス（TSC デーモン、TSC ドメインマネージャ、および TSC レギュレータ）、サーバアプリケーションプロセス、および TSCORB コネクタを監視できます。プロセス監視方法の詳細については、マニュアル「TPBroker ユーザーズガイド」を参照してください。

TPBroker の運用支援機能を使用してプロセス監視する場合、プロセス監視定義の定義ファイルを作成してください。プロセス監視定義に記述する項目を次の表に示します。

表 4-2 プロセス監視定義に記述する項目

記述項目	監視対象プロセス					
	TSC デー モン	TSC ド メイン マネ ジャ	TSC レ ギュ レー タ 1	サー バア プ リ ケ ー シ ョ ン ²	TSCOR B コ ネ ク タ ³	
	間 接 起 動 方 式	間 接 起 動 方 式	直 接 起 動 方 式	直 接 起 動 方 式	間 接 起 動 方 式	間 接 起 動 方 式
監視対象プロセスを正常開始させる場合のコマンド名	tscstart コ マ ン ド	tscdmst art コ マ ン ド	tscreglt d コ マ ン ド	実行形 式ファ イル名 称 (Java の場 合は Java インタ プリタ 名称)	tscstart pre コ マ ン ド	tscstart gw コ マ ン ド
監視対象プロセスのプロセス ID を取得するコマンド名	tscgetpi d コ マ ン ド	tscgetpi d コ マ ン ド	-	-	tscgetpi d コ マ ン ド	tscgetpi d コ マ ン ド
正常終了または強制終了させるコマンド名	tscstop コ マ ン ド	tscdmst op コ マ ン ド	-	-	tscstop pre コ マ ン ド	tscstop gw コ マ ン ド

（凡例）

- : 該当しません。

注 1

TSC デーモンの開始時に -TSCRegStart オプションを指定して TSC レギュレータを開始する場合は、TSC レギュレータに関するプロセス監視定義を定義ファイルに記述する必要はありません。

注 2

サーバアプリケーションは、間接起動方式および直接起動方式のどちらでも開始できます。

注 3

TSCORB コネクタを使用する場合は、OTM - Connector for ORB が必要です。

OTM - Connector for ORB については、サポート時期をご確認ください。

4.5 OTM の障害対策

OTM の障害と取得する情報について説明します。

OTM が出力するメッセージについては、「6. メッセージ」を参照してください。

4.5.1 OTM で発生する障害の種類

OTM で発生する障害の種類を次の表に示します。

表 4-3 OTM の障害

障害内容	OTM の処理	ユーザの処理
TSC デモンの異常終了	再開時に、異常終了した TSC デモンに対して接続したアプリケーションプロセスを終了します。前回要求をされていたリクエスト情報はすべて破棄します。	tscstart コマンドを実行して再開します。
TSC ドメインマネージャの異常終了	正常開始の場合、共用メモリなどの内部資源を引き継いで再開します。ただし、管理情報が破壊されているときは、正常開始では TSC ドメインマネージャを再開できません。 -TSCForceStart オプションを指定して強制正常開始する場合、前回に管理していた TSC デモンを終了し、共用メモリを再作成します。このとき、TSC デモンを再開しません。 -TSCAutoForce オプションを指定して自動強制開始する場合、管理情報を引き継がないときは自動的に強制正常開始になります。	tscdmstart コマンドを実行して再開します。 TPBroker の運用支援機能と連携する場合は、 -TSCForceStart オプションまたは -TSCAutoForce オプションを指定してください。
システムプロセス (TSC レギュレータ) の異常終了	TSC デモンから自動開始したシステムプロセスの場合、登録時のコマンドオプションを使用して TSC デモンが再開します。	ユーザが開始したシステムプロセスの場合、tscregltd コマンドを実行して再開します。
サーバアプリケーション異常終了	ありません。	サーバアプリケーションを再開します。

4.5.2 OTM で取得できる障害情報

OTM は、TSCSPOOL 環境変数に指定したディレクトリに障害情報を出力します。出力メッセージの詳細については、「6. メッセージ」を参照してください。

5

コマンドリファレンス

この章では、運用コマンドおよびアプリケーションプログラム開始時のオプションについて、コマンド名のアルファベット順に説明します。ただし、アプリケーションプログラムの作成時に使用するコマンド（`tscidl2cpp` および `tscidl2j`）については、マニュアル「TPBroker Object Transaction Monitor プログラマーズガイド」で説明しています。

運用コマンドの一覧

クライアントアプリケーションの開始

サーバアプリケーションの開始

運用コマンドの一覧

運用コマンドの一覧を次の表に示します。

表 5-1 運用コマンドの一覧

コマンド名	説明	OTM	OTM - Client	
システム管理	tscdminfo	TSC ドメイン情報の表示と削除		×
	tscdmstart	TSC ドメインマネージャの開始		×
	tscdmstop	TSC ドメインマネージャの終了		×
	tscgetpid	プロセス ID の取得		×
	tsclogcat	メッセージの編集と出力		
	tscscon	コネクション情報の出力		×
	tscprcls	プロセス情報の出力		×
	tscrasget	トラブルシュート情報の取得と出力		
	tscregltd	TSC レギュレータの開始		×
	tscridinfo	TSC ドメイン情報の出力		×
	tscstart	TSC デーモンの開始		×
	tscstop	TSC デーモンの終了		×
	tscver	実行形式ファイルおよびライブラリのバージョン情報の出力		
	tscjver	jar ファイルのバージョン情報の出力		
稼働統計情報の出力	tscstsed	稼働統計情報の編集と出力		×
	tscstsstart	稼働統計情報の取得の開始		×
	tscstsstop	稼働統計情報の取得の終了		×
アプリケーションプログラムの作成	tscidl2cpp ¹	C++ 用トランザクションフレームの出力		
	tscidl2j ¹	Java 用トランザクションフレームの出力		
アプリケーションプログラムの運用	tscgetref	接続先情報の取得と編集		2

コマンド名		説明	OTM	OTM - Client
	tsholdra	TSC ルートアクセプタの閉塞		×
	tsclsra	TSC ルートアクセプタ情報の出力		×
	tscrlesra	TSC ルートアクセプタの閉塞解除		×
	tscstartprc	サーバアプリケーションの開始		×
	tscstopprc	サーバアプリケーションの終了		×
TSCORB コネクタの運用	tscstartgw	TSCORB コネクタの開始	3	×
	tscstopgw	TSCORB コネクタの終了	3	×
	tscgetior	IOR 文字列の取得	3	×

(凡例)

○ : 提供します。

× : 提供しません。

注 1

アプリケーションプログラムの作成時に使用するコマンド (tscidl2cpp および tscidl2j) については、マニュアル「TPBroker Object Transaction Monitor プログラマーズガイド」を参照してください。

注 2

OTM - Client では、接続先情報の取得はできません。接続先情報ファイルを入力とする接続先情報の編集だけができます。

注 3

TSCORB コネクタの運用に使用するコマンド (tscstartgw , tscstopgw , tscgetior) の実行には、OTM - Connector for ORB が必要です。OTM - Connector for ORB については、サポート時期をご確認ください。

tscdminfo (TSC ドメイン情報の表示と削除)

形式

```
tscdminfo [-h] [-d]
```

機能

コンピュータ内の OTM 環境に登録済みの TSC ドメインに関する情報を表示または削除します。

オプション

-h

コマンドの使用方法が表示されます。

-d

コンピュータ内の OTM 環境に登録済みの TSC ドメイン情報を削除します。

"tscdminfo -d"の実行後に、現在の OTM 環境に登録済みの TSCSPOOL と TSC ドメイン名称の一覧が表示されます。ユーザは一覧から削除したい組み合わせの番号を選択し、コマンドの指示に従って入力します。

使用例

-d オプション指定時の使用例を次に示します。

1. tscdminfo コマンドを実行します。

```
% tscdminfo -d
```

2. TSC ドメイン名称および TSCSPOOL の一覧が表示されます。

```
No TSC-Domain      Port      TSCSPOOL
1 hitachi          9496     /home1/hitachi/tsc/spool
2 TSCDOMAIN        9497     /MAKE_DIR/TSC0100/spool
Please input Number which you wish delete. ->
```

3. 削除したい環境を表す数字を入力します。

```
Please input Number which you wish delete. ->2
```

4. 問題がなければ 'y' を入力します。

```
2 TSCDOMAIN        9497     /MAKE_DIR/TSC0100/spool

We will delete this combination of Domain_name and TSCSPOOL
directory.
OK ? (y/n) -> y
```

戻り値

このコマンドは次に示す戻り値をシェルに返してから、処理を終了します。

戻り値	意味
0	正常終了しました。
0 以外	コマンド処理中にエラーが発生したために異常終了しました。 出力されたメッセージに従って対策したあと、再度、コマンドを実行してください。

tscdmstart (TSC ドメインマネージャの開始)

形式

```
tscdmstart [-h] [-TSCDomain TSCドメイン名称]
            [-TSCMaxTSC TSCデーモン管理最大数]
            [-TSCPort TSCドメイン管理ポート番号]
            [-TSCMyHost ホスト名称またはIPアドレス]
            [-TSCSendInterval TSCノード情報の送信間隔]
            [-TSCSubnetMask サブネットマスク]
            [-TSCSendHost ホスト名称またはIPアドレス
                [,ホスト名称またはIPアドレス,...]]
            [-TSCSendHostInterval 異なるネットワークセグメントへの
                TSCノード情報の送信間隔]
            [-TSCForceStart | -TSCAutoForce]
```

機能

TSC ドメインマネージャを開始します。

オプション

-h

コマンドの使用方法が表示されます。

-TSCDomain TSC ドメイン名称

~ < 1 ~ 31 文字の英数字 > 《TSCDOMAIN》

TSC ドメインマネージャが属する TSC ドメイン名称を指定します。"TSC" または "tsc" で始まる文字列は指定しないでください。省略した場合は "TSCDOMAIN" が設定されま

す。

-TSCMaxTSC TSC デーモン管理最大数

~ < 符号なし整数 > ((1 ~ 1024)) 《64》

TSC ドメイン内で管理する TSC デーモンの最大数を指定します。省略した場合は "64" が設定されます。

-TSCPort TSC ドメイン管理ポート番号

~ < 符号なし整数 > ((5001 ~ 65535)) 《20137》

TSC ドメインマネージャが負荷情報を交換するために使用するポート番号を指定します。省略した場合はサービス名ファイルが参照されます。サービス名ファイルに指定がないときは "20137" が設定されます。

サービス名ファイルへの記述の形式を次に示します。

TSC_TSCドメイン名称 ポート番号/tcp

例えば, "tscdmstart -TSCDomain JAPAN" と指定して TSC ドメインマネージャを開始する場合, サービス名ファイルの, 次のように指定したエントリが読み込まれます。

TSC_JAPAN 10021/tcp

TSC ドメインマネージャは一つのポート番号で tcp および udp の両方のプロトコルを使用します。そのため, TSC ドメインマネージャが使用するポート番号をほかの udp のポート番号として使用しないでください。

-TSCPort オプションの指定値は共用メモリの作成時に使用されるため, 異常終了後に TSC ドメインマネージャを再開するときは変更しないでください。変更したい場合は, TSC ドメインマネージャを正常終了させてから変更してください。また, 共通の TSC ドメイン名称を持つ TSC ドメインマネージャを複数のコンピュータで開始する場合, -TSCPort オプションに同じ値を指定してください。

-TSCMyHost ホスト名称または IP アドレス

~ < 1 ~ 64 文字の文字列 > 《hostname コマンドで取得されるホスト名称》

マルチホームドホスト環境で OTM が使用するホスト名称または IP アドレスを指定します。省略した場合は, hostname コマンドで取得されるホスト名称が設定されます。

-TSCMyHost オプションの指定値は共用メモリの作成時に使用されるため, 異常終了後に TSC ドメインマネージャを再開するときは変更しないでください。変更したい場合は, TSC ドメインマネージャを正常終了させてから変更してください。

-TSCSendInterval TSC ノード情報の送信間隔

~ < 符号なし整数 > ((0 ~ 65535)) 《60》(単位 : 秒)

TSC ドメインマネージャが TSC ノード情報をネットワークセグメント内にブロードキャストする間隔を秒単位で指定します。省略した場合は "60 (秒)" が設定されます。

TSC ノード情報によって, 他 TSC ノードの TSC ドメインマネージャにアプリケーションプログラムの位置情報をブロードキャストします。そのため, 大きな値を指定すると, 他 TSC ノードの TSC ドメインマネージャに自 TSC ノード情報をブロードキャストする間隔が長くなり, 他 TSC ノードから自 TSC ノードへのアプリケーションプログラムのメソッド要求に失敗することがあります。小さな値を指定する場合, 他 TSC ノードの TSC ドメインマネージャに TSC ノード情報をブロードキャストする間隔が短くなります。その結果, ネットワークの負荷が高くなる可能性があります。

TSC ノード情報をブロードキャストしない場合は, "0" を指定してください。

-TSCSubnetMask サブネットマスク

~ (nnn.nnn.nnn.nnn) < 符号なし整数 > ((0 ~ 255)) 《255.255.255.0》

ネットワークのサブネットマスクを指定します。省略した場合は "255.255.255.0" が設定されます。

-TSCSendHost ホスト名称または IP アドレス [, ホスト名称または IP アドレス ,...]

~ < 英数字 , ピリオド (.) , およびハイフン (-) >

TSC デモン情報の配布先となる、異なるネットワークセグメントにある TSC ドメインマネージャが動作するホストのホスト名称または IP アドレスを指定します。この指定によって、複数のネットワークセグメントで構成した TSC ドメインで、TSC ノード情報を配布できるようになります。詳細については、「2.6 複数のネットワークセグメントで構成する TSC ドメイン」を参照してください。

ホスト名称を指定する場合は、/etc/hosts に設定されているホスト名称を指定してください。また、ハイフン (-) で始まるホスト名称または IP アドレスは指定しないでください。指定した場合の動作は保証しません。

複数のホストを指定する場合は、指定値をコンマ (,) で区切ります。次に指定例を示します。

-TSCSendHost HOSTA,HOSTB

-TSCSendHostInterval 異なるネットワークセグメントへの TSC ノード情報の送信間隔

~ < 符号なし整数 > ((0 ~ 65535)) 《-TSCSendInterval オプションに指定した値》(単位 : 秒)

異なるネットワークセグメントにある TSC ドメインマネージャが動作するホストに TSC ノード情報を送信する送信間隔を秒単位で指定します。

TSC ノード情報によって、他 TSC ノードの TSC ドメインマネージャに、アプリケーションプログラムの位置情報を送信します。そのため、大きな値を指定すると、他 TSC ノードの TSC ドメインマネージャに自 TSC ノード情報を送信する間隔が長くなり、他 TSC ノードから自 TSC ノードへのアプリケーションプログラムのメソッド要求に失敗することがあります。小さな値を指定する場合、他 TSC ノードの TSC ドメインマネージャに TSC ノード情報を送信する間隔が短くなります。その結果、ネットワークの負荷が高くなる場合があります。

TSC ノード情報を送信しない場合は、"0" を指定してください。

-TSCSendHostInterval コマンドオプションに値を指定しても、-TSCSendHost オプションを指定しなければ無効になります。また、-TSCSendHost、-TSCSendInterval、および -TSCSendHostInterval オプションの指定値によって TSC ノード情報の送信の有

無は異なります。

コマンドオプションの指定値と TSC ノード情報の送信の関係を次の表に示します。

表 5-2 tscdmstart コマンドのオプション指定値と TSC ノード情報の送信の関係

-TSCSendHost	-TSCSendHostInterval	-TSCSendInterval	TSC ノード情報の送信
指定あり	0	0	×
		0 以外	×
	0 以外	0	
		0 以外	
	指定なし	0	×
		0 以外	
指定なし	無効	無効	×

(凡例)

○ : 送信されます。

× : 送信されません。

注

-TSCSendInterval オプションを指定しない場合のデフォルト値 (60 秒) を含みません。

-TSCForceStart

前回は強制終了または異常終了している場合に、TSC ドメインマネージャを強制的に正常開始します。前回は管理していた TSC デーモンを強制終了し、管理情報を初期化して TSC ドメインマネージャを開始します。このとき、TSC デーモンは再開始されません。

前回は正常終了している場合は正常開始するため、-TSCForceStart オプションの指定は無視されます。

-TSCForceStart オプションは、-TSCAutoForce オプションと同時に指定できません。詳細については、「2.5.1 TSC ドメインマネージャの開始形態」を参照してください。

-TSCAutoForce

前回は強制終了または異常終了している場合に、前回の管理情報を引き継いで TSC ドメインマネージャを自動強制開始します。前回の管理情報を正常に引き継げる場合は、終了前のシステム状態を復元できます。前回の管理情報を引き継げない場合は、-TSCForceStart オプションが指定されたときと同様の動作をします。

前回は正常終了している場合は正常開始するため、-TSCAutoForce オプションの指定は無視されます。

-TSCAutoForce オプションは、-TSCForceStart オプションと同時に指定できません。
詳細については、「2.5.1 TSC ドメインマネージャの開始形態」を参照してください。

戻り値

このコマンドは次に示す戻り値をシェルに返してから、処理を終了します。

戻り値	意味
0	正常終了しました。
0以外	コマンド処理中にエラーが発生したために異常終了しました。 出力されたメッセージに従って対策したあと、再度、コマンドを実行してください。

注意事項

TSC ドメインマネージャは開始後にカレントディレクトリを移動します。そのため、次に示す場合のコアファイルの出力先は、実行形式ファイルのディレクトリとは異なります。

- "tscdmstop -TSCForce -TSCDebugInfo" または kill コマンドなどで TSC ドメインマネージャを強制終了した場合
- TSC ドメインマネージャが異常終了した場合

移動後のカレントディレクトリは、UNIX の場合は "\$TSCSPOOL/adm/TSC ドメイン名称" で、Windows の場合は "%TSCSPOOL%\adm\TSC ドメイン名称" です。

tscdmstop (TSC ドメインマネージャの終了)

形式

```
tscdmstop [-h] [-TSCDomain TSCドメイン名称]
           [-TSCMyHost ホスト名称またはIPアドレス]
           [-TSCForce] [-TSCDebugInfo]
```

機能

TSC ドメインマネージャを終了します。

オプション

-h

コマンドの使用方法が表示されます。

-TSCDomain TSC ドメイン名称

~ < 1 ~ 31 文字の英数字 > 《TSCDOMAIN》

TSC ドメインマネージャが属する TSC ドメイン名称を指定します。"TSC" または "tsc" で始まる文字列は指定しないでください。省略した場合は "TSCDOMAIN" が設定されます。

-TSCMyHost ホスト名称または IP アドレス

~ < 1 ~ 64 文字の文字列 > 《hostname コマンドで取得されるホスト名称》

マルチホームドホスト環境で OTM が使用するホスト名称または IP アドレスを指定します。省略した場合は、hostname コマンドで取得されるホスト名称が設定されます。

-TSCForce

TSC ドメインマネージャを強制終了します。

-TSCForce オプションで TSC ドメインマネージャを強制終了させたあとの再開では、tscdmstart コマンドに -TSCForceStart オプションを指定して開始してください。

-TSCDebugInfo

TSC ドメインマネージャの強制終了時にデバッグ情報として、共用メモリダンプおよびコアを取得します。-TSCDebugInfo オプションは、-TSCForce オプションの指定時だけ有効です。

戻り値

このコマンドは次に示す戻り値をシェルに返してから、処理を終了します。

tscdmstop (TSC ドメインマネジャの終了)

戻り値	意味
0	正常終了しました。
0 以外	コマンド処理中にエラーが発生したために異常終了しました。 出力されたメッセージに従って対策したあと、再度、コマンドを実行してください。

tscgetior (IOR 文字列の取得)

形式

```
tscgetior [-h] |
  [{-TSCDomain TSCドメイン名称 [,TSCドメイン名称...] |
  -TSCDomainAll}]
  [{-TSCID TSC識別子 [,TSC識別子...] | -TSCIDAll}]
  [-TSCHost ホスト名称またはIPアドレス
  [,ホスト名称またはIPアドレス...]]
  [-TSCMyHost ホスト名称またはIPアドレス]
  [-TSCInterfaceName インタフェース名 [,インタフェース名...]]
  [-TSCObjectName オブジェクト名 [,オブジェクト名...]]
  [-TSCOutPath 出力ディレクトリ名]
```

機能

TSCORB コネクタに登録されているインタフェースの IOR 文字列を取得します。

tscgetior コマンドを実行すると、オプションに指定された条件に従って、出力ファイルにテキスト形式で IOR 文字列を抽出します。この IOR 文字列を使用することで、クライアントアプリケーションは任意の TSCORB コネクタに接続できます。

なお、tscgetior コマンドを使用して IOR 文字列を取得するには、OTM - Connector for ORB が必要です。OTM - Connector for ORB については、サポート時期をご確認ください。

オプション

-h

コマンドの使用方法が表示されます。

-TSCDomain TSC ドメイン名称 [,TSC ドメイン名称 ...]

~ < 1 ~ 31 文字の英数字 > 《TSCDOMAIN》

IOR 文字列の取得の対象となる TSCORB コネクタが属する TSC ドメイン名称を指定します。"TSC" または "tsc" で始まる文字列は指定しないでください。省略した場合は "TSCDOMAIN" が設定されます。

-TSCDomain オプションには、複数の TSC ドメイン名称を指定できます。tscgetior コマンドと同じホストで開始済みの TSC ドメイン名称を指定してください。

なお、-TSCDomain オプションは -TSCDomainAll オプションと同時に指定できません。

-TSCDomainAll

tscgetior コマンドを実行するホストにある、すべての TSC ドメインを対象として IOR 文字列を取得します。省略した場合は、-TSCDomain オプションで指定された TSC ドメインを対象として IOR 文字列を取得します。-TSCDomainAll オプションは、-TSCDomain オプションと同時に指定できません。

なお、自ホストにある TSC ドメインを確認するには tscdminfo コマンドを実行してください。

-TSCID TSC 識別子 [,TSC 識別子 ...]

~ < 1 ~ 31 文字の英数字、およびピリオド (.) > 《IP アドレス》

IOR 文字列の取得の対象となる TSCORB コネクタが属する TSC ノードの識別子を指定します。"TSC" または "tsc" で始まる TSC 識別子は指定しないでください。ピリオド (.) は、IP アドレスを指定する場合だけ使用できます。-TSCID オプションには、複数の TSC 識別子を指定できます。省略した場合は、IP アドレスが TSC 識別子に設定されます。例えば、ホストの IP アドレスが "172.17.112.43" のとき、"172.17.112.43" という文字列が TSC 識別子となります。

-TSCID オプションは、-TSCIDAll オプションと同時に指定できません。

-TSCIDAll

指定した TSC ドメイン内のすべての TSC 識別子を対象として、IOR 文字列を取得します。

省略した場合は、-TSCID オプションで指定した TSC 識別子を対象として IOR 文字列を取得します。-TSCIDAll オプションは、-TSCID オプションと同時に指定できません。

-TSCHost ホスト名称または IP アドレス [, ホスト名称または IP アドレス...]

~ < 1 ~ 64 文字の文字列 >

IOR 文字列の取得の対象となるホストのホスト名称または IP アドレスを指定します。

-TSCDomain オプション (または -TSCDomainAll オプション) および -TSCID オプション (または -TSCIDAll オプション) で指定された TSC ノードのうち、-TSCHost オプションで指定されたホストにある TSC ノードを対象として、IOR 文字列を取得します。-TSCHost オプションには、複数のホスト名称または IP アドレスを指定できます。

省略した場合、tscgetior コマンドで IOR 文字列の取得を要求したホストの TSC ドメインマネージャが認識できる、ネットワーク内にあるすべてのホストを対象とします。

-TSCMyHost ホスト名称または IP アドレス

~ < 1 ~ 64 文字の文字列 > 《hostname コマンドで取得されるホスト名称》

マルチホームドホスト環境で OTM が使用するホスト名称または IP アドレスを指定します。省略した場合は、hostname コマンドで取得されるホスト名称が設定されます。

-TSCID オプションを省略して、-TSCMyHost オプションを指定した場合、デフォルトの TSC 識別子は -TSCMyHost オプションで指定した IP アドレスになります。

-TSCInterfaceName インタフェース名 [, インタフェース名...]

~ <英数字, およびスラント (/) >

IOR 文字列の取得の対象となるインタフェース名を指定します。モジュール名付きのインタフェース名を指定する場合は、"モジュール名/インタフェース名"の形式で指定してください。-TSCInterfaceName オプションには、複数のインタフェース名を指定できません。

省略した場合、該当する TSC ドメインが認識する TSCORB コネクタに登録されている、すべてのインタフェース名を対象とします。

-TSCObjectName オブジェクト名 [, オブジェクト名...]

~ <英数字, ピリオド (.), プラス (+), およびハイフン (-) >

IOR 文字列の取得の対象となるオブジェクト名を指定します。英数字以外のオブジェクト名は指定しないでください。また、"TSC" または "tsc" で始まるオブジェクト名は指定しないでください。-TSCObjectName オプションには、複数のオブジェクト名を指定できません。

省略した場合、該当する TSC ドメインが認識する TSCORB コネクタに指定されている、すべてのオブジェクト名を対象とします。

-TSCOutPath 出力ディレクトリ名

UNIX の場合 ~ <英数字, ピリオド (.), スラント (/), プラス (+), およびハイフン (-) > 《カレントディレクトリ》

Windows の場合 ~ <英数字, ピリオド (.), 円符号 (¥), コロン (:), プラス (+), およびハイフン (-) > 《カレントディレクトリ》

IOR 文字列ファイルを出力するディレクトリ名称を絶対パスまたは相対パスで指定します。省略した場合は、tscgetior コマンドを実行したときのカレントディレクトリに出力します。

IOR 文字列ファイルは、取得の対象となる TSCORB コネクタに登録されている一つのインタフェースにつき一つずつ出力されます。各インタフェースの IOR 文字列ファイルの出力先ディレクトリと、IOR 文字列ファイルの名称を次に示します。

-TSCOutPathの指定ディレクトリ/
[aa..aa/]bb..bb/[cc..cc]/dd..dd#ee..ee.ior

(凡例)

記号	説明
スラント (/)	ディレクトリの区切りを表します。Windows の場合は、円符号 (¥) で表します。
aa..aa	TSCORB コネクタに登録されているインタフェースのモジュール名です。モジュール名称がない場合は、このディレクトリは作成されません。
bb..bb	TSCORB コネクタに登録されているインタフェースのインタフェース名です。
cc..cc	TSCORB コネクタの開始時に、tscstartgw コマンドの -TSCObjectName オプションで指定したオブジェクト名です。-TSCObjectName オプションの指定を省略した場合は、このディレクトリは作成されません。
dd..dd	TSCORB コネクタが動作しているホストの IP アドレスです。
ee..ee	TSCORB コネクタの開始時に、-TSCPort オプションで指定したポート番号です。

戻り値

このコマンドは次に示す戻り値をシェルに返してから、処理を終了します。

戻り値	意味
0	正常終了しました。
0 以外	コマンド処理中にエラーが発生したために異常終了しました。 出力されたメッセージに従って対策したあと、再度、コマンドを実行してください。

注意事項

コマンドオプション引数を複数指定する場合、区切り文字としてコンマ (,) を指定してください。また、引数とコンマ (,) の間には空白を入れしないでください。

出力先ディレクトリにすでに同名のファイルがある場合は上書きされるため、IOR 文字列ファイルを残しておきたい場合は、そのつどファイルをコピーしておくか、新しく作成したディレクトリを出力先ディレクトリに指定してください。

次に示す状態のときは、IOR 文字列を取得できません。

- tscgetior コマンドを実行するホストで、該当する TSC ドメインマネージャが開始されていない状態
- 他ホスト上の TSCORB コネクタの IOR 文字列を取得する場合に、該当する TSC ドメインマネージャが開始されていない状態
- 他ホスト上の TSCORB コネクタの IOR 文字列を取得する場合に、TSCORB コネクタの実行ホストで、該当する TSC ドメインマネージャが開始されていない状態

IOR 文字列ファイルの出力処理で障害が発生した場合は、IOR 文字列ファイルの出力内容が不正となることがあります。

tscgetpid (プロセス ID の取得)

形式

```
tscgetpid { [-h] |
            -TSCDomain TSCドメイン名称 [-TSCID TSC識別子]
            [-TSCMyHost ホスト名称またはIPアドレス] |
            -TSCDomain TSCドメイン名称 -TSCID TSC識別子
            -TSCAPID アプリケーション識別子
            [-TSCMyHost ホスト名称またはIPアドレス] }
```

機能

TPBroker が提供する運用支援機能のプロセス監視を実行する場合に使用します。

TSC ドメインマネージャ, TSC デーモン, サーバアプリケーション, または TSCORB コネクタのプロセス ID を取得します。

オプション

-h

コマンドの使用方法が表示されます。

-TSCDomain TSC ドメイン名称

~ < 1 ~ 31 文字の英数字 >

-TSCID オプションを指定する場合, プロセス ID 取得対象となる TSC デーモンが属する TSC ドメイン名称を指定します。

-TSCID オプションを省略する場合, プロセス ID 取得対象となる TSC ドメインマネージャの TSC ドメイン名称を指定します。

-TSCAPID オプションを指定する場合, プロセス ID 取得対象となるサーバアプリケーションまたは TSCORB コネクタを管理する TSC デーモンの TSC ドメイン名称を指定します。

指定するときは, "TSC" または "tsc" で始まる TSC ドメイン名称は指定しないでください。

-TSCID TSC 識別子

~ < 1 ~ 31 文字の英数字, およびピリオド (.) >

プロセス ID 取得対象となる TSC デーモンの TSC 識別子を指定します。"TSC" または "tsc" で始まる TSC 識別子は指定しないでください。ピリオド (.) は, IP アドレスを指定する場合だけ使用できます。省略した場合は, -TSCDomain オプションで指定された TSC ドメインマネージャのプロセス ID が取得されます。

-TSCAPID オプションを指定する場合、プロセス ID の取得対象となるサーバアプリケーションまたは TSCORB コネクタを管理する TSC デーモンの TSC 識別子を指定します。この場合、-TSCID オプションは省略できません。

-TSCAPID アプリケーション識別子

~ < 1 ~ 32 文字の英数字 >

プロセス ID 取得対象となるサーバアプリケーション、または TSCORB コネクタのアプリケーション識別子を指定します。

-TSCMyHost ホスト名称または IP アドレス

~ < 1 ~ 64 文字の文字列 > 《hostname コマンドで取得されるホスト名称》

マルチホームドホスト環境で OTM が使用するホスト名称または IP アドレスを指定します。省略した場合は、hostname コマンドで取得されるホスト名称が設定されます。

戻り値

このコマンドは次に示す戻り値をシェルに返してから、処理を終了します。

戻り値	意味
0	正常終了しました。
0 以外	コマンド処理中にエラーが発生したために異常終了しました。 出力されたメッセージに従って対策したあと、再度、コマンドを実行してください。

tscgetref (接続先情報の取得と編集)

形式

```
tscgetref [-h] |
  [{-TSCDomain TSCドメイン名称 [,TSCドメイン名称...] |
  -TSCDomainAll}]
  [{-TSCID TSC識別子 [,TSC識別子...] | -TSCIDA11}]
  [-TSCHost ホスト名称またはIPアドレス
  [,ホスト名称またはIPアドレス...]]
  [-TSCMyHost ホスト名称またはIPアドレス]
  [-TSCRequestWay 0 | 1 ] [-TSCConnectType bind | ior]
  [-TSCInFile 入力ファイル名称 [,入力ファイル名称...]]
  [{-TSCPrint | -TSCOutFile 出力ファイル名称}]
```

機能

TSC ノードにある TSC デモンまたは TSC レギュレータの接続先情報を取得および編集します。

tscgetref コマンドの出力ファイルである接続先情報ファイルの内容に応じて、クライアントアプリケーションまたはサーバアプリケーションは TSC ノードへ接続します。この場合のサーバアプリケーションは、TSCAdm クラスの getTSCClient メソッドを発行する、クライアントアプリケーションの機能を持つアプリケーションプログラムに限られます。接続先情報ファイルの詳細については、「2.2 接続先情報ファイルを使用する接続」を参照してください。

なお、tscgetref コマンドで接続先情報を取得できるのは OTM だけです。OTM - Client では、接続先情報ファイルを入力元とする接続先情報の編集だけができます。

オプション

-h

コマンドの使用方法が表示されます。

-TSCDomain TSC ドメイン名称 [,TSC ドメイン名称 ...]

~ < 1 ~ 31 文字の英数字 > 《TSCDOMAIN》

接続先情報の取得または編集の対象となる TSC ドメイン名称を指定します。"TSC" または "tsc" で始まる文字列は指定しないでください。省略した場合は "TSCDOMAIN" が設定されます。

-TSCDomain オプションには、複数の TSC ドメイン名称を指定できます。接続先情報を取得する場合は、tscgetref コマンドと同じホストで開始済みの、すべての TSC ドメイン名称を指定してください。

なお、-TSCDomain オプションは -TSCDomainAll オプションと同時に指定できません。

ん。

-TSCDomainAll

すべての TSC ドメインを対象として、接続先情報を取得または編集します。接続先情報を取得する場合は、tscgetref コマンドを実行するホストにあるすべての TSC ドメインを対象とします。接続先情報を編集する場合は、入力元となる接続先情報ファイル内に記述されたすべての TSC ドメインを対象とします。

省略した場合は、-TSCDomain オプションで指定された TSC ドメインを対象として、接続先情報を取得または編集します。-TSCDomainAll オプションは、-TSCDomain オプションと同時に指定できません。

なお、自ホストにある TSC ドメインを確認するには tscdminfo コマンドを実行してください。

-TSCID TSC 識別子 [,TSC 識別子 ...]

~ < 1 ~ 31 文字の英数字、およびピリオド (.) > 《IP アドレス》

接続先情報の取得または編集の対象となる TSC デーモンの識別子を指定します。"TSC" または "tsc" で始まる TSC 識別子は指定しないでください。ピリオド (.) は、IP アドレスを指定する場合だけ使用できます。-TSCID オプションには、複数の TSC 識別子を指定できます。省略した場合は、IP アドレスが TSC 識別子に設定されます。例えば、ホストの IP アドレスが "172.17.112.43" のとき、"172.17.112.43" という文字列が TSC 識別子となります。

-TSCID オプションは、-TSCIDAll オプションと同時に指定できません。

-TSCIDAll

指定した TSC ドメイン内のすべての TSC 識別子を対象として、接続先情報を取得または編集します。接続先情報を取得する場合は、指定された TSC ドメインが認識するすべての TSC 識別子を対象とします。接続先情報を編集する場合は、入力元となる接続先情報ファイル内に記述された接続先情報のうち、指定された TSC ドメイン内のすべての TSC 識別子を対象とします。

省略した場合は、-TSCID オプションで指定した TSC 識別子を対象として接続先情報を取得または編集します。-TSCIDAll オプションは、-TSCID オプションと同時に指定できません。

-TSCHost ホスト名称または IP アドレス [, ホスト名称または IP アドレス ...]

~ < 1 ~ 64 文字の文字列 >

接続先情報の取得または編集の対象となるホストのホスト名称または IP アドレスを指定します。

-TSCDomain オプション (または -TSCDomainAll オプション) および -TSCID オプション (または -TSCIDAll オプション) で指定された TSC ノードのうち、-TSCHost オプションで指定されたホストにある TSC ノードを対象として、接続先情報を取得または編集します。-TSCHost オプションには、複数のホスト名称または IP アドレスを指定できます。

省略した場合、接続先情報を取得するときは、tscgetref コマンドで接続先情報の取得を要求したホストの TSC ドメインマネージャが認識できる、ネットワーク内にあるすべてのホストを対象とします。接続先情報を編集するときは、入力元となる接続先情報ファイル内に記述されたすべてのホストを対象とします。

-TSCMyHost ホスト名称または IP アドレス

~ < 1 ~ 64 文字の文字列 > 《hostname コマンドで取得されるホスト名称》

マルチホームドホスト環境で OTM が使用するホスト名称または IP アドレスを指定します。省略した場合は、hostname コマンドで取得されるホスト名称が設定されます。

-TSCID オプションを省略して、-TSCMyHost オプションを指定した場合、デフォルトの TSC 識別子は -TSCMyHost オプションで指定した IP アドレスになります。

-TSCRequestWay 0 | 1

~ < 符号なし整数 > ((0 | 1))

接続先情報の取得または編集の対象となる接続経路を指定します。

"0" を指定すると、TSC デーモンに直結したリクエストの情報を取得または編集します。

"1" を指定すると、TSC レギュレータを経由して TSC デーモンに接続したリクエストの情報を取得または編集します。

省略した場合は、両方の方法に該当する情報を取得または編集します。

-TSCConnectType bind | ior

~ < 英字 > ((bind | ior))

接続先情報の取得または編集の対象となる接続方法を指定します。

"bind" を指定すると、TPBroker のスマートエージェントを使用した接続先の情報を取得または編集します。

"ior" を指定すると、IOR 文字列の使用を使用した接続先の情報を取得または編集します。

省略した場合は、両方の接続方法に該当する情報を取得または編集します。

-TSCInFile 入力ファイル名称 [, 入力ファイル名称 ...]

UNIX の場合 ~ < 英数字 , ピリオド (.) , スラント (/) , プラス (+) , およびハイフン (-) >

Windows の場合 ~ < 英数字 , ピリオド (.) , 円符号 (¥) , コロン (:) , プラス (+) , およびハイフン (-) >

編集の対象となる接続先情報ファイル名称を絶対パスまたは相対パスで指定します。省略した場合は , tscgetref コマンドの実行時にオンラインとなっている TSC ノードの情報が取得されます。-TSCInFile オプションには , 複数の接続先情報ファイル名称を指定できます。

ただし , OTM · Client では -TSCInFile オプションは省略できません。

-TSCPrint

接続先情報の取得または編集時に , 出力結果を標準出力に文字形式で出力します。出力結果は接続先情報ファイルとは形式が異なるので , 接続先情報ファイルとしては使用できません。また , -TSCPrint オプションを指定した場合は , 接続先情報ファイルは出力されません。-TSCPrint オプションは , -TSCOutFile オプションと同時に指定できません。

省略した場合は , 標準出力には何も出力されません。

-TSCOutFile 出力ファイル名称

UNIX の場合 ~ < 英数字 , ピリオド (.) , スラント (/) , プラス (+) , およびハイフン (-) > 《tscrefinfo.txt》

Windows の場合 ~ < 英数字 , ピリオド (.) , 円符号 (¥) , コロン (:) , プラス (+) , およびハイフン (-) > 《tscrefinfo.txt》

接続先情報を出力する接続先情報ファイル名称を絶対パスまたは相対パスで指定します。省略した場合は , tscgetref コマンドを実行したときのカレントディレクトリに "tscrefinfo.txt" という名前で接続先情報ファイルが作成されます。

出力先ディレクトリにすでに同名のファイルがある場合は上書きされるため , 接続先情報を残しておきたい場合はファイルをコピーしておいてください。接続先情報ファイルはテキスト形式で出力されます。

-TSCOutFile オプションは , -TSCPrint オプションと同時に指定できません。

戻り値

このコマンドは次に示す戻り値をシェルに返してから , 処理を終了します。

戻り値	意味
0	正常終了しました。
0 以外	コマンド処理中にエラーが発生したために異常終了しました。 出力されたメッセージに従って対策したあと、再度、コマンドを実行してください。

出力形式

このコマンドの出力形式を次に示します。出力例については、「付録 B tscgetref コマンドの出力例」を参照してください。

-TSCOutFile オプションを指定した場合の接続先情報ファイルの出力形式

```
aa..aa
bb..bb#cc..cc#dd..dd#ee..ee#ff..ff#gg..gg###hh..hh
```

(凡例)

記号	説明
aa..aa	ファイル管理情報です。OTM で使用する管理情報なので、変更、削除、および移動はしないでください。
bb..bb	TSC ドメイン名称 (1 ~ 31 文字の英数字) です。
cc..cc	TSC 識別子 (1 ~ 31 文字の英数字) です。
dd..dd	接続経路を次の値で示します。 0 : TSC デーモンに直結 1 : TSC レギュレータを経由して TSC デーモンに接続
ee..ee	接続方法を次の値で示します。 0 : TPBroker のスマートエージェントの使用 1 : IOR 文字列の使用
ff..ff	IP アドレスです。
gg..gg	ポート番号です。接続経路が "1" で接続方法が "0" の場合、"0" が出力されます。
hh..hh	IOR 文字列です。接続方法が "1" の場合、IOR 文字列が出力されます。

注

bb..bb ~ hh..hh の行は、接続先情報の数だけ繰り返されます。

-TSCPrint オプションを指定した場合の出力形式

```

version = aa..aa
---- record : bb..bb ----
TSCDomain   = cc..cc
TSCID       = dd..dd
RequestWay  = ee..ee
ConnectType = ff..ff
Ippaddress  = gg..gg
Port        = hh..hh
    
```

(凡例)

記号	説明
aa..aa	ファイル管理情報です。
bb..bb	レコード通番です。
cc..cc	TSC ドメイン名称 (1 ~ 31 文字の英数字) です。
dd..dd	TSC 識別子 (1 ~ 31 文字の英数字) です。
ee..ee	接続経路を次の値で示します。 tscd : TSC デーモンに直結 reg : TSC レギュレータを経由して TSC デーモンに接続
ff..ff	接続方法を次の値で示します。 bind : TPBroker のスマートエージェントの使用 ior : IOR 文字列の使用
gg..gg	IP アドレスです。
hh..hh	ポート番号です。

注

接続先情報の数だけ繰り返します。

ソートキー

一つのオプションに複数指定できるコマンドオプション引数は、接続先情報の取得または編集時のソートキーとして使用されます。つまり、接続先情報ファイルの各レコードは、コマンドオプション引数の指定順序に従ってソートされます。

また、各オプションにはソートキーとしての優先順位があります。コマンドオプション引数の指定順序に従うソートは、より優先順位の高いオプションのコマンドオプション引数が同じレコードに対して実行されます。接続先情報ファイルのソートキーの優先順位を高い順に示します。

1. -TSCDomain オプション
2. -TSCHost オプション
3. -TSCID オプション

-TSCDomainAll オプションを指定したときに取得される TSC ドメイン名称、

-TSCIDAll オプションを指定したときに取得される TSC 識別子、および -TSCHost オプションを省略したときに取得されるホスト名ではソートされません。ソートキーとして有効なオプションでソートされる以外では順序は不定です。

注意事項

コマンドオプション引数を複数指定する場合、区切り文字としてコンマ(,)を指定してください。また、引数とコンマ(,)の間には空白を入れないでください。

-TSCInFile オプションを省略して接続先情報を取得する場合、次に示す状態のときは接続先情報を取得できません。

- tscgetref コマンドを実行するホストで、該当する TSC ドメインマネージャが開始されていない状態
- 該当する TSC ドメインが他ホストにあるときに、そのホストで TSC ドメインマネージャが開始されていない状態

-TSCInFile オプションを指定して接続先情報を編集する場合、複数の接続先情報ファイル内に同じデータがあるときは、重複を削除して出力されます。

tscholdra (TSC ルートアクセプタの閉塞)

形式

```
tscholdra [-h] [-TSCDomain TSCドメイン名称] [-TSCID TSC識別子]
           { -TSCRootAcceptor TSCルートアクセプタ登録名称 |
             -TSCRootAcceptorALL }
           [-TSCMyHost ホスト名称またはIPアドレス]
           [[-TSCRequestLeave] [-TSCTimeOut タイムアウト値] |
            [-TSCForceSession]]
```

機能

TSC ルートアクセプタを閉塞します。このあと、リクエストにエラーが返るようになります。コマンド実行時に `-TSCRequestLeave` オプションが指定されている場合、キューイング中のリクエストはサーバアプリケーションに振り分けられます。`-TSCRequestLeave` オプションが指定されていない場合は、例外で返ります。

オプション

`-h`

コマンドの使用方法が表示されます。

`-TSCDomain` TSC ドメイン名称

~ < 1 ~ 31 文字の英数字 > 《TSCDOMAIN》

TSC ルートアクセプタを閉塞する TSC デーモンの TSC ドメイン名称を指定します。

"TSC" または "tsc" で始まる文字列は指定しないでください。省略した場合は "TSCDOMAIN" が設定されます。

`-TSCID` TSC 識別子

~ < 1 ~ 31 文字の英数字, およびピリオド (.) > 《IP アドレス》

TSC ルートアクセプタを閉塞する TSC デーモンの TSC 識別子を指定します。"TSC" または "tsc" で始まる TSC 識別子は指定しないでください。ピリオド (.) は、IP アドレスを指定する場合だけ使用できます。省略した場合は、IP アドレスで TSC 識別子が設定されます。

例えば、ホストの IP アドレスが "172.17.112.43" のとき、"172.17.112.43" という文字列が TSC 識別子となります。

`-TSCRootAcceptor` TSC ルートアクセプタ登録名称

~ < 1 ~ 31 文字の英数字 >

閉塞する TSC ルートアクセプタ登録名称を指定します。

-TSCRootAcceptorALL

すべての TSC ルートアクセプタを一括して閉塞します。このとき、閉塞済みの TSC ルートアクセプタも含まれます。

-TSCMyHost ホスト名称または IP アドレス

～ < 1 ~ 64 文字の文字列 > 《hostname コマンドで取得されるホスト名称》

マルチホームドホスト環境で OTM が使用するホスト名称または IP アドレスを指定します。省略した場合は、hostname コマンドで取得されるホスト名称が設定されます。

-TSCRequestLeave

TSC ルートアクセプタ閉塞時に、スケジュール用キューにキューイング中のリクエストを破棄しないで TSC ルートアクセプタを閉塞します。破棄しなかったリクエストはサーバアプリケーションが取り出します。

tscholdra コマンドはキューイング中のすべてのリクエストが取り出されるのを、
-TSCTimeOut オプションの指定値まで待ち合わせます。リクエストの待ち合わせ中に tscrlesra コマンドによって閉塞解除された場合は、リクエストの受け付けを可能にして、tscholdra コマンドは正常終了します。

01-01 以前のバージョンの TSC ルートアクセプタに対して -TSCRequestLeave オプションを指定した場合、-TSCRequestLeave オプションは無視されて、滞留しているリクエストは破棄されます。

-TSCRequestLeave オプションは、-TSCForceSession オプションと同時に指定できません。

-TSCTimeOut タイムアウト値

～ < 符号なし整数 > ((0 ~ 2147483647)) 《180》(単位 : 秒)

TSC ルートアクセプタの閉塞処理の待ち時間を指定します。"0" を指定した場合、無限に待ち続けます。省略した場合は "180" が設定されます。

-TSCTimeOut オプションは、-TSCForceSession オプションと同時に指定できません。

-TSCForceSession

TSC ルートアクセプタの閉塞と同時に、強制的にセッションを閉塞します。

-TSCForceSession オプションを省略した場合は、開始しているセッションは引き続き呼び出せますが、新規のセッションは呼び出せません。

-TSCForceSession オプションは、-TSCRequestLeave オプションまたは -TSCTimeOut オプションと同時に指定できません。

戻り値

このコマンドは次に示す戻り値をシェルに返してから、処理を終了します。

戻り値	意味
0	正常終了しました。
1	コマンド処理中にエラーが発生したために異常終了しました。 出力されたメッセージに従って対策したあと、再度、コマンドを実行してください。
2	指定された TSC ルートアクセプタはすでに閉塞しています。 この戻り値は <code>-TSCRootAcceptor</code> オプションを指定した場合だけ返されます。

tscjver (jar ファイルのバージョン情報の出力)

形式

```
tscjver -h | ファイルパス
```

機能

OTM が提供する jar ファイルのバージョン情報を出力します。

オプション

-h

コマンドの使用方法が表示されます。

コマンド引数

ファイルパス

~ <文字列>

対象となるファイルパス (相対パスおよび絶対パス) を指定します。

戻り値

このコマンドは次に示す戻り値をシェルに返してから、処理を終了します。

戻り値	意味
0	正常終了しました。
0 以外	コマンド処理中にエラーが発生したために異常終了しました。 出力されたメッセージに従って対策したあと、再度、コマンドを実行してください。

出力形式

```
Information For : aa..aa
Product Name   : bb..bb
PP Number      : cc..cc
Version        : dd..dd
Copyright      : ee..ee
Company        : ff..ff
Build OS       : gg..gg
Build Compiler : hh..hh
Build Date     : ii..ii jj..jj
```

(凡例)

記号	説明
aa..aa	ファイル名 (文字列) です。
bb..bb	製品名称 (文字列) です。
cc..cc	PP 形名 (文字列) です。
dd..dd	バージョンおよびリビジョン (文字列) です。
ee..ee	コピーライト (文字列) です。
ff..ff	会社名 (文字列) です。
gg..gg	ビルド OS (文字列) です。
hh..hh	ビルドコンパイラ (文字列) です。
ii..ii	ビルド日付 (文字列) です。
jj..jj	ビルド時刻 (文字列) です。

tsclogcat (メッセージの編集と出力)

形式

```
tsclogcat [-h] | [-d | -t TSC識別子]
```

機能

OTM で出力したメッセージを編集して標準出力に出力します。オプションによって TSC ログを出力するか、TSC ドメインログを出力するかを指定できます。

オプション

-h

コマンドの使用方法が表示されます。

-d

TSC ドメインログを出力します。

-t TSC 識別子

~ < 1 ~ 31 文字の英数字 > 《IP アドレス》

指定する TSC 識別子の TSC ログを出力します。

マルチホームドホスト環境で TSC ログを出力する場合は、-t オプションは省略できません。

すべてのオプションを省略した場合は、IP アドレスで TSC 識別子が設定されます。例えば、ホストの IP アドレスが "192.17.113.19" の場合、"-t 192.17.113.19" が設定されます。

戻り値

このコマンドは次に示す戻り値をシェルに返してから、処理を終了します。

戻り値	意味
0	正常終了しました。 出力するデータがない (KFOT73591-I メッセージが出力された) 場合、および -h オプションを指定した場合も 0 が返されます。
0 以外	コマンド処理中にエラーが発生したために異常終了しました。 出力されたメッセージに従って対策したあと、再度、コマンドを実行してください。

注意事項

tsclogcat コマンド実行時には、TSCSPOOL 環境変数が設定されている必要があります。

す。各オプション指定時の TSCSPOOL 環境変数の設定方法を次に示します。

- -t オプションを指定する場合、該当する TSC 識別子の動作している実行環境に設定した TSCSPOOL 環境変数が、tslogcat コマンドの実行環境にも必要です。
- -d オプションを指定する場合、TSC ドメインマネージャ実行時に指定した TSCSPOOL 環境変数が必要です。

tsclscon (コネクション情報の出力)

形式

```
tsclscon { [-h] |
           [-TSCDomain TSCドメイン名称] [-TSCID TSC識別子]
           [-TSCMyHost ホスト名称またはIPアドレス]
           [-TSCList] [-TSCReg プロセスID [,プロセスID...]]
           [-TSCOrb アプリケーション識別子 [,アプリケーション識別子...]] }
```

機能

コンピュータ内の TSC デモン、および TSC デモンに接続するプロセスについての情報を出力します。

オプション

-h

コマンドの使用方法が表示されます。

-TSCDomain TSC ドメイン名称

~ < 1 ~ 31 文字の英数字 > 《TSCDOMAIN》

情報取得対象の TSC デモンが属する TSC ドメイン名称を指定します。"TSC" または "tsc" で始まる名称は指定しないでください。省略した場合は "TSCDOMAIN" が設定されます。

-TSCID TSC 識別子

~ < 1 ~ 31 文字の英数字、およびピリオド(.) > 《IP アドレス》

情報取得対象の TSC デモンの TSC 識別子を指定します。"TSC" または "tsc" で始まる TSC 識別子は指定しないでください。ピリオド(.) は、IP アドレスを指定する場合だけ使用できます。省略した場合は、IP アドレスで TSC 識別子が設定されます。

例えば、ホストの IP アドレスが "172.17.112.43" のとき、"172.17.112.43" という文字列が TSC 識別子となります。

-TSCMyHost ホスト名称または IP アドレス

~ < 1 ~ 64 文字の文字列 > 《hostname コマンドで取得されるホスト名称》

マルチホームドホスト環境で OTM が使用するホスト名称または IP アドレスを指定します。省略した場合は、hostname コマンドで取得されるホスト名称が設定されます。

-TSCList

接続プロセス情報がリスト形式で出力されます。

-TSCReg プロセス ID[, プロセス ID...]

~ < 符号なし整数 >

TSC レギュレータ情報として出力する TSC レギュレータのプロセス ID を指定します。指定したプロセス ID で TSC デーモンへの接続が完了している TSC レギュレータの情報だけが出力されます。

-TSCList オプションとともに指定した場合には、TSC レギュレータの情報と、TSC レギュレータの接続プロセス情報が出力されます。

プロセス ID を複数指定する場合、区切り文字としてコンマ (,) を指定してください。また、プロセス ID とコンマ (,) の間には空白を入れないでください。

-TSCORb アプリケーション識別子 [, アプリケーション識別子 ...]

~ < 1 ~ 32 文字の英数字 >

TSCORB コネクタ情報として出力する、TSCORB コネクタのアプリケーション識別子を指定します。指定したアプリケーション識別子で TSC デーモンへの接続が完了している TSCORB コネクタの情報だけが出力されます。

-TSCList オプションとともに指定した場合には、TSCORB コネクタの情報と、TSCORB コネクタへ接続している ORB クライアントアプリケーションの詳細情報が出力されます。

なお、-TSCORb オプションを使用して TSCORB コネクタの情報を取得するには、OTM - Connector for ORB が必要です。OTM - Connector for ORB については、サポート時期をご確認ください。

戻り値

このコマンドは次に示す戻り値をシェルに返してから、処理を終了します。

戻り値	意味
0	正常終了しました。
0 以外	コマンド処理中にエラーが発生したために異常終了しました。 出力されたメッセージに従って対策したあと、再度、コマンドを実行してください。

出力形式

```

*** Command Edit Time aa..aa ***
*** tscd Information ***
TSCDomain  bb..bb
TSCID      cc..cc
Pid        dd..dd
IP         ee..ee
Start-Time ff..ff
CL-Max     gg..gg
CL-Peak    hh..hh
CL-Peak-Time ii..ii
SV-Max     jj..jj
SV-Peak    kk..kk
SV-Peak-Time ll..ll
CL-Count   mm..mm
SV-Count   nn..nn
REG-Count  oo..oo
OGW-Count  pp..pp
TCL-Count  qq..qq
OCL-Count  rr..rr
*** Connection Information ***
  No Kind  Pid IP      Connect-Time
ssss tttt dd..dd ee..ee  uu..uu
          :
ssss tttt dd..dd ee..ee  uu..uu
-----
*** tscrgltd[vv..vv] Information ***
Pid      dd..dd
IP       ee..ee
Connect-Time uu..uu
CL-Max   gg..gg
CL-Peak  hh..hh
CL-Peak-Time ii..ii
CL-Count mm..mm
*** Connection Information ***
  No Kind  Pid IP      Connect-Time
ssss tttt dd..dd ee..ee  uu..uu
          :
ssss tttt dd..dd ee..ee  uu..uu
-----
*** tscorbd[ww..ww] Information ***
APID     xx..xx
Pid      dd..dd
IP       ee..ee
Connect-Time uu..uu
CL-Max   gg..gg
CL-Peak  hh..hh
CL-Peak-Time ii..ii
CL-Count mm..mm
*** Connection Information ***
  No Kind  Port IP      Connect-Time
ssss tttt yy..yy ee..ee  uu..uu
          :
ssss tttt yy..yy ee..ee  uu..uu
-----

```

TSCデーモン情報 (1)

TSCデーモンの
接続プロセス情報 (2)

TSCレギュレータ情報 (3)

TSCレギュレータの
接続プロセス情報 (4)

TSCORBコネクタ情報 (5)

TSCORBコネクタの
接続プロセス情報 (6)

(凡例)

記号	説明
aa..aa	コマンド実行日時 (文字列) です。
bb..bb	TSC デーモンの TSC ドメイン名称 (文字列) です。
cc..cc	TSC デーモンの TSC 識別子 (文字列) です。
dd..dd	プロセス ID (数字) です。
ee..ee	IP アドレス (文字列) です。
ff..ff	TSC デーモンが開始した日時 (文字列) です。
gg..gg	TSC デーモン情報の場合は、tscstart コマンドの -TSCClientConnectCount オプションで指定した接続クライアントの最大数 (数字) です。TSC レギュレータ情報または TSCORB コネクタ情報の場合は、tscregld コマンドまたは tscstartgw コマンドの -TSCClientConnectCount オプションで指定したコネクション数 (数字) です。
hh..hh	TSC デーモンの接続プロセス情報の場合は、TSC デーモンに直結していたクライアントアプリケーション、TSC レギュレータ、および TSCORB コネクタの接続ピーク値 (数字) です。TSC レギュレータの接続プロセス情報の場合は、TSC レギュレータに接続していたクライアントアプリケーションの接続ピーク値 (数字) です。TSCORB コネクタの接続プロセス情報の場合は、TSCORB コネクタに接続していた ORB クライアントアプリケーションの接続ピーク値 (数字) です。
ii..ii	接続ピーク値が記録された日時 (文字列) です。
jj..jj	TSC デーモンの開始時に、tscstart コマンドの -TSCServerConnectCount オプションで指定した接続サーバアプリケーションの最大数 (数字) です。
kk..kk	TSC デーモンに接続したサーバアプリケーションの接続ピーク値 (数字) です。
ll..ll	接続ピーク値が記録された日時 (文字列) です。
mm..mm	TSC デーモン情報の場合は、コマンド実行時に TSC デーモンに直結していたクライアントアプリケーション数 (数字) です。TSC レギュレータ情報の場合は、コマンド実行時に TSC レギュレータに接続していたクライアントアプリケーション数 (数字) です。TSCORB コネクタ情報の場合は、コマンド実行時に TSCORB コネクタに接続していた ORB クライアントアプリケーション数 (数字) です。
nn..nn	コマンド実行時に TSC デーモンに接続していたサーバアプリケーション数 (数字) です。
oo..oo	コマンド実行時に TSC デーモンに接続していた TSC レギュレータ数 (数字) です。
pp..pp	コマンド実行時に TSC デーモンに接続していた TSCORB コネクタ数 (数字) です。

記号	説明
qq..qq	コマンド実行時にすべての TSC レギュレータに接続していたすべてのクライアントアプリケーション数 (数字) です。
rr..rr	コマンド実行時にすべての TSCORB コネクタに接続していたすべての ORB クライアントアプリケーション数 (数字) です。
ssss	通番 (数字) です。9999 を超えて出力される場合は、0 に戻ります。
tttt	<p>プロセス種別を次の文字列で示します。</p> <p>REGD : TSC レギュレータ</p> <p>OGWD : TSCORB コネクタ</p> <p>SERV : サーバアプリケーション</p> <p>FCLT : TSC デーモンに接続しているクライアントアプリケーション</p> <p>TCLT : TSC レギュレータに接続しているクライアントアプリケーション</p> <p>TCLT は、TSC レギュレータの接続プロセス情報に出力されます。</p> <p>OCLT : TSCORB コネクタに接続している ORB クライアントアプリケーション</p> <p>OCLT は、TSCORB コネクタの接続プロセス情報に出力されます。</p> <p>UNKW : プロセス種別判定不可能</p> <p>UNKW は、次に示すどちらかの場合にも出力されることがあります。</p> <ul style="list-style-type: none"> • TSC デーモンに接続するクライアントアプリケーションまたはサーバアプリケーションが 02-00 以前のライブラリを使用している場合 • 上記以外のプロセスが TSC デーモンまたは TSC レギュレータに接続している場合
uu..uu	TSC デーモンまたは TSC レギュレータへの接続日時 (文字列) です。
vv..vv	TSC レギュレータの数を表す通番 (数字) です。
ww..ww	TSCORB コネクタの数を表す通番 (数字) です。
xx..xx	TSCORB コネクタの開始時のアプリケーション識別子 (文字列) です。
yy..yy	ORB クライアントの送信側のポート番号 (数字) です。TPBroker のクライアントアプリケーションと TSCORB コネクタが同じマシンにある場合に、UNIX ドメインプロトコルが使用されたときは、ポート番号には "0" が出力されます。

注

-TSCList オプション、-TSCReg オプション、および -TSCOrb オプションの組み合わせによって、出力される内容が異なります。オプションの指定内容と、出力形式で示した (1) ~ (6) の部分の出力の有無の関係を次の表に示します。

オプションの指定			出力される内容					
-TSCList	-TSCReg	-TSCOrb	(1)	(2)	(3)	(4)	(5)	(6)
なし	なし	なし		x	1	x	2	x
あり	なし	なし			1	1	2	2
なし	あり	なし	x	x	3	x	x	x
あり	あり	なし	x	x	3	3	x	x
なし	なし	あり	x	x	x	x	4	x
あり	なし	あり	x	x	x	x	4	4
なし	あり	あり	x	x	3	x	4	x
あり	あり	あり	x	x	3	3	4	4

(凡例)

: 出力されます。

x : 出力されません。

注 1

接続している TSC レギュレータ数の分だけ出力されます。

注 2

接続している TSCORB コネクタ数の分だけ出力されます。

注 3

指定したプロセス ID のうち、TSC デーモンへ接続している数の分だけ出力されません。

注 4

指定したアプリケーション識別子のうち、TSC デーモンへ接続している数の分だけ出力されます。

注意事項

tsclscon コマンドは、情報取得対象となる TSC デーモン開始時の TSCSPOOL 環境変数に指定されたディレクトリを基に情報を取得します。そのため、対象となる TSC デーモンと同じマシン上、かつ同じ TSCSPOOL 環境変数の環境で、tsclscon コマンドを実行してください。

OTM - Client の Java アプリケーションが TSC デーモンまたは TSC レギュレータに接続している場合、接続プロセス情報のプロセス ID に "0" が出力されます。

プロセス種別が UNKW の場合、接続プロセス情報のプロセス ID に "0" が出力されることがあります。

tsclscon コマンドでは、TSC デーモンのオンライン中の処理性能を確保するために、共用メモリの参照で排他処理は実施しません。また、接続していたプロセスの異常終了を検知する前に tsclscon コマンドを実行した場合、更新されていない情報が出力されることがあります。

クライアントアプリケーションが 02-00 以前のライブラリを使用して TSC レギュレータに接続している場合、接続プロセス情報のプロセス ID に "0" が出力されることがあります。

クライアントアプリケーションが 02-00 以前のライブラリを使用して TSC レギュレータに接続している場合、接続プロセス情報の IP アドレスに "0.0.0.0" が出力されることがあります。

ORB クライアントと TSCORB コネクタとの間に、NAT などの IP アドレスの変換機能がある場合、クライアントに割り当てた実際の IP アドレスは出力されません。また、TPBroker の ORB クライアントを使用して、同じマシンから OTM のサーバアプリケーションのメソッドを呼び出した場合は、IP アドレスがループバックアドレス (127.0.0.1) で出力されることがあります。

tsclsra (TSC ルートアクセプタ情報の出力)

形式

```
tsclsra [-h] [-TSCDomain TSCドメイン名称] [-TSCID TSC識別子]
         [-TSCRootAcceptor TSCルートアクセプタ登録名称]
         [-TSCMyHost ホスト名称またはIPアドレス]
         [[-TSCInterface [インタフェース名称]]
          [-TSCHighPrior] | [-TSCSession]]
```

機能

TSC ルートアクセプタ内のスケジューリング情報を出力します。

オプション

-h

コマンドの使用方法が表示されます。

-TSCDomain TSC ドメイン名称

~ < 1 ~ 31 文字の英数字 > 《TSCDOMAIN》

TSC ルートアクセプタ情報を出力する TSC デーモンの TSC ドメイン名称を指定します。
"TSC" または "tsc" で始まる名称は指定しないでください。省略した場合は
"TSCDOMAIN" が設定されます。

-TSCID TSC 識別子

~ < 1 ~ 31 文字の英数字 , およびピリオド (.) > 《IP アドレス》

TSC ルートアクセプタ情報を出力する TSC デーモンの識別子を指定します。"TSC" または "tsc" で始まる TSC 識別子は指定しないでください。ピリオド (.) は、IP アドレスを指定する場合だけ使用できます。省略した場合は、IP アドレスで TSC 識別子が設定されます。

例えば、ホストの IP アドレスが "172.17.112.43" のとき、"172.17.112.43" という文字列が TSC 識別子となります。

-TSCRootAcceptor TSC ルートアクセプタ登録名称

~ < 1 ~ 31 文字の英数字 >

出力する TSC ルートアクセプタの登録名称を指定します。省略した場合は、すべての TSC ルートアクセプタの情報が出力されます。

-TSCMyHost ホスト名称または IP アドレス

~ < 1 ~ 64 文字の文字列 > 《hostname コマンドで取得されるホスト名称》

マルチホームドホスト環境で OTM が使用するホスト名称または IP アドレスを指定します。省略した場合は、hostname コマンドで取得されるホスト名称が設定されます。

-TSCInterface インタフェース名称

~ < 1 文字以上の文字列 >

出力するインタフェースの名称を指定します。インタフェース名称を指定した場合、指定したインタフェース名称と TSC アクセプタ名称が出力されます。

-TSCInterface オプションだけを指定した場合、すべてのインタフェース名称と TSC アクセプタ名称が出力されます。

-TSCInterface オプションは、-TSCSession オプションと同時に指定できません。

-TSCHighPrior

TSC デーモンの開始時に tsctest コマンドの -TSCHighPriorBorder オプションに指定された値以下の、高いプライオリティのリクエストに関する情報を出力します。

-TSCHighPrior オプションは、-TSCSession オプションと同時に指定できません。

-TSCSession

起動している TSC ルートアクセプタごとのセッションに関する情報を出力します。

-TSCSession オプションは、-TSCInterface オプションまたは -TSCHighPrior オプションと同時に指定できません。

戻り値

このコマンドは次に示す戻り値をシェルに返してから、処理を終了します。

戻り値	意味
0	正常終了しました。
0 以外	コマンド処理中にエラーが発生したために異常終了しました。 出力されたメッセージに従って対策したあと、再度、コマンドを実行してください。

出力形式

-TSCInterface オプションを指定しない場合の出力形式

```
Domain_Name:aa..aa TSCID:bb..bb
RootAcceptor_Name Status RA_Count Request_Count Request_Max
cc..cc          d      ee..ee ff..ff          gg..gg
```

(凡例)

記号	説明
aa..aa	指定した TSC ドメイン名称 (1 ~ 31 文字の英数字) です。
bb..bb	指定した TSC 識別子 (1 ~ 31 文字の英数字) です。
cc..cc	指定した条件と一致する TSC ルートアクセプタ名称 (1 ~ 31 文字の英数字) です。
d	指定した条件と一致する TSC ルートアクセプタの状態を次の値で示します。 S : TSC ルートアクセプタ準備中 A : スケジューリング可能状態 E : TSC ルートアクセプタ終了処理中 H : TSC ルートアクセプタ閉塞中
ee..ee	TSC ルートアクセプタの登録数 (10 けたの 10 進数) です。
ff..ff	スケジューリング用キューに滞留しているリクエスト数 (10 けたの 10 進数) です。
gg..gg	スケジューリング用キューに滞留したリクエスト最大数 (10 けたの 10 進数) です。

注

指定した条件と一致する TSC ルートアクセプタが複数ある場合、一致する TSC ルートアクセプタの数だけ、cc..cc ~ gg..gg の行が繰り返し出力されます。

-TSCInterface オプションを指定する場合の出力形式

```
Domain_Name : aa..aa TSCID : bb..bb
RootAcceptor_Name Status RA_Count Request_Count Request_Max Acceptor_Name Interface_Name
cc..cc          d      ee..ee ff..ff          gg..gg          ll..ll          mm..mm
```

(凡例)

記号	説明
aa..aa	指定した TSC ドメイン名称 (1 ~ 31 文字の英数字) です。
bb..bb	指定した TSC 識別子 (1 ~ 31 文字の英数字) です。

記号	説明
cc..cc	指定した条件と一致する TSC ルートアクセプタ名称 (1 ~ 31 文字の英数字) です。
d	指定した条件と一致する TSC ルートアクセプタの状態を次の値で示します。 S : TSC ルートアクセプタ準備中 A : スケジューリング可能状態 E : TSC ルートアクセプタ終了処理中 H : TSC ルートアクセプタ閉塞中
ee..ee	TSC ルートアクセプタの登録数 (10 けたの 10 進数) です。
ff..ff	スケジューリング用キューに滞留しているリクエスト数 (10 けたの 10 進数) です。
gg..gg	スケジューリング用キューに滞留したリクエスト最大数 (10 けたの 10 進数) です。
ll..ll	mm..mm に出力されているインタフェースの TSC アクセプタ名称 (1 ~ 31 文字の英数字) です。
mm..mm	指定した条件と一致するインタフェース名称 (1 文字以上の英数字) です。

注

指定した条件と一致するインタフェースが複数ある場合、一致するインタフェースの数だけ、枠で囲んだ部分が繰り返し出力されます。

-TSCHighPrior オプションを指定する場合の出力形式

```
Domain_Name : aa..aa  TSCID : bb..bb
RootAcceptor_Name Status RA_Count Request_Count Request_Max HPBorder HPRequest HPRequest_Count HPRequest_Max
cc..cc          d      ee..ee  ff..ff          gg..gg          h          ii..ii          jj..jj          kk..kk
Accepter_Name Interface_Name
ll..ll          mm..mm
```

(凡例)

記号	説明
aa..aa	指定した TSC ドメイン名称 (1 ~ 31 文字の英数字) です。
bb..bb	指定した TSC 識別子 (1 ~ 31 文字の英数字) です。
cc..cc	指定した条件と一致する TSC ルートアクセプタ名称 (1 ~ 31 文字の英数字) です。
d	指定した条件と一致する TSC ルートアクセプタの状態を次の値で示します。 S : TSC ルートアクセプタ準備中 A : スケジューリング可能状態 E : TSC ルートアクセプタ終了処理中 H : TSC ルートアクセプタ閉塞中
ee..ee	TSC ルートアクセプタの登録数 (10 けたの 10 進数) です。

記号	説明
ff..ff	スケジュール用キューに滞留しているリクエスト数 (10 けたの 10 進数) です。
gg..gg	スケジュール用キューに滞留したリクエスト最大数 (10 けたの 10 進数) です。
h	TSC デーモンの開始時に、tscstart コマンドの -TSCHighPriorBorder オプションで指定した高プライオリティしきい値 (1 けたの 10 進数) です。
ii..ii	TSC デーモンの開始時に、tscstart コマンドの -TSCHighPriorRequestCount オプションで指定した高プライオリティリクエスト数 (5 けたの 10 進数) です。
jj..jj	スケジュール用キューに滞留している -TSCHighPriorBorder オプション指定値以上のプライオリティが指定されたリクエスト数 (5 けたの 10 進数) です。
kk..kk	スケジュール用キューに滞留している -TSCHighPriorBorder オプション指定値以上のプライオリティが指定されたリクエスト最大数 (5 けたの 10 進数) です。
ll..ll	mm..mm に出力されているインタフェースの TSC アクセプタ名称 (1 ~ 31 文字の英数字) です。
mm..mm	指定した条件と一致するインタフェース名称 (1 文字以上の英数字) です。

注

指定した条件と一致する TSC ルートアクセプタが複数ある場合、一致する TSC ルートアクセプタの数だけ、cc..cc ~ mm..mm の行が繰り返し出力されます。また、指定した条件と一致するインタフェースが複数ある場合、一致するインタフェースの数だけ、枠で囲んだ部分が繰り返し出力されます。

-TSCSession オプションを指定する場合の出力形式

```

*** tscd Information ***
TSCDomain          aa..aa
TSCID              bb..bb
-----
*** RootAcceptor Information ***
RootAcceptor       cc..cc
Status             dd
OccupyThread       ee..ee
MaxThread          ff..ff
*** Session Information ***
CL_IPAddr CL_ProcessId SesStart WaitStart SesInterval
  gg..gg   hh..hh   ii..ii   jj..jj   kk..kk
  :       :       :       :       :
  :       :       :       :       :
  gg..gg   hh..hh   ii..ii   jj..jj   kk..kk
-----
*** RootAcceptor Information ***
RootAcceptor       cc..cc
Status             dd
OccupyThread       ee..ee
MaxThread          ff..ff
*** Session Information ***
CL_IPAddr CL_ProcessId SesStart WaitStart SesInterval
  gg..gg   hh..hh   ii..ii   jj..jj   kk..kk
  :       :       :       :       :
  :       :       :       :       :
  gg..gg   hh..hh   ii..ii   jj..jj   kk..kk
-----
:
:

```

} TSCデーモン情報
 } TSCルートアクセプタ情報 (1)
 } TSCルートアクセプタのセッション情報 (2)
 } TSCルートアクセプタ情報 (1)
 } TSCルートアクセプタのセッション情報 (2)

(凡例)

記号	説明
aa..aa	指定した TSC ドメイン名称 (1 ~ 31 文字の英数字) です。
bb..bb	指定した TSC 識別子 (1 ~ 31 文字の英数字) です。
cc..cc	指定した条件と一致する TSC ルートアクセプタ名称 (1 ~ 31 文字の英数字) です。
dd	指定した条件と一致する TSC ルートアクセプタの状態を次の値で示します。 S : TSC ルートアクセプタ準備中 A : スケジューリング可能状態 E : TSC ルートアクセプタ終了処理中 H : TSC ルートアクセプタ閉塞中 HS : セッション呼び出し閉塞中
ee..ee	セッション呼び出し機能で占有されたスレッド数 (5 けたの 10 進数) です。
ff..ff	TSC ルートアクセプタが管理しているスレッド数 (5 けたの 10 進数) です。

記号	説明
gg..gg	セッション呼び出し元クライアントの IP アドレス (文字列) です。
hh..hh	セッション呼び出し元クライアントのプロセス ID (10 けたの 10 進数) です。
ii..ii	セッション情報を生成した時刻 (19 文字の文字列) です。
jj..jj	セッション呼び出しインターバル時間監視を開始した時刻 (19 文字の文字列) です。リクエスト待ちをしていない場合には、 - (ハイフン) が出力されます。
kk..kk	セッション呼び出しインターバル監視時間 (10 けたの 10 進数) です。

注

- 指定した条件に一致する TSC ルートアクセプタが複数ある場合、一致する TSC ルートアクセプタの数だけ、(1) および (2) の情報が繰り返し出力されます。
- セッション呼び出しが行われている場合、呼び出しが行われているセッションの数だけ、gg..gg ~ kk..kk の行が繰り返し出力されます。
- OTM - Client の Java アプリケーションが TSC デーモンまたは TSC レギュレータに接続している場合、hh..hh (セッション呼び出し元クライアントのプロセス ID) に "0" が出力されます。

tscprcls (プロセス情報の出力)

形式

```
tscprcls [-h] [-TSCID TSC識別子] [-TSCDomain TSCドメイン名称]
          [-TSCtscd] [-TSCMyHost ホスト名称またはIPアドレス]
```

機能

コンピュータ内の TSC デーモンおよび TSC ドメインマネージャが管理するプロセスに関する情報を出力します。TSC ドメインマネージャが管理するプロセスは、自マシンでの該当ドメインに属する TSC デーモンを指します。TSC デーモンが管理するプロセスについては、「2.4.6 プロセスの管理」を参照してください。

オプション

-h

コマンドの使用方法が表示されます。

-TSCID TSC 識別子

~ < 1 ~ 31 文字の英数字、およびピリオド (.) > 《IP アドレス》

出力対象の TSC デーモンの TSC 識別子を指定します。"TSC" および "tsc" で始まる文字列は指定しないでください。ピリオド (.) は、IP アドレスを指定する場合だけ使用できます。省略した場合は、IP アドレスで TSC 識別子が設定されます。

例えば、ホストの IP アドレスが "172.17.112.43" のとき、"172.17.112.43" という文字列が TSC 識別子となります。

-TSCDomain TSC ドメイン名称

~ < 1 ~ 31 文字の英数字 > 《TSCDOMAIN》

-TSCtscd オプションを省略する場合、出力対象の TSC デーモンが属する TSC ドメイン名称を指定します。

-TSCtscd オプションを指定する場合、出力対象の TSC ドメインマネージャの TSC ドメイン名称を指定します。

指定するときは、"TSC" および "tsc" で始まる文字列は指定しないでください。省略した場合は "TSCDOMAIN" が設定されます。

-TSCtscd

TSC ドメインマネージャに登録済みのプロセスに関する情報を出力します。省略した場合は、指定された TSC デーモンに登録済みのプロセスに関する情報が出力されます。

-TSCMyHost ホスト名称または IP アドレス

~ < 1 ~ 64 文字の文字列 > 《hostname コマンドで取得されるホスト名称》

マルチホームドホスト環境で OTM が使用するホスト名称または IP アドレスを指定します。省略した場合は、hostname コマンドで取得されるホスト名称が設定されます。

戻り値

このコマンドは次に示す戻り値をシェルに返してから、処理を終了します。

戻り値	意味
0	正常終了しました。
0 以外	コマンド処理中にエラーが発生したために異常終了しました。 出力されたメッセージに従って対策したあと、再度、コマンドを実行してください。

出力形式

TSC デーモンが管理するプロセス情報の出力形式

```
No |PID |KIND|S-TIME |DIR-ID |WITH |SV_NAME|APID |STATUS
aaaa|bbbb|ccc |dd..dd |ee..ee |ff..ff |gg..gg |hh..hh |iiiiii
:
```

(凡例)

記号	説明
aaaaa	項番 (数字) です。
bbbbbb	プロセス ID (文字列) です。
ccc	プロセス種別 (文字列) を次の値で示します。 <ul style="list-style-type: none"> • Srv : サーバアプリケーション • Clt : クライアントアプリケーション • Reg : TSC レギュレータ • Prf : tscprfd デーモン • Orb : TSCORB コネクタ tscstartprc コマンドで開始するアプリケーションプログラムの場合、アプリケーションプロセスが開始され、種別が特定できるまで "-" が出力されることがあります。
dd..dd	プロセスの開始時刻 (文字列) です。
ee..ee	カレントディレクトリ ID です。
ff..ff	内部情報です。

記号	説明
gg..gg	<p>実行形式ファイル名称 (文字列) です。</p> <p>tscstartprc コマンドで開始するアプリケーションプログラムの場合は、-TSCAPName オプションに指定した値が出力されま す。クライアントアプリケーション、および tscstartprc コマ ンドで開始しないアプリケーションプログラムの場合は、argv[0] の値が出力されます。Java アプリケーションの場合は "-" が出 力されます。</p> <p>なお、フルパス指定の場合のディレクトリ名は出力されません。 実行形式ファイル名称が 31 文字を超える場合、先頭 31 文字が 出力されます。</p>
hh..hh	<p>アプリケーション識別子 (文字列) です。</p> <p>クライアントアプリケーション、およびサーバアプリケーション の場合、-TSCAPID オプションが指定されなかったときは "-" が出力されます。</p>
iiiiii	<p>プロセスステータス (文字列) です。</p> <p>クライアントアプリケーションの場合は、次の値で示します。</p> <ul style="list-style-type: none"> • LIVING : initClient() から終了要求受付まで • DYING : 終了要求受付から endClient() まで • DEAD : initClient() 以前または endClient() 以降 <p>サーバアプリケーションの場合は、次の値で示します。</p> <ul style="list-style-type: none"> • LIVING : initServer() から serverMainloop() まで • ACTIVE : serverMainloop() から終了要求受付まで • DYING : 終了要求受付から endServer() まで • DEAD : initServer() 以前または endServer() 以降 <p>TSC レギュレータまたは TSCORB コネクタの場合は、次の値 で示します。</p> <ul style="list-style-type: none"> • LIVING : 正常開始処理中 • ACTIVE : オンライン稼働中 • DYING : 正常終了処理中 • DEAD : 開始処理以前、または停止処理以降

TSC ドメインマネージャが管理する TSC デーモンの情報の出力形式

```

No      | PID      | KIND      | S-TIME    | TSCID
aa..aa  | bb..bb   | cc..cc    | dd..dd    | ee..ee
:
```

(凡例)

記号	説明
aa..aa	項番 (数字) です。
bb..bb	プロセス ID (文字列) です。
cc..cc	プロセス種別 (文字列) を次の値で示します。 Tscd : TSC デーモン

記号	説明
dd..dd	TSC デーモンの開始時刻 (文字列) です。
ee..ee	TSC 識別子 (文字列) です。

注意事項

tscprcls コマンドは、TSCSPOOL 環境変数に指定されたディレクトリを基に情報を取得するため、TSC ドメイン名称と TSCSPOOL の関係が正しい状態でコマンドを実行してください。

組み合わせが誤っている場合、エラーメッセージを出力して処理を終了します。

-TSCWithSystem オプションに "0" を指定して開始したクライアントアプリケーションの情報は出力されません。

tscrasget (トラブルシューティング情報の取得と出力)

形式

```
tscrasget { [-h] | [-p] [-s] [-TSCSpool TSCスプールディレクトリ |  
            -TSCDomain TSCドメイン名称] 出力先ディレクトリ }
```

機能

tscrasget コマンドを実行するホスト上で OTM が必要とするトラブルシューティング情報をファイルに取得し、取得した情報の一覧を標準出力または標準エラー出力に出力します。

オプション

-h

コマンドの使用方法が表示されます。

-p

TPBroker についてのトラブルシューティング情報を出力先ディレクトリに出力します。あらかじめ、TPBroker の VBROKER_ADM 環境変数を設定してください。

-s

共通メッセージ (UNIX の場合は syslog, Windows の場合は eventlog) の取得を抑制します。

-TSCSpool TSC スプールディレクトリ

トラブルシューティング情報を取得する TSC スプールディレクトリを指定します。

-TSCDomain TSC ドメイン名称

~ < 1 ~ 31 文字の英数字 >

トラブルシューティング情報を取得する TSC ドメイン名称を指定します。省略した場合は、コンピュータ内の OTM 環境に登録済みのすべての TSC ドメインからトラブルシューティング情報を取得します。

コマンド引数

出力先ディレクトリ

トラブルシューティング情報を出力するファイルのディレクトリ名 (相対パスまたは絶対パス) を指定します。

戻り値

このコマンドは次に示す戻り値をシェルに返してから、処理を終了します。

戻り値	意味
0	正常終了しました。
0 以外	コマンド処理中にエラーが発生したために異常終了しました。 出力されたメッセージに従って対策したあと、再度、コマンドを実行してください。

出力形式

```
TPBroker Object Transaction Monitor: RAS completed collection of
aa..aa
```

(凡例)

記号	説明
aa..aa	収集する情報、ファイル名称などです。

注意事項

出力先ディレクトリにすでに同名のファイルがある場合は上書きされるため、トラブルシュート情報を残しておきたい場合は、ファイルをコピーしておくか、新しく作成したディレクトリを出力先ディレクトリに指定してください。

OTM - Client の環境で TSC スプールディレクトリの情報を取得するには、
-TSCSpool オプションを指定してください。OTM - Client の環境で -TSCSpool オプションの指定がない場合は、TSCSPOOL 環境変数で指定したディレクトリの情報を取得します。

UNIX では、アプリケーションプログラムの異常終了時にコアファイルが出力されません。しかし、コアファイルは任意の実行ディレクトリに出力されるため、tscrasget コマンドではその情報を取得できません。そのため、障害対策時には、tscrasget コマンドの実行のほかにアプリケーションプログラムのコアファイルの取得が必要な場合があります。

tscregltd (TSC レギュレータの開始)

形式

```
tscregltd [-h] [-TSCID TSC識別子] [-TSCDomain TSCドメイン名称]
          [-TSCMyHost ホスト名称またはIPアドレス]
          [-TSCClientConnectCount コネクション数]
          [-TSCPort ポート番号]
          [-TSCIORConnect]
          [-TSCPool]
```

機能

クライアントアプリケーションからの要求をレギュレートするために、TSC レギュレータを開始します。TSC レギュレータは TSC デモンが動作するコンピュータと同じコンピュータで開始してください。

tscregltd コマンドはコマンド入力画面で動作しますので、必要に応じてバックグラウンドで実行してください。

オプション

-h

コマンドの使用方法が表示されます。

-TSCID TSC 識別子

~ < 1 ~ 31 文字の英数字、およびピリオド (.) > 《IP アドレス》

TSC デモンの識別子を指定します。"TSC" または "tsc" で始まる TSC 識別子は指定しないでください。ピリオド (.) は、IP アドレスを指定する場合だけ使用できます。省略した場合は、IP アドレスが TSC 識別子に設定されます。例えば、ホストの IP アドレスが "172.17.112.43" のとき、"172.17.112.43" という文字列が TSC 識別子となります。

-TSCDomain TSC ドメイン名称

~ < 1 ~ 31 文字の英数字 > 《TSCDOMAIN》

TSC デモンが属する TSC ドメイン名称を指定します。"TSC" または "tsc" で始まる TSC ドメイン名称は指定しないでください。省略した場合は "TSCDOMAIN" が設定されます。

-TSCMyHost ホスト名称または IP アドレス

~ < 1 ~ 64 文字の文字列 > 《hostname コマンドで取得されるホスト名称》

マルチホームドホスト環境で OTM が使用するホスト名称または IP アドレスを指定します。省略した場合は、hostname コマンドで取得されるホスト名称が設定されます。

-TSCID オプションを省略して -TSCMyHost オプションを指定した場合、デフォルトの TSC 識別子は -TSCMyHost オプションで指定した IP アドレスになります。

-TSCClientConnectCount コネクション数

~ < 符号なし整数 > ((1 ~ 32767)) 《64》

TSC レギュレータに接続できるコネクション数を指定します。コネクション数は、クライアントアプリケーションからの接続数です。省略した場合は "64" が設定されます。

-TSCPort ポート番号

~ < 符号なし整数 > ((5001 ~ 65535))

TSC レギュレータの新しいコネクションを探すときに使用するポート番号を指定します。省略した場合は、OS が自動的に割り当てたポート番号を使用します。

-TSCIORConnect

TSC レギュレータの IOR 接続を可能にし、IOR 文字列を接続先情報ファイルへ出力します。-TSCIORConnect オプションを指定する場合は、-TSCPort オプションも指定してください。-TSCIORConnect オプションを省略した場合は、TSC レギュレータの IOR 文字列は接続先情報ファイルへ出力されません。

-TSCTPool

TSC レギュレータをスレッドプーリング (ThreadPool) モデルで動作させます。省略した場合は、スレッドパーセッション (ThreadSession) で動作させます。スレッドプーリングおよびスレッドパーセッションの詳細については、マニュアル「Borland Enterprise Server VisiBroker デベロッパーズガイド」を参照してください。

戻り値

このコマンドは次に示す戻り値をシェルに返してから、処理を終了します。

戻り値	意味
0	正常終了しました。
0 以外	コマンド処理中にエラーが発生したために異常終了しました。 出力されたメッセージに従って対策したあと、再度、コマンドを実行してください。

注意事項

TSC レギュレータのオプションは、同じ TSC ノードに属する TSC デーモンの開始時に指定できます。TSC レギュレータの開始時に指定しなかったオプションのデフォルトには、TSC デーモンの開始時に指定された値が引き継がれます。

ただし、次に示すオプションについては、同じ TSC ノードの TSC デーモンの開始時に指定されていても引き継がれません。

- -TSCDomain オプション
- -TSCID オプション
- -TSCMyHost オプション
- -TSCPort オプション

TSC レギュレータは開始後にカレントディレクトリを移動します。そのため、次に示す場合のコアファイルの出力先は、実行形式ファイルのディレクトリとは異なります。

- "tscstop -TSCForce -TSCDebugInfo" または kill コマンドなどで TSC レギュレータを強制終了した場合
- TSC レギュレータが異常終了した場合

移動後のカレントディレクトリは、UNIX の場合は "\$TSCSPOOL/adm/TSC 識別子 / reg/ カレントディレクトリ ID" で、Windows の場合は "%TSCSPOOL%\adm\TSC 識別子 ¥reg¥ カレントディレクトリ ID" です。

カレントディレクトリ ID については、この章の「tscprcls (プロセス情報の出力)」で、出力形式を参照してください。

tscridinfo (TSC ドメイン情報の出力)

形式

```
tscridinfo [-h] | [-TSCDomain TSCドメイン名称]
            [-TSCHost ホストの完全修飾名またはIPアドレス]
            [-TSCMyHost ホスト名称またはIPアドレス]
```

機能

TSC ドメイン情報を出力します。

オプション

-h

コマンドの使用方法が表示されます。

-TSCDomain TSC ドメイン名称

~ < 1 ~ 31 文字の英数字 > 《TSCDOMAIN》

TSC ドメイン情報を出力する TSC ドメインマネージャの TSC ドメイン名称を指定します。"TSC" または "tsc" で始まる TSC ドメイン名称は指定しないでください。省略した場合は "TSCDOMAIN" が設定されます。

-TSCHost ホストの完全修飾名または IP アドレス

~ < 1 ~ 64 文字の文字列 > 《自ホストの IP アドレス》

-TSCDomain オプションで指定した TSC ドメイン名称に対応する TSC ドメインマネージャがあるホストの完全修飾名または IP アドレスを指定します。マルチホームドホスト環境では、OTM の使用する -TSCMyHost オプションに指定するホスト名称または IP アドレスを指定してください。

省略した場合は、自ホストの IP アドレスが設定されます。マルチホームドホスト環境では、省略時に設定される IP アドレスは OS の動作に依存します。

-TSCMyHost ホスト名称または IP アドレス

~ < 1 ~ 64 文字の文字列 > 《hostname コマンドで取得されるホスト名称》

マルチホームドホスト環境で OTM が使用するホスト名称または IP アドレスを指定します。省略した場合は、hostname コマンドで取得されるホスト名称が設定されます。

戻り値

このコマンドは次に示す戻り値をシェルに返してから、処理を終了します。

戻り値	意味
0	正常終了しました。
0 以外	コマンド処理中にエラーが発生したために異常終了しました。 出力されたメッセージに従って対策したあと、再度、コマンドを実行してください。

tscrsra (TSC ルートアクセプタの閉塞解除)

形式

```
tscrsra [-h] [-TSCDomain TSCドメイン名称] [-TSCID TSC識別子]
        { -TSCRootAcceptor TSCルートアクセプタ登録名称 |
          -TSCRootAcceptorALL }
        [-TSCMyHost ホスト名称またはIPアドレス]
```

機能

TSC ルートアクセプタの閉塞を解除します。このあと、リクエストの受付ができます。

オプション

-h

コマンドの使用方法が表示されます。

-TSCDomain TSC ドメイン名称

~ < 1 ~ 31 文字の英数字 > 《TSCDOMAIN》

TSC ルートアクセプタを閉塞解除する TSC デーモンの TSC ドメイン名称を指定します。"TSC" または "tsc" で始まる名称は指定しないでください。省略した場合は "TSCDOMAIN" が設定されます。

-TSCID TSC 識別子

~ < 1 ~ 31 文字の英数字, およびピリオド (.) > 《IP アドレス》

TSC ルートアクセプタを閉塞解除する TSC デーモンの識別子を指定します。"TSC" または "tsc" で始まる TSC 識別子は指定しないでください。ピリオド (.) は、IP アドレスを指定する場合だけ使用できます。省略した場合は、IP アドレスが TSC 識別子に設定されます。例えば、ホストの IP アドレスが "172.17.112.43" のとき、"172.17.112.43" という文字列が TSC 識別子となります。

-TSCRootAcceptor TSC ルートアクセプタ登録名称

~ < 1 ~ 31 文字の英数字 >

閉塞解除する TSC ルートアクセプタの登録名称を指定します。

-TSCRootAcceptorALL

すべての TSC ルートアクセプタを一括して閉塞解除します。閉塞解除済みの TSC ルートアクセプタを含めることができます。

-TSCMyHost ホスト名称または IP アドレス

~ < 1 ~ 64 文字の文字列 > 《hostname コマンドで取得されるホスト名称》

マルチホームドホスト環境で OTM が使用するホスト名称または IP アドレスを指定します。省略した場合は、hostname コマンドで取得されるホスト名称が設定されます。

戻り値

このコマンドは次に示す戻り値をシェルに返してから、処理を終了します。

戻り値	意味
0	正常終了しました。
1	コマンド処理中にエラーが発生したために異常終了しました。 出力されたメッセージに従って対策したあと、再度、コマンドを実行してください。
2	指定された TSC ルートアクセプタはすでに閉塞解除しています。 この戻り値は、-TSCRootAcceptor オプションを指定した場合だけ返されます。

tscstart (TSC デーモンの開始)

形式

```
tscstart [-h] [-TSCDomain TSCドメイン名称] [-TSCID TSC識別子]
[-TSCPort ポート番号]
[-TSCMyHost ホスト名称またはIPアドレス]
[-TSCEntryCount 登録プロセス数] [-TSCNice nice値]
[-TSCRegStart プロセス数]
[-TSCRegOption コマンドオプションファイル名称]
[-TSCEnviron ユーザ環境変数定義ファイル名称]
[-TSCStartTimeOut タイムアウト値]
[-TSCClientConnectCount 接続クライアントの最大数]
[-TSCServerConnectCount 接続サーバアプリケーションの最大数]
[-TSCLogFileSize TSCログファイル最大サイズ]
[-TSCLogFileCount TSCログファイル最大数]
[-TSCRootAcceptorCount TSCルートアクセプタ数]
[-TSCRootAcceptorRegistCount TSCルートアクセプタ登録可能数]
[-TSCServerCacheSize サーバキャッシュサイズ]
[-TSCDispatchPolicy normalDispatch | priorDispatch]
[-TSCDispatchParallelCount スレッドの最大値]
[-TSCMaxRequestCount リクエスト数]
[-TSCQueueDeleteWait TSCルートアクセプタキュー保留時間]
[-TSCHighPriorRequestCount 高プライオリティリクエスト数]
[-TSCHighPriorBorder 高プライオリティしきい値]
[-TSCStatsUse Y | N] [-TSCStatsFileCount ファイル数]
[-TSCStatsFileSize ファイルサイズ]
```

機能

リクエストのキューイングや振り分けをするための TSC デーモンを開始します。

オプション

-h

コマンドの使用方法が表示されます。

-TSCDomain TSC ドメイン名称

~ < 1 ~ 31 文字の英数字 > 《TSCDOMAIN》

TSC デーモンが属する TSC ドメイン名称を指定します。"TSC" または "tsc" で始まる TSC ドメイン名称は指定しないでください。また、TSC 識別子と TSC ドメイン名称には同じ値を指定しないでください。省略した場合は "TSCDOMAIN" が設定されます。

-TSCID TSC 識別子

~ < 1 ~ 31 文字の英数字、およびピリオド (.) > 《IP アドレス》

TSC デーモンの識別子を指定します。"TSC" または "tsc" で始まる TSC 識別子は指定しないでください。また、同一 TSC ドメイン内で重複する TSC 識別子は使用できません。

ピリオド (.) は、IP アドレスを指定する場合だけ使用できます。省略した場合は、IP アドレスが TSC 識別子に設定されます。例えば、ホストの IP アドレスが "172.17.112.43" のときは、"172.17.112.43" という文字列が TSC 識別子となります。

-TSCPort ポート番号

~ < 符号なし整数 > ((5001 ~ 65535)) 《20138》

TSC デーモンの新しいコネクションを探すときに使用するポート番号を指定します。省略した場合はサービス名ファイルが参照されます。サービス名ファイルに指定がないときは "20138" が設定されます。

サービス名ファイルへの記述の形式を次に示します。

TSC_TSCドメイン名称TSC識別子 ポート番号/tcp

例えば、"tscstart -TSCDomain JAPAN -TSCID Hitachi" と指定して TSC デーモンを開始する場合、サービス名ファイルの、次のように指定したエントリが読み込まれます。

TSC_JAPANHitachi 10021/tcp

-TSCMyHost ホスト名称または IP アドレス

~ < 1 ~ 64 文字の文字列 > 《hostname コマンドで取得されるホスト名称》

マルチホームドホスト環境で OTM が使用するホスト名称または IP アドレスを指定します。省略した場合は、hostname コマンドで取得されるホスト名称が設定されます。

-TSCID オプションを省略して -TSCMyHost オプションを指定した場合、デフォルトの TSC 識別子は -TSCMyHost オプションで指定した IP アドレスになります。

-TSCEntryCount 登録プロセス数

~ < 符号なし整数 > ((32 ~ 32767)) 《オプションの合計値 + TSC システムプロセス数》

TSC デーモンで管理するプロセスの数を指定します。管理するプロセスについては、「2.4.6 プロセスの管理」を参照してください。

-TSCEntryCount オプションを省略した場合、-TSCServerConnectCount オプションと -TSCClientConnectCount オプションの指定値の合計に、1 (TSC システムプロセス数) を加算した値が設定されます。

-TSCNice nice 値

~ < 符号なし整数 > 《0》

TSC デーモンが管理するサーバで使用する -TSCNice オプションの指定値のデフォルトを指定します。省略した場合は "0" が設定されます。指定値の範囲は OS が提供する

nice の指定範囲に従います。

-TSCRegStart プロセス数

~ <符号なし整数> ((1 ~ 32767))

TSC デーモン開始時に自動開始する TSC レギュレータのプロセス数を指定します。TSC デーモンは TSC レギュレータの動作を監視し、終了を検出すると再開します。省略した場合は、TSC デーモン開始時に TSC レギュレータは自動開始されません。

-TSCRegOption コマンドオプションファイル名称

UNIX の場合 ~ <英数字, ピリオド (.), スラント (/), プラス (+), およびハイフン (-)>

Windows の場合 ~ <英数字, ピリオド (.), 円符号 (¥), コロン (:), プラス (+), およびハイフン (-)>

TSC レギュレータを自動開始するときに指定するコマンドオプションを記述したファイル名称を指定します。コマンドオプションファイル名称はユーザ任意の名称です。コマンドオプションファイルには英数字, 空白, ハイフン (-), ピリオド (.), およびスラント (/) だけを記述できます。コマンドオプション以外は記述しないでください。例えば, 次のように指定します。

-TSCClientConnectCount 64

コマンドオプションファイル名称を "regoption" とする場合は, TSC デーモン開始時に次のように指定します。

```
tscstart -TSCRegOption regoption -TSCRegStart 10
```

自動開始する複数の TSC レギュレータのポート番号を固定の番号にしたい場合は, コマンドオプションファイル内で -TSCPort オプションを指定します。指定された -TSCPort オプションの値を基底値とし, 1 ずつ加算した値が各 TSC レギュレータへの -TSCPort オプションに指定されます。

-TSCRegOption オプションを省略した場合, TSC レギュレータには -TSCDomain オプション, -TSCID オプション, および -TSCMyHost オプション以外のコマンドオプションは指定されません。-TSCDomain オプション, -TSCID オプション, および -TSCMyHost オプションには, tscstart コマンドで指定した TSC ドメイン名称, TSC 識別子, およびホスト名称がそれぞれ設定されます。なお, -TSCRegStart オプションが指定されていない場合は, -TSCRegOption オプションは無視されます。

-TSCEnviron ユーザ環境変数定義ファイル名称

UNIX の場合 ~ <英数字, ピリオド (.), スラント (/), プラス (+), およびハイフン

(-) >

Windows の場合 ~ < 英数字, ピリオド (.), 円符号 (¥), コロン (:), プラス (+), およびハイフン (-) >

開始するプロセスが使用する環境変数を記述したユーザ環境変数定義ファイル名称をフルパスで指定します。ユーザ環境変数定義ファイルの詳細については、「4.1.4 ユーザ環境変数定義ファイル」を参照してください。

-TSCStartTimeOut タイムアウト値

~ < 符号なし整数 > ((0 ~ 2147483647)) 180 (単位: 秒)

TSC デーモンの開始処理の待ち時間を指定します。"0" を指定した場合, 無限に待ち続けます。省略した場合は "180 (秒)" が設定されます。

-TSCClientConnectCount 接続クライアントの最大数

~ < 符号なし整数 > ((0 ~ 32767)) 《64》

自 TSC デーモンに接続するクライアントアプリケーションおよび TSC レギュレータの最大数を指定します。省略した場合は "64" が設定されます。

-TSCServerConnectCount 接続サーバアプリケーションの最大数

~ < 符号なし整数 > ((0 ~ 32767)) 《64》

自 TSC デーモンに接続するサーバアプリケーションの最大数を指定します。省略した場合は "64" が設定されます。

-TSCLogFileSize TSC ログファイル最大サイズ

~ < 符号なし整数 > ((1 ~ 3)) 《1》(単位: メガバイト)

TSC ログファイルの 1 ファイルの最大サイズをメガバイト単位で指定します。省略した場合は "1 (メガバイト)" が設定されます。(-TSCLogFileSize オプションの指定値) × (-TSCLogFileCount オプションの指定値) メガバイト分のメッセージを TSC ログファイルに保存できます。TSC ログファイルに保存するメッセージの容量を変更するには, まず -TSCLogFileCount オプションの指定値を変更してください。-TSCLogFileCount オプションの指定値の変更だけでは不十分な場合にだけ, -TSCLogFileSize オプションを指定してください。

-TSCLogFileCount TSC ログファイル最大数

~ < 符号なし整数 > ((2 ~ 32)) 《2》

TSC ログファイルの最大ファイル数を指定します。省略した場合は "2" が設定されます。(-TSCLogFileSize オプションの指定値) × (-TSCLogFileCount オプションの指定値) メガバイト分のメッセージを TSC ログファイルに保存できます。TSC ログに保存する

メッセージの容量を変更するには、まず -TSCLogFileCount オプションの指定値を変更してください。-TSCLogFileCount オプションの指定値の変更だけでは不十分な場合にだけ、-TSCLogFileSize オプションを指定してください。

TSC ログファイルのサイズが -TSCLogFileSize オプションの指定値を超えると、TSC ログファイルは次のファイルに切り替えられます。TSC ログファイルの切り替え時に、TSC ログファイルの総数が -TSCLogFileCount オプションの指定値を超えていると、最も古い TSC ログファイルの内容が削除されます。

上記の内容に留意し、TSC ログファイルの容量を見積もってください。

-TSCRootAcceptorCount TSC ルートアクセプタ数

~ < 符号なし整数 > ((1 ~ 32767)) 《128》

TSC デーモンに登録できる TSC ルートアクセプタ数を指定します。省略した場合は "128" が設定されます。

-TSCRootAcceptorRegistCount TSC ルートアクセプタ登録可能数

~ < 符号なし整数 > ((1 ~ 32767)) 《64》

同じ TSC ルートアクセプタの重複登録可能数を指定します。省略した場合は "64" が設定されます。

-TSCServerCacheSize サーバキャッシュサイズ

~ < 符号なし整数 > ((1 ~ 32767)) 《128》 (単位 : キロバイト)

TSC ドメイン内のサーバ情報を格納するキャッシュテーブルのサイズを指定します。省略した場合は "128 (キロバイト)" が設定されます。

サーバキャッシュテーブルのサイズは次に示す計算式で求めてください。

$$\begin{aligned} & \{ \quad 208 \times \text{同じTSCドメイン内のTSCID数} \\ & \quad + 144 \times \text{同じTSCドメイン内の全TSCルートアクセプタ数} \\ & \quad + 144 \times \text{同じTSCドメイン内のTSCルートアクセプタに登録する全TSCユーザアクセプ} \\ & \quad \text{タ数} \\ & \quad + (128 + \text{最大インタフェース名長}) \times \text{同じTSCドメイン内のインタフェース数} \\ & \quad \} \times 1.5 \quad (\text{バイト}) \end{aligned}$$

-TSCDispatchPolicy normalDispatch | priorDispatch

~ < 英字 > ((normalDispatch | priorDispatch)) 《normalDispatch》

リクエストのスケジュールポリシーを次の値で指定します。

"normalDispatch" を指定すると、TSC ルートアクセプタの負荷状況に応じたリクエストをスケジュールします。

"priorDispatch" を指定すると、リクエストを受け付けた TSC デーモンに登録されている TSC ルートアクセプタに、優先的にリクエストをスケジュールします。ただし、その TSC デーモンに登録されている TSC ルートアクセプタが高負荷状態、および閉塞状態の場合は、ほかの TSC デーモンに登録している TSC ルートアクセプタにリクエストをスケジュールします。

省略した場合は "normalDispatch" が設定されます。

-TSCDispatchParallelCount スレッドの最大値

~ <符号なし整数> ((0 ~ 32767)) 《255》

自 TSC デーモン内でクライアントからの要求メッセージをサーバアプリケーションに振り分けるスレッドの最大値、つまり、自 TSC デーモンに登録される RootAcceptor オブジェクトの平行カウント (常駐するスレッド数) の和を指定します。省略した場合は "255" が設定されます。

-TSCMaxRequestCount リクエスト数

~ <符号なし整数> ((1 ~ 32767)) 《50》

TSC デーモンに登録する TSC ルートアクセプタが、同時に登録できるリクエスト数を指定します。省略した場合は "50" が設定されます。

-TSCQueueDeleteWait TSC ルートアクセプタキュー保留時間

~ <符号なし整数> ((0 ~ 2147483647)) 《0》 (単位 : 秒)

異常終了したサーバアプリケーションが登録していた TSC ルートアクセプタのスケジュール用キューを削除しないで保留し、サーバアプリケーションの再開始を待つ場合の待ち時間を指定します。

指定した時間内にサーバアプリケーションが再開始された場合は、保留中のスケジュール用キューを再利用し、キューイングされていたリクエストがスケジュールされます。

指定時間を経過してもサーバアプリケーションが再開始されない場合は、TSC ルートアクセプタのスケジュール用キューを閉塞したあと、TSC ルートアクセプタの登録を削除します。なお、スケジュール用キューの閉塞、および TSC ルートアクセプタの登録の削除は、インターバル処理で実行されるため、指定時間よりも遅れることがあります。

省略した場合、または "0" が指定された場合は、スケジュール用キューを保留しないで TSC ルートアクセプタの登録を削除します。

-TSCHighPriorRequestCount 高プライオリティリクエスト数

~ <符号なし整数> ((0 ~ 32766)) 《0》

スケジュール用キューに登録できるリクエスト数のうち、プライオリティの高いリクエストだけを登録する数を指定します。-TSCHighPriorBorder オプションの指定値以下の

プライオリティ値が設定された、プライオリティの高いリクエストが対象となります。

-TSCHighPriorRequestCount オプションを指定すると、プライオリティの低いリクエストを登録できる数に制限が発生します。つまり、-TSCMaxRequestCount オプションの指定値から -TSCHighPriorRequestCount オプションの指定値を差し引いた値を超えて、プライオリティの低いリクエストが発生した場合は例外が返ります。一方、プライオリティの高いリクエストは、-TSCMaxRequestCount オプション指定値を超えて発生した場合にだけ例外が返ります。

-TSCHighPriorRequestCount オプションには、-TSCMaxRequestCount オプション指定値以上の値は指定できません。省略した場合は "0" が設定されます。

-TSCHighPriorBorder 高プライオリティしきい値

~ < 符号なし整数 > ((1 ~ 7)) 《3》

-TSCHighPriorBorder オプションに指定するプライオリティ値以下の値を持つ、プライオリティの高いリクエストを -TSCHighPriorRequestCount オプションで指定する対象とします。省略した場合は "3" が設定されます。

-TSCStatsUse Y | N

~ < 英字 > ((Y | N)) 《Y》

該当する TSC ノードで稼働統計情報を取得するかどうかを指定します。小文字での指定もできます。省略した場合は "Y" が設定され、該当する TSC ノードで稼働統計情報を取得します。

-TSCStatsFileCount ファイル数

~ < 符号なし整数 > ((3 ~ 256)) 《3》

稼働統計情報ファイルの世代数を指定します。省略した場合は "3" が設定されます。

-TSCStatsFileSize ファイルサイズ

~ < 符号なし整数 > ((1 ~ 1024)) 《1》 (単位 : メガバイト)

稼働統計情報ファイルのサイズをメガバイト単位で指定します。省略した場合は "1 (メガバイト)" が設定されます。

戻り値

このコマンドは次に示す戻り値をシェルに返してから、処理を終了します。

戻り値	意味
0	正常終了しました。
0 以外	コマンド処理中にエラーが発生したために異常終了しました。 出力されたメッセージに従って対策したあと、再度、コマンドを実行してください。

注意事項

TSC デーモンの開始時に tscstart コマンドで指定したオプションの値は、同じ TSC ノードに属する TSC レギュレータ、TSCORB コネクタ、およびサーバアプリケーションの開始コマンドのオプションのデフォルトとして引き継がれます。ただし、一部のオプションの値は引き継がれません。詳細は、この章の「tscregltd (TSC レギュレータの開始)」、「tscstartgw (TSCORB コネクタの開始)」、「サーバアプリケーションの開始」の注意事項を参照してください。

-TSCRegStart オプションに指定する自動開始の TSC レギュレータの数が増えると、TSC デーモンの開始処理に時間が掛かる場合があります。その場合は、-TSCStartTimeOut オプションで開始処理の待ち時間を調整してください。

TSC デーモンは、開始後にカレントディレクトリを移動します。そのため、次に示す場合のコアファイルの出力先は、実行形式ファイルのディレクトリとは異なります。

- "tscstop -TSCForce -TSCDebugInfo" または kill コマンドなどで TSC デーモンを強制終了した場合
- TSC デーモンが異常終了した場合

移動後のカレントディレクトリは、UNIX の場合は "\$TSCSPOOL/adm/TSC 識別子 " で、Windows の場合は "%TSCSPOOL%\adm\TSC 識別子 " です。

-TSCRegStart オプションを指定した場合、TSC レギュレータの自動開始が一つでも異常終了すると、TSC デーモンは開始処理失敗と判断してエラーメッセージを出力し、開始処理を中止します。

-TSCRegOption オプションで指定したコマンドオプションファイル内に記述したコマンドオプションの正当性を、自動開始した TSC レギュレータがチェックします。そのため、コマンドオプション不正、および -TSCPort オプション指定によるポート番号の重複は、自動開始した TSC レギュレータで出力したエラーメッセージを参照して対策してください。

TSC デーモンの開始完了前のメッセージには、-TSCLogFileSize オプションおよび -TSCLogFileCount オプションが有効にならない場合があります。

tscstartgw (TSCORB コネクタの開始)

形式

```
tscstartgw [-h] | -TSCAPID アプリケーション識別子
[-TSCDomain TSCドメイン名称] [-TSCID TSC識別子]
[-TSCUid ユーザID] [-TSCGid グループID]
[-TSCMyHost ホスト名称またはIPアドレス]
[-TSCEnviron ユーザ環境変数定義ファイル名称]
[-TSCStartTimeOut TSCORBコネクタ開始時のタイムアウト値]
[-TSCTimeOut タイムアウト値]
[-TSCRequestPriority プライオリティ値]
[-TSCExceptConvertFile 例外変換テーブルファイル]
[-TSCIDLFile IDLファイル[, IDLファイル...]]
[-TSCIDLPath IDLファイルのディレクトリ
[:IDLファイルのディレクトリ...]]
[[-I インクルードファイルのディレクトリ
[:インクルードファイルのディレクトリ...]]...]
[-TSCObjectName オブジェクト名]
[-TSCIORConnect] [-TSCAgent 1 | 0]
[-TSCSendAcceptor TSCアクセプタ名称]
[-TSCClientConnectCount コネクション数]
[-TSCPort ポート番号]
```

機能

CORBA に準拠するクライアントアプリケーション、つまり、TPBroker クライアントを始めとする ORB クライアントからのリクエストを OTM で処理するための TSCORB コネクタを開始します。tscstartgw コマンドは、TSCORB コネクタがリクエスト受付待ち状態になったことを確認して、リターンします。

なお、tscstartgw コマンドを使用して TSCORB コネクタを開始するには、OTM - Connector for ORB が必要です。OTM - Connector for ORB については、サポート時期をご確認ください。

オプション

-h

コマンドの使用方法が表示されます。

-TSCAPID アプリケーション識別子

~ < 1 ~ 32 文字の英数字 >

開始する TSCORB コネクタのアプリケーション識別子を指定します。-TSCDomain オプションおよび -TSCID オプションで指定した TSC ノード内で、サーバアプリケーションのアプリケーション識別子も含めて一意となるように指定してください。

-TSCDomain TSC ドメイン名称

~ < 1 ~ 31 文字の英数字 > 《TSCDOMAIN》

TSCORB コネクタを管理する TSC デーモンが属する TSC ドメイン名称を指定します。"TSC" または "tsc" で始まる TSC ドメイン名称は指定しないでください。省略した場合は "TSCDOMAIN" が設定されます。

-TSCID TSC 識別子

~ < 1 ~ 31 文字の英数字, およびピリオド (.) > 《IP アドレス》

TSCORB コネクタを管理する TSC デーモンの識別子を指定します。"TSC" または "tsc" で始まる TSC 識別子は指定しないでください。ピリオド (.) は, IP アドレスを指定する場合だけ使用できます。省略した場合は, IP アドレスが TSC 識別子に設定されます。例えば, ホストの IP アドレスが "172.17.112.43" のときは, "172.17.112.43" という文字列が TSC 識別子となります。

-TSCUid ユーザ ID

~ < 符号なし整数 > ((0 ~ 2147483647)) 《tscstartgw コマンドを実行したユーザのユーザ ID》

開始する TSCORB コネクタのユーザ ID を指定します。-TSCUid オプションの指定値を使用して TSCORB コネクタを開始した場合の処理は, 動作環境に依存します。なお, Windows の場合, -TSCUid オプションは無効です。

-TSCUid オプションを省略した場合は, tscstartgw コマンドを実行したユーザのユーザ ID で TSCORB コネクタを開始します。

-TSCGid グループ ID

~ < 符号なし整数 > ((0 ~ 2147483647)) 《tscstartgw コマンドを実行したユーザのグループ ID》

開始する TSCORB コネクタのグループ ID を指定します。-TSCGid オプションの指定値を使用して TSCORB コネクタを開始した場合の処理は, 動作環境に依存します。なお, Windows の場合, -TSCGid オプションは無効です。

-TSCGid オプションを省略した場合は, tscstartgw コマンドを実行したユーザのグループ ID で TSCORB コネクタを開始します。

-TSCMyHost ホスト名称または IP アドレス

~ < 1 ~ 64 文字の文字列 > 《hostname コマンドで取得されるホスト名称》

マルチホームドホスト環境で OTM が使用するホスト名称または IP アドレスを指定します。省略した場合は, hostname コマンドで取得されるホスト名称が設定されます。

-TSCID オプションを省略して -TSCMyHost オプションを指定した場合, デフォルトの TSC 識別子は -TSCMyHost オプションで指定した IP アドレスになります。

-TSCEnviron ユーザ環境変数定義ファイル名称

UNIX の場合 ~ < 英数字 , ピリオド (.) , スラント (/) , プラス (+) , およびハイフン (-) >

Windows の場合 ~ < 英数字 , ピリオド (.) , 円符号 (¥) , コロン (:) , プラス (+) , およびハイフン (-) >

開始する TSCORB コネクタが使用する環境変数を記述したユーザ環境変数定義ファイル名称をフルパスで指定します。ユーザ環境変数定義ファイルの詳細については、「4.1.4 ユーザ環境変数定義ファイル」を参照してください。

-TSCStartTimeOut TSCORB コネクタ開始時のタイムアウト値

~ < 符号なし整数 > ((0 ~ 2147483647)) 180 (単位 : 秒)

TSCORB コネクタの開始処理の待ち時間を指定します。"0" を指定した場合、無限に待ち続けます。省略した場合は "180 (秒)" が設定されます。

-TSCTimeOut タイムアウト値

~ < 符号なし整数 > ((0 ~ 2147483647)) 《180》(単位 : 秒)

サービスの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指定します。"0" を指定した場合、無限に待ち続けます。省略した場合は "180 (秒)" が設定されます。

-TSCRequestPriority プライオリティ値

~ < 符号なし整数 > ((1 ~ 8)) 《4》

スケジュール用キューに滞留したリクエストを引き出すときのプライオリティ (優先順位) を指定します。指定値が小さいほどプライオリティは高くなります。

省略した場合は "4" が設定されます。

-TSCExceptConvertFile 例外変換テーブルファイル

UNIX の場合 ~ < 英数字 , ピリオド (.) , スラント (/) , コロン (:) , プラス (+) , およびハイフン (-) >

Windows の場合 ~ < 英数字 , ピリオド (.) , 円符号 (¥) , コロン (:) , プラス (+) , およびハイフン (-) >

TSCORB コネクタ経由で接続している ORB クライアントに例外を通知するときは、OTM のシステム例外を CORBA 例外に変換して通知します。-TSCExceptConvertFile オプションは、デフォルトの例外変換以外の方法で変換する場合だけ指定します。この場合、使用する例外変換テーブルファイル名称を絶対パスで指定してください。

例外変換テーブルファイルには、OTM 例外から CORBA 例外に変換する方法を記述します。例外変換テーブルファイルについては、「付録 D 例外変換テーブルファイルの作成

方法」を参照してください。

-TSCIDLFile IDL ファイル [,IDL ファイル ...]

UNIX の場合 ~ < 英数字, ピリオド (.), スラント (/), プラス (+), およびハイフン (-) >

Windows の場合 ~ < 英数字, ピリオド (.), 円符号 (¥), コロン (:), プラス (+), およびハイフン (-) >

TSCORB コネクタのインタフェースを定義した IDL ファイルを指定します。

-TSCIDLPath オプションを指定しない場合は絶対パスで指定し、-TSCIDLPath を指定する場合はそのディレクトリからの相対パスで指定してください。

IDL ファイルを複数指定する場合、区切り文字としてコンマ (,) を指定してください。また、IDL ファイルとコンマ (,) の間には空白を入れないでください。

なお、-TSCIDLFile オプションは省略できません。

-TSCIDLFile オプションには、開始する TSCORB コネクタを経由して呼び出すサーバアプリケーションの作成時に使用した IDL ファイルを指定してください。内容が異なる IDL ファイルを使用すると、動作が不正になることがあります。また、IDL ファイルを複数指定する場合に、各 IDL ファイルの定義で、同一のモジュール中に同一のインタフェース定義が記述されているときは、そのインタフェース定義に対して作成される CORBA オブジェクトは一つだけとなります。ただし、同じインタフェース定義の中でオペレーションの定義に不整合があった場合、つまり、異なるオペレーションが定義されていた場合は、TSCORB コネクタの開始時にエラーになるか、動作が不正になります。

-TSCIDLPath IDL ファイルのディレクトリ [:IDL ファイルのディレクトリ ...]

UNIX の場合 ~ < 英数字, ピリオド (.), スラント (/), プラス (+), およびハイフン (-) >

Windows の場合 ~ < 英数字, ピリオド (.), 円符号 (¥), コロン (:), プラス (+), およびハイフン (-) >

-TSCIDLFile オプションで指定する IDL ファイルの格納ディレクトリを、絶対パスで指定します。複数指定する場合、UNIX の場合はコロン (:), Windows の場合はセミコロン (;) で区切って指定してください。

TSCORB コネクタは、-TSCIDLPath オプションに指定したディレクトリからの相対パスで、-TSCIDLFile オプションに指定した IDL ファイルを検索します。このとき、最初に検索されたディレクトリの IDL ファイルが有効となります。

-I インクルードファイルのディレクトリ [: インクルードファイルのディレクトリ ...]

UNIX の場合 ~ < 英数字 , ピリオド (.) , スラント (/) , プラス (+) , およびハイフン (-) >

Windows の場合 ~ < 英数字 , ピリオド (.) , 円符号 (¥) , コロン (:) , プラス (+) , およびハイフン (-) >

-TSCIDLFile で指定した IDL ファイル中に記述されているインクルードファイルの格納ディレクトリを , 絶対パスで指定します。複数指定する場合 , UNIX の場合はコロン (:) , Windows の場合はセミコロン (;) で区切るか , または -I オプションを複数回指定してください。

-TSCObjectName オブジェクト名

~ < 英数字 , ピリオド (.) , プラス (+) , およびハイフン (-) > 《空文字列》

開始する TSCORB コネクタに対応する CORBA オブジェクトの名称を指定します。先頭は英数字で指定してください。なお , ほかの TSCORB コネクタと重複するオブジェクト名も指定できます。

省略した場合は , オブジェクト名は空文字列となります。

-TSCIORConnect

開始する TSCORB コネクタに対応する CORBA オブジェクトに対して , IOR 文字列を使用した接続をする場合に指定します。-TSCIORConnect オプションを指定すると , TSCORB コネクタの開始後に , tscgetior コマンドを使用した IOR 文字列の取得ができるようになります。

-TSCIORConnect オプションを指定した場合は , -TSCPort オプションを省略できません。また , -TSCIORConnect オプションを省略した場合は , -TSCAgent オプションには "0" を指定できません。

-TSCAgent 1 | 0

~ < 符号なし整数 > ((1 | 0)) 《1》

開始する TSCORB コネクタに対応する CORBA オブジェクトを TPBroker の osagent に登録するかどうかを指定します。

"1" を指定すると , 対応する CORBA オブジェクトを TPBroker の osagent に登録します。この場合は , TPBroker のクライアントアプリケーションから TSCORB コネクタへの接続で , TPBroker の _bind を使用できるようになります。

"0" を指定すると , 対応する CORBA オブジェクトを TPBroker の osagent に登録しません。この場合は , -TSCIORConnect オプションの指定を省略できません。

-TSCAgent オプションを省略した場合は "1" が設定されます。

-TSCSendAcceptor TSC アクセプタ名称

~ < 1 ~ 31 文字の英数字 >

TSCORB コネクタからサーバオブジェクトへの呼び出しを識別するための TSC アクセプタ名称を指定します。すべての呼び出しに対して、ここで指定した同じ TSC アクセプタ名称が適用されます。

-TSCSendAcceptor オプションを省略した場合は、"TSC アクセプタ名称なし" としてサーバオブジェクトが呼び出されます。

-TSCClientConnectCount コネクション数

~ < 符号なし整数 > ((1 ~ 32767)) 《64》

TSCORB コネクタに接続できる、ORB クライアントアプリケーションからの最大コネクション数を指定します。-TSCClientConnectCount オプションの指定値を超えて接続した場合、リクエストの呼び出し時に例外が発生します。

-TSCPort ポート番号

~ < 符号なし整数 > ((5001 ~ 65535))

TSCORB コネクタの新しいコネクションを探すときに使用するポート番号を指定します。省略した場合は、OS が自動的に割り当てたポート番号を使用します。

戻り値

このコマンドは次に示す戻り値をシェルに返してから、処理を終了します。

戻り値	意味
0	正常終了しました。
0 以外	コマンド処理中にエラーが発生したために異常終了しました。 出力されたメッセージに従って対策したあと、再度、コマンドを実行してください。 ただし、このコマンドが異常終了した場合も、TSCORB コネクタの開始処理が続行されるときがあります。

注意事項

tscstartgw コマンドのオプション引数に使用できない文字を指定した場合は、コマンドオプションエラーとならないで、不正な動作をする場合があります。

tscstartgw コマンドが正常終了しなかった場合でも、TSCORB コネクタプロセスが生成、つまり、KFOT70606-I メッセージ出力された場合、TSCORB コネクタの開始処理は続行されます。

TSCORB コネクタのオプションは、同じ TSC ノードに属する TSC デーモンの開始時に指定できます。TSCORB コネクタの開始時に指定しなかったオプションのデフォ

トには、TSC デーモンの開始時に指定された値が引き継がれます。
ただし、次に示すオプションについては、同じ TSC ノードの TSC デーモンの開始時に指定されていても引き継がれません。

- -I オプション
- -TSCID オプション
- -TSCDomain オプション
- -TSCMyHost オプション
- -TSCAPID オプション

TSCORB コネクタは開始後にカレントディレクトリを移動します。そのため、次に示す場合のコアファイルの出力先は、実行形式ファイルのディレクトリとは異なります。

- "tscstopgw -TSCForce -TSCDebugInfo", "tscstop -TSCForce -TSCDebugInfo", または kill コマンドなどで TSCORB コネクタを強制終了した場合
- TSCORB コネクタが異常終了した場合

移動後のカレントディレクトリは、UNIX の場合は "\$ TSCSPOOL/adm/TSC 識別子 / orbd/ カレントディレクトリ ID" で、Windows の場合は "%TSCSPOOL%¥adm¥TSC 識別子 ¥orbd¥ カレントディレクトリ ID" です。

カレントディレクトリ ID については、この章の「tscprcls (プロセス情報の出力)」で、出力形式を参照してください。

tscstartgw コマンドは、実行中にカレントディレクトリに一時ファイルを作成します。そのため、次のことに注意してください。

- tscstartgw コマンドは、読み取りまたは書き込み権限がないディレクトリで実行するとエラーになります。必ず読み取りおよび書き込み権限があるディレクトリで実行してください。
- tscstartgw コマンドを kill コマンドなどで強制停止した場合、カレントディレクトリに tscstartgw コマンドが作成する一時ファイルが残ることがあります。

tscstartprc (サーバアプリケーションの開始)

形式

```
tscstartprc [-h] -TSCAPID アプリケーション識別子
             -TSCAPName サーバアプリケーションの実行形式ファイル名称
             [-TSCDomain TSCドメイン名称] [-TSCID TSC識別子]
             [-TSCPort ポート番号] [-TSCStartTimeOut タイムアウト値]
             [-TSCWatchTime クラス監視時間]
             [-TSCWatchMethod メソッド監視時間]
             [-TSCRequestWay 1 | 0]
             [-TSCEnviron ユーザ環境変数定義ファイル名称]
             [-TSCRootAcceptor TSCルートアクセプタ登録名称]
             [-TSCNice nice値] [-TSCuid ユーザID]
             [-TSCGidグループID] [-TSCTimeOut タイムアウト値]
             [-TSCParallelCount 常駐スレッド数]
             [-TSCRequestPriority プライオリティ値]
             [-TSCAcceptor TSCアクセプタ名称]
             [-TSCClientMessageBufferCount クライアント通信用バッファ数]
             [-TSCRebindTimes リバインド回数]
             [-TSCRebindInterval リバインド間隔]
             [-TSCMyHost ホスト名称またはIPアドレス]
             [-TSCRetryReference 接続先情報ファイル名称]
             [-TSCRetryWay マルチノードリトライ接続の接続方式]
```

機能

サーバアプリケーションを開始します。tscstartprc コマンドは TSCAdm クラスの serverMainloop メソッドの発行を確認してリターンします。tscstartprc コマンドで開始するプロセス数は 1 プロセスです。また、tscstartprc コマンドを使用して開始するサーバアプリケーションのコマンドラインには、tscstartprc コマンドに指定したコマンドライン引数がすべて指定されます。サーバアプリケーションでは、プロセス開始時のコマンドライン情報のすべてを TSCAdm クラスの initServer メソッドに指定するようにしてください。

オプション

ここで説明していないオプションについては、この章の「サーバアプリケーションの開始」を参照してください。

-h

コマンドの使用方法が表示されます。

-TSCAPID アプリケーション識別子

~ < 1 ~ 32 文字の英数字 >

開始するプロセスのアプリケーション識別子を指定します。-TSCDomain オプションおよび -TSCID オプションで指定した TSC ノード内で、TSCORB コネクタのアプリケーション識別子も含めて一意となるように指定してください。

-TSCAPName サーバアプリケーションの実行形式ファイル名称

UNIX の場合 ~ <英数字, 空白, ピリオド (.), スラント (/), コロン (:), チルダ (~), プラス (+), ハイフン (-), および等号 (=) >

Windows の場合 ~ <英数字, 空白, ピリオド (.), スラント (/), 円符号 (¥), コロン (:), チルダ (~), プラス (+), ハイフン (-), および等号 (=) >

注

UNIX の場合の ":", "~", "=", および Windows の場合の "/", "~", "=" は, Java のシステムプロパティとしてだけ使用できます。

開始するサーバアプリケーションの実行形式ファイル名称を指定します。フルパス名で記述した場合, フルパス名で実行します。実行形式ファイル名称だけを指定した場合, PATH 環境変数を使用して実行します。PATH 環境変数は tscstart コマンドを実行した際の PATH 環境変数を使用します。

Java アプリケーションの場合は空白を使用できますが, この場合も, 空白は, Java 実行コマンド, クラス名称, およびシステムプロパティの区切り文字としてだけ使用してください。この場合, 実行形式ファイル名称に空白を使用するには, 引用符 (") で囲む必要があります。Java アプリケーションの開始の例として, Java アプリケーション実行形式ファイル名称が "Server.class" の場合の指定例を次に示します。

-TSCAPName "vbj Server"

-TSCStartTimeOut タイムアウト値

~ <符号なし整数> ((0 ~ 2147483647)) 180 (単位: 秒)

サーバアプリケーションの開始処理の待ち時間を指定します。"0" を指定した場合, 無限に待ち続けます。省略した場合は "180 (秒)" が設定されます。

-TSCEnviron ユーザ環境変数定義ファイル名称

UNIX の場合 ~ <英数字, ピリオド (.), スラント (/), プラス (+), およびハイフン (-) >

Windows の場合 ~ <英数字, ピリオド (.), 円符号 (¥), コロン (:), プラス (+), およびハイフン (-) >

開始するプロセスが使用する環境変数を記述したユーザ環境変数定義ファイル名称をフルパスで指定します。ユーザ環境変数定義ファイルの詳細については, 「4.1.4 ユーザ環境変数定義ファイル」を参照してください。

-TSCUId ユーザ ID

~ <符号なし整数> ((0 ~ 2147483647)) 《tscstartprc コマンドを実行したユーザのユー

ユーザ ID

開始するサーバアプリケーションのユーザ ID を指定します。-TSCUId オプションの指定値を使用してサーバアプリケーションを開始した場合の処理は、動作環境に依存します。なお、Windows の場合、-TSCUId オプションは無効です。

-TSCUId オプションを省略した場合は、tscstartprc コマンドを実行したユーザのユーザ ID でサーバアプリケーションを開始します。

-TSCGid グループ ID

~ <符号なし整数> ((0 ~ 2147483647)) 《tscstartprc コマンドを実行したユーザのグループ ID》

開始するサーバアプリケーションのグループ ID を指定します。-TSCGid オプションの指定値を使用してサーバアプリケーションを開始した場合の処理は、動作環境に依存します。なお、Windows の場合、-TSCGid オプションは無効です。

-TSCGid オプションを省略した場合は、tscstartprc コマンドを実行したユーザのグループ ID でサーバアプリケーションを開始します。

戻り値

このコマンドは次に示す戻り値をシェルに返してから、処理を終了します。

戻り値	意味
0	正常終了しました。
0 以外	コマンド処理中にエラーが発生したために異常終了しました。出力されたメッセージに従って対策したあと、再度、コマンドを実行してください。ただし、このコマンドが異常終了した場合も、サーバアプリケーションの開始処理が続行されるときがあります。

注意事項

tscstartprc コマンドが正常終了しなかった場合でも、サーバアプリケーションプロセスが生成、つまり、KFOT70606-I メッセージ出力された場合、サーバアプリケーションの開始処理は続行されます。

サーバアプリケーションのオプションは、同じ TSC ノードに属する TSC デーモンの開始時に指定できます。サーバアプリケーションの開始時に指定しなかったオプションのデフォルトには、TSC デーモンの開始時に指定された値が引き継がれます。ただし、次に示すオプションについては、同じ TSC ノードの TSC デーモンの開始時に指定されていても引き継がれません。

- -TSCAPID オプション
- -TSCDomain オプション
- -TSCID オプション

- -TSCPort オプション
- -TSCStartTimeOut オプション
- -TSCMyHost オプション
- -TSCRetryReference オプション
- -TSCRetryWay オプション

tscstartprc コマンドで開始したサーバアプリケーションは、開始後にカレントディレクトリを移動します。そのため、次に示す場合のコアファイルの出力先は、実行形式ファイルのディレクトリとは異なります。

- "tscstopprc -TSCForce -TSCDebugInfo", "tscstop -TSCForce -TSCDebugInfo", または kill コマンドなどでサーバアプリケーションを強制終了した場合
- サーバアプリケーションが異常終了した場合

移動後のカレントディレクトリは、UNIX の場合は "\$TSCSPOOL/adm/TSC 識別子 / ap/ カレントディレクトリ ID" で、Windows の場合は "%TSCSPOOL%\adm\TSC 識別子 \ap\ カレントディレクトリ ID" です。カレントディレクトリ ID については、この章の「tscprels (プロセス情報の出力)」で、出力形式を参照してください。

tscstartprc コマンドは OTM のサーバアプリケーションを開始するコマンドです。次に示すプログラムなどを開始する場合には、tscstartprc コマンドはアプリケーションプログラムの開始失敗として、エラーメッセージを出力して異常終了します。

- OTM 以外のアプリケーションプログラム
- TSCAdm クラスの serverMainloop メソッドを発行しないアプリケーションプログラム
- TSC レギュレータ
- TSCORB コネクタ

tscstop (TSC デーモンの終了)

形式

```
tscstop [-h] [-TSCForce] [-TSCID TSC識別子]
        [-TSCDomain TSCドメイン名称] [-TSCDebugInfo]
        [-TSCMyHost ホスト名称またはIPアドレス]
```

機能

リクエストのキューイングやリクエストの振り分けをするための TSC デーモンを終了します。

オプション

-h

コマンドの使用方法が表示されます。

-TSCForce

TSC デーモンを強制終了します。

-TSCID TSC 識別子

~ < 1 ~ 31 文字の英数字, およびピリオド (.) > 《IP アドレス》

TSC デーモンの識別子を指定します。"TSC" または "tsc" で始まる TSC 識別子は指定しないでください。ピリオド (.) は, IP アドレスを指定する場合だけ使用できます。省略した場合は, IP アドレスが TSC 識別子に設定されます。例えば, ホストの IP アドレスが "172.17.112.43" のとき, "172.17.112.43" という文字列が TSC 識別子となります。

-TSCDomain TSC ドメイン名称

~ < 1 ~ 31 文字の英数字 > 《TSCDOMAIN》

TSC デーモンが属する TSC ドメイン名称を指定します。"TSC" または "tsc" で始まる TSC ドメイン名称は指定しないでください。省略した場合は "TSCDOMAIN" が設定されます。

-TSCDebugInfo

TSC デーモンの強制終了時にデバッグ情報として, 共用メモリダンプおよびコアを取得します。

-TSCDebugInfo オプションは, -TSCForce オプションを指定したときだけ有効です。

tscstop (TSC デーモンの終了)

-TSCMyHost ホスト名称または IP アドレス

~ < 1 ~ 64 文字の文字列 > 《hostname コマンドで取得されるホスト名称》

マルチホームドホスト環境で OTM が使用するホスト名称または IP アドレスを指定します。省略した場合は、hostname コマンドで取得されるホスト名称が設定されます。

-TSCID オプションを省略して -TSCMyHost オプションを指定した場合、デフォルトの TSC 識別子は -TSCMyHost オプションで指定した IP アドレスになります。

戻り値

このコマンドは次に示す戻り値をシェルに返してから、処理を終了します。

戻り値	意味
0	正常終了しました。
0 以外	コマンド処理中にエラーが発生したために異常終了しました。 出力されたメッセージに従って対策したあと、再度、コマンドを実行してください。

tscstopgw (TSCORB コネクタの終了)

形式

```
tscstopgw [-h] -TSCAPID アプリケーション識別子
          [-TSCDomain TSCドメイン名称] [-TSCID TSC識別子]
          [-TSCMyHost ホスト名称またはIPアドレス]
          [-TSCForce] [-TSCDebugInfo]
```

機能

TSCORB コネクタを終了します。指定されたアプリケーション識別子を持つ TSCORB コネクタに終了要求を通知し、TSCORB コネクタが終了するのを待ち合わせます。

なお、tscstopgw コマンドを使用して TSCORB コネクタを終了するには、OTM - Connector for ORB が必要です。OTM - Connector for ORB については、サポート時期をご確認ください。

オプション

-h

コマンドの使用方法が表示されます。

-TSCAPID アプリケーション識別子

~ < 1 ~ 32 文字の英数字 >

終了する TSCORB コネクタのアプリケーション識別子を指定します。

-TSCDomain TSC ドメイン名称

~ < 1 ~ 31 文字の英数字 > 《TSCDOMAIN》

TSCORB コネクタを管理する TSC デーモンが属する TSC ドメイン名称を指定します。"TSC" または "tsc" で始まる TSC ドメイン名称は指定しないでください。省略した場合は "TSCDOMAIN" が設定されます。

-TSCID TSC 識別子

~ < 1 ~ 31 文字の英数字、およびピリオド (.) > 《IP アドレス》

TSCORB コネクタを管理する TSC デーモンの識別子を指定します。"TSC" または "tsc" で始まる TSC 識別子は指定しないでください。ピリオド (.) は、IP アドレスを指定する場合だけ使用できます。省略した場合は、IP アドレスが TSC 識別子に設定されます。例えば、ホストの IP アドレスが "172.17.112.43" のときは、"172.17.112.43" という文字列が TSC 識別子となります。

-TSCMyHost ホスト名称または IP アドレス

~ < 1 ~ 64 文字の文字列 > 《hostname コマンドで取得されるホスト名称》

マルチホームドホスト環境で OTM が使用するホスト名称または IP アドレスを指定します。省略した場合は、hostname コマンドで取得されるホスト名称が設定されます。

-TSCID オプションを省略して -TSCMyHost オプションを指定した場合、デフォルトの TSC 識別子は -TSCMyHost オプションで指定した IP アドレスになります。

-TSCForce

指定された TSCORB コネクタを強制終了します。

-TSCDebugInfo

TSCORB コネクタの強制終了時に、デバッグ情報として共用メモリダンプおよびコアファイルを取得します。-TSCDebugInfo オプションは -TSCForce オプションの指定時だけ有効です。

戻り値

このコマンドは次に示す戻り値をシェルに返してから、処理を終了します。

戻り値	意味
0	正常終了しました。
0 以外	コマンド処理中にエラーが発生したために異常終了しました。 出力されたメッセージに従って対策したあと、再度、コマンドを実行してください。 ただし、このコマンドが異常終了した場合も、TSCORB コネクタの終了処理が続行されるときがあります。

tscstopprc (サーバアプリケーションの終了)

形式

```
tscstopprc [-h] -TSCAPID アプリケーション識別子
            [-TSCDomain TSCドメイン名称] [-TSCID TSC識別子]
            [-TSCForce] [-TSCDebugInfo]
            [-TSCMyHost ホスト名称またはIPアドレス]
```

機能

サーバアプリケーションを終了します。指定されたアプリケーション識別子を持つサーバアプリケーションに終了要求を通知し、サーバアプリケーションが終了するのを待ち合わせます。tscstopprc コマンドは TSCAdm クラスの endServer メソッドの発行を確認してリターンします。

オプション

-h

コマンドの使用方法が表示されます。

-TSCAPID アプリケーション識別子

~ < 1 ~ 32 文字の英数字 >

終了するプロセスのアプリケーション識別子を指定します。

-TSCDomain TSC ドメイン名称

~ < 1 ~ 31 文字の英数字 > 《TSCDOMAIN》

終了するサーバアプリケーションを管理する TSC ドメイン名称を指定します。"TSC" または "tsc" で始まる文字列は指定しないでください。省略した場合は "TSCDOMAIN" が設定されます。

-TSCID TSC 識別子

~ < 1 ~ 31 文字の英数字, およびピリオド (.) > 《IP アドレス》

終了するアプリケーションを管理する TSC 識別子を指定します。"TSC" または "tsc" で始まる文字列は指定しないでください。ピリオド (.) は、IP アドレスを指定する場合だけ使用できます。省略した場合は、IP アドレスで TSC 識別子が設定されます。

-TSCForce

指定されたサーバアプリケーションを強制終了します。

-TSCDebugInfo

サーバアプリケーションの強制終了時に、デバッグ情報として共用メモリダンプおよびコアファイルを取得します。-TSCDebugInfo オプションは -TSCForce オプションの指定時だけ有効です。

-TSCMyHost ホスト名称または IP アドレス

~ < 1 ~ 64 文字の文字列 > 《hostname コマンドで取得されるホスト名称》

マルチホームドホスト環境で OTM が使用するホスト名称または IP アドレスを指定します。省略した場合は、hostname コマンドで取得されるホスト名称が設定されます。

-TSCID オプションを省略して -TSCMyHost オプションを指定した場合、デフォルトの TSC 識別子は -TSCMyHost オプションで指定した IP アドレスになります。

戻り値

このコマンドは次に示す戻り値をシェルに返してから、処理を終了します。

戻り値	意味
0	正常終了しました。
0 以外	コマンド処理中にエラーが発生したために異常終了しました。 出力されたメッセージに従って対策したあと、再度、コマンドを実行してください。 ただし、このコマンドが異常終了した場合も、サーバアプリケーションの終了処理が続行されるときがあります。

tscstsed (稼働統計情報の編集と出力)

形式

```
tscstsed [-h] | [-TSCEdit 編集形式 [, 編集形式...]]
  [-TSCInterval 時間間隔] [-TSCTime [開始時刻] [, 終了時刻]]
  [-TSCRootAcceptor TSCルートアクセプタ名 [, TSCルートアクセプタ名...]]
  [-TSCOperation オペレーション名 [, オペレーション名...]]
  [-TSCCsv] [-TSCStdout]
  [稼働統計情報ファイル名 [稼働統計情報ファイル名...]]
```

機能

稼働統計情報ファイルに取得した稼働統計情報を編集して標準出力に出力します。tscstsed コマンドでは、次の 3 種類の出力ができます。

- TSC ノード全体でまとめた情報
- TSC ルートアクセプタ単位のスケジュールキュー情報
- サーバアプリケーションのオペレーション単位のリクエスト情報

稼働統計情報の詳細については、「2.9 稼働統計情報の取得」を参照してください。

オプション

-h

コマンドの使用方法が表示されます。

-TSCEdit 編集形式 [, 編集形式 ...]

~ <文字列>

稼働統計情報を編集および出力する形式を指定します。

"nod" を指定すると TSC ノード全体でまとめた情報を編集および出力します。

"rac" を指定すると TSC ルートアクセプタ単位のスケジュールキュー情報を編集および出力します。

"opr" を指定するとサーバアプリケーションのオペレーション単位のリクエスト情報を編集および出力します。

省略した場合は、"nod", "rac", および "opr" のすべてが設定されます。

-TSCInterval 時間間隔

~ <符号なし整数> ((1 ~ 1440)) 《10》 (単位: 分)

稼働統計情報を編集出力する時間の間隔を、分単位で指定します。tscststart コマンドの -TSCInterval オプションに指定した時間間隔の整数倍を指定してください。整数倍を

指定しない場合は、編集結果が正しく出力されません。

省略した場合は "10 (分)" が設定されます。

-TSCTime [開始時刻][, 終了時刻]

~ < 文字列 >

稼働統計情報を編集および出力する範囲を、データの取得時刻で指定します。時分月日年を hhmm[MMDD[YYYY]] の形式で指定します。月 (MM), 日 (DD), および年 (YYYY) を省略すると、現在の年月日が設定されます。

記号	意味	値の範囲	省略の可否
hh	時	00 hh 23	省略できません。
mm	分	00 mm 59	省略できません。
MM	月	01 MM 12	省略できます。
DD	日	01 DD 31	省略できます。
YYYY	年	1970 YYYY	省略できます。

開始時刻を省略した場合は、ファイルの先頭から終了時刻までが出力範囲になります。終了時刻を省略した場合は、開始時刻からファイルの最後までが出力範囲になります。開始時刻と終了時刻の両方を省略した場合は、稼働統計情報ファイルのすべての情報が編集および出力されます。

-TSCRootAcceptor TSC ルートアクセプタ名 [,TSC ルートアクセプタ名 ...]

~ < 1 ~ 31 文字の英数字 >

稼働統計情報を編集および出力する TSC ルートアクセプタ名を指定します。

-TSCRootAcceptor オプションは -TSCEdit オプションに "rac" を指定する場合、または -TSCEdit オプションを省略する場合に指定できます。

-TSCOperation オペレーション名 [, オペレーション名 ...]

~ < 英数字 >

稼働統計情報を編集および出力するオペレーション名を指定します。

-TSCOperation オプションは、-TSCEdit オプションに "opr" を指定する場合、または -TSCEdit オプションを省略する場合に指定できます。

-TSCCsv

編集および出力の結果を CSV 形式 (コンマで区切ったテキスト形式) で出力します。

-TSCStdout

編集および出力の結果を標準出力に出力します。

コマンドオプションの組み合わせ

複数のコマンドオプションを指定する場合に指定できる tscstsed コマンドのコマンドオプションの組み合わせを次の表に示します。

表 5-3 tscstsed コマンドのコマンドオプションの組み合わせ

コマンドオプション		TSCEdit			TSCInterval	TSCTime	TSCRootAcceptor	TSCOperation	TSCCsv	TSCStdout
		nod	rac	opr						
TSCEdit	nod	-					x	x		
TSCEdit	rac		-					x		
TSCEdit	opr			-			x			
TSCInterval					-					
TSCTime						-				
TSCRootAcceptor		x		x			-			
TSCOperation		x	x					-		
TSCCsv									-	
TSCStdout										-

(凡例)

- : 組み合わせられます。
- × : 組み合わせられません。
- : 該当しません。

注

-h オプションとほかのオプションを指定する場合、-h が優先されます。

コマンド引数

稼働統計情報ファイル名 [稼働統計情報ファイル名 ...]

~ <文字列>

編集および出力の入力となる稼働統計情報ファイル名を指定します。半角スペースで区切って、最大 256 個のファイル名を指定できます。

省略した場合は、標準入力から入力します。

戻り値

このコマンドは次に示す戻り値をシェルに返してから、処理を終了します。

戻り値	意味
0	正常終了しました。
0以外	コマンド処理中にエラーが発生したために異常終了しました。 出力されたメッセージに従って対策したあと、再度、コマンドを実行してください。

出力形式

稼働統計情報の出力形式を -TSCEdit オプションで指定する情報の形式ごとに示します。
出力例については、「付録 C tscstsed コマンドの出力例」を参照してください。

TSC ノード全体でまとめた情報の出力形式

-TSCEdit オプションに "nod" を指定して、TSC ノード全体でまとめた情報を出力する形式を次に示します。

```

*** Statistics Information by TSCNode ***
TSCDomain :aa..aa
TSCID      :bb..bb
-----
** Edit time cc..cc - dd..dd **
statistical data : ee..ee - ee..ee
no data          : ff..ff - ff..ff
-----
Event                Count      Maximum      Minimum      Average (unit)
Request Response Time gg..gg      gg..gg      gg..gg      gg..gg (usec)
Request Execution Time hh..hh      hh..hh      hh..hh      hh..hh (usec)
Request Time Out     ii..ii      -            -            - (count)
Request Error        jj..jj      -            -            - (count)
Request Response Time (ORB) kk..kk      kk..kk      kk..kk      kk..kk (usec)
Request Time Out (ORB) ll..ll      -            -            - (count)
Request Error (ORB)  mm..mm      -            -            - (count)
Session Start Count  nn..nn      -            -            - (count)
Session Count        -           oo..oo      oo..oo      oo..oo (count)
Schedule Wait Count  pp..pp      pp..pp      pp..pp      pp..pp (count)
Schedule Wait Count (HPR) qq..qq      qq..qq      qq..qq      qq..qq (count)
Schedule Wait Time   rr..rr      rr..rr      rr..rr      rr..rr (usec)
Request Overflow     ss..ss      -            -            - (count)
Request Overflow (LPR) tt..tt      -            -            - (count)
AP down              uu..uu      -            -            - (count)
Regulator down       vv..vv      -            -            - (count)
ORBConnector down    ww..ww      -            -            - (count)
Regulator-AP         -           xx..xx      xx..xx      xx..xx (count)
TSCDaemon-ClientAP  -           yy..yy      yy..yy      yy..yy (count)
TSCDaemon-ServerAP (Ctr) -          zz..zz      zz..zz      zz..zz (count)
TSCDaemon-ServerAP (Req) -          AA..AA      AA..AA      AA..AA (count)
TSCDaemon-Regulator -          BB..BB      BB..BB      BB..BB (count)
TSCDaemon-ORBConnector -          CC..CC      CC..CC      CC..CC (count)

```

(凡例)

記号	説明
aa..aa	TSC ドメイン名称 (1 ~ 31 文字の英数字) です。
bb..bb	TSC 識別子 (1 ~ 31 文字の英数字) です。
cc..cc	編集開始時間です。
dd..dd	編集終了時間です。
ee..ee	編集時間内の稼働統計情報が取得されている時間帯です。
ff..ff	編集時間内で稼働統計情報が取得されていない時間帯です。
gg..gg	レスポンス時間 (11 けたの 10 進数) です。 出力する値が 100000000000 以上の場合, "*****" が出力されます。
hh..hh	リクエスト実行時間 (11 けたの 10 進数) です。 出力する値が 100000000000 以上の場合, "*****" が出力されます。
ii..ii	リクエストタイムアウト (11 けたの 10 進数) です。 出力する値が 100000000000 以上の場合, "*****" が出力されます。
jj..jj	リクエスト障害 (11 けたの 10 進数) です。 出力する値が 100000000000 以上の場合, "*****" が出力されます。
kk..kk	TSCORB コネクタのレスポンス時間 (11 けたの 10 進数) です。 出力する値が 100000000000 以上の場合, "*****" が出力されます。
ll..ll	TSCORB コネクタのリクエストタイムアウト (11 けたの 10 進数) です。 出力する値が 100000000000 以上の場合, "*****" が出力されます。
mm..mm	TSCORB コネクタのリクエスト障害 (11 けたの 10 進数) です。 出力する値が 100000000000 以上の場合, "*****" が出力されます。
nn..nn	セッション開始メソッド発行数 (11 けたの 10 進数) です。 出力する値が 100000000000 以上の場合, "*****" が出力されます。
oo..oo	セッション数 (11 けたの 10 進数) です。 出力する値が 100000000000 以上の場合, "*****" が出力されます。
pp..pp	スケジュール待ち件数 (11 けたの 10 進数) です。 出力する値が 100000000000 以上の場合, "*****" が出力されます。
qq..qq	プライオリティの高いスケジュールの待ち件数 (11 けたの 10 進数) です。 出力する値が 100000000000 以上の場合, "*****" が出力されます。

記号	説明
rr..rr	スケジュール待ち時間 (11 けたの 10 進数) です。 出力する値が 100000000000 以上の場合, "*****" が出力されます。
ss..ss	スケジュールできなかったリクエスト数 (11 けたの 10 進数) です。 出力する値が 100000000000 以上の場合, "*****" が出力されます。
tt..tt	スケジュールできなかったプライオリティの低いリクエスト数 (11 けたの 10 進数) です。 出力する値が 100000000000 以上の場合, "*****" が出力されます。
uu..uu	アプリケーションプログラム異常終了の件数 (11 けたの 10 進数) です。 出力する値が 100000000000 以上の場合, "*****" が出力されます。
vv..vv	TSC レギュレータ異常終了の件数 (11 けたの 10 進数) です。 出力する値が 100000000000 以上の場合, "*****" が出力されます。
ww..ww	TSCORB コネクタ異常終了の件数 (11 けたの 10 進数) です。 出力する値が 100000000000 以上の場合, "*****" が出力されます。
xx..xx	TSC レギュレータとアプリケーションプログラム間のコネクション件数 (11 けたの 10 進数) です。TSC レギュレータの開始と、稼働統計情報のファイル出力処理が重なった場合、この TSC レギュレータデーモンと TSC デーモンのコネクションは、TSC デーモン - TSC レギュレータコネクション件数に加算されないで、TSC デーモン - クライアントアプリケーションコネクション件数に加算されることがあります。 出力する値が 100000000000 以上の場合, "*****" が出力されます。
yy..yy	TSC デーモンとクライアントアプリケーション間のコネクション件数 (11 けたの 10 進数) です。 出力する値が 100000000000 以上の場合, "*****" が出力されます。
zz..zz	TSC デーモンとサーバアプリケーション間の制御用のコネクション件数 (11 けたの 10 進数) です。 出力する値が 100000000000 以上の場合, "*****" が出力されます。
AA..AA	TSC デーモンとサーバアプリケーション間のリクエスト用のコネクション件数 (11 けたの 10 進数) です。 出力する値が 100000000000 以上の場合, "*****" が出力されます。

記号	説明
BB..BB	TSC デーモンと TSC レギュレータ間のコネクション件数 (11 けたの 10 進数) です。 出力する値が 100000000000 以上の場合, "*****" が出力されます。
CC..CC	TSC デーモンと TSCORB コネクタ間のコネクション件数 (11 けたの 10 進数) です。 出力する値が 100000000000 以上の場合, "*****" が出力されます。

TSC ノード全体でまとめた情報の CSV 形式での出力形式

-TSCEdit オプションに "nod" を指定し, かつ -TSCCsv オプションを指定して, TSC ノード全体でまとめた情報を CSV 形式で出力する形式を次に示します。

```
*** Statistics Information by TSCNode ***
TSCDomain, TSCID, start, end, Event, Count, Maximum, Minimum, Average, unit
aa..aa, bb..bb, cc..cc, dd..dd, ee..ee, ff..ff, gg..gg, hh..hh, ii..ii, jj..jj
aa..aa, bb..bb, cc..cc, dd..dd, ee..ee, ff..ff, gg..gg, hh..hh, ii..ii, jj..jj
```

(凡例)

記号	説明
aa..aa	TSC ドメイン名称 (1 ~ 31 文字の英数字) です。
bb..bb	TSC 識別子 (1 ~ 31 文字の英数字) です。
cc..cc	編集開始時間です。
dd..dd	編集終了時間です。
ee..ee	イベント内容です。
ff..ff	発生件数 (11 けたの 10 進数) です。 出力する値が 100000000000 以上の場合, "*****" が出力されます。
gg..gg	最大値 (11 けたの 10 進数) です。 出力する値が 100000000000 以上の場合, "*****" が出力されます。
hh..hh	最小値 (11 けたの 10 進数) です。 出力する値が 100000000000 以上の場合, "*****" が出力されます。
ii..ii	平均値 (11 けたの 10 進数) です。 出力する値が 100000000000 以上の場合, "*****" が出力されます。
jj..jj	単位です。

注

-TSCsv オプションを指定すると、編集時間内の統計情報データが取得されている時間帯、および編集時間内で稼働統計が取得されていない時間帯は出力されません。

TSC ルートアクセプタ単位の情報出力形式

-TSCedit オプションに "rac" を指定して、TSC ルートアクセプタ単位でスケジュールキュー情報を出力する形式を次に示します。

```

*** Statistics Information by RootAcceptor ***
TSCDomain :aa..aa
TSCID      :bb..bb
-----
** Edit time cc..cc - dd..dd **
statistical data : ee..ee - ee..ee
no data          : ff..ff - ff..ff
-----
RootAcceptor : gg..gg
-----
Event          Count      Maximum    Minimum    Average (unit)
Schedule Wait Count  hh..hh    hh..hh    hh..hh    hh..hh(count)
Schedule Wait Count(HPR) ii..ii    ii..ii    ii..ii    ii..ii(usec)
Schedule Wait Time   jj..jj    jj..jj    jj..jj    jj..jj(count)
Request Overflow     kk..kk    -          -          -(count)
Request Overflow(LPR) ll..ll    -          -          -(count)
TSCDaemon-ServerAP(Req) -          mm..mm    mm..mm    mm..mm(count)
    
```

※

(凡例)

記号	説明
aa..aa	TSC ドメイン名称 (1 ~ 31 文字の英数字) です。
bb..bb	TSC 識別子 (1 ~ 31 文字の英数字) です。
cc..cc	編集開始時間です。
dd..dd	編集終了時間です。
ee..ee	編集時間内の稼働統計情報が取得されている時間帯です。
ff..ff	編集時間内で稼働統計情報が取得されていない時間帯です。
gg..gg	TSC ルートアクセプタ名 (1 ~ 31 文字) です。
hh..hh	スケジュール待ち件数 (11 けたの 10 進数) です。 出力する値が 100000000000 以上の場合、"*****" が出力されます。
ii..ii	プライオリティの高いスケジュールの待ち件数 (11 けたの 10 進数) です。 出力する値が 100000000000 以上の場合、"*****" が出力されます。
jj..jj	スケジュール待ち時間 (11 けたの 10 進数) です。 出力する値が 100000000000 以上の場合、"*****" が出力されます。

記号	説明
kk..kk	スケジュールできなかったリクエスト数 (11 けたの 10 進数) です。 出力する値が 100000000000 以上の場合, "*****" が出力されます。
ll..ll	スケジュールできなかったプライオリティの低いリクエスト数 (11 けたの 10 進数) です。 出力する値が 100000000000 以上の場合, "*****" が出力されます。
mm..mm	TSC デーモン - サーバアプリケーションリクエスト用コネクション件数 (11 けたの 10 進数) です。 出力する値が 100000000000 以上の場合, "*****" が出力されます。

注

TSC ルートアクセプタ単位に繰り返し出力されます。

TSC ルートアクセプタ単位の情報の CSV 形式での出力形式

-TSCedit オプションに "rac" を指定し, かつ -TSCcsv オプションを指定して, TSC ルートアクセプタ単位のスケジュールキュー情報を CSV 形式で出力する形式を次に示します。

```
*** Statistics Information by RootAcceptor ***
TSCDomain, TSCID, RootAcceptor, start, end, Event, Count, Maximum, Minimum, Average, unit
aa. .aa, bb. .bb, cc. .cc, dd. .dd, ee. .ee, ff. .ff, gg. .gg, hh. .hh, ii. .ii, jj. .jj, kk. .kk
aa. .aa, bb. .bb, cc. .cc, dd. .dd, ee. .ee, ff. .ff, gg. .gg, hh. .hh, ii. .ii, jj. .jj, kk. .kk
```

(凡例)

記号	説明
aa..aa	TSC ドメイン名称 (1 ~ 31 文字の英数字) です。
bb..bb	TSC 識別子 (1 ~ 31 文字の英数字) です。
cc..cc	TSC ルートアクセプタ名称 (1 ~ 31 文字の英数字) です。
dd..dd	編集開始時間です。
ee..ee	編集終了時間です。
ff..ff	イベント内容です。
gg..gg	発生件数 (11 けたの 10 進数) です。 出力する値が 100000000000 以上の場合, "*****" が出力されます。
hh..hh	最大値 (11 けたの 10 進数) です。 出力する値が 100000000000 以上の場合, "*****" が出力されます。

記号	説明
ii..ii	最小値 (11 けたの 10 進数) です。 出力する値が 100000000000 以上の場合, "*****" が出力されます。
jj..jj	平均値 (11 けたの 10 進数) です。 出力する値が 100000000000 以上の場合, "*****" が出力されます。
kk..kk	単位です。

注

-TSCCsv オプションを指定すると、編集時間内の統計情報データが取得されている時間帯、および編集時間内で稼働統計が取得されていない時間帯は出力されません。

サーバアプリケーションのオペレーション単位の情報の出力形式

-TSCEdit オプションに "opr" を指定して、サーバアプリケーションのオペレーション単位でリクエスト情報を出力する形式を次に示します。

TSCDomain :aa..aa
TSCID :bb..bb

** Edit time cc..cc - dd..dd **
statistical data : ee..ee - ee..ee
no data : ff..ff - ff..ff

Interface :gg..gg
Operation :hh..hh

Event	Count	Maximum	Minimum	Average (unit)
Kind of Request				
Synchronized	ii..ii	-	-	-(count)
Oneway	jj..jj	-	-	-(count)
Session Sync	kk..kk	-	-	-(count)
Request Response Time	ll..ll	ll..ll	ll..ll	ll..ll (usec)
Request Execution Time	mm..mm	mm..mm	mm..mm	mm..mm (usec)
Request Time Out	nn..nn	-	-	-(count)
Request Error	oo..oo	-	-	-(count)
Request Response Time(ORB)	pp..pp	pp..pp	pp..pp	pp..pp (usec)
Request Time Out(ORB)	qq..qq	-	-	-(count)
Request Error(ORB)	rr..rr	-	-	-(count)

※

(凡例)

記号	説明
aa..aa	TSC ドメイン名称 (1 ~ 31 文字の英数字) です。
bb..bb	TSC 識別子 (1 ~ 31 文字の英数字) です。
cc..cc	編集開始時間です。
dd..dd	編集終了時間です。

記号	説明
ee..ee	編集時間内の稼働統計情報が取得されている時間帯です。
ff..ff	編集時間内で稼働統計情報が取得されていない時間帯です。
gg..gg	インタフェース名 (1 ~ 31 文字の英数字) です。 31 文字を超える場合は, "先頭から 15 文字" + "*" + "最後から 15 文字" の形式で出力されます。
hh..hh	オペレーション名 (1 ~ 31 文字の英数字) です。 31 文字を超える場合は, "先頭から 15 文字" + "*" + "最後から 15 文字" の形式で出力されます。
ii..ii	同期型呼び出し回数 (11 けたの 10 進数) です。 出力する値が 100000000000 以上の場合, "*****" が出力されます。
jj..jj	非応答型呼び出し回数 (11 けたの 10 進数) です。 出力する値が 100000000000 以上の場合, "*****" が出力されます。
kk..kk	セッション型呼び出し回数 (11 けたの 10 進数) です。 出力する値が 100000000000 以上の場合, "*****" が出力されます。
ll..ll	レスポンス時間 (11 けたの 10 進数) です。 出力する値が 100000000000 以上の場合, "*****" が出力されます。
mm..mm	リクエスト実行時間 (11 けたの 10 進数) です。 出力する値が 100000000000 以上の場合, "*****" が出力されます。
nn..nn	リクエストタイムアウト (11 けたの 10 進数) です。 出力する値が 100000000000 以上の場合, "*****" が出力されます。
oo..oo	リクエスト障害 (11 けたの 10 進数) です。 出力する値が 100000000000 以上の場合, "*****" が出力されます。
pp..pp	TSCORB コネクタのレスポンス時間 (11 けたの 10 進数) です。 出力する値が 100000000000 以上の場合, "*****" が出力されます。
qq..qq	TSCORB コネクタのリクエストタイムアウト (11 けたの 10 進数) です。 出力する値が 100000000000 以上の場合, "*****" が出力されます。
rr..rr	TSCORB コネクタのリクエスト障害 (11 けたの 10 進数) です。 出力する値が 100000000000 以上の場合, "*****" が出力されます。

注

サーバアプリケーションのオペレーション単位に繰り返し出力されます。

サーバアプリケーションのオペレーション単位の情報の CSV 形式での出力形式

-TSCEdit オプションに "rac" を指定し、かつ -TSCCsv オプションを指定して、サーバアプリケーションのオペレーション単位のリクエスト情報を CSV 形式で出力する形式を次に示します。

```
*** Statistics Information by Operation ***
TSCDomain, TSCID, Interface, Operation, start, end, Event, Count, Maximum, Minimum, Average, unit
aa..aa, bb..bb, cc..cc, dd..dd, ee..ee, ff..ff, gg..gg, hh..hh, ii..ii, jj..jj, kk..kk, ll..ll
aa..aa, bb..bb, cc..cc, dd..dd, ee..ee, ff..ff, gg..gg, hh..hh, ii..ii, jj..jj, kk..kk, ll..ll
```

(凡例)

記号	説明
aa..aa	TSC ドメイン名称 (1 ~ 31 文字の英数字) です。
bb..bb	TSC 識別子 (1 ~ 31 文字の英数字) です。
cc..cc	インタフェース名 (1 ~ 31 文字の英数字) です。
dd..dd	オペレーション名 (1 ~ 31 文字の英数字) です。
ee..ee	編集開始時間です。
ff..ff	編集終了時間です。
gg..gg	イベント内容です。
hh..hh	発生件数 (11 けたの 10 進数) です。 出力する値が 100000000000 以上の場合, "*****" が出力されます。
ii..ii	最大値 (11 けたの 10 進数) です。 出力する値が 100000000000 以上の場合, "*****" が出力されます。
jj..jj	最小値 (11 けたの 10 進数) です。 出力する値が 100000000000 以上の場合, "*****" が出力されます。
kk..kk	平均値 (11 けたの 10 進数) です。 出力する値が 100000000000 以上の場合, "*****" が出力されます。
ll..ll	単位です。

注

-TSCCsv オプションを指定すると、編集時間内の統計情報データが取得されている時間帯、および編集時間内で稼働統計が取得されていない時間帯は出力されません。

tscstsstart (稼働統計情報の取得の開始)

形式

```
tscstsstart [-h] | [-TSCDomain TSCドメイン名称] [-TSCID TSC識別子]
            [-TSCInterval 時間間隔] [-TSCCheckInterval 時間間隔]
```

機能

稼働統計情報ファイルへの情報の取得を開始します。TSC デーモンを開始する前に tscstsstart コマンドを実行するとエラーになります。

稼働統計情報の詳細については、「2.9 稼働統計情報の取得」を参照してください。

オプション

-h

コマンドの使用方法が表示されます。

-TSCDomain TSC ドメイン名称

~ < 1 ~ 31 文字の英数字 > 《TSCDOMAIN》

TSC ドメインマネージャが属する TSC ドメイン名称を指定します。"TSC" または "tsc" で始まる TSC ドメイン名称は指定しないでください。省略した場合は "TSCDOMAIN" が設定されます。

-TSCID TSC 識別子

~ < 1 ~ 31 文字の英数字, およびピリオド (.) > 《IP アドレス》

稼働統計情報ファイルへの情報の取得を開始する TSC ノードの識別子を指定します。"TSC" または "tsc" で始まる文字列は指定しないでください。ピリオド (.) は、IP アドレスを指定する場合だけ使用できます。マルチホームドホスト環境で使用する場合は、-TSCID オプションは省略できません。省略した場合は、IP アドレスで TSC 識別子が設定されます。

-TSCInterval 時間間隔

~ < 符号なし整数 > ((1 ~ 1440)) 《1》(単位: 分)

稼働統計情報ファイルに情報を取得する時間の間隔を分単位で指定します。省略した場合は "1 (分)" が設定されます。

-TSCCheckInterval 時間間隔

~ < 符号なし整数 > ((1 ~ 60)) 《60》(単位: 秒)

コネクション情報を取得する時間の間隔を秒単位で指定します。コネクション数が多くなると、コネクション情報の取得に時間が掛かるようになります。そのため、コネクション数が多い場合は指定する値を大きくしてください。省略した場合は "60 (秒)" が設定されます。

戻り値

このコマンドは次に示す戻り値をシェルに返してから、処理を終了します。

戻り値	意味
0	正常終了しました。
0 以外	コマンド処理中にエラーが発生したために異常終了しました。 出力されたメッセージに従って対策したあと、再度、コマンドを実行してください。

注意事項

tscstsstart コマンドを実行して -TSCInterval オプションに指定した時間が経過した時点で、稼働統計情報ファイルに情報が出力されるとは限りません。それは、tscstsstart コマンドを実行して最初に XX 時 YY 分 00 秒 (X, Y : 符号なし整数) となったときから、TSC ノードからの稼働統計情報の取得が開始されるためです。この取得開始から -TSCInterval オプションに指定した時間が経過した時点で、稼働統計情報ファイルに情報が出力されます。

例えば、tscstsstart コマンドに -TSCInterval 1 を指定して、18 時 10 分 30 秒に実行すると、稼働統計情報の取得は 18 時 11 分 00 秒に開始されます。つまり、稼働統計情報ファイルが出力されるのは 18 時 12 分 00 秒となります。

tscstsstop (稼働統計情報の取得の終了)

形式

```
tscstsstop [-h] | [-TSCDomain TSCドメイン名称] [-TSCID TSC識別子]
```

機能

稼働統計情報ファイルへの情報の取得を終了します。

稼働統計情報の詳細については、「2.9 稼働統計情報の取得」を参照してください。

オプション

-h

コマンドの使用方法が表示されます。

-TSCDomain TSC ドメイン名称

~ < 1 ~ 31 文字の英数字 > 《TSCDOMAIN》

TSC ドメインマネージャが属する TSC ドメイン名称を指定します。"TSC" または "tsc" で始まる TSC ドメイン名称は指定しないでください。省略した場合は "TSCDOMAIN" が設定されます。

-TSCID TSC 識別子

~ < 1 ~ 31 文字の英数字, およびピリオド (.) > 《IP アドレス》

稼働統計情報ファイルへの情報の取得を終了する TSC ノードの識別子を指定します。"TSC" または "tsc" で始まる文字列は指定しないでください。ピリオド (.) は、IP アドレスを指定する場合だけ使用できます。マルチホームドホスト環境で使用する場合は、-TSCID オプションは省略できません。省略した場合は、IP アドレスで TSC 識別子が設定されます。

戻り値

このコマンドは次に示す戻り値をシェルに返してから、処理を終了します。

戻り値	意味
0	正常終了しました。
0 以外	コマンド処理中にエラーが発生したために異常終了しました。 出力されたメッセージに従って対策したあと、再度、コマンドを実行してください。

tscstsstop (稼働統計情報の取得の終了)

注意事項

tscstsstart コマンドの -TSCInterval オプションに指定した時間間隔の経過前に tscstsstop コマンドを実行すると、tscstsstop コマンドの実行までに取得した情報がすべてファイルに出力されます。

tscver (実行形式ファイルおよびライブラリのバージョン情報の出力)

形式

tscver -h | ファイルパス

機能

OTM が提供する実行形式ファイルおよびライブラリのバージョン情報を出力します。

オプション

-h

コマンドの使用方法が表示されます。

コマンド引数

ファイルパス

~ <文字列>

対象となるファイルパス (相対パスおよび絶対パス) を指定します。

戻り値

このコマンドは次に示す戻り値をシェルに返してから、処理を終了します。

戻り値	意味
0	正常終了しました。
0 以外	コマンド処理中にエラーが発生したために異常終了しました。 出力されたメッセージに従って対策したあと、再度、コマンドを実行してください。

出力形式

```
Information For : aa..aa
Product Name   : bb..bb
PP Number      : cc..cc
Version        : dd..dd
Copyright      : ee..ee
Company        : ff..ff
Build OS       : gg..gg
Build Compiler : hh..hh
Build Date     : ii..ii jj..jj
```

(凡例)

tscver (実行形式ファイルおよびライブラリのバージョン情報の出力)

記号	説明
aa..aa	ファイル名称 (文字列) です。
bb..bb	製品名称 (文字列) です。
cc..cc	PP 形名 (文字列) です。
dd..dd	バージョンおよびリビジョン (文字列) です。
ee..ee	コピーライト (文字列) です。
ff..ff	会社名 (文字列) です。
gg..gg	ビルド OS (文字列) です。
hh..hh	ビルドコンパイラ (文字列) です。
ii..ii	ビルド日付 (文字列) です。
jj..jj	ビルド時刻 (文字列) です。

クライアントアプリケーションの開始

形式

クライアントアプリケーション名称

```

[-TSCDomain TSCドメイン名称] [-TSCID TSC識別子]
[-TSCMyHost ホスト名称またはIPアドレス]
[-TSCTimeOut タイムアウト値] [-TSCNice nice値]
[-TSCRequestPriority プライオリティ値]
[-TSCRequestWay 1 | 0]
[-TSCClientMessageBufferCount クライアント通信用バッファ数]
[-TSCRebindTimes リバインド回数]
[-TSCRebindInterval リバインド間隔]
[-TSCTimeOutRebind]
[-TSCPort ポート番号]
[-TSCRetryReference 接続先情報ファイル名称]
[-TSCRetryWay マルチノードリトライ接続の接続方式]
[-TSCDebugInfo]
[-TSCWithSystem 1 | 0]
[-TSCSessionInterval セッション呼び出しインターバル監視時間]

```

オプション

-TSCDomain TSC ドメイン名称

~ < 1 ~ 31 文字の英数字 > 《TSCDOMAIN》

自プロセスが属する TSC ドメイン名称を指定します。"TSC" または "tsc" で始まる TSC ドメイン名称は指定しないでください。省略した場合は "TSCDOMAIN" が設定されません。

-TSCID TSC 識別子

~ < 1 ~ 31 文字の英数字, およびピリオド (.) > 《IP アドレス》

自プロセスが属する TSC 識別子を指定します。ピリオド (.) は、IP アドレスを指定する場合だけ使用できます。省略した場合は、IP アドレスが TSC 識別子に設定されます。例えば、ホストの IP アドレスが "172.17.112.43" のとき、"172.17.112.43" という文字列が TSC 識別子となります。

-TSCMyHost ホスト名称または IP アドレス

~ < 1 ~ 64 文字の文字列 > 《hostname コマンドで取得されるホスト名称》

マルチホームドホスト環境で OTM が使用するホスト名称または IP アドレスを指定します。省略した場合は、hostname コマンドで取得されるホスト名称が設定されます。

-TSCID オプションを省略して -TSCMyHost オプションを指定した場合、デフォルトの TSC 識別子は -TSCMyHost オプションで指定した IP アドレスになります。

ただし、Java アプレットでは -TSCMyHost オプションの指定は無効です。-TSCID オプ

ションを省略して `-TSCMyHost` オプションを指定した場合に `-TSCMyHost` オプションの値はデフォルトになりません。必要に応じて、`-TSCID` オプションに IP アドレスを指定してください。

`-TSCTimeOut` タイムアウト値

~ <符号なし整数> ((0 ~ 2147483647)) 《180》(単位: 秒)

サービスの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指定します。また、接続・切断処理やリバインドで発生する通信の時間を監視するためにも使用されます。"0" を指定した場合、無限に待ち続けます。省略した場合は "180 (秒)" が設定されます。

`-TSCNice nice` 値

~ <符号なし整数> 《0》

プロセスの `nice` 値の増分を指定します。省略した場合は "0" が設定されます。指定値の範囲は OS が提供する `nice` の指定範囲に従います。

`-TSCRequestPriority` プライオリティ値

~ <符号なし整数> ((1 ~ 8)) 《4》

スケジューリングキューに滞留したリクエストを引き出すときのプライオリティ (優先順位) を指定します。指定値が小さいほどプライオリティは高くなります。

省略した場合は "4" が設定されます。

`-TSCRequestWay 1 | 0`

~ <符号なし整数> ((1 | 0)) 《0》

クライアントアプリケーションからの接続経路を指定します。なお、シングルスレッドライブラリを使用するアプリケーションプログラムの場合、TSC デーモンに直結したリクエストはできません。

"0" を指定すると、TSC デーモンに直結してリクエストします。

"1" を指定すると、TSC レギュレータを経由してリクエストします。TSC レギュレータによって接続のレギュレートを実行します。

省略した場合は "0" が設定されます。

`-TSCClientMessageBufferCount` クライアント通信用バッファ数

~ <符号なし整数> ((1 ~ 64)) 《1》

TSC デーモンに直結するクライアントアプリケーションの通信用バッファの数を指定します。

TSC デーモンに直結するクライアントアプリケーションの場合、
 -TSCClientMessageBufferCount オプションで指定した値に従ってクライアント通信用バッファが生成されます。TSC レギュレータを経由するクライアントアプリケーションの場合、-TSCClientMessageBufferCount オプションは無視されます。

クライアント通信用バッファはクライアントアプリケーションがリクエストに対する応答を TSC デーモンから受信するためのバッファです。クライアントアプリケーションプロセス内で、特定の TSC デーモンに同時に要求するスレッド数が増加すると、通信用バッファに対して多数のスレッドが同時に要求するようになり、スレッド間での競合が増加します。そこで、同時に要求するスレッド数に応じてクライアント通信用バッファ数を増加することによって競合を減少できます。スレッド間での競合を増加させないで性能を維持するためには、要求するスレッド数 "4" に対して、クライアント通信用バッファ数 "1" を目安に指定することをお勧めします。

省略した場合は "1" が設定されます。

-TSCRebindTimes リバインド回数

~ <符号なし整数> ((0 ~ 255)) 《0》

1 回のメソッド呼び出しでリバインドを繰り返す回数を指定します。"0" を指定した場合はリバインドしないで、例外がクライアントアプリケーションに返されます。省略した場合は "0" が設定されます。

オートリバインド機能の詳細については、「2.1.3 TSC ノードとのリバインド」を参照してください。

-TSCRebindInterval リバインド間隔

~ <符号なし整数> ((0 ~ 65535)) 《15》(単位: 秒)

リバインドが失敗してから、次のリバインドを実行するまでの間隔を指定します。省略した場合は "15 (秒)" が設定されます。

-TSCTimeOutRebind

TSCNoResponseException クラスの TIMED_OUT 例外が発生した場合にリバインドを実行します。-TSCTimeOutRebind オプションは、-TSCRebindTimes オプションの指定時だけ有効です。詳細については、「2.1.3 TSC ノードとのリバインド」を参照してください。

-TSCPort ポート番号

~ <符号なし整数> ((5001 ~ 65535))

クライアントアプリケーションの新しい接続を探すときに使用するポート番号を指定します。省略した場合は、OS が自動的に割り当てたポート番号を使用します。

-TSCRetryReference 接続先情報ファイル名称

UNIX の場合 ~ <英数字, ピリオド (.), スラント (/), コロン (:), チルダ (~), プラス (+), およびハイフン (-)>

Windows の場合 ~ <英数字, ピリオド (.), スラント (/), 円符号 (¥), コロン (:), チルダ (~), プラス (+), およびハイフン (-)>

注

UNIX の場合の ":", "~", および Windows の場合の "/", "~" は, Java アプリケーションまたは Java アプレットで, 接続先情報ファイルを URL で指定する場合だけ使用できます。

クライアントアプリケーションからリクエストを送信するときの, TSC ノードへの接続時に使用される接続先情報ファイルを指定します。接続先情報ファイルは相対パスまたは絶対パスで指定できます。

Java アプリケーションまたは Java アプレットの場合には, WWW サーバ上の接続先情報ファイルを URL で指定できます。接続先情報ファイルの URL は次に示す形式で指定してください。

<http>://<ホスト名またはIPアドレス>[:<ポート番号>]/<パス>

ただし, ローカルコンピュータ内のファイルを指定する場合は, 接続先情報ファイル名称を上記の形式にはしないでください。また, Java アプレットに対してローカルファイルを指定する場合, または Java アプレット自身のダウンロード元以外のコンピュータにあるファイルを URL で指定する場合には, Java アプレットでの読み込みが許可されるように署名を設定する必要があります。

接続先情報ファイルは tscgetref コマンドで作成してください。接続先情報ファイルの詳細については, 「2.2 接続先情報ファイルを使用する接続」を参照してください。

-TSCRetryWay マルチノードリトライ接続の接続方式

~ < 4 けたの符号なし整数 > 《0000》

接続先情報ファイルを指定してマルチノードリトライ接続を実行する場合に, その接続方式を指定します。省略した場合は "0000 (ファイル検索方式かつ順次方式)" が設定されます。

上位 1 けた目 ~ 2 けた目

"00" を指定してください。

上位 3 けた目

マルチノードリトライ方式を指定します。

"0" を指定すると, ファイル検索方式で接続します。この場合, TSCAdm クラスの

getTSCClient() メソッドに指定した条件に従って、接続先情報ファイルから接続対象が選択されます。

"1" を指定すると、API 優先方式で接続します。この場合、最初の接続では、TSCDomain オブジェクトに指定した TSC ドメイン名称と TSC 識別子を持つ TSC ノードに対して TPBroker のスマートエージェントを使用します。接続に失敗した場合は、接続先情報ファイルから接続対象が選択されます。

上位 4 けた目

マルチノードリトライ順序を指定します。

"0" を指定すると、接続先情報ファイルの先頭から順に接続します（順次方式）。

"1" を指定すると、接続先情報ファイルからランダムに接続対象を選択して接続します（ランダム方式）。

-TSCRetryWay オプションの指定値と、実際に選択される接続対象の関係については、「2.2.4 マルチノードリトライ接続の接続対象」を参照してください。なお、-TSCRetryWay オプションは、-TSCRetryReference オプションで接続先情報ファイルが正しく指定されている場合だけ有効となります。

-TSCDebugInfo

OTM - Client の Java アプリケーションまたは Java アプレットの使用時に、OTM - Client が提供する C++ ライブラリからログメッセージおよびトレース情報を出力します。出力先は TSCSPOOL 環境変数に指定されるディレクトリです。OTM - Client の障害情報については、「3.3.2 OTM - Client で取得できる障害情報」を参照してください。

-TSCWithSystem 1 | 0

~ <符号なし整数> ((1 | 0)) 《1》

クライアントアプリケーションを TSC デーモンと連動させるかどうかを指定します。省略した場合は "1 (連動させる)" が設定されます。OTM - Client の場合は常に "0" が設定されます。

"1" を指定すると、クライアントアプリケーションのプロセスは TSC デーモンに登録されます。TSC デーモンは停止時に、登録されたクライアントアプリケーションの停止を待ちます。

"0" を指定すると、そのクライアントアプリケーションは、TSC デーモンとは独立して動作します。そのため、TSC デーモンは停止時に、クライアントアプリケーションの停止を待ちません。クライアントアプリケーションのプロセスを停止するタイミングはユーザ任意となります。なお、"0" を指定して開始したクライアントアプリケーションのプロセスは管理元の TSC デーモンに登録されないため、tscprecls コマンドの出力結果には表示されません。また、異常終了した場合も、稼働統計情報の「アプリケーションプログラム異常終了」の件数には加算されません。さらに、\$TSCSPOOL の有無もチェックされないため、\$TSCSPOOL が設定されていなくてもエラーになりません。ただし、

\$TSCSPOOL が設定されているときは、障害情報などは出力できます。

-TSCSessionInterval セッション呼び出しインターバル監視時間

~ <符号なし整数> ((1 ~ 2147483647)) 《180》(単位：秒)

セッション呼び出し中のオブジェクトが、応答を返してから次のリクエストまたはセッション呼び出しの終了要求を受け取るまでを監視する時間（最大時間間隔）を指定します。省略した場合は "180 (秒)" が設定されます。

セッション呼び出しインターバル監視時間がタイムアウトになった場合、TSCRootAcceptor クラスの deactivate() メソッドと同じ処理をして、セッション呼び出し待ち状態だったオブジェクトが使用されないようにします。セッション呼び出しインターバル時間監視の詳細については、「2.4.8 セッション呼び出しインターバル時間監視」を参照してください。

サーバアプリケーションの開始

形式

サーバアプリケーション名称

```

[-TSCDomain TSCドメイン名称] [-TSCID TSC識別子]
[-TSCAPID アプリケーション識別子]
[-TSCMyHost ホスト名称またはIPアドレス]
[-TSCRootAcceptor TSCルートアクセプタ登録名称]
[-TSCNice nice値] [-TSCTimeOut タイムアウト値]
[-TSCParallelCount 常駐スレッド数]
[-TSCRequestPriority プライオリティ値]
[-TSCAcceptor TSCアクセプタ名称]
[-TSCRequestWay 1 | 0]
[-TSCClientMessageBufferCount クライアント通信用バッファ数]
[-TSCRebindTimes リバインド回数]
[-TSCRebindInterval リバインド間隔]
[-TSCTimeOutRebind]
[-TSCWatchTime クラス監視時間] [-TSCWatchMethod メソッド監視時間]
[-TSCPort ポート番号]
[-TSCRetryReference 接続先情報ファイル名称]
[-TSCRetryWay マルチノードリトライ接続の接続方式]
[-TSCSessionInterval セッション呼び出しインターバル監視時間]

```

オプション

-TSCDomain TSC ドメイン名称

~ < 1 ~ 31 文字の英数字 > 《TSCDOMAIN》

自プロセスを管理する TSC ドメイン名称を指定します。"TSC" または "tsc" で始まる TSC ドメイン名称は指定しないでください。省略した場合は "TSCDOMAIN" が設定されます。

-TSCID TSC 識別子

~ < 1 ~ 31 文字の英数字, およびピリオド(.) > 《IP アドレス》

自プロセスを管理する TSC 識別子を指定します。ピリオド(.) は、IP アドレスを指定する場合だけ使用できます。同じ TSC ドメインと TSC 識別子を持つ TSC デモンが、同じコンピュータ内で開始している必要があります。

省略した場合は、IP アドレスが TSC 識別子に設定されます。例えば、ホストの IP アドレスが "172.17.112.43" のとき、"172.17.112.43" という文字列が TSC 識別子となります。

-TSCAPID アプリケーション識別子

~ < 1 ~ 32 文字の英数字 >

開始するプロセスのアプリケーション識別子を指定します。-TSCDomain オプション、

および -TSCID オプションで指定した TSC ノード内で一意となるように指定してください。

-TSCAPID オプションを指定したサーバアプリケーションは、tscstopprc コマンドで終了できます。

-TSCMyHost ホスト名称または IP アドレス

~ < 1 ~ 64 文字の文字列 > 《hostname コマンドで取得されるホスト名称》

マルチホームドホスト環境で OTM が使用するホスト名称または IP アドレスを指定します。省略した場合は、hostname コマンドで取得されるホスト名称が設定されます。

-TSCID オプションを省略して -TSCMyHost オプションを指定した場合、デフォルトの TSC 識別子は -TSCMyHost オプションで指定した IP アドレスになります。

-TSCRootAcceptor TSC ルートアクセプタ登録名称

~ < 1 ~ 31 文字の英数字 > 《default》

TSC ルートアクセプタ登録名称を指定します。

-TSCRootAcceptor オプションの指定値は、TSC ルートアクセプタ登録名称を指定しないで、TSC ルートアクセプタの状態を active 状態に変更する場合に利用されます。省略した場合は "default" が設定されます。

-TSCNice nice 値

~ < 符号なし整数 > 《0》

自プロセスの nice 値の増分を指定します。省略した場合は、"0" が設定されます。指定値の範囲は OS が提供する nice の指定範囲に従います。

-TSCTimeOut タイムアウト値

~ < 符号なし整数 > ((0 ~ 2147483647)) 《180》(単位：秒)

サービスの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指定します。また、接続・切断処理やリバインドで発生する通信の時間を監視するためにも使用されます。"0" を指定した場合、無限に待ち続けます。省略した場合は "180 (秒)" が設定されます。

-TSCParallelCount 常駐スレッド数

~ < 符号なし整数 > ((1 ~ 127)) 《1》

TSC ルートアクセプタ単位に常駐するスレッド数 (パラレルカウント) を指定します。

省略した場合は "1" が設定されます。

-TSCRequestPriority プライオリティ 値

~ < 符号なし整数 > ((1 ~ 8)) 《4》

スケジューリング用キューに滞留したリクエストを引き出すときのプライオリティ（優先順位）を指定します。指定値が小さいほどプライオリティは高くなります。

省略した場合は "4" が設定されます。

-TSCAcceptor TSC アクセプタ名称

~ < 1 ~ 31 文字の英数字 >

TSC アクセプタ名称を指定します。

TSC アクセプタ名称を指定しないで TSC アクセプタのインスタンスを生成する場合に利用されます。

-TSCRequestWay 1 | 0

~ < 符号なし整数 > ((1 | 0)) 《0》

サーバアプリケーションからの接続経路を指定します。なお、シングルスレッドライブラリを使用するアプリケーションプログラムの場合、TSC デーモンに直結したリクエストはできません。

"0" を指定すると、TSC デーモンに直結してリクエストします。

"1" を指定すると、TSC レギュレータを経由してリクエストします。TSC レギュレータによってコネクションのレギュレートを実行します。

省略した場合は "0" が設定されます。

-TSCClientMessageBufferCount クライアント通信用バッファ数

~ < 符号なし整数 > ((1 ~ 64)) 《1》

TSC デーモンに直結するクライアントアプリケーションの通信用バッファの数を指定します。

TSC デーモンに直結するクライアントアプリケーションの場合、

-TSCClientMessageBufferCount オプションで指定した値に従ってクライアント通信用バッファが生成されます。TSC レギュレータを経由するクライアントアプリケーションの場合、-TSCClientMessageBufferCount オプションは無視されます。

クライアント通信用バッファはクライアントアプリケーションがリクエストに対する応答を TSC デーモンから受信するためのバッファです。クライアントアプリケーションプロセス内で、特定の TSC デーモンに同時に要求するスレッド数が増加すると、通信用バッファに対して多数のスレッドが同時に要求するようになり、スレッド間での競合が増加します。そこで、同時に要求するスレッド数に応じてクライアント通信用バッファ

数を増加することによって競合を減少できます。スレッド間での競合を増加させないで性能を維持するためには、要求するスレッド数 "4" に対して、クライアント通信用バッファ数 "1" を目安に指定することをお勧めします。

省略した場合は "1" が設定されます。

-TSCRebindTimes リバインド回数

~ < 符号なし整数 > ((0 ~ 255)) 《0》

1 回のメソッド呼び出しでリバインドを繰り返す回数を指定します。"0" を指定した場合はリバインドしないで、例外がクライアントアプリケーションに返されます。省略した場合は "0" が設定されます。

オートリバインド機能の詳細については、「2.1.3 TSC ノードとのリバインド」を参照してください。

-TSCRebindInterval リバインド間隔

~ < 符号なし整数 > ((0 ~ 65535)) 《15》(単位: 秒)

リバインドが失敗してから、次のリバインドを実行するまでの間隔を指定します。省略した場合は "15 (秒)" が設定されます。

-TSCTimeOutRebind

TSCNoResponseException クラスの TIMED_OUT 例外が発生した場合にリバインドを実行します。-TSCTimeOutRebind オプションは、-TSCRebindTimes オプションの指定時だけ有効です。詳細については、「2.1.3 TSC ノードとのリバインド」を参照してください。

-TSCWatchTime クラス監視時間

~ < 符号なし整数 > ((0 ~ 2147483647)) 《300》(単位: 秒)

TSCWatchTime クラスの監視時間を指定します。監視時間を指定しないで TSCWatchTime クラスのインスタンスを生成する場合に使用します。"0" を指定した場合、監視をしません。省略した場合は "300 (秒)" が設定されます。

サーバアプリケーションの時間監視の機能については、「2.7.6 サーバアプリケーションの時間監視」を参照してください。

-TSCWatchMethod メソッド監視時間

~ < 符号なし整数 > ((0 ~ 2147483647)) 《0》(単位: 秒)

IDL ファイルに記述したメソッド単位の実行監視時間を指定します。

指定した場合には、監視時間内にメソッドの処理が終了しないとき、プロセスを異常終了します。-TSCWatchMethod オプションと TSCWatchTime クラスとは併用できます。

また、TSCWatchTime クラスをアプリケーションプログラム中で記述しない場合でも、メソッドの実行監視時間については -TSCWatchMethod オプションで設定できます。

"0" を指定した場合は監視をしません。省略した場合は "0" が設定されます。

-TSCPort ポート番号

~ <符号なし整数> ((5001 ~ 65535))

サーバアプリケーションの新しいコネクションを探すときに使用するポート番号を指定します。省略した場合は、OS が自動的に割り当てたポート番号を使用します。

-TSCRetryReference 接続先情報ファイル名称

UNIX の場合 ~ <英数字, ピリオド (.), スラント (/), コロン (:), チルダ (~), プラス (+), およびハイフン (-)>

Windows の場合 ~ <英数字, ピリオド (.), スラント (/), 円符号 (¥), コロン (:), チルダ (~), プラス (+), およびハイフン (-)>

注

UNIX の場合の ":", "~", および Windows の場合の "/", "~" は、Java アプリケーションで、接続先情報ファイルを URL で指定する場合だけ使用できます。

サーバアプリケーションからリクエストを送信するときの、TSC ノードへの接続時に使用される接続先情報ファイルを指定します。接続先情報ファイルは相対パスまたは絶対パスで指定できます。

Java アプリケーションの場合には、WWW サーバ上の接続先情報ファイルを URL で指定できます。接続先情報ファイルの URL は次に示す形式で指定してください。

<http>://<ホスト名またはIPアドレス>[:<ポート番号>]/<パス>

ただし、ローカルコンピュータ内のファイルを指定する場合は、接続先情報ファイル名称を上記の形式にはしないでください。

接続先情報ファイルは tscgetref コマンドで作成してください。接続先情報ファイルの詳細については、「2.2 接続先情報ファイルを使用する接続」を参照してください。

-TSCRetryWay マルチノードリトライ接続の接続方式

~ <4 けたの符号なし整数> 《0000》

接続先情報ファイルを指定してマルチノードリトライ接続を実行する場合に、その接続方式を指定します。省略した場合は "0000 (ファイル検索方式かつ順次方式)" が設定されます。

上位 1 けた目 ~ 2 けた目

"00" を指定してください。

上位 3 けた目

マルチノードリトライ方式を指定します。

"0" を指定すると、ファイル検索方式で接続します。この場合、TSCAdm クラスの getTSCClient() メソッドに指定した条件に従って、接続先情報ファイルから接続対象が選択されます。

"1" を指定すると、API 優先方式で接続します。この場合、最初の接続では、TSCDomain オブジェクトに指定した TSC ドメイン名称と TSC 識別子を持つ TSC ノードに対して TPBroker のスマートエージェントを使用します。接続に失敗した場合は、接続先情報ファイルから接続対象が選択されます。

上位 4 けた目

マルチノードリトライ順序を指定します。

"0" を指定すると、接続先情報ファイルの先頭から順に接続します（順次方式）。

"1" を指定すると、接続先情報ファイルからランダムに接続対象を選択して接続します（ランダム方式）。

-TSCRetryWay オプションの指定値と、実際に選択される接続対象の関係については、「2.2.4 マルチノードリトライ接続の接続対象」を参照してください。なお、

-TSCRetryWay オプションは、-TSCRetryReference オプションで接続先情報ファイルが正しく指定されている場合だけ有効となります。

-TSCSessionInterval セッション呼び出しインターバル監視時間

~ <符号なし整数> ((1 ~ 2147483647)) 《180》(単位: 秒)

セッション呼び出し中のオブジェクトが、応答を返してから次のリクエストまたはセッション呼び出しの終了要求を受け取るまでを監視する時間（最大時間間隔）を指定します。省略した場合は "180 (秒)" が設定されます。

セッション呼び出しインターバル監視時間がタイムアウトになった場合、TSCRootAcceptor クラスの deactivate() メソッドと同じ処理をして、セッション呼び出し待ち状態だったオブジェクトが使用されないようにします。セッション呼び出しインターバル時間監視の詳細については、「2.4.8 セッション呼び出しインターバル時間監視」を参照してください。

注意事項

サーバアプリケーションのオプションは、同じ TSC ノードに属する TSC デーモンの開始時に指定できます。サーバアプリケーションの開始時に指定しなかったオプションのデフォルトには、TSC デーモンの開始時に指定された値が引き継がれます。

ただし、次に示すオプションについては、同じ TSC ノードの TSC デーモンの開始時に指定されていても引き継がれません。

- -TSCDomain オプション

- -TSCID オプション
- -TSCAPID オプション
- -TSCMyHost オプション
- -TSCPort オプション
- -TSCRetryReference オプション
- -TSCRetryWay オプション

6

メッセージ

この章では、OTM が出力するメッセージについて説明します。

6.1 メッセージの形式

6.2 メッセージ一覧

6.1 メッセージの形式

OTM で出力されるメッセージの形式を示します。

6.1.1 メッセージの出力形式

メッセージの出力先が標準出力および標準エラー出力の場合、または `tslogcat` コマンドで標準出力へ出力する場合は、メッセージ ID とメッセージテキストのほかに、付加情報が出力されます。標準出力に出力される付加情報とメッセージの形式を次に示します。

```
DDD MMM dd hh:mm:ss yyyy: KFOT00000-X AAAAA BBBBB:YY...YY
```

表 6-1 標準出力に出力される付加情報とメッセージ

分類	記号	意味	出力される文字種別と文字数
付加情報	DDD	曜日	半角英字 3 文字
	MMM	月	半角数字 3 文字
	dd	日	半角数字 2 文字
	hh:mm:ss	時分秒	半角数字 8 文字
	yyyy	年	半角数字 4 文字
	AAAAA	プロセス ID	半角数字 5 文字
	BBBBB	スレッド ID	半角数字
メッセージ固定部	KFOT00000-X	メッセージ ID	半角英数字 11 文字
	YY...YY	メッセージテキスト	制限なし

6.1.2 メッセージの記述形式

このマニュアルでの記述形式を次に示します。

KFOTnnnnn - X

メッセージテキスト (Y)

メッセージの意味を説明します。

(S) システムがメッセージを出力したあとの主な処理を示します。

(O) メッセージ確認時、オペレータがする必要のある処置を示します。

【対策】 メッセージ確認時のシステム管理者の処置を示します。

注

メッセージ中の **(O)** または **【対策】** で、「保守員に連絡してください。」とは、当社社員、または当社営業担当部署に連絡していただくことを示します。

表 6-2 メッセージ ID 記号の意味

記号	意味
KFOT	OTM のメッセージであることを示します。
nnnn	メッセージの通し番号です。
X	<p>メッセージの種類を次の値で示します。</p> <p>E :</p> <p>各ライブラリ、コマンド、サーバの機能が動かない障害が起きたことを示します。 または、コマンドのオペランド指定誤りによって、動作ができないことを示します。</p> <p>W :</p> <p>各ライブラリ、コマンド、およびサーバからのメモリの使用状況などについての警告を示しています。 または、コマンドのオペランド指定誤りがありましたが、値を仮定して動作を続行することを示しています。</p> <p>I :</p> <p>上記 E、W に該当しないメッセージで、単純に動作を示しています。</p> <p>R :</p> <p>出力メッセージに対するユーザからの応答を待っていることを示しています。</p>
Y	<p>メッセージの出力先種別を次の値で示します。</p> <p>C : 標準出力</p> <p>E : 標準エラー出力</p> <p>L : メッセージログファイル</p> <p>S : syslog</p> <p>一つのメッセージが複数の出力種別を持つ場合は、出力される可能性のある種別を " + " でつないで示します。また、tslogcat コマンドでの標準出力への出力は "L" となります。</p>

6.2 メッセージ一覧

OTM で出力するメッセージを次に示します。

これら以外にも保守用のメッセージが出力されることがあります。保守用のメッセージは障害対策に必要となることがありますので、出力内容を保存してください。

6.2.1 KFOT26000 ~ KFOT26999

KFOT26949 - E

"aa....aa(bb....bb)" failed.errno=cc....cc (S + E)

システムコールでエラーが発生しました。

aa....aa : エラーとなったシステムコール

bb....bb : tscd 内部情報 (システムコール呼び出し元ファイル名)

cc....cc : システムコールの errno 値

(S) 処理を中止します。

【対策】 システムコールがエラーになった原因を調査してください。

KFOT26975 - E

"malloc(aa....aa)" failed.pid=bb....bb,type=cc....cc (S)

メモリ不足が発生しました。

aa....aa : 確保しようとしたメモリサイズ

bb....bb : プロセス ID

cc....cc : TSC デーモン内部情報

(S) 処理を中止します。

【対策】 メモリを再度、見積もり直してください。

KFOT26977 - E

***** (pid=aa....aa) killed by code=bb....bb (S + E)

プロセスが異常終了しました。

aa....aa : プロセス ID

bb....bb : アポートコード (TSC デーモン内部情報)

(S) 処理を中止します。

(O) システム管理者に連絡してください。

【対策】 保守員に連絡してください。アポートコードについて、「付録 A アポートコー

ド一覧」を参照してください。

KFOT26978 - E

The specified variable value is incorrect. file-name:aa....aa line-bb....bb cc....cc (S)

定義解析で変数の指定値に誤りがありました。

aa....aa : エラーとなった定義ファイル名称

bb....bb : エラーを検出した行番号

cc....cc : エラーとなった定義句

(S) 処理を中止します。

(O) システム管理者に連絡してください。

【対策】 定義を見直し、再度実行してください。

KFOT26979 - E

The definition file could not be opened. definition-file-name:aa....aa error-number=bb....bb (S)

定義ファイルのオープンに失敗しました。

aa....aa : エラーとなった定義ファイル名称

bb....bb : エラー番号

(S) 処理を中止します。

(O) システム管理者に連絡してください。

【対策】 定義を見直し、再度実行してください。

KFOT26980 - E

The specified variable name contains an error. file-name:aa....aa line-bb....bb (S)

定義解析で定義句名に誤りがありました。

aa....aa : エラーとなった定義ファイル名称

bb....bb : エラーを検出した行番号

(S) 処理を中止します。

(O) システム管理者に連絡してください。

【対策】 定義を見直し、再度実行してください。

KFOT26981 - E

Definition file analysis cannot be performed due to insufficient memory space.

Required-memory-space=aa....aa (S)

定義解析でメモリ不足が発生しました。

6. メッセージ

aa....aa : 確保しようとしたメモリ量

(S) 処理を中止します。

(O) システム管理者に連絡してください。

【対策】メモリを再度、見積もり直してください。

KFOT26984 - E

Record length exceeds the limits. file-name:aa....aa line-bb....bb (S)

定義情報のレコード長が最大レコード長(80バイト)を超えています。

aa....aa : エラーとなった定義ファイル名称

bb....bb : エラーとなった行番号

(S) 処理を中止します。

(O) システム管理者に連絡してください。

【対策】定義を見直し、再度実行してください。

KFOT26985 - E

invalid option is specified with this command. (E)

該当するコマンドでは、指定できないオプションが指定されています。

(S) 処理を中止します。

(O) システム管理者に連絡してください。

【対策】コマンドオプションを見直し、再度実行してください。

KFOT26986 - E

number of command arguments or options exceeds the limit. (E)

コマンド引数またはオプションの数が許容範囲を超えています。

(S) 処理を中止します。

(O) システム管理者に連絡してください。

【対策】コマンド引数を見直し、再度実行してください。

KFOT26987 - E

length of command arguments or option arguments exceeds the limit. (E)

コマンド引数またはオプション引数の長さが許容範囲を超えています。

(S) 処理を中止します。

(O) システム管理者に連絡してください。

【対策】コマンド引数を見直し、再度実行してください。

KFOT26991 - W

The log file is decided as aa....aa because the file used in the last employment at the end can't be referred to. The final filename: aa....aa (S + E)

前回，最後に使用したログファイルに障害が発生したため，最初に使用するログファイルを決定します。

aa....aa：前回の運用で最後に使用されたログファイル名称

(S) 処理を続行します。

(O) システム管理者に連絡してください。

【対策】 障害となった原因を調査してください。

KFOT26992 - E

An error occurred in the log file (aa....aa). Output to the log file is aborted. (S + E)

ログファイルに入出力障害が発生したため，このプロセスからの以降のログメッセージはログファイルには出力しないで，syslog ファイルにだけ出力します。

aa....aa：障害となったログファイル名称

(S) 処理を続行します。

(O) システム管理者に連絡してください。

【対策】 障害となった原因を調査してください。

KFOT26993 - E

An error occurred in the log administration file. Output to the log file is aborted. (S + E)

ログ管理ファイルに入出力障害が発生したため，このプロセスからの以降のログメッセージはログファイルには出力しないで，syslog ファイルにだけ出力します。

(S) 処理を続行します。

(O) システム管理者に連絡してください。

【対策】 障害となった原因を調査してください。

KFOT26994 - E

An error occurred in the Initialization of the message log circumstance. (S + E)

ログメッセージ環境の初期化処理で障害が発生しました。メッセージの直前に出力されているメッセージを参照して障害を除去してください。

(S) 処理を続行します。

(O) システム管理者に連絡してください。

【対策】 障害となった原因を調査してください。

6.2.2 KFOT27000 ~ KFOT27999

KFOT27102 - W

prp tracing service cannot be started. service name:aa....aa reason code=bb....bb (S + E)

prf トレースサービスの開始で障害が発生しました。prp は prf トレースサービスを意味します。

aa....aa : 障害の発生した TSC 識別子を示します。

bb....bb : 理由コード

101 : メモリ不足が発生しました。

102 : メモリ不足が発生しました。

103 : ファイルの検定時、または新規作成時に I/O エラーが発生しました。

104 : msgget システムコールでの、新しいメッセージ待ち行列 ID が割り当てられませんでした。ipcs コマンドで使用状況を確認し、不要なメッセージキューがある場合は、ipcrm コマンドによって削除する必要があります。

106 : TSCSPOOL 環境変数に示すディレクトリが正しくありません。また、TSCSPOOL 環境変数を省略した場合、TSCDIR 環境変数に示すディレクトリを見直す必要があります。

112 : 指定されたトレースサービスは、すでに開始されています。または、トレースサービスは終了しているが、トレースを取得していたプロセスが開始中のため、新たにトレースサービスを開始できません。

114 : prf トレースサービスで使用する共用メモリが破壊されています。

115 : prf トレースサービスで使用する共用メモリのアドレスが取得できません。

401 : prf トレースサービスの内部制御ファイルのオープン時にエラーが発生しました。

402 : prf トレースサービスの内部制御ファイルが破壊されています。

403 : prf トレースサービスの内部制御ファイルでエラーが発生しました。TSCSPOOL 環境変数に示すディレクトリが正しく作成されていない可能性があります。また、TSCSPOOL 環境変数を省略した場合、TSCDIR 環境変数に示すディレクトリを見直す必要があります。

404 : トレースファイル作成ディレクトリの指定に誤りがあります。

1601 : システムコールでエラーが発生しました。

(S) 障害の発生した TSC デーモンでの prf トレースサービスを中止します。

(O) 理由コードによって原因を調査してください。

【対策】 理由コードによる障害の原因を取り除いてください。

KFOT27103 - W

prp tracing service cannot be continued. service name: aa....aa reason code=bb....bb (S + E)

prf トレースサービスの、情報取得中に障害が発生しました。prp は prf トレースサービスを意味します。

aa....aa : 障害の発生した TSC 識別子を示します。

bb....bb : 理由コード

101 : メモリ不足が発生しました。

103 : ファイルで I/O エラーが発生しました。

104 : msgget システムコールでの、新しいメッセージ待ち行列 ID が割り当てられませんでした。ipcs コマンドで使用状況を確認し、不要なメッセージキューがある場合は、ipcrm コマンドによって削除する必要があります。

1601 : システムコールエラー発生しました。

(S) 障害の発生した prf トレースサービス、またはシステム統計情報取得サービスを中止します。

(O) 理由コードを確認してください。

【対策】 理由コードによる障害の原因を取り除いてください。

KFOT27104 - E

cannot execute aa....aa command. reason code=bb....bb-cc....cc (E)

コマンドが実行できませんでした。

aa....aa : 実行できなかったコマンド名称

bb....bb : 理由コード

101 : メモリ不足が発生しました。

103 : トレースファイルに I/O エラーが発生しました。

106 : TSCSPOOL 環境変数で示すディレクトリが正しく作成されていない可能性があります。また、TSCSPOOL 環境変数を省略した場合、TSCDIR 環境変数に示すディレクトリを見直す必要があります。

または、指定されたサービス名称はありません。

107 : トレースファイルでオープンエラーが発生しました。

113 : 次の原因が考えられます。

- 指定されたサービス名称はありません。
- トレースサービス中ではありません。
- トレースサービスは開始しているが、トレースサービスプロセスの初期化処理をしており、コマンドが受け付けられる状態ではありません。

802 : 内部制御情報ファイルで I/O エラーが発生しました。

6. メッセージ

804 : トレースファイルが見つかりませんでした。

1601 : システムコールでエラーが発生しました。

cc....cc : システム内部コード

(S) コマンドをエラー終了します。

(O) 理由コードを確認してください。

【対策】 理由コードによる障害を取り除いたあと、再度コマンドを実行してください。

KFOT27105 - E

invalid command format. (E)

コマンドの形式が誤っています。

(S) コマンドをエラー終了します。

(O) 指定したオプションに誤りがないか見直してください。

【対策】 指定オプションを修正後、再度コマンドを実行してください。

KFOT27106 - E

failure to open trace file.filename:aa....aa (E)

トレースファイルのオープンに失敗しました。

aa....aa : オープンに失敗したファイル名

(S) 処理を続行します。

(O) 直前に出力されたシステムコールエラーのメッセージによって、障害の原因を調査してください。

【対策】 オープン失敗の原因を取り除いてください。

KFOT27107 - E

error occurred while handling trace file. (E)

トレースファイルをアクセス中に障害が発生しました。

(S) 処理を続行します。

(O) 直前に出力されたシステムコールエラーのメッセージによって、障害の原因を調査してください。

【対策】 オープン失敗の原因を取り除いてください。

KFOT27108 - I

invalid trace data was detected while reading trace file. (S + E)

トレースファイルをアクセス中にデータ不正が発生しました。

(S) データ不正となったデータを読み飛ばし、処理を続行します。

KFOT27600 - I

aa...aa was assigned as current trace file. ID:bb....bb (L + C)

稼働統計情報を取得するファイルを割り当てました。

aa....aa : 割り当てたファイル名

bb....bb : サービス識別子 (TSC 識別子)

(S) 処理を続行します。

6.2.3 KFOT70000 ~ KFOT70999**KFOT70001 - E**

unusable definition : aa....aa (C)

定義 aa....aa は使用できません。

aa....aa : 使用できない定義

(S) 処理を中止します。

(O) システム管理者に連絡してください。

【対策】 IDL 定義を見直してください。

KFOT70002 - E

cannot open file : aa....aa (C)

ファイル aa....aa がオープンできません。

aa....aa : オープンできないファイル名称

(S) 処理を中止します。

(O) システム管理者に連絡してください。

【対策】 ファイルがあるか、読み込み権限があるかを確認してください。

KFOT70003 - E

invalid command argument. (C)

コマンド引数が間違っています。

(S) 処理を中止します。

(O) システム管理者に連絡してください。

【対策】 コマンド引数を見直してください。

KFOT70004 - E

TSCDIR variable undefined. (C)

6. メッセージ

TSCDIR 環境変数が設定されていません。

(S) 処理を中止します。

(O) システム管理者に連絡してください。

【対策】 TSCDIR 環境変数を設定してください。

KFOT70005 - I

```
usage:tscidl2cpp [-h][-TSCclient_ext ext-string] [-TSCserver_ext ext-string]
[-TSCtemplate_ext ext-string] [-TSCsession_ext ext-string] [-TSChdr_suffix hdr_suffix]
[-TSCsrc_suffix src_suffix] [-client_ext ext-string] [-server_ext ext-string]
[-hdr_suffix hdr_suffix] [-src_suffix src_suffix] [-idl2cpp] [-template] [-TSCspxy]
[-TSCexport tag] [-TSCexport_skel tag] [-export tag] [-export_skel tag]
[-TSCroot_dir directory] [-TSCsrc_dir directory] [-TSChdr_dir directory] [-root_dir directory]
[-TSCno_proxy] [-TSCno_skel] [-TSCidl2cppfix fix-flag] [-type_code_info] [-namespace]
[-TSCcompat] [[-I directory[:directory...]]...] [-A] idfile (C)
```

tscidl2cpp コマンドの使用方法を示します。コマンドの形式が不正な場合、または -h オプションが指定された場合に出力します。-I オプションのコロン (:) は、Windows ではセミコロン (;) で表示されます。

(S) コマンドをエラー終了します。

(O) 指定したオプションに誤りがないか見直してください。

【対策】 指定オプションを修正後、再度コマンドを実行してください。

KFOT70006 - I

```
usage:tscidl2j [-h] | [-idl2java] [-template] [-TSCspxy] [-package package_name]
[-TSCroot_dir directory] [-root_dir directory] [-TSCno_proxy] [-TSCno_skel]
[[-I directory[:directory...]]...] [-A] idfile (C)
```

tscidl2j コマンドの使用方法を示します。コマンドの形式が不正な場合、または -h オプションが指定された場合に出力します。-I オプションのコロン (:) は、Windows ではセミコロン (;) で表示されます。

(S) コマンドをエラー終了します。

(O) 指定したオプションに誤りがないか見直してください。

【対策】 指定オプションを修正後、再度コマンドを実行してください。

KFOT70008 - I

no output file. (C)

出力するファイルがありません。

(S) コマンドを終了します。

(O) 指定したオプションを見直してください。

【対策】 指定オプションを修正後、再度コマンドを実行してください。

KFOT70100 - E

cannot create temporary file : aa....aa. (reason=bb....bb) (C)

一時ファイルを作成できません。

aa....aa : 作成できないファイルの名称

bb....bb : 作成できない理由

(S) 処理を中止します。

(O) システム管理者に連絡してください。

【対策】 ファイルおよびディレクトリのパーミッションなどを見直してください。

KFOT70101 - E

cannot open temporary file : aa....aa. (reason=bb....bb) (C)

一時ファイルをオープンできません。

aa....aa : オープンできないファイルの名称

bb....bb : オープンできない理由

(S) 処理を中止します。

(O) システム管理者に連絡してください。

【対策】 ファイルおよびディレクトリのパーミッションなどを見直してください。

KFOT70102 - E

cannot write file : aa....aa. (reason=bb....bb) (C)

ファイルの書き込みができません。

aa....aa : 書き込みができないファイルの名称

bb....bb : 書き込みができない理由

(S) 処理を中止します。

(O) システム管理者に連絡してください。

【対策】 ファイルおよびディレクトリのパーミッションなどを見直してください。

KFOT70103 - E

cannot delete file : aa....aa. (reason=bb....bb) (C)

6. メッセージ

ファイルの削除ができません。

aa....aa : 削除できないファイルの名称

bb....bb : 削除できない理由

(S) 処理を中止します。

(O) システム管理者に連絡してください。

【対策】 ファイルおよびディレクトリのパーミッションなどを見直し、削除できなかったファイルを削除してください。

KFOT70300 - E

inside error occurred. error code=aa....aa (C)

内部エラーが発生しました。

aa....aa : エラーコード

(S) 処理を中止します。

(O) システム管理者に連絡してください。

【対策】 トランザクションフレームジェネレータで使用できない IDL 定義をしていないか確認してください。IDL 定義に問題がない場合は、保守員に連絡してください。

KFOT70322 - E

not enough memory. (C)

メモリ不足が発生しました。

(S) 処理を中止します。

(O) システム管理者に連絡してください。

【対策】 ほかのプログラムを終了させるなどして、十分なメモリを確保してから再度実行してください。

KFOT70327 - E

cannot open output file "aa....aa".bb....bb. (C)

出力ファイルをオープンできません。

aa....aa : オープンできないファイルの名称

bb....bb : オープンできない理由

(S) 処理を中止します。

(O) システム管理者に連絡してください。

【対策】 ファイルおよびディレクトリのパーミッションなどを見直してください。

KFOT70328 - E

cannot write output file "aa...aa".bb...bb. (C)

出力ファイルの書き込みができません。

aa...aa : 書き込みできないファイルの名称

bb...bb : 書き込みできない理由

(S) 処理を中止します。

(O) システム管理者に連絡してください。

【対策】ファイルおよびディレクトリのパーミッションなどを見直してください。

KFOT70332 - E

aa...aa failed.bb...bb. (E)

内部コマンドの実行に失敗しました。

aa...aa : 失敗したコマンドの名称

bb...bb : 失敗した理由 (コマンドからのメッセージ)

(S) 処理を中止します。

(O) システム管理者に連絡してください。

【対策】bb...bb を参考に失敗の原因を取り除いてください。

KFOT70400 - E

invalid option for the connector for ORB.option=aa...aa value=bb...bb reason=cc...cc. (L + E)

TSCORB コネクタを開始する tscstartgw コマンドの引数の指定に誤りがあります。

aa...aa : 誤りがあるオプションの名称

cc...cc が NO_OPT_ARG, OVER_LIMIT, または INV_VAL 以外の場合は, 何も表示されないことがあります。

bb...bb : aa...aa で示されるオプションに指定した値

cc...cc が OVER_LIMIT または INV_VAL 以外の場合は, 何も表示されないことがあります。

cc...cc : エラー発生要因

NO_OPTION

指定を省略できないオプションが指定されていません。

NO_OPT_ARG

必要なオプション引数がありません。

OVER_LIMIT

オプション引数の値が指定できる範囲を超えています。

INV_VAL

6. メッセージ

オプション引数の値が不正です。

上記以外

オプションの解析中に内部エラーが発生しました。

(S) 処理を中断します。

(O) tscstartgw コマンドに指定したオプションに誤りがないかどうかを見直してください。

【対策】 指定オプションを修正後、再度コマンドを実行してください。内部エラーの場合は保守員に連絡してください。

KFOT70401 - E

cannot access IDL file. file=aa...aa dir=bb...bb kind=cc...cc. (L + E)

IDL ファイルのアクセスに失敗しました。

aa...aa : アクセスに失敗した IDL ファイル名

bb...bb : IDL ファイルの格納ディレクトリ

- -TSCIDLPath オプションを指定しなかった場合は、"-" が表示されます。
- -TSCIDLPath オプションを指定し、かつ cc...cc が NOT_FOUND の場合は、-TSCIDLPath オプションで指定したすべてのディレクトリが表示されます。
- -TSCIDLPath を指定し、かつ cc...cc が NOT_FOUND 以外の場合は、アクセスに失敗した IDL ファイルのパスが表示されます。

cc...cc : エラー発生要因

NOT_FOUND :

IDL ファイルを検索できませんでした。

PERMISSION :

IDL ファイルに読み込み権限がありません。

INV_SIZE :

IDL ファイルのサイズが不正です。IDL ファイルが空である可能性があります。

DERECTORY :

指定した IDL ファイルはディレクトリです。

上記以外 :

その他のエラーが発生しました。

(S) 処理を中断します。

(O) 次について確認してください。

1. TSCORB コネクタの開始時に、tscstartgw コマンドの -TSCIDLFile オプションおよび -TSCIDLPath オプションに指定した値が正しいか。または、IDL ファイルが正しい場所にあるか。
2. tscstartgw コマンドに指定した -TSCUId オプションのユーザまたは -TSCGid オ

- プシヨンのグループに読み込み権限があるか。
- IDL ファイルが空のファイルでないか。または、tscstartgw コマンドの -TSCIDLFile オプションの指定値と同じ名称のディレクトリがないか。

【対策】それぞれ、次のとおり対策してください。

- cc....cc が NOT_FOUND, INV_SIZE, または DERECTORY の場合は、IDL ファイルの格納ディレクトリを変更するか、tscstartgw コマンドの -TSCIDLFile オプションまたは -TSCIDLPath オプションの指定値を修正して、再度 TSCORB コネクタを開始してください。
- cc....cc が PERMISSION の場合は、IDL ファイルに読み込み権限を設定し、再度 TSCORB コネクタを開始してください。読み込み権限は、tscstartgw コマンドの -TSCUId オプションおよび -TSCGid オプションで指定したユーザおよびグループに設定します。これらのオプションを省略した場合は、tscstartgw コマンドを実行したユーザおよびグループに設定してください。
- cc....cc がその他の場合は、このメッセージの前に出力されているメッセージ (KFOT75500-E) を参照して、障害の原因を取り除いてください。

KFOT70402 - E

the error occurred while analyzing IDL file. file=aa....aa kind=bb....bb inter=cc....cc detail=dd....dd.
(L + E)

IDL ファイルの解析中にエラーが発生しました。

aa....aa : エラーが発生した IDL ファイル名
絶対パスで表示されます。

bb....bb : エラー発生要因

EXEC_ERR :

内部で実行する TPBroker のコマンドが実行できませんでした。TPBroker が正しくインストールされていないか、TPBroker のインストールディレクトリの bin が PATH 環境変数に設定されていない可能性があります。

CMD_ERR :

内部コマンドがエラーとなりました。IDL ファイルの定義内容が不正であるか、インクルードファイルを検索できなかった可能性があります。

ORB_EXCEPT :

IDL ファイルの解析処理中に CORBA 例外が発生しました。

TMP_FILE :

一時ファイルのアクセスに失敗しました。

MEMORY :

メモリ不足が発生しました。

上記以外 :

その他のエラーが発生しました。

cc....cc : 保守情報

dd....dd : エラー詳細情報

(S) 処理を中断します。

(O) IDL ファイル, エラー発生要因, 保守情報, エラー詳細情報を取得し, システム管理者に連絡してください。

【対策】次のとおり対策してください。

- bb....bb が EXEC_ERR の場合は, TPBroker が正しくインストールされているかどうか, TPBroker のインストールディレクトリの bin が PATH 環境変数に設定されているかどうかを確認してください。
TSCORB コネクタの PATH 環境変数は, TSCORB コネクタの開始時に tscstartgw コマンドの -TSCEnviron オプションで PATH が指定されている場合はその指定値です。PATH が指定されていない場合は tscstart コマンド実行時の PATH 環境変数の値です。
- bb....bb が CMD_ERR の場合は, dd....dd の内容に従って IDL ファイルの内容を見直し, 誤りがあれば修正してください。また, IDL ファイルのインクルードファイルがあるディレクトリを, 正しく -I オプションに指定しているかどうかを確認してください。誤りがあった場合は -I オプションを修正するか, インクルードファイルを格納するディレクトリを変更してください。
IDL ファイルの内容, およびインクルードファイルの格納ディレクトリに誤りがない場合は, TPBroker の環境が正しく設定されているかどうかを確認してください。
- bb....bb が上記以外の場合は, IDL ファイル, エラー種別, 保守情報, および詳細情報を取得し, 保守員に連絡してください。

KFOT70403 - E

the IDL definition is invalid. reason=aa....aa rep_id=bb....bb. (L + E)

IDL 定義が不正です。

aa....aa : エラー発生要因

NO_IF_DEF

インタフェース定義がありません。

IF_CONFLICT

インタフェース定義に不整合がありました。

bb....bb : 不整合のあったリポジトリ ID

aa....aa が NO_IF_DEF の場合はハイフン (-) が表示されます。

(S) 処理を中断します。

(O) 使用した IDL ファイルと, bb....bb に表示されたりポジトリ ID を取得して, システム管理者に連絡してください。

【対策】次のとおり対策してください。

- aa....aa が NO_IF_DEF の場合は、TSCORB コネクタの開始時に tscstartgw コマンドの -TSCIDLFile オプションおよび -TSCIDLPath オプションに指定した値に誤りがないかどうかを確認してください。
誤りがない場合は、IDL ファイルの内容にインタフェース定義が含まれているかどうかを確認してください。
- aa....aa が IF_CONFLICT の場合は、bb....bb で示されたりポジトリ ID を持つ定義が IDL ファイルに複数定義されていないかどうかを確認してください。同じリポジトリ ID となるインタフェースの定義は、定義内容をすべて一致させるか、または異なる定義内容のインタフェースは別のリポジトリ ID に変更する形で IDL 定義ファイルを修正し、再度 TSCORB コネクタを開始してください。

KFOT70404 - E

the error occurred while initializing the connector for ORB. kind=aa....aa proc=bb....bb inter=cc....cc detail=dd....dd. (L + E)

TSCORB コネクタの初期化中にエラーが発生しました。

aa....aa : エラー発生要因

ORB_EXCEPT :

CORBA 例外が発生しました。

MEMORY :

メモリ不足が発生しました。

上記以外 :

内部エラーが発生しました。

bb....bb : エラー発生個所

OPTION : オプション解析処理

IDL_ANA : IDL ファイル解析処理

上記以外 : その他の内部処理

cc....cc : 保守情報

dd....dd : エラー詳細情報

(S) 処理を中断します。

(O) エラー発生要因, エラー発生個所, 保守情報, およびエラー詳細情報を取得し, システム管理者に連絡してください。

【対策】次のとおり対策してください。

- aa....aa が MEMORY の場合は、不要なプロセスを終了させるか、またはメモリを増やしてください。
- aa....aa が MEMORY 以外の場合は、エラー発生要因, エラー発生個所, 保守情報, およびエラー詳細情報を取得し, 保守員に連絡してください。

KFOT70405 - W

cannot stop internal process. pid=aa....aa start_time=bb....bb. (L + E)

内部コマンドを終了できません。

aa....aa : 終了できなかったプロセスのプロセス ID

bb....bb : 終了できなかったプロセスのプロセス開始時刻

(S) 処理を続行します。

【対策】 該当するプロセス ID, 開始時刻のプロセスが存在する場合は, kill コマンドなどで終了させてください。ゾンビプロセスで残る場合は, 一度 TSCORB コネクタを終了させて, 再度 TSCORB コネクタを開始してください。

KFOT70406 - W

the IDL definition was ignored. reason=NO_OP_DEF rep_id=aa....aa. (L + E)

オペレーション定義がないため, IDL 定義は無視されました。

aa....aa : 無視された IDL 定義のリポジトリ ID

(S) 処理を続行します。

【対策】 TSCORB コネクタの開始時に tscstartgw コマンドの -TSCIDLFile オプションで指定した IDL ファイルに, オペレーション定義を持たないインタフェース定義がないかどうかを確認してください。必要であれば IDL ファイルを修正し, 再度 TSCORB コネクタを開始してください。

KFOT70500 - I

now starting tscdmd. (S + L + C)

TSC ドメインマネージャ開始中です。

(S)TSC ドメインマネージャ開始処理を続行します。

KFOT70501 - E

cannot start tscdmd because another tscdmd with the same domain name is operating. domain name = aa....aa (S + L + E)

同じ TSC ドメイン名称の TSC ドメインマネージャが動作中のため, 開始できません。

aa....aa : TSC ドメイン名称

(S)TSC ドメインマネージャの開始処理を中断します。

(O) 動作中の TSC ドメインマネージャの終了を待って, 再度開始してください。または, TSC ドメイン名称を変更して, 再度開始してください。

KFOT70502 - I

tscdmd is now online. (S + L + C)

TSC ドメインマネージャ開始が完了しました。

(S)TSC ドメインマネージャの処理を続行します。

KFOT70503 - I

now terminating tscdmd. terminate type=aa....aa (S + L + C)

TSC ドメインマネージャ終了中です。

aa....aa : 終了形態

NORMAL STOP : 正常終了

FORCE STOP : 強制正常終了

(S)TSC ドメインマネージャの処理を続行します。

KFOT70504 - I

tscdmd stop. (S + L + C)

TSC ドメインマネージャが終了しました。

(S)TSC ドメインマネージャの処理を続行します。

KFOT70505 - I

now starting tscd. (S + L + C)

TSC デーモン開始中です。

(S)TSC デーモンの処理を続行します。

KFOT70506 - E

cannot start tscd because another tscd with the same TSCID is operating in the same domain.

TSCID=aa....aa (S + L + E)

同じ TSC 識別子の TSC デーモンが動作中のため、開始できません。

aa....aa : TSC 識別子

(S)TSC デーモンの開始処理を中断します。

(O)動作中の TSC デーモンの終了を待って再度開始してください。または、TSC 識別子を変更して再度開始してください。

KFOT70507 - E

the combination of domain name and TSCSPOOL directory is invalid. (S + L + E)

TSC ドメイン名称と TSCSPOOL の組み合わせが誤っています。

6. メッセージ

(S)TSC ドメインマネージャおよび TSC デーモンの開始処理を中断します。

(O)TSC ドメイン名称と TSCSPOOL の組み合わせを確認してください。

【対策】 TSC ドメイン名称と TSCSPOOL の組み合わせで次に当てはまるものがないか調べてください。

- 指定された TSCSPOOL がすでにほかの TSC ドメイン名称を持つ TSC ドメインマネージャに使用されています。
- 指定された TSC ドメイン名称を持つ TSC ドメインマネージャが別の TSCSPOOL を使用してすでに動作しています。

正しい組み合わせを指定し直して、再度開始してください。

ただし、TSC ドメインマネージャ開始時に環境を調べて、該当する TSCSPOOL ディレクトリが未使用の場合で、新しい組み合わせで動作させたいときは、`tscdminfo` コマンドで TSC ドメイン情報を削除して、再度、TSC ドメインマネージャを開始してください。

KFOT70508 - W

TSCSPOOL directory is not specified. \$TSCDIR/spool suppose to \$TSCSPOOL. (L + C)
TSCSPOOL 環境変数が設定されていません。

(S)\$ TSCDIR/spool ディレクトリを \$ TSCSPOOL と仮定して処理を続行します。

KFOT70509 - E

cannot start aa....aa because of improper operational environment. reason code=bb....bb (S + L + E)

動作環境が整っていないので、デーモンを開始できません。

aa....aa : デーモン名称

bb....bb : 開始できない要因

MEMORY :

メモリが不足しています。

ENV_TSCDIR :

TSCDIR 環境変数が設定されていません。または、そのディレクトリを参照できません。

ENV_TSCSPOOL :

TSCSPOOL が参照できません。

(S)デーモン開始処理を中断します。

(O)デーモン動作環境を修正後、再度開始してください。

【対策】 デーモン動作環境を見直してください。

MEMORY :

OS の実メモリサイズ, スワップメモリサイズが十分あるか調べてください。

ENV_TSCDIR :

TSCDIR 環境変数を設定してください。または, TSCDIR ディレクトリがアクセスできるか調べてください。

ENV_TSCSPOOL :

TSCSPOOL ディレクトリがアクセスできるか調べてください。

KFOT70510 - I

tscd is now online. (S + L + C)

TSC デーモン開始が完了し, オンライン業務を開始できます。

(S)TSC デーモンの処理を続行します。

KFOT70511 - I

now terminating tscd. terminate type=aa....aa (S + L + C)

TSC デーモン終了中です。

aa....aa : 終了形態

NORMAL STOP : 正常終了

FORCE STOP : 強制終了

(S)TSC デーモンの処理を続行します。

KFOT70512 - I

tscd stop. (S + L + C)

TSC デーモンが終了しました。

(S)TSC デーモンの処理を続行します。

KFOT70513 - E

command argument is invalid. option=aa....aa (E)

コマンド引数の指定に誤りがあります。

aa....aa : 誤りがあるオプションの名称

(S) コマンドをエラー終了します。

(O) 次について見直してください。

- 指定したオプションに誤りがいないか。
- 省略できないオプションを省略していないか。
- 指定したオプションがほかのオプションと同時に指定できるか。

6. メッセージ

【対策】 指定オプションを修正し、正しい組み合わせでオプションを指定したあと、再度コマンドを実行してください。

KFOT70514 - I

```
usage:tscstart [-h] [-TSCDomain domain_name] [-TSCID tscid] [-TSCPort port_no]
[-TSCMyHost host_name] [-TSCEntryCount count] [-TSCNice nice] [-TSCRegStart count]
[-TSCRegOption file] [-TSCEnviron file] [-TSCStartTimeOut time]
[-TSCClientConnectCount count] [-TSCServerConnectCount count]
[-TSCLogFileSize file_size] [-TSCLogFileCount file_count]
[-TSCRootAcceptorCount count] [-TSCRootAcceptorRegistCount count]
[-TSCServerCacheSize size] [-TSCDispatchPolicy normalDispatch |priorDispatch]
[-TSCDispatchParallelCount count] [-TSCMaxRequestCount count]
[-TSCQueueDeleteWait time] [-TSCHighPriorRequestCount count]
[-TSCHighPriorBorder count] [-TSCStatsUse Y|N] [-TSCStatsFileCount file_count]
[-TSCStatsFileSize file_size] ( E )
```

tscstart コマンドの使用方法を示します。コマンドの形式が不正な場合、または -h オプションが指定された場合に出力します。

(S) コマンドをエラー終了します。

(O) 指定したオプションに誤りがないか見直してください。

【対策】 指定オプションを修正後、再度コマンドを実行してください。

KFOT70515 - I

```
usage: tscstop [-h] [-TSCForce] [-TSCDebugInfo] [-TSCDomain domain_name] [-TSCID tscid]
[-TSCMyHost host_name] ( E )
```

tscstop コマンドの使用方法を示します。コマンドの形式が不正な場合、または -h オプションが指定された場合に出力します。

(S) コマンドをエラー終了します。

(O) 指定したオプションに誤りがないか見直してください。

【対策】 指定オプションを修正後、再度コマンドを実行してください。

KFOT70516 - I

```
usage:tscdmstart [-h] [-TSCDomain domain_name] [-TSCMaxTSC tscd_cnt] [-TSCPort port_no]
[-TSCMyHost host_name] [-TSCForceStart | -TSCAutoForce] [-TSCSendInterval value]
[-TSCSubnetMask value] [-TSCSendHost host_name] [-TSCSendHostInterval value] ( E )
```

tscdmstart コマンドの使用方法を示します。コマンドの形式が不正な場合、または -h オ

プションが指定された場合に出力します。

(S) コマンドをエラー終了します。

(O) 指定したオプションに誤りがないか見直してください。

【対策】 指定オプションを修正後、再度コマンドを実行してください。

KFOT70517 - I

usage:tscdstop [-h] [-TSCDomain domain_name] [-TSCMyHost host_name] [-TSCForce]

[-TSCDebugInfo] (E)

tscdstop コマンドの使用方法を示します。コマンドの形式が不正な場合、または -h オプションが指定された場合に出力します。

(S) コマンドをエラー終了します。

(O) 指定したオプションに誤りがないか見直してください。

【対策】 指定オプションを修正後、再度コマンドを実行してください。

KFOT70518 - E

the number of processes (ClientRegulator and AP) is over connection count (S + L + E)

接続プロセス数が、コネクション数を超過しました。

(S) 登録処理は中断し、TSC デーモンは処理を続行します。

(O) 接続プロセス数が、-TSCEntryCount オプションで指定した値を超えていないかどうか調べてください。

【対策】 接続コネクション数を超過していないかどうか調べてください。

KFOT70519 - E

cannot start tscprfd. (S + L + E)

tscprfd が開始できません。

(S) TSC デーモンは処理を続行します。

KFOT70520 - E

internal error occurred. err_code=aa....aa-bb....bb exception=cc....cc (S + E + L)

内部処理で例外を受け取りました。

aa....aa : 要因コード 1

bb....bb : 要因コード 2

cc....cc : 例外

(S) デーモンの処理を中断します。

(O) 要因コード 1, 要因コード 2, および例外情報を取得してシステム管理者に連絡して

6. メッセージ

ください。

KFOT70522 - E

the number of tscd processes is over connection count. (S + L + E)

TSC デーモンの登録数がコネクション数を超過しました。

(S) 登録処理は中断し、TSC デーモンは処理を続行します。

(O) 接続プロセス数がコネクション数を超過していないかどうかを調べてください。

KFOT70523 - E

cannot access shared memory. (S + L + E)

共用メモリへのアクセスでエラーが発生しました。

(S) デーモンの処理を中断します。

(O) 次について確認してください。

1. 共用メモリが作成されているか。
2. 共用メモリサイズが取得できる最大サイズを超えていないか。
3. -TSCDomain オプションに指定された TSC ドメイン名称と TSCSPOOL との関係に誤りがないか。

【対策】 それぞれ、次のとおり対策してください。

1. デーモンが開始されていることを確認し、再度、コマンドを実行してください。
2. メモリを増やしてください。または、取得するサイズを変更してください。
3. -TSCDomain オプションに適切な値を指定してください。

KFOT70524 - E

TSCDIR directory is not specified. (S + L + E)

TSCDIR 環境変数が指定されていません。

(S) デーモンの処理を中断します。

(O) 環境変数を確認してください。

KFOT70525 - E

too many file is open in this process. (S + L + E)

プロセス内でオープンできるファイルの数が最大値を超過しました。

(S) デーモンの場合はエラーメッセージを出力して処理を継続します。コマンドの場合は処理を中断します。

(O) プロセス内でオープンできるファイルの最大値を確認してください。

KFOT70526 - E

internal error occurred. err_code=aa....aa-bb....bb (S + L + E)

内部処理エラーが発生しました。

aa....aa : 要因コード 1

bb....bb : 要因コード 2

(S) デーモンの処理を中断します。

(O) 要因コード 1 および要因コード 2 を取得してシステム管理者に連絡してください。

KFOT70527 - E

cannot find tscd with specified TSCID in this domain. TSCID=aa....aa. (S + L + E)

指定された TSC 識別子を持つ TSC デーモンが見つかりません。

aa....aa : TSC 識別子

(S) デーモンの処理を中断します。

(O) 指定された TSC 識別子を持つ TSC デーモンが開始されているかどうかを確認してください。

【対策】 このメッセージは TSC デーモンがある場合も出力されることがあります。オプションの指定が誤っているか TSC デーモンへの通信ができない状態になっていることが考えられます。

このメッセージが出力されたプロセスによって次のとおり対策してください。

1. tscstop コマンドの場合

-TSCDomain オプションおよび -TSCID オプションが正しいか確認してください。それ以外の場合には "tscstop -TSCForce -TSCDebugInfo" を実行して TSC デーモンを強制終了してください。

2. サーバアプリケーション、クライアントアプリケーション、および tscregltd コマンドの場合

-TSCDomain オプションおよび -TSCID オプションが正しいかを確認してください。それ以外の場合には "tscstop -TSCForce -TSCDebugInfo" を実行して TSC デーモンを強制終了し、資料採取後 TSC デーモンを再開始してください。その後、再度アプリケーションプログラムまたは TSC レギュレータを開始してください。

KFOT70528 - E

cannot find tscdmd with specified domain name. domain name = aa....aa. (S + L + E)

指定された TSC ドメイン名称を持つ TSC ドメインマネージャが見つかりません。

aa....aa : TSC ドメイン名称

(S) TSC ドメインマネージャでこのメッセージが出力された場合は、デーモンの処理を続

6. メッセージ

行します。それ以外の場合はコマンドまたはデーモンの処理を中断します。

(O) 指定された TSC ドメイン名称を持つ TSC ドメインマネージャが開始されているかどうかを確認してください。

【対策】 このメッセージは TSC ドメインマネージャがある場合も出力されることがあります。オプションの指定が誤っているか TSC ドメインマネージャへの通信ができない状態になっていることが考えられます。

このメッセージが出力されたプロセスによって次のとおり対策してください。

1. `tscdmstop` コマンドの場合

-TSCDomain オプションが正しいか確認してください。それ以外の場合には "`tscdmstop -TSCForce -TSCDebugInfo`" を実行して TSC ドメインマネージャを強制終了してください。

2. `tscstart` コマンドの場合

-TSCDomain オプションが正しいかを確認してください。それ以外の場合には "`tscdmstop -TSCForce -TSCDebugInfo`" を実行して TSC ドメインマネージャを強制終了し、資料採取後 TSC ドメインマネージャを再開してください。

その後、再度 TSC デーモンを開始してください。

3. TSC ドメインマネージャの場合

他ホストにある TSC ドメインマネージャが開始されているか確認してください。

その後、必要に応じて他ホストで TSC ドメインマネージャを開始または強制終了し、資料採取後に TSC ドメインマネージャを再開してから `tscgetref` コマンドを実行してください。

KFOT70529 - E

the specified TSCPort number may be repeated. (S + L + E)

-TSCPort オプションに指定したポート番号はすでに使用されている可能性があります。

(S) デーモンの処理を中断します。

(O) ポート番号を変更して、再度デーモンを開始してください。

KFOT70530 - E

the initialization of domain manager is failed. error_code1=aa....aa error_code2=bb....bb. (S + L + E)

TSC ドメインマネージャの初期化処理に失敗しました。

aa....aa : 保守情報 1

bb....bb : 保守情報 2

(S) デーモンの処理を中断します。

(O) このメッセージの前に出力されているエラーメッセージを参照して対処してください。-TSCPort オプションで指定したポート番号が、ほかの TSC ドメインマネージャです

でに使用されているなどの可能性があります。エラーメッセージから対処方法がわからない場合は error_code1 と error_code2 の値を取得してシステム管理者に連絡してください。

KFOT70531 - E

cannot start sending UDP. (S + L + E)

TSC ドメインマネージャが UDP の送信開始に失敗しました。

(S) デーモンの処理を中断します。

(O) このメッセージの前に出力されているエラーメッセージを参照して対処してください。

KFOT70532 - E

the initialization of tscd is failed. error_code1=aa....aa error_code2=bb....bb. (S + L + E)

TSC デーモンが初期化処理に失敗しました。

aa....aa : 保守情報 1

bb....bb : 保守情報 2

(S) デーモンの処理を中断します。

(O) このメッセージの前に出力されているエラーメッセージを参照して対処してください。エラーメッセージから対処方法がわからない場合は error_code1 と error_code2 の値を取得してシステム管理者に連絡してください。

KFOT70533 - I

TSCDIR directory=aa....aa

TSC version=bb....bb

TSCSPOOL directory=cc....cc

TSCDomain=dd....dd

TSCPort=ee....ee (S + L + C)

開始中の TSC ドメインマネージャに関する情報を出力します。

aa....aa : TSCDIR ディレクトリ

bb....bb : バージョン

cc....cc : TSCSPOOL ディレクトリ

dd....dd : TSC ドメイン名称

ee....ee : TSCPort 指定値

(S) TSC ドメインマネージャ開始処理を続行します。

KFOT70534 - I

TSCDIR directory=aa....aa

TSC version=bb....bb

TSCSPOOL directory=cc....cc

TSCDomain=dd....dd

TSCID=ee....ee

TSCPort=ff....ff (S + L + C)

開始中の TSC デーモンに関する情報を出力します。

aa....aa : TSCDIR ディレクトリ

bb....bb : バージョン

cc....cc : TSCSPOOL ディレクトリ

dd....dd : TSC ドメイン名称

ee....ee : TSCPort 指定値

(S)TSC デーモン開始処理を続行します。

KFOT70535 - I

tscdmd determined start type. start type=aa....aa. (L + C)

TSC ドメインマネージャの開始形態を決定しました。

aa....aa : 開始形態

NORMAL START : 正常開始

FORCE START : 強制正常開始

AUTO FORCE START : 自動強制開始

(S)TSC ドメインマネージャの処理を続行します。

KFOT70536 - E

too many file is open in this system. (S + L + E)

システム内でオープンできるファイルの数が最大値を超えました。

(S) デーモンの場合はエラーメッセージを出力して処理を続行します。コマンドの場合は処理を中断します。

(O) システム内でオープンできるファイルの最大値を確認してください。

KFOT70537 - E

cannot get system resources. resource_code = aa....aa resource_value = bb....bb. (S + L + E)

システムのリソース取得処理が失敗しました。

aa....aa : リソース名称

RLIMIT_NOFILE : ファイルディスクリプタ

bb....bb : 確保しようとしている値

(S) デーモンの処理を中断します。

(O) システム内のリソースの最大値と確保しようとしているリソースの値を確認してください。

KFOT70538 - E

tscd with specified TSCID is on the other host. (S + L + E)

指定された TSC 識別子を持つ TSC デーモンは別のホストで開始されています。

(S) デーモンの処理を中断します。

(O) 指定された TSC 識別子を持つ TSC デーモンが同じホストで開始されているかどうかを確認してください。

KFOT70539 - I

now starting tscregltd. (S + L + C)

TSC レギュレータ開始中です。

(S) TSC レギュレータ開始処理を続行します。

KFOT70540 - I

usage:tscregltd [-h] [-TSCDomain domain_name] [-TSCID tscid] [-TSCMyHost host_name]

[-TSCPort port_no] [-TSCClientConnectCount count] [-TSCIORConnect] (E)

tscregltd コマンドの使用方法を示します。コマンドの形式が不正な場合、または `-h` オプションが指定された場合に出力します。

(S) コマンドをエラー終了します。

(O) 指定したオプションに誤りがないか見直してください。

【対策】 指定オプションを修正後、再度コマンドを実行してください。

KFOT70541 - I

usage:tscprcls [-h] [-TSCID tscid] [-TSCDomain domain_name] [-TSCMyHost host_name]

[-TSCTscd] (E)

tscprcls コマンドの使用方法を示します。コマンドの形式が不正な場合、または `-h` オプションが指定された場合に出力します。

(S) コマンドをエラー終了します。

(O) 指定したオプションに誤りがないか見直してください。

6. メッセージ

【対策】 指定オプションを修正後、再度コマンドを実行してください。

KFOT70542 - I

tscregltd is now online. (S + L + C)

TSC レギュレータ開始が完了し、オンライン業務を開始できます。

(S)TSC レギュレータの処理を続行します。

KFOT70543 - E

the error occurred while accessing domain manager information file. (S + L + E)

TSCSPOOL 情報ファイルへのアクセスでエラーが起きました。

(S) コマンドおよびデーモンの処理を中断します。

(O) 次について確認してください。

1. TSCSPOOL 情報ファイルに対するロックの取得に失敗した可能性があります。このメッセージを出力したコマンドまたはデーモンと同時に、10 秒間以上実行中であるコマンド、および 10 秒間以上開始処理中であるデーモンが存在しないか確認してください。
2. TSCSPOOL 情報ファイルが存在しない可能性があります。インストール後 TSC ドメインマネージャを一度も起動していないか確認してください。
3. このメッセージの前に出力されているメッセージがないか確認してください。

【対策】 次のとおり対策してください。

1. 実行中のコマンドおよび開始処理中のデーモンが存在しない状態で、再度コマンドまたはデーモンを開始してください。
2. TSC ドメインマネージャを開始してください。このメッセージの前に出力されているメッセージを参照して、障害の原因を取り除いてください。

KFOT70544 - E

input number is invalid. (E)

tscdminfo コマンドの入力番号が不正です。

(S) コマンドの処理を中断します。

(O) 正しい番号を入力してください。

KFOT70545 - I

now terminating tscregltd (S + L + C)

TSC レギュレータが終了中です。

(S)TSC レギュレータの処理を続行します。

KFOT70546 - I

tscregltd stop (S + L + C)

TSC レギュレータが終了しました。

(S) TSC レギュレータの処理を続行します。

KFOT70547 - W

tscdmd is restarted, so -TSCMaxTSC value is ignored. now -TSCMaxTSC value is aa....aa (S + L + C)

TSC ドメインマネージャを再開始します。その場合 -TSCMaxTSC オプションに指定された値は無視され、前回の開始時に使用された -TSCMaxTSC オプションの値が使用されます。

aa....aa : 今回の開始で使用される -TSCMaxTSC オプションの値

(S) TSC ドメインマネージャの処理を続行します。

(O) -TSCMaxTSC オプションに指定された値を有効にする場合は -TSCForceStart オプションを指定して、再度 tscdmstart コマンドを実行してください。

KFOT70548 - E

cannot stop aa....aa. reason code=bb....bb. (S + L + E)

デーモンが終了できません。

aa....aa : デーモン

bb....bb : 終了できない要因

SHARED_MEMORY :

共用メモリが参照できません。

STATUS :

デーモンは開始中または終了中です。

DAEMON_STARTING :

デーモンは開始中です。

DAEMON_TERMINATING :

デーモンは終了中です。

PRC_STARTING :

登録プロセスが開始中です。TSC デーモン終了時にだけ出力されます。

PRC_TERMINATING :

登録プロセスが終了中です。TSC デーモン終了時にだけ出力されます。

MAYBE_DOWN :

終了中のデーモンが強制終了または異常終了した可能性があります。

6. メッセージ

SYSTEM_CALL :

デーモン終了中にコマンドがシステムコールエラーを検知しました。デーモンは正常終了している可能性があります。

(S)デーモン終了処理を中断します。

(O)デーモン動作状況を確認後、デーモンが終了していない場合は再度終了してください。

【対策】デーモン動作状況を見直してください。

SHARED_MEMORY :

TSC デーモンが動作中か、\$TSCSPOOL 配下のファイルが削除されていないか調べてください。

STATUS :

デーモンが開始中または終了中でないか調べてください。

DAEMON_STARTING :

デーモンが開始中でないか調べてください。デーモンの開始処理完了後、再度終了処理を実行してください。

DAEMON_TERMINATING :

デーモンが終了中でないか調べてください。

PRC_STARTING :

登録プロセスが開始中でないか調べてください。登録プロセスの開始処理完了後、再度終了処理を実行してください。

PRC_TERMINATING :

登録プロセスが終了中でないか調べてください。登録プロセスの終了処理完了後、再度終了処理を実行してください。

MAYBE_DOWN :

aa...aa が tscdmd の場合は、TSC ドメインマネージャの次回開始時に強制開始してください。

SYSTEM_CALL :

デーモンが正常終了しているかどうか確認してください。異常終了している場合は、メッセージを参照して障害を取り除いてください。

KFOT70549 - I

there is no tscspool information. (C)

自ホストには TSC ドメインおよび \$TSCSPOOL 情報が登録されていません。

(S)tscdminfo の処理を終了します。

KFOT70550 - E

cannot delete tscspool information, because now tscdmd is running. (E)

指定された TSC ドメイン名称を持つ TSC ドメインマネージャが動作中のため、TSC ドメインおよび TSCSPOOL 情報は削除できません。

(S) tscdminfo の処理を終了します。

(O) TSC ドメインマネージャの動作状況を確認後、再度コマンドを実行してください。

KFOT70551 - E

the error occurred while terminating process. name=aa....aa reason=bb....bb. (L + E)

TSC デーモン登録プロセスの終了中にエラーが発生しました。

aa....aa : 実行形式ファイル名称

該当するプロセスが Java アプリケーションの場合は "-" を出力します。

bb....bb : エラー発生要因

TIMEOUT :

時間内に TSC デーモン登録プロセスの終了が完了しません。

MAYBE_DOWN :

TSC デーモン登録プロセスに対して終了要求が実行できませんでした。登録プロセスが異常終了した可能性があります。

(S) 該当する TSC デーモン登録プロセスの終了処理を中断します。

(O) エラーが発生したプロセスの実行形式ファイル名称および要因を確認してください。

【対策】 それぞれ次のとおり対策してください。

TIMEOUT :

登録プロセスの終了処理に時間が掛かっています。必要に応じて tscstopprc コマンドで強制終了してください。

MAYBE_DOWN :

登録プロセスが異常終了していないかどうか確認してください。異常終了している場合は例外などを参照して障害を取り除いてください。

KFOT70552 - I

usage:tscdminfo [-h] [-d] (E)

tscdminfo コマンドの使用方を示します。コマンドの形式が不正な場合、または -h オプションが指定された場合に出力します。

(S) コマンドをエラー終了します。

(O) 指定したオプションに誤りがないか見直してください。

【対策】 指定オプションを修正後、再度コマンドを実行してください。

KFOT70553 - E

the initialization of tscregltd is failed. error_code1=aa....aa error_code2=bb....bb. (S + L + E)
tscregltd の初期化処理で失敗しました。

aa....aa : 保守情報 1

bb....bb : 保守情報 2

(S) デーモンの処理を中断します。

(O) このメッセージの前に出力されているエラーメッセージを参照して対処してください。エラーメッセージから対処方法がわからない場合は error_code1 と error_code2 の値を取得してシステム管理者に連絡してください。

KFOT70554 - E

cannot stop tscdmd, because tscd is now running. (L + E)

登録済みの tscd が開始中のため、tscdmd を終了できませんでした。

(S) コマンドの処理を中断します。

(O) 登録済みの tscd の動作状況を確認後、tscd を終了してから再度コマンドを実行してください。

KFOT70556 - E

cannot entry this process, because of status. (S + L + E)

登録先の tscd が開始中または終了中であるため登録処理ができませんでした。

(S) デーモンの処理を中断します。

(O) 登録先の tscd が開始中のときはオンラインになってから、終了中のときは再度 tscd を開始してから実行してください。

KFOT70558 - E

cannot get shared memory, because its size is too large. Size = aa....aa. (S + L + E)

取得しようとした共用メモリサイズがシステムによって決められた最大値を超えています。

aa....aa : 取得しようとした共用メモリサイズ

(S) デーモンの処理を中断します。

(O) 共用メモリサイズ算出式によって取得する共用メモリサイズを調整するか、またはシステムの最大値を変更してください。

KFOT70559 - E

the error occurred while starting tscd. process terminated. (S + L + E)

tscd 開始中にエラーが発生しました。

(S) コマンドの処理を中断します。

(O) このメッセージの前に出力されているメッセージに対する対策を実行してください。メッセージが出力されていない場合は tscd が異常終了している可能性があるため \$TSCSPOOL 配下のファイルを取得し、システム管理者に連絡してください。

KFOT70560 - E

the error occurred while starting tscdmd. process terminated. (S + L + E)

tscdmd 開始中にエラーが発生しました。

(S) コマンドの処理を中断します。

(O) このメッセージの前に出力されているメッセージに対する対策を実行してください。メッセージが出力されていない場合は tscdmd が異常終了している可能性があるため \$TSCSPOOL 配下のファイルを取得し、システム管理者に連絡してください。

KFOT70561 - E

memory allocation failed. (S + L + E)

メモリの取得に失敗しました。

(S) デーモンおよびコマンドの処理を中断します。

(O) システム管理者に連絡してください。

【対策】 不要なプロセスを終了させるか、またはメモリを増やしてください。

KFOT70566 - W

an error occurred during the starting of aa....aa. pid=bb....bb (S + L + E)

システムプロセスの自動開始または再開に失敗しました。

aa....aa : プロセス名称

bb....bb : 開始失敗プロセスのプロセス ID

(S) TSC デーモン開始時は TSC デーモンの処理を終了します。システムプロセス再開時は TSC デーモンの処理を続行します。

(O) 指定された TSC レギュレータのコマンドオプションファイルが正しいか確認し、正しく指定して再度 TSC デーモンを開始してください。

KFOT70567 - W

an error occurred during the accessing of the option file for aa....aa.pid=bb....bb (S + L + E)

システムプロセスのコマンドオプションファイルのアクセスでエラーが発生しました。

aa....aa : プロセス名称

6. メッセージ

bb....bb : 開始失敗プロセスのプロセス ID

(S)TSC デーモン開始時は TSC デーモンの処理を終了します。システムプロセス再開始時は TSC デーモンの処理を続行します。

(O)TSC デーモン開始時は指定された TSC レギュレータのコマンドオプションファイルが正しいか確認し、再度 TSC デーモンを開始してください。TSC デーモン開始済みの場合は \$TSCSPOOL 配下のファイルが削除されていないか確認してください。

KFOT70568 - W

the specified option for aa....aa is invalid. pid=bb....bb (S + L + E)

システムプロセスのコマンドオプションファイルの記述内容が不正です。

aa....aa : プロセス名称

bb....bb : 開始失敗プロセスのプロセス ID

(S)TSC デーモン開始時は TSC デーモンの処理を終了します。システムプロセス再開始時は TSC デーモンの処理を続行します。

(O)TSC デーモン開始時の場合は、指定された TSC レギュレータのコマンドオプションファイルが正しいか確認してください。システムプロセス再開始時はシステム管理者に連絡してください。

【対策】TSC デーモン開始時の場合は次について確認して、再度 TSC デーモンを開始してください。

- TSC レギュレータのコマンドオプションファイルの記述内容が正しいか。
コマンドオプションファイル内に使用できる文字は、英数字、空白、ハイフン (-)、ピリオド (.), およびスラント (/) です。
- 指定したコマンドオプションファイル名称が正しいか。

KFOT70569 - I

restart the system process. name=aa....aa. (L + E)

異常終了したシステムプロセスを再開始します。

aa....aa : プロセスの実行形式ファイル名称

(S)システムプロセスを再開始します。

KFOT70570 - E

cannot restart system process. name=aa....aa. (L + E)

システムプロセスの再開始に失敗しました。

aa....aa : プロセスの実行形式ファイル名称

(S)TSC デーモンの処理を続行します。

(O)このメッセージの前に出力されているエラーメッセージなどを参照して障害を取り

除いたあと、必要に応じてシステムプロセスを再開始してください。

エラーメッセージが何も出力されていない場合はシステム管理者に連絡してください。

システムプロセスを直接開始で再開始した場合、システムプロセスが異常終了しても再開始されません。システムプロセスが異常終了した場合に再開始させたいときは TSC デーモンを終了し、再開始してください。

KFOT70571 - I

usage: tscrasget { [-h] | [-p] [-s] [-TSCSpool spool_directory | -TSCDomain domain_name]

dest_directory } (C + E)

tscrasget コマンドの使用法を示します。コマンドの形式が不正な場合、または -h オプションが指定された場合に出力します。

(S) コマンドをエラー終了します。

(O) 指定したオプションに誤りがないか見直してください。

【対策】 指定オプションを修正後、再度コマンドを実行してください。

KFOT70572 - I

TPBroker Object Transaction Monitor: RAS completed collection of aa....aa (C + E)

tscrasget コマンドの情報、メッセージを出力します。

aa....aa : 収集する情報、メッセージなど

KFOT70573 - E

cannot create the process. process_name=aa....aa. (S + L + E)

プロセスの生成に失敗しました。

aa....aa : プロセス名称

tscd : TSC デーモン

tscdmd : ドメインマネージャ

tscregltd : TSC レギュレータ

tscprfd : TSC システムプロセス

(S) 各デーモンの開始処理を終了します。

(O) 子プロセスの生成に失敗しました。次について確認するか、または時間をおいて再開始してください。

1. システムファイルリソースが不足していないか。
2. 子プロセスの生成に失敗していないか。
3. メモリが不足していないか。

KFOT70574 - I

usage:tscgetpid {[-h] | -TSCDomain domain_name [-TSCID tscid] [-TSCMyHost host_name] |
-TSCDomain domain_name -TSCID tscid -TSCAPID application_id

[-TSCMyHost host_name]} (C)

tscgetpid コマンドの使用方法を示します。コマンドの形式が不正な場合、または -h オプションが指定された場合に出力します。

(S) コマンドをエラー終了します。

(O) 指定したオプションに誤りがないか見直してください。

【対策】 指定オプションを修正後、再度コマンドを実行してください。

KFOT70575 - E

the error occurred while getting process ID. (L + E)

プロセス ID 取得時にエラーが発生しました。

(S) コマンドの処理を終了します。

(O) 取得対象の TSC デーモンおよび TSC ドメインマネージャが開始されているか、または \$ TSCPOOL 配下のファイルが削除されていないか確認してください。

KFOT70576 - I

now starting the application. TSCAPID=aa...aa TSCAPName=bb...bb. (L + C)

アプリケーションプログラムまたは TSCORB コネクタの開始処理を開始します。

aa...aa : アプリケーション識別子

bb...bb : 実行形式ファイル名称

開始対象プログラムが TSCORB コネクタの場合は、"tscorbd" を出力します。

(S) プログラムの開始処理を続行します。

KFOT70577 - E

the error occurred while starting the application. TSCAPID=aa...aa TSCAPName=bb...bb

pid=cc...cc reason=dd...dd. (L + E)

アプリケーションプログラムまたは TSCORB コネクタの開始処理中にエラーが発生しました。

aa...aa : アプリケーション識別子

bb...bb : 実行形式ファイル名称

開始対象プログラムが TSCORB コネクタの場合は、"tscorbd" を出力します。

cc...cc : プログラムのプロセス ID

プログラムのプロセス ID が特定できない場合は、"-1" を出力します。

dd....dd : エラー発生要因

SHM_LOCK_ERROR :

共用メモリのアクセスロックに失敗しました。

AP_MAYBE_DOWN :

プロセスの生成に失敗しました。またはプロセスの終了を検出しました。

NOT_ENOUGH_MEMORY :

メモリが不足しています。

ENV_ERROR :

動作環境が整っていません。

PRC_NUM_OVER :

登録プロセス数が上限を超えました。

SAME_APID :

すでに同一のアプリケーション識別子を持つプログラムが登録されています。

TIMEOUT :

時間内にプログラムの開始が完了しません。

TSCD_STARTING :

TSC デーモンが開始処理中です。

TSCD_TERMINATING :

TSC デーモンが終了処理中です。

TSCD_NOT_FOUND :

TSC デーモンが開始していません。

INVALID_SHM :

共用メモリが不正です。

INTERNAL_ERROR :

内部処理エラーが発生しました。

ENV_ANALYZE :

ユーザ環境変数定義の解析でエラーが発生しました。

PRE_PROC :

プロセスの生成の前処理に失敗しました。

(S) コマンドの処理を中断します。ただし、このメッセージが出力された場合でも、プログラムの開始処理が続行される場合があります。

プログラムの開始完了を確認する場合はメッセージログを参照し、KFOT70594-I のメッセージが出力されていることを確認してください。

(O) 要因を確認してください。

【対策】 要因および前に出力されるメッセージの内容を参照して、障害の原因を取り除いてください。ただし、プログラムは TSC デーモンの子プロセスとして開始されます。そ

6. メッセージ

のため、プログラムからの標準エラー出力および標準出力へのメッセージは TSC デーモンの標準エラー出力および標準出力に出力されます。このメッセージの前にメッセージが出力されていない場合は、メッセージログを参照してください。

KFOT70578 - I

the application is now online. TSCAPID=aa....aa TSCAPName=bb....bb pid=cc....cc. (L + C)

アプリケーションプログラムまたは TSCORB コネクタの開始処理が完了しました。

aa....aa : アプリケーション識別子

bb....bb : 実行形式ファイル名称

開始対象プログラムが TSCORB コネクタの場合は、"tscorbd" を出力します。

cc....cc : プログラムのプロセス ID

(S) プログラムの開始処理を完了します。

KFOT70579 - I

usage:tscstartprc [-h] -TSCAPID application_id -TSCAPName application_name
[-TSCDomain domain_name] [-TSCID tscid] [-TSCUid user_id] [-TSCGid group_id]
[-TSCNice nice] [-TSCPort port_no] [-TSCMyHost host_name] [-TSCWatchTime watch_time]
[-TSCWatchMethod watch_time] [-TSCRequestWay way] [-TSCStartTimeOut time]
[-TSCEnviron file] [-TSCRetryReference file] [-TSCRetryWay retry_way]
[-TSCTimeOut time] [-TSCRequestPriority value] [-TSCParallelCount count]
[-TSCRootAcceptor root_acceptor_name] [-TSCAcceptor acceptor_name]
[-TSCClientMessageBufferCount count] [-TSCRebindTimes rebind_times]
[-TSCRebindInterval rebind_interval] (E)

tscstartprc コマンドの使用方法を示します。コマンドの形式が不正な場合、または -h オプションが指定された場合に出力します。

(S) コマンドを終了します。

(O) 指定したオプションに誤りがないか見直してください。

【対策】 指定オプションを修正後、再度コマンドを実行してください。

KFOT70580 - I

now terminating the application. TSCAPID=aa....aa terminate type=bb....bb. (L + C)

アプリケーションプログラムまたは TSCORB コネクタの終了処理を開始します。

aa....aa : アプリケーション識別子

bb....bb : 終了形態

NORMAL STOP : 正常終了

FORCE STOP : 強制終了

(S)プログラムの終了処理を続行します。

KFOT70581 - E

the error occurred while terminating the application.TSCAPID=aa....aa pid=bb....bb
reason=cc....cc. (L + E)

アプリケーションプログラムまたは TSCORB コネクタの終了処理中にエラーが発生しました。

aa....aa : アプリケーション識別子

bb....bb : プログラムのプロセス ID

プログラムのプロセス ID が特定できない場合 "-1" を出力します。

cc....cc : エラー発生要因

SHM_LOCK_ERROR :

共用メモリのアクセスロックに失敗しました。

AP_MAYBE_DOWN :

プログラムの異常終了を検出しました。

NOT_ENOUGH_MEMORY :

メモリが不足しています。

ENV_ERROR :

動作環境が整っていません。

APID_NOT_FOUND :

指定されたアプリケーション識別子を持つプログラムは登録されていません。

TIMEOUT :

時間内にプログラムの終了が完了しませんでした。

TSCD_STARTING :

TSC デーモンが開始処理中です。

TSCD_TERMINATING :

TSC デーモンが終了処理中です。

TSCD_NOT_FOUND :

TSC デーモンが開始していません。

INVALID_SHM :

共用メモリが不正です。

AP_STARTING :

指定されたアプリケーション識別子を持つプログラムが開始処理中です。

AP_TERMINATING :

6. メッセージ

指定されたアプリケーション識別子を持つプログラムが終了処理中です。

INTERNAL_ERROR :

内部処理エラーが発生しました。

(S) コマンドの処理を中断します。ただし、このメッセージが出力された場合でも、プログラムの終了処理が続行される場合があります。

プログラムの終了完了を確認する場合はメッセージログを参照し、KFOT70596-Iのメッセージが出力されていることを確認してください。

(O) 要因を確認してください。

【対策】 要因および前に出力されるメッセージの内容を参照して、障害の原因を取り除いてください。ただし、プログラムは TSC デーモンの子プロセスとして開始されます。そのため、プログラムからの標準エラー出力および標準出力へのメッセージは TSC デーモンの標準エラー出力および標準出力に出力されます。このメッセージの前にメッセージが出力されていない場合はメッセージログを参照してください。

KFOT70582 - I

the application stop. TSCAPID=aa....aa pid=bb....bb. (L + C)

アプリケーションプログラムまたは TSCORB コネクタの終了処理が完了しました。

aa....aa : アプリケーション識別子

bb....bb : プログラムのプロセス ID

(S) プログラムの終了処理を完了します。

KFOT70583 - I

usage:tscstopprc -TSCAPID application_id [-TSCDomain domain_name] [-TSCID tscid]

[-TSCMyHost host_name] [-TSCForce] [-TSCDebugInfo] [-h] (E)

tscstopprc コマンドの使用方法を示します。コマンドの形式が不正な場合、または -h オプションが指定された場合に出力します。

(S) コマンドを終了します。

(O) 指定したオプションに誤りがないか見直してください。

【対策】 指定オプションを修正後、再度コマンドを実行してください。

KFOT70584 - E

cannot start the application process because tscd is now starting. TSCAPID=aa....aa. (L + E)

TSC デーモンが開始処理中のため、アプリケーションプログラムまたは TSCORB コネクタを開始できませんでした。

aa....aa : アプリケーション識別子

(S) プログラムの開始処理を中断します。

(O)TSC デーモンの開始が完了したことを確認してプログラムを再度、開始してください。

KFOT70585 - E

cannot start the application process because tscd is now terminating. TSCAPID=aa....aa. (L + E)

TSC デーモンが終了処理中のため、アプリケーションプログラムまたは TSCORB コネクタを開始できませんでした。

aa....aa : アプリケーション識別子

(S) プログラムの開始処理を中断します。

(O)TSC デーモンの終了完了後、TSC デーモンを再開してプログラムを再度、開始してください。

KFOT70586 - E

the application which has same application id is already exist. TSCAPID=aa....aa. (L + E)

すでに同一のアプリケーション識別子を持つアプリケーションプログラムまたは TSCORB コネクタが登録されているため、プログラムを開始できませんでした。

aa....aa : アプリケーション識別子

(S) プログラムの開始処理を中断します。

(O) 同一アプリケーション識別子を持つプログラムを終了するか、またはアプリケーション識別子を変更してプログラムを再度、開始してください。

KFOT70587 - E

cannot create the application process. TSCAPID=aa....aa.reason=bb....bb. (L + E)

アプリケーションプログラムまたは TSCORB コネクタのプロセス生成に失敗しました。

aa....aa : アプリケーション識別子

アプリケーションでない場合は "-" を出力します。

bb....bb : エラー発生要因

NAME_TOO_LONG :

プログラムが移動しようとしたディレクトリへのパスが長過ぎます。

DIR_NOENTRY :

プログラムが移動しようとしたディレクトリの構成要素がありません。

IS_NOT_DIR :

プログラムが移動しようとしたディレクトリの構成要素がディレクトリではありません。

DIR_ACCESS :

6. メッセージ

プログラムが移動しようとしたディレクトリに対してのパーミッションがありません。

INVALID_ID :

tscstartpre コマンドまたは tscstartgw コマンドに指定した -TSCUId オプションまたは -TSCGid オプションに指定した値が不正です。

OTHERS :

システムリソースの不足、または内部エラーが発生しました。

(S) プロセスの開始処理を中断します。

(O) 要因を確認してください。

【対策】 それぞれ、次のとおり対策してください。

NAME_TOO_LONG :

プログラムが移動する先のディレクトリのパス名称長がシステムで定義するファイルパス名称最大長を超えていないか確認してください。

DIR_NOENTRY, IS_NOT_DIR, DIR_ACCESS :

システム管理者に連絡してください。

INVALID_ID :

tscstartpre コマンドまたは tscstartgw コマンドに指定した -TSCUId オプションまたは -TSCGid オプションに指定した値を確認してください。問題がない場合はシステム管理者に連絡してください。

OTHERS :

メモリが不足していないか、またはプロセス数がシステムによって決められた上限を超えていないか確認してください。問題がない場合はシステム管理者に連絡してください。

KFOT70588 - E

the application process may be down. TSCAPID=aa....aa pid=bb....bb dir_id=cc....cc
end_state=dd....dd. (L)

アプリケーションプログラムまたは TSCORB コネクタの終了を検出しました。アプリケーションプログラムの場合、TSCAdm クラスの endServer メソッドまたは endClient メソッドを発行しないで終了したときにこのメッセージが出力されます。ただし、OTM - Client のクライアントアプリケーション、または -TSCWithSystem オプションに "0" を指定して開始したクライアントアプリケーションの終了は検出されません。

aa....aa : アプリケーション識別子

アプリケーション識別子を持たないプログラムの場合は "-" を出力します。

bb....bb : 終了を検出したプログラムのプロセス ID

cc....cc : カレントディレクトリ ID

クライアントアプリケーションまたは tscstartpre コマンドを使用しないで開始した

サーバアプリケーションの場合は "-" を出力します。

dd....dd : 終了状態

tscstartprc コマンドで開始したサーバアプリケーションまたは TSCORB コネクタの場合 :

TSC デーモンがオンライン中のときはプロセスの終了状態を出力します。TSC デーモンが開始中または終了 (強制終了を含む) 中のときは "*" を出力します。

クライアントアプリケーションまたは tscstartprc コマンドを使用しないで開始したサーバアプリケーションの場合 :

"-" を出力します。

(S)tscstartprc コマンドによるサーバアプリケーションの開始中, または tscstartgw コマンドによる TSCORB コネクタの開始中の場合は, プログラムの開始処理を中断します。TSC デーモンがオンライン中の場合は TSC デーモンの処理を続行します。

(O) 出力されたアプリケーション識別子またはプロセス ID を持つアプリケーションプログラムまたは TSCORB コネクタが異常終了していないかどうかを確認してください。異常終了している場合は, 例外や出力されているエラーメッセージなどを参照して障害を取り除いて再度, 開始してください。

KFOT70589 - E

the error occurred in lock control for shared memory access. (S + L + E)

共用メモリアクセス時のロック取得または解放でエラーが発生しました。

(S) 処理を中止します。

(O) 次について確認してください。

1. メモリが不足していないか。
2. TSCSPOOL ディレクトリ下のファイルを削除したり, アクセス権を未変更でないか。

【対策】 それぞれ, 次のとおり対策してください。

1. 不要なプロセスを終了するか, またはメモリを増やしてください。
2. TSC デーモンおよび TSC ドメインマネージャを強制終了後, 再開始してください。

KFOT70590 - E

cannot stop the application process because tscd is now starting. TSCAPID=aa....aa. (L + E)

TSC デーモンが開始処理中のため, アプリケーションプログラムまたは TSCORB コネクタを終了できませんでした。

aa....aa : アプリケーション識別子

(S) プログラムの終了処理を中断します。

KFOT70591 - E

cannot stop the application process because tscd is now terminating. TSCAPID=aa....aa. (L + E)

TSC デーモンが終了処理中のため、アプリケーションプログラムまたは TSCORB コネクタを終了できませんでした。

aa....aa : アプリケーション識別子

(S) プログラムの終了処理を中断します。

KFOT70592 - E

the application process which has specified application id is not exist. TSCAPID=aa....aa. (L + E)

指定されたアプリケーション識別子を持つアプリケーションプログラムまたは TSCORB コネクタは登録されていません。

aa....aa : アプリケーション識別子

(S) プログラムの終了処理を中断します。

(O) 指定したアプリケーション識別子を持つプログラムが登録されているかどうかを確認してください。

KFOT70593 - I

the entry of process is completed. process_kind=aa....aa TSCAPID=bb....bb pid=cc....cc. (L)

開始プロセスの TSC デーモンへの登録処理が完了しました。

aa....aa : プロセス種別

Srv : サーバアプリケーション

Clt : クライアントアプリケーション

Reg : TSC レギュレータ

Orb : TSCORB コネクタ

Prf : TSC システムプロセス

bb....bb : アプリケーション識別子

開始プロセスがサーバアプリケーションで、かつアプリケーション識別子が指定されていた場合、または TSCORB コネクタの場合に出力します。それ以外の場合は、 "-" を出力します。

cc....cc : 開始プロセスのプロセス ID

(S) プロセスの開始処理を続行します。

KFOT70594 - I

the process is ready for request. process_kind=aa....aa TSCAPID=bb....bb pid=cc....cc. (L)

TSC デーモン登録プロセスが要求待ち状態になりました。

aa....aa : プロセス種別

Srv : サーバアプリケーション

Reg : TSC レギュレータ

Orb : TSCORB コネクタ

bb....bb : アプリケーション識別子

開始プロセスがサーバアプリケーションで、かつアプリケーション識別子が指定されていた場合、または TSCORB コネクタの場合に出力されます。それ以外の場合には、 "-" を出力します。

cc....cc : 開始プロセスのプロセス ID

(S) プロセスの開始処理を続行します。

KFOT70595 - I

the request for terminating application process is received. TSCAPID=aa....aa. pid=bb....bb. (L)

アプリケーションプログラムまたは TSCORB コネクタの終了要求を受信しました。

aa....aa : アプリケーション識別子

bb....bb : プログラムのプロセス ID

(S) プログラムの終了処理を続行します。

KFOT70596 - I

the release of process is completed. process_kind=aa....aa TSCAPID=bb....bb pid=cc....cc. (L)

TSC デーモン登録プロセスの終了処理が完了しました。

aa....aa : プロセス種別

Srv : サーバアプリケーション

Clnt : クライアントアプリケーション

Reg : TSC レギュレータ

Orb : TSCORB コネクタ

Prf : TSC システムプロセス

bb....bb : アプリケーション識別子

終了プロセスがサーバアプリケーションで、かつアプリケーション識別子が指定されていた場合、または TSCORB コネクタの場合に出力されます。それ以外の場合には、 "-" を出力します。

6. メッセージ

cc....cc : 終了処理を完了したプロセスのプロセス ID
(S)TSC デーモンの処理を続行します。

KFOT70597 - E

the timeout occurred. thread-ID=aa....aa watch_time= bb....bb (S + L + E)
時間監視機能によってタイムアウトを検出しました。

aa....aa : タイムアウトしたスレッド ID

bb....bb : 監視時間

(S) 処理を中止します。

(O) システム管理者に連絡してください。

【対策】タイムアウトが発生した要因を調査し、OTM またはアプリケーションプログラムを再開始してください。

KFOT70598 - E

cannot stop the application process because the application which has specified application id is now starting. TSCAPID=aa...aa. (L + E)

指定されたアプリケーション識別子を持つアプリケーションプログラムまたは TSCORB コネクタは、開始処理中のため終了できません。

aa....aa : アプリケーション識別子

(S) プログラムの終了処理を中断します。

(O) プログラムの開始完了後、再度、終了処理を実行してください。または指定したアプリケーション識別子が正しいことを確認してください。

KFOT70599 - E

cannot stop the application process because the application which has specified application id is now terminating. TSCAPID=aa....aa. (L + E)

指定されたアプリケーション識別子を持つアプリケーションプログラムまたは TSCORB コネクタは、終了処理中のため終了できません。

aa....aa : アプリケーション識別子

(S) プログラムの終了処理を中断します。

(O) 指定したアプリケーション識別子が正しいことを確認してください。

KFOT70600 - I

the request for starting application process is received. TSCAPID=aa....aa pid=bb....bb. (L)

TSC デーモンへのアプリケーションプログラムまたは TSCORB コネクタの開始要求が完了しました。

aa....aa : アプリケーション識別子
bb....bb : プログラムのプロセス ID
(S) プログラムの開始処理を続行します。

KFOT70601 - E

execution of process is failed. name=aa....aa reason=bb....bb. (L)

開始プロセスの実行に失敗しました。

aa....aa : 実行形式ファイル名称

bb....bb : エラー発生要因

(S) プロセスの開始処理を中断します。

(O) このメッセージの前に出力されているメッセージ (KFOT75500-E) を参照してください。

【対策】 エラー発生要因は動作環境で提供されているシステムコール `execvp(2)` の `errno` 値を示しています。このメッセージの前に出力されている KFOT75500-E のメッセージ、または各動作環境に応じたエラー発生要因を参照して対策してください。なお、Windows を使用する場合、このメッセージは出力されません。

KFOT70603 - W

shared memory is invalid. initialize shared memory. reason=aa....aa. kind=bb....bb. (L + C)

共有メモリが不正です。新規に共有メモリを作成し処理を続行します。

aa....aa : エラー発生要因

`INVALID_VERSION` : 共有メモリのバージョンが不正です。

`CRITICAL_NG` : 共有メモリ内の管理情報が壊れています。

bb....bb : 共有メモリ種別

`tscd` : TSC デーモン用共有メモリ

`tscdmd` : TSC ドメインマネージャ用共有メモリ

(S) デーモンの開始処理を続行します。

KFOT70604 - E

shared memory is invalid. process terminated. reason=aa....aa. kind=bb....bb. (L + E)

参照しようとした共有メモリが不正です。処理を終了します。

aa....aa : エラー発生要因

`INVALID_VERSION` : 共有メモリのバージョンが不正です。

`CRITICAL_NG` : 共有メモリ内の管理情報が壊れています。

bb....bb : 共用メモリ種別

tscd : TSC デーモン用共用メモリ

tscdmd : TSC ドメインマネージャ用共用メモリ

(S) コマンドまたはデーモンの処理を終了します。

(O) エラー発生要因ごとに、次の事項について確認してください。

INVALID_VERSION :

1. 現在開始されている TSC ドメインマネージャおよび TSC デーモンのバージョンがこのメッセージを出力したプロセスのバージョンと異なっていないか。
2. TSC ドメインマネージャおよび TSC デーモンは開始されていないが、前回どちらかのデーモンが異常終了、または強制終了しているため前回の共用メモリが残っているのではないか。
3. TSC ドメインマネージャの再開始時に前回の共用メモリを参照し、その際にバージョン不一致を検出したのではないか。

CRITICAL_NG :

TSC デーモンまたは TSC ドメインマネージャが、異常終了または強制終了していないか。

【対策】 それぞれ、次のとおり対策してください。

1 または 2 の場合

このメッセージを出力したプロセスと、作成されている共用メモリのバージョンが異なるため、共用メモリが使用できません。共用メモリを使用する場合はこのメッセージを出力したプロセスと同じバージョンの TSC ドメインマネージャおよび TSC デーモンを再開始してください。

3 または CRITICAL_NG の場合

使用しようとしている共用メモリは再利用できません。tscdmstart コマンドに -TSCForceStart オプションまたは -TSCAutoForce オプションを指定して、TSC ドメインマネージャを開始してください。

KFOT70605 - E

this process cannot start by command. TSCAPName aa....aa. (L + E)

このプロセスは、tscstartpre コマンドによる開始はできません。処理を終了します。

aa....aa : 開始されたプロセスの実行形式ファイル名称

(S) プロセスの処理を終了します。

(O) このプロセスは、tscstartpre コマンドによる開始はできません。開始したプロセスが TSC レギュレータの場合は、直接、手動で開始してください。開始したプロセスが TSCORB コネクタの場合は、tscstartgw コマンドで開始してください。

KFOT70606 - I

the creation of process is succeeded. TSCAPID=aa....aa TSCAPName=bb....bb pid=cc....cc dir_id=dd....dd. (L)

プロセス生成に成功しました。

aa....aa : アプリケーション識別子

アプリケーション識別子を持たない場合は "-" を出力します。

bb....bb : 実行形式ファイル名称

cc....cc : 生成したプロセスのプロセス ID

dd....dd : カレントディレクトリ ID

TSC システムプロセス, クライアントアプリケーション, またはサーバアプリケーション名称で開始するサーバアプリケーションの場合は "-" を出力します。

(S)TSC デモン登録プロセスの開始処理を続行します。

KFOT70607 - E

the error occurred while accessing environment definition file. TSCEnviron=aa....aa (L + E)

ユーザ環境変数定義ファイルへのアクセスでエラーが発生しました。

aa....aa : ユーザ環境変数定義ファイル名称

(S) コマンドの処理を終了します。

(O) このメッセージの前出力されているメッセージ (KFOT75500-E) を参照して, 障害の原因を取り除いてください。

KFOT70608 - E

the definition of environment is invalid. TSCEnviron=aa....aa line=bb....bb. (L + E)

ユーザ環境変数定義ファイルの記述内容に誤りがあります。

aa....aa : ユーザ環境変数定義ファイル名称

bb....bb : 解析エラーとなった行番号

継続符号 (¥) によって一つの定義が複数行に記述されている場合は, エラーとなった定義の最初の行の行番号が出力されます。

(S) コマンドの処理を終了します。

(O) ユーザ環境変数定義ファイルの記述内容を, 次について確認してください。

1. イコール (=) の記述が抜けている行がないか。
2. 行の先頭または最後がイコール (=) の行がないか。
3. 先頭が "TSC" または "tsc" で始まる環境変数名称がないか。
4. 空白またはタブだけの行がないか。

【対策】 ユーザ環境変数定義ファイルの記述内容を修正してください。

KFOT70609 - E

the error occurred while accessing environment definition file. TSCAPID=aa....aa
TSCEnviron=bb....bb (L + E)

ユーザ環境変数定義ファイルへのアクセスでエラーが発生しました。

aa....aa : アプリケーション識別子

bb....bb : ユーザ環境変数定義ファイル名称

(S) コマンドの処理を終了します。

(O) このメッセージの前に出力されているメッセージ (KFOT75500-E) を参照して、障害の原因を取り除いてください。

KFOT70610 - E

the definition of environment is invalid. TSCAPID=aa....aa TSCEnviron=bb....bb line=cc....cc. (L + E)

ユーザ環境変数定義ファイルの記述内容に誤りがあります。

aa....aa : アプリケーション識別子

bb....bb : ユーザ環境変数定義ファイル名称

cc....cc : 解析エラーとなった行番号

継続符号 (¥) によって一つの定義が複数行に記述されている場合は、エラーとなった定義の最初の行の行番号が出力されます。

(S) コマンドの処理を終了します。

(O) 次について確認してください。

1. ユーザ環境変数定義ファイルの記述内容について確認してください。
イコール (=) の記述が抜けている行がないか。
行の先頭または最後がイコール (=) の行がないか。
先頭が "TSC" または "tsc" で始まる環境変数名称がないか。
空白またはタブだけの行がないか。
2. ユーザ環境変数定義ファイル名称にディレクトリを指定していないか確認してください。

【対策】 それぞれ、次のとおり対策してください。

1. ユーザ環境変数定義ファイルの記述内容を修正してください。
2. 正しいユーザ環境変数定義ファイルを指定してください。

KFOT70611 - W

load library failed. library=aa....aa reason=bb....bb. (L)

ライブラリのロードに失敗しました。

aa....aa : ロードに失敗したライブラリ名

bb....bb : ロードの失敗要因

unsatisfied : ライブラリのシンボルが解決できません。

security : ライブラリのロードが許可されていません。

init : ライブラリの初期化に失敗しました。

(S) 処理を続行します。

(O) システム管理者に連絡してください。

【対策】 次のとおり対策してください。

- Java アプリケーションを使用し、かつ bb....bb が unsatisfied の場合は、ライブラリ格納ディレクトリを指定する環境変数に誤りがあるか、または不正なバージョンの TPBroker を使用している可能性があります。ライブラリ格納ディレクトリを指定する環境変数の修正後、または正しいバージョンの TPBroker をインストール後、再度実行してください。
- Java アプレットを使用し、かつ bb....bb が security の場合は、OTM の Java アーカイブファイルを使用した可能性があります。Java アプレットは OTM - Client の Java アーカイブファイルを使用してください。
- その他の場合は保守員に連絡してください。

KFOT70612 - E

the fatal error occurred. so process is aborted. reason=aa....aa. (S + L + E)

致命的なエラーが発生しました。実行中のプロセスを終了します。

aa....aa : エラー発生要因

SHM_LOCK_ERROR :

共用メモリのロック制御に失敗しました。

(S) 実行中のプロセスを中断します。

(O) 次について確認してください。

1. メモリが不足していないか。
2. TSCSPOOL ディレクトリ下のファイルを削除したり、アクセス権の変更をしていないか。

【対策】 それぞれ、次のとおり対策してください。

1. 不要なプロセスを終了するか、またはメモリを増やしてください。
2. TSC ドメインマネージャおよび TSC デモンを強制終了後、再開始してください。

KFOT70613 - E

system process may be down. name=aa....aa pid=bb....bb end_status=cc....cc. (L + E)

システムプロセスの異常終了を検出しました。

aa....aa : プロセスの実行形式ファイル名称

6. メッセージ

bb....bb : 異常終了を検出したプロセスのプロセス ID

cc....cc : 終了状態

TSC デーモン開始時に自動開始されたシステムプロセスの場合 :

TSC デーモンがオンライン中のときはプロセスの終了状態を表示します。TSC デーモンが開始中または終了(強制終了を含む)中のときは "*" を表示します。

上記以外のシステムプロセスの場合 :

"-" を表示します。

(S) 異常終了したプロセスが TSC デーモン開始時に自動開始するシステムプロセスであった場合、再開処理を実行します。それ以外の場合は TSC デーモンの処理を続行します。

(O) このメッセージのあとに KFOT70569-I が出力されているか確認してください。KFOT70569-I が出力されている場合、該当するシステムプロセスは再開されています。出力されていない場合はこのメッセージの前に出力されているエラーメッセージなどを参照して障害を取り除いてから必要に応じて再開してください。該当するシステムプロセスが再開されてなく、かつエラーメッセージが何も出力されていない場合はシステム管理者に連絡してください。

KFOT70614 - E

the format of retry reference file is invalid. name=aa....aa line=bb....bb reason=cc....cc. (L + E)

指定した接続先情報ファイルの形式が不正です。処理を終了します。

aa....aa : 接続先情報ファイル名称

bb....bb : 解析エラーとなった行番号

cc....cc : エラー発生要因

INVALID_VERSION :

このバージョンで解析できない形式の接続先情報ファイルを指定しました。

INVALID_FILE_FORMAT :

接続先情報ファイルの形式が不正です。

INVALID_DATA :

接続先情報ファイル中のデータが不正です。

(S) コマンドまたはアプリケーションプログラムの処理を終了します。

(O) 要因を確認してください。

【対策】 それぞれ、次のとおり対策してください。

INVALID_VERSION :

接続先情報ファイルの内容を不当に変更した、または接続情報ファイル作成時に tscgetref コマンドにこのバージョンで解析できない形式を指定した可能性が

あります。正しい形式の接続先情報ファイルを指定してください。

INVALID_FILE_FORMAT :

tscgetref コマンドで作成した以外のファイルを接続先情報ファイルとして指定したか、接続先情報ファイルの内容を不当に変更した可能性があります。
tscgetref コマンドで作成したファイルを指定するか、接続先情報ファイルを再作成してください。また、接続先情報ファイル名称にディレクトリを指定した可能性があります。接続先情報ファイルを指定してください。

INVALID_DATA :

接続先情報ファイルの内容を不当に変更した可能性があります。接続先情報ファイルを再作成してください。

注

このメッセージは、アプリケーションプログラムから出力された場合は TSC ログに、tscgetref コマンドから出力された場合は標準エラー出力に出力されます。

KFOT70615 - E

an error occurred during the access of the retry reference file. name=aa....aa. (L + E)

接続先情報ファイルへのアクセスでエラーが発生しました。

aa....aa : 接続先情報ファイル名称

(S) 処理を終了します。

(O) Java アプリケーションまたは Java アプレットで、接続先情報ファイルとして URL を指定した場合はシステム管理者に連絡してください。それ以外の場合は次に示すとおり対策してください。

【対策】 それぞれ、次のとおり対策してください。

アプリケーションプログラムから出力された場合 :

1. Java アプリケーションプログラムまたは Java アプレット
ローカルファイルまたは URL で示される WWW サーバ上のファイルがあり、かつ読み取り可能であるか確認してください。
また、Java アプレットに対して URL でないローカルファイルまたはアプレットダウンロード元以外のファイルの URL を指定した場合は、Java アプレットが署名によってセキュリティの制限が解除されているか確認してください。
2. C++ アプリケーションプログラム
このメッセージの前に出力されているメッセージ (KFOT75500-E) を参照して、障害の原因を取り除いてください。

tscgetref コマンドから出力された場合 :

このメッセージの前に出力されているメッセージ (KFOT75500-E) を参照して、障害の原因を取り除いてください。

注

6. メッセージ

このメッセージは、アプリケーションプログラムから出力された場合は TSC ログに、tscgetref コマンドから出力された場合は標準エラー出力に出力されます。

KFOT70616 - I

```
tscgetref [-h] | [ { -TSCDomain domain_name[,domain_name...] | -TSCDomainAll } ]  
[ { -TSCID tscid[,tscid...] | -TSCIDAll } ] [-TSCHost host_name[,host_name...]]  
[-TSCMyHost host_name] [-TSCRequestWay 0 | 1] [-TSCConnectType bind | ior]  
[-TSCInFile file[,file...]] [ { -TSCPrint | -TSCOutFile file } ] ( E )
```

tscgetref コマンドの使用方法を示します。コマンドの形式が不正な場合、または -h オプションが指定された場合に出力します。

(S) コマンドをエラー終了します。

(O) 指定したオプションに誤りがないか見直してください。

【対策】 指定オプションを修正後、再度コマンドを実行してください。

KFOT70617 - E

```
cannot get the connecting information from specified domain. domain_name=aa....aa  
ip_address=bb....bb reason=cc....cc. ( L + E )
```

TSC ドメインの接続先情報を取得できませんでした。

aa....aa : TSC ドメイン名称

bb....bb : TSC ドメインマネージャがあるホストの IP アドレス

tscgetref コマンドまたは tscgetior コマンドでこのメッセージを出力する場合、"" が出力されます。

cc....cc : エラー発生要因

COMM_FAILURE :

該当する TSC ドメインマネージャへの接続処理で障害が発生しました。

GET_INFO_ERROR :

該当する TSC ドメインでの接続先情報取得に失敗しました。

INVALID_VERSION :

該当する TSC ドメインマネージャは接続先情報取得機能を持ちません。

TIMEOUT :

時間内に接続先情報取得処理が完了しませんでした。

STATUS :

TSC ドメインマネージャが開始中または終了中です。

INTERNAL_ERROR :

内部処理エラーが発生しました。

(S) ほかに指定された TSC ドメインがある場合は、継続して次の TSC ドメインの接続先情報を取得します。

(O) 要因を確認してください。

なお、tscgetref コマンドまたは tscgetior コマンドでこのメッセージが出力された場合は、コマンドと同じホストの TSC ドメインマネージャの TSC ドメインログを参照してください。TSC ドメインマネージャでこのメッセージが出力された場合は、出力された IP アドレスを持つホスト上の TSC ドメインマネージャで障害が発生している可能性があります。該当する TSC ドメインログを参照してください。

【対策】 必要に応じて、それぞれ次のとおり対策してください。

GET_INFO_ERROR, COMM_FAILURE, TIMEOUT :

接続先情報取得時の障害について各 TSC ドメインマネージャの TSC ドメインログを参照し、障害を取り除いてください。

エラー発生要因が COMM_FAILURE または TIMEOUT の場合、TSC ドメインログに障害情報が出力されていないことがあります。この場合は、TSC ドメインマネージャへの接続で障害が発生した可能性があるため、必要に応じて、該当する TSC ドメインマネージャを強制終了し、資料採取後 TSC ドメインマネージャを再開してください。

INVALID_VERSION :

該当する TSC ドメインマネージャのバージョンを確認してください。

OTM 02-01 以降の TSC ドメインマネージャの TSC ドメイン名称を指定してください。

STATUS :

TSC ドメインマネージャが開始中の場合は、オンラインになるまで待ってください。終了中の場合は必要に応じて TSC ドメインマネージャを再開してください。

KFOT70618 - W

there is no connecting information in specified domain. domain_name= aa....aa
ip_address=bb....bb. (C + L)

該当する TSC ドメインには、指定された条件を満たす接続先情報がありません。

aa....aa : TSC ドメイン名称

bb....bb : TSC ドメインマネージャがあるホストの IP アドレス

(S) コマンドまたはデーモンの処理を続行します。ほかに指定された TSC ドメインがある場合は、継続して次の TSC ドメインの接続先情報を取得します。

(O) tscgetref コマンドまたは tscgetior コマンドで指定された条件が満たされているかどうかを確認してください。例えば、次に示す項目を確認してください。

tscgetref コマンドの場合

6. メッセージ

- 該当する TSC ドメインで TSC デーモンが開始しているか。
- -TSCRequestWay オプションに "1" を指定した場合に、該当する TSC ドメインで TSC レギュレータが開始しているか。
- -TSCRequestWay オプションに "1" を、また -TSCConnectType オプションに "ior" を指定した場合に、該当する TSC ドメインで -TSCIORConnect オプションを指定した TSC レギュレータが開始しているか。

tscgetior コマンドの場合

- 該当する TSC ドメインで TSC デーモンが開始しているか。
- -TSCInterfaceName オプションを指定した場合に、その指定値と一致するインタフェースの TSCORB コネクタが、該当する TSC ドメインで開始しているか。
- -TSCObjectName オプションを指定した場合に、その指定値と一致するオブジェクト名を -TSCObjectName オプションに指定した TSCORB コネクタが、該当する TSC ドメインで開始しているか。
- -TSCIORConnect オプションを指定した TSCORB コネクタが開始しているか。

【対策】 コマンドオプションを見直したり、動作環境を見直して、必要に応じて再度コマンドを実行してください。

KFOT70619 - E

there is no connecting information in the specified domains. (E)

指定された条件を満たす接続先情報を一つも取得できませんでした。

(S) コマンドの処理を終了します。

(O) tscgetref コマンドまたは tscgetior コマンドに指定された条件が動作環境で満たされているか、または接続先情報取得処理で障害が発生していないか確認してください。

【対策】 tscgetref コマンドまたは tscgetior コマンドを実行した動作環境を確認してください。確認項目については、メッセージ KFOT70618-W の対策を参照してください。確認した結果、指定された条件が動作環境で問題ない場合は、接続先情報の取得先である TSC ドメインマネージャで障害が発生していないか確認してください。各 TSC ドメインマネージャで発生した障害の詳細内容については、各 TSC ドメインマネージャの標準エラー出力または TSC ドメインログを参照してください。

KFOT70620 - I

the writing in the retry reference file is completed. (C)

接続先情報ファイルを作成しました。

(S) コマンドの処理を終了します。

KFOT70621 - E

an error occurred during the acquiring of connecting information. reason=aa....aa. (L + E)

TSC ドメインマネージャでの接続先情報の取得処理に失敗しました。

aa....aa : エラー発生要因

INVALID_DATA :

接続先情報の内容が不正です。

ACCESS_ERROR :

接続先情報へのアクセス処理で障害が発生しました。

NOT_ENOUGH_MEMORY :

メモリ不足

(S) デーモンでの接続先情報取得処理を中断します。

(O) システム管理者に連絡してください。

【対策】 必要に応じて、それぞれ次のとおり対策してください。

INVALID_DATA :

該当する TSC ドメインの TSC ドメインマネージャ、TSC デーモン、および TSC レギュレータを終了し、再開始してください。

ACCESS_ERROR :

該当する TSC ドメインマネージャが使用する TSCSPOOL ディレクトリ下のファイルを削除したり、アクセス権の変更をしていないか確認してください。

NOT_ENOUGH_MEMORY :

不要なプロセスを終了させるか、またはメモリを増やしてください。

KFOT70622 - W

an error occurred during the initialization of connecting information. reason=aa....aa. (L + E)

接続先情報の初期化に失敗しました。

aa....aa : エラー発生要因

ACCESS_ERROR :

接続先情報へのアクセス処理で障害が発生しました。

NOT_ENOUGH_MEMORY :

メモリが不足しています。

(S) デーモンの処理を中断します。

(O) システム管理者に連絡してください。

【対策】 それぞれ、次のとおり対策してください。

ACCESS_ERROR :

該当する TSC ドメインマネージャが使用する TSCSPOOL ディレクトリ下のファ

6. メッセージ

イルを削除したり、アクセス権の変更をしていないか確認してください。

NOT_ENOUGH_MEMORY :

不要なプロセスを終了させるか、またはメモリを増やしてください。

KFOT70623 - W

the error occurred while deleting the connecting information.reason=aa....aa (L + E)

接続先情報の削除に失敗しました。

aa....aa : エラー発生要因

ACCESS_ERROR :

接続先情報へのアクセス処理で障害が発生しました。

NOT_ENOUGH_MEMORY :

メモリが不足しています。

(S) デーモンの処理を続行します。

(O) システム管理者に連絡してください。

【対策】 それぞれ、次のとおり対策してください。

ACCESS_ERROR :

該当する TSC ドメインマネージャが使用する TSCSPOOL ディレクトリ下のファイルを削除したり、アクセス権の変更をしていないか確認してください。

NOT_ENOUGH_MEMORY :

不要なプロセスを終了させるか、またはメモリを増やしてください。

KFOT70624 - W

the error occurred while updating the connecting information. type=aa....aa reason=bb....bb. (L + E)

接続先情報の更新に失敗しました。

aa....aa : 更新種別

ADD_INFORMATION :

接続先情報の追加

DELETE_INFORMATION :

接続先情報の削除

bb....bb : エラー発生要因

INVALID_DATA :

接続先情報の内容が不正です。

ACCESS_ERROR :

接続先情報へのアクセス処理で障害が発生しました。

NOT_ENOUGH_MEMORY :

メモリが不足しています。

(S) デーモンの処理を続行します。

(O) システム管理者に連絡してください。

【対策】 必要に応じて、それぞれ次のとおり対策してください。

INVALID_DATA, ACCESS_ERROR :

該当する TSC ドメインマネージャが使用する TSCSPOOL ディレクトリ下のファイルを削除したり、アクセス権の変更をしていないか確認してください。また、TSCSPOOL ディレクトリがあるファイルシステムの空き容量が十分に確認してください。

NOT_ENOUGH_MEMORY :

不要なプロセスを終了させるか、またはメモリを増やしてください。

KFOT70625 - W

the information for IOR string in the retry reference file is invalid. domain_name=aa....aa
tscid=bb..bb. (L + E)

接続先情報ファイルの IOR 文字列に対する、TSC ドメイン名称および TSC 識別子が不正です。

aa....aa : 不正なレコードの TSC ドメイン名称

bb....bb : 不正なレコードの TSC 識別子

(S) 処理を続行します。

(O) 接続先情報ファイルの内容を、不当に変更していないか確認してください。

【対策】 正しい内容の接続先情報ファイルを再作成し、再度実行してください。

KFOT70627 - I

now starting the connector for ORB. (L + C)

TSCORB コネクタを開始中です。

(S) TSCORB コネクタの開始処理を続行します。

KFOT70628 - I

the connector for ORB is now online. (L + C)

TSCORB コネクタの開始が完了しました。オンライン業務を開始できます。

(S) TSCORB コネクタの処理を続行します。

KFOT70629 - I

now terminating the connector for ORB. (L + C)

6. メッセージ

TSCORB コネクタの終了処理を開始します。

(S)TSCORB コネクタの終了処理を続行します。

KFOT70630 - I

the connector for ORB stop. (L + C)

TSCORB コネクタの終了処理が完了しました。

(S)TSCORB コネクタの終了処理を完了します。

KFOT70631 - E

the initialization of the connector for ORB is failed. error_code1=aa....aa error_code2=bb....bb.

(L + E)

TSCORB コネクタの初期化処理に失敗しました。

aa....aa : 保守情報 1

bb....bb : 保守情報 2

(S)TSCORB コネクタの開始処理を中断します。

(O) このメッセージの前に出力されているエラーメッセージを参照して対処してください。エラーメッセージから対処方法がわからない場合は error_code1 と error_code2 の値を取得してシステム管理者に連絡してください。

KFOT70632 - I

TSC version=aa....aa

TSCObjectName=bb....bb

TSCPort=cc....cc. (L + C)

開始中の TSCORB コネクタに関する情報を出力します。

aa....aa : TSCORB コネクタのバージョン

bb....bb : オブジェクト名称

TSCORB コネクタの開始時に、-TSCObjectName オプションの指定を省略した場合は、"- " を出力します。

cc....cc : ポート番号

TSCORB コネクタの開始時に、-TSCPort オプションの指定を省略した場合は、"- " を出力します。

(S)TSCORB コネクタの開始処理を続行します。

KFOT70633 - E

Object Transaction Monitor - Connector for Object Request Broker is not installed normally.
reason=aa...aa. (L + E)

TSCORB コネクタが正しくインストールされていません。

aa...aa : エラー発生要因

INSTALL :

OTM - Connector for ORB がインストールされていません。

INVALID :

OTM - Connector for ORB のインストール情報が不正です。

VERSION :

OTM - Connector for ORB のバージョンが不正です。

(S)TSCORB コネクタのコマンド処理を中断します。

(O) 要因を確認してください。

【対策】 それぞれ次のとおり対策してください。

INSTALL :

OTM - Connector for ORB をインストールしてください。

INVALID :

OTM - Connector for ORB を再インストールしてください。

VERSION :

正しいバージョンの OTM - Connector for ORB を再インストールしてください。

KFOT70634 - E

the timeout occurred. (L + E)

tscstart コマンドの実行時に -TSCStartTimeOut オプションで指定したタイムアウト時間が経過しても、TSC デーモンの開始処理が終了しませんでした。

(S) コマンドの処理を中断します。

(O) コマンド実行時の -TSCStartTimeOut オプションの指定値を見直してください。

【対策】 タイムアウト値の指定を修正後、再度コマンドを実行してください。

KFOT70635 - I

usage:tscstartgw [-h] -TSCAPID application_id [-TSCDomain domain_name] [-TSCID tscid]
[-TSCUid user_id] [-TSCGid group_id] [-TSCMyHost host_name] [-TSCEnviron file]
[-TSCStartTimeOut time] [-TSCTimeOut time] [-TSCRequestPriority value]
[-TSCExceptConvertFile file] -TSCIDLFile file[,file...] [-TSCIDLPath path[:path...]]

6. メッセージ

```
[-I directory[:directory...]] [-TSCObjectName object_name] [-TSCIORConnect]
[-TSCAgent 0|1] [-TSCSendAcceptor acceptor_name] [-TSCClientConnectCount count]
[-TSCPort port_no] ( E )
```

tscstartgw コマンドの使用方法を示します。コマンドの形式が不正な場合、または `-h` オプションが指定された場合に出力します。

(S) コマンドをエラー終了します。

(O) 指定したオプションに誤りがないか見直してください。

【対策】 指定オプションを修正後、再度コマンドを実行してください。

KFOT70636 - I

```
usage:tscstopgw [-h] -TSCAPID application_id [-TSCDomain domain_name]
```

```
[-TSCID tscid] [-TSCMyHost host_name] [-TSCForce ] [-TSCDebugInfo] ( E )
```

tscstopgw コマンドの使用方法を示します。コマンドの形式が不正な場合、または `-h` オプションが指定された場合に出力します。

(S) コマンドをエラー終了します。

(O) 指定したオプションに誤りがないか見直してください。

【対策】 指定オプションを修正後、再度コマンドを実行してください。

KFOT70638 - I

```
tscgetior [-h] | [{"-TSCDomain domain_name[,domain_name...]} | -TSCDomainAll]
```

```
["{-TSCID tscid[,tscid...]} | -TSCIDAll"][-TSCHost host_name[,host_name...]]
```

```
[-TSCMyHost host_name] [-TSCInterfaceName interface_name[,interface_name...]]
```

```
[-TSCObjectName object_name[,object_name...]] [-TSCOutPath directory] ( E )
```

tscgetior コマンドの使用方法を示します。コマンドの形式が不正な場合、または `-h` オプションが指定された場合に出力されます。

(S) コマンドをエラー終了します。

(O) 指定したオプションに誤りがないか見直してください。

【対策】 指定オプションを修正後、再度コマンドを実行してください。

KFOT70639 - E

```
the error occurred while accessing file. name=aa....aa reason=bb....bb. ( E )
```

ファイルへのアクセスでエラーが発生しました。

aa....aa : ファイル名称

bb....bb : エラー発生要因

SYSTEM_CALL :

システムコールエラーが発生しました。

PATH_LENGTH :

ファイル名称が長過ぎます。

(S) 処理を終了します。

(O) 要因を確認してください。

【対策】それぞれ次のとおり対策してください。

SYSTEM_CALL :

このメッセージの前に出力されているメッセージ (KFOT75500-E) を参照して、障害の原因を取り除いてください。

PATH_LENGTH :

ファイル名称が長過ぎます。-TSCOutPath オプションの指定値を見直してください。

KFOT70640 - I

the writing in the IOR string file is completed. (C)

IOR 文字列ファイルの作成に成功しました。

(S) 処理を終了します。

KFOT70641 - W

cannot terminate tsacd, because shared memory has been broken. error_code=aa....aa. (S + L + E)

TSC ドメインマネージャが強制終了または異常終了したため、共有メモリの管理情報の一部が壊れ、前回管理していた TSC デーモンを強制終了できません。

aa....aa : 保守情報

(S) TSC ドメインマネージャの開始処理を続行します。

(O) TSC ドメインマネージャ開始コマンドに指定した TSC ドメイン内に、前回開始した TSC デーモンがないか確認してください。

【対策】 tsacdstart コマンドに指定した TSC ドメイン内に、前回開始した TSC デーモンが見つかった場合、その TSC デーモンを直接終了させてください。

KFOT70642 - W

cannot restore management information. kind=aa....aa. (L + E)

前回の管理情報を引き継げませんでした。

aa....aa : 引き継げなかった情報の種類

CONNECTING_INFORMATION : 接続先情報および IOR 文字列

6. メッセージ

(S)TSC ドメインマネージャの開始処理を続行します。

CONNECTING_INFORMATION :

開始した TSC ドメインマネージャと同一マシンかつ同一 TSC ドメイン内のプロセスのうち、TSC ドメインマネージャ開始後に開始したプロセスの接続先情報および IOR 文字列しか取得できません。

(O) 次について確認してください。

CONNECTING_INFORMATION :

TSCSPOOL 内の情報を不当に削除していないか。

【対策】 必要に応じて、それぞれ次のとおり対策してください。

CONNECTING_INFORMATION :

接続先情報および IOR 文字列を取得したい場合は、そのプロセスを正常停止させたあと、再開始してください。

KFOT70643 - I

tscdmd determined restart mode. mode= aa....aa. (L + C)

TSC ドメインマネージャの自動強制開始時に、前回のシステム状態を引き継いで開始したのか、またはシステム状態を引き継げないため初期状態で開始したのかを示します。

aa....aa : 前回のシステム状態の引き継ぎ状況

REUSE :

管理情報を引き継げたので、前回終了前のシステム状態を復元して開始します。

INITIALIZE :

管理情報を引き継げなかったので、初期状態で開始します。

(S)TSC ドメインマネージャの開始処理を続行します。

6.2.4 KFOT71000 ~ KFOT71999

KFOT71000 - E

memory allocation error. size : aa....aa (L + E + S)

メモリの取得に失敗しました。

aa....aa : 確保領域サイズ

(S) 処理を続行します。

(O) システム管理者に連絡してください。

【対策】 不要なプロセスを終了させるか、またはメモリを増やしてください。

KFOT71001 - E

can't start thread. maintenance code : aa....aa (L)

スレッドを開始できません。

aa....aa : 保守コード

(S) 処理を続行します。

(O) 次の事項について確認してください。

1. -TSCMaxTSC , -TSCClientConnectCount , および -TSCServerConnectCount オプションの指定値が不正でないか。
2. メモリが不足していないか。

【対策】 それぞれ、次のとおり対策してください。

1. -TSCMaxTSC , -TSCClientConnectCount , および -TSCServerConnectCount オプションに適切な値を指定してください。
2. 不要なプロセスを終了させるか、またはメモリを増やしてください。

KFOT71003 - W

option error:aa....aa (L)

オプションの取得に失敗しました。

aa....aa : オプション名称

(S) デフォルト値を設定して処理を続行します。

KFOT71004 - E

invalid option value:aa....aa (L + E)

オプションの指定値が不正です。

aa....aa : オプション名称

(O) オプションの指定値を確認してください。

KFOT71005 - W

failed in initializing communication module. reason code : aa....aa (L + E + S)

通信モジュールの初期化に失敗しました。

aa....aa : 初期化に失敗した要因コード

- 1 : メモリ不足が発生しました。
- 2 : CORBA の例外を受けました。
- 3 : 予期しない例外を受けました。
- 4 : ロックエリアの初期化に失敗しました。
- 5 : ロックの確保に失敗しました。

6. メッセージ

6 : ロックの解放に失敗しました。

7 : 通信データが不正です。

8 : 処理できない通信データを受け取りました。

9 : 予期しないエラーが発生しました。

10 : TPBroker のプロパティが正しく指定されていません。

(S) 処理を続行します。

KFOT71051 - W

communication manager not found. communication manager name : aa....aa host name :bb....bb
maintenance code1 : cc....cc code2 : dd....dd (L + E + S)

通信管理オブジェクトが見つかりません。

aa....aa : 通信管理オブジェクト名称

bb....bb : ホスト名称

通信管理オブジェクトの検索を特定のホストに実行していない場合は , "*****"
を表示します。

cc....cc : 保守コード

dd....dd : 保守コード

(S) 処理を続行しますが , 接続の確立ができません。

(O) 次について確認してください。

1. osagent が異常終了していないか。
2. osagent の負荷が高過ぎないか。
3. OSAGENT_PORT 環境変数 , OSAGENT_ADDR 環境変数 ,
OSAGENT_ADDR_FILE 環境変数および ORB オプションに誤りがないか。
4. TSC デーモンまたは TSC ドメインマネージャが異常終了していないか。
5. 指定した TSCID および TSC ドメイン名称に誤りがないか。

【対策】 それぞれ , 次のとおり対策してください。

1. osagent を再開始してください。
2. 不要なプロセスを終了してください。
3. 正しい OSAGENT_PORT 環境変数 , OSAGENT_ADDR 環境変数 ,
OSAGENT_ADDR_FILE 環境変数および ORB オプションを指定してください。
4. OTM の規定に従い TSC デーモンまたは TSC ドメインマネージャを再開始してく
ださい。
5. 正しい TSCID および TSC ドメイン名称を指定してください。

対策しても , 引き続きこのメッセージが出力される場合は , osfind コマンドを実行
し , その結果を保守員に連絡してください。

KFOT71052 - W

convert an object reference to string/convert string to an object reference error for CORBA object.
maintenance code : aa....aa (L)

CORBA オブジェクトの文字列化または復元に失敗しました。

aa....aa : 保守コード

(S) 処理を続行しますが、通信できません。

(O) 次について確認してください。

1. -TSCMaxTSC, -TSCClientConnectCount, および -TSCServerConnectCount オプションの指定値が不正でないか。
2. メモリが不足していないか。
3. ネットワークインタフェースが異常終了していないか。
4. オブジェクトがあるプロセス (TSC デモンまたは TSC ドメインマネージャ) が異常終了していないか。
5. オブジェクトがあるプロセスが、異なるポート番号で再開始していないか。

【対策】それぞれ、次のとおり対策してください。

1. -TSCMaxTSC, -TSCClientConnectCount, および -TSCServerConnectCount オプションに適切な値を指定してください。
2. 不要なプロセスを終了させるか、またはメモリを増やしてください。
3. ネットワークインタフェースの設定を通信可能状態にしてください。
4. OTM の規定に従い、プロセスを再開始してください。
5. プロセスに -TSCPort オプションを指定してください。

KFOT71053 - E

failed in initializing communication module. maintenance code1 : aa....aa code2 : bb....bb (L + E + S)

通信モジュールの初期化処理に失敗しました。

aa....aa : 保守コード 1

bb....bb : 保守コード 2

(S) 処理を続行します。

(O) 次について確認してください。

1. メモリが不足していないか。
2. -TSCPort オプションの指定値がほかのプロセスで使用されていないか。

【対策】それぞれ、次のとおり対策してください。

1. 不要なプロセスを終了させるか、メモリを増やしてください。
2. -TSCPort オプションの指定値をほかのプロセスで使用していない値に変更してください。

KFOT71100 - W

request of communication manager timeout. communication manager name : aa....aa timeout : bb....bb (L)

通信管理オブジェクトの呼び出しがタイムアウトしました。

aa....aa : 通信管理オブジェクト名称

bb....bb : タイムアウト値

(S) 処理を続行します。

(O) 次について確認してください。

1. 通信管理オブジェクトがあるプロセスの負荷が高過ぎないか。
2. 接続が物理的に切断されていないか。

KFOT71101 - W

can't register communication object : aa....aa,bb....bb (L)

通信オブジェクトを登録できません。

aa....aa : 通信管理オブジェクト名称

bb....bb : 通信オブジェクト名称

(S) 処理を続行しますが、その通信オブジェクトへのメッセージを送信できません。

(O) メモリが不足していないかを確認してください。

【対策】 不要なプロセスを終了させるか、またはメモリを増やしてください。

KFOT71102 - W

can't register communication object : aa....aa (L)

通信オブジェクトを登録できません。

aa....aa : 通信管理オブジェクト名称

(S) 処理を続行しますが、その通信オブジェクトへのメッセージを送信できません。

(O) 次について確認してください。

1. 登録する通信オブジェクトが多過ぎないか。
2. メモリが不足していないか。

【対策】 それぞれ、次のとおり対策してください。

1. 一つの TSC デーモンに接続するサーバアプリケーションのオブジェクト数を減らしてください。
2. 不要なプロセスを終了させるか、またはメモリを増やしてください。

KFOT71103 - W

communication object not registered : aa....aa,bb....bb (L)

通信オブジェクトが登録されていません。

aa....aa : 通信管理オブジェクト名称

bb....bb : 通信オブジェクト名称

(S) 処理を続行しますが、その通信オブジェクトへのメッセージを送信できません。

(O) 通信オブジェクトの登録に失敗していないか (KFOT71101 または KFOT71102) を確認してください。

【対策】 KFOT71101 または KFOT71102 の対策を参照してください。

KFOT71150 - W

cannot open a new connection for a thread. maintenance code1 : aa....aa (L)

接続の確立に失敗しました。

aa....aa : 保守コード

(S) 処理を続行しますが、リクエストを送信できません。

(O) 次について確認してください。

1. -TSCMaxTSC, -TSCClientConnectCount, および -TSCServerConnectCount オプションの指定値が不正でないか。
2. メモリが不足していないか。
3. ネットワークインタフェースが異常終了していないか。

【対策】 それぞれ、次のとおり対策してください。

1. -TSCMaxTSC, -TSCClientConnectCount, および -TSCServerConnectCount オプションに適切な値を指定してください。
2. 不要なプロセスを終了させるか、またはメモリを増やしてください。
3. ネットワークインタフェースの設定を通信可能状態にしてください。

KFOT71151 - W

static connection request timeout. timeout : aa....aa (L)

リクエストがタイムアウトしました。

aa....aa : タイムアウト値

(S) 処理を続行します。

(O) 次について確認してください。

1. 送信先プロセスの負荷が高過ぎないか。
2. リクエストの処理に時間を要していないか。

KFOT71152 - W

static connection request failure. maintenance code : aa....aa (L)

リクエストの処理に失敗しました。

aa....aa : 保守コード

6. メッセージ

(S) 処理を続行しますが、リクエストの処理結果を取得できません。

(O) 次について確認してください。

1. プロセスが異常終了していないか。
2. -TSCMaxTSC, -TSCClientConnectCount, および -TSCServerConnectCount オプションの指定値が不正でないか。

【対策】それぞれ、次のとおり対策してください。

1. OTM の規定に従ってプロセスを再開始してください。
2. -TSCMaxTSC, -TSCClientConnectCount, および -TSCServerConnectCount オプションに適切な値を指定してください。

KFOT71153 - W

cannot narrow an object reference to a sub-type. maintenance code : aa....aa (L)

オブジェクトの変換に失敗しました。

aa....aa : 保守コード

(S) 処理を続行しますが、リクエストを送信できません。

(O) プロセスが異常終了していないかを確認してください。

【対策】OTM の規定に従ってプロセスを再開始してください。

KFOT71200 - W

dynamic connection request failure. maintenance code : aa....aa (L)

リクエストの処理に失敗しました。

aa....aa : 保守コード

(S) 処理を続行しますが、リクエストの処理結果を取得できません。

(O) 次について確認してください。

1. プロセスが異常終了していないか。
2. -TSCMaxTSC, -TSCClientConnectCount, および -TSCServerConnectCount オプションの指定値が不正でないか。

【対策】それぞれ、次のとおり対策してください。

1. OTM の規定に従ってプロセスを再開始してください。
2. -TSCMaxTSC, -TSCClientConnectCount, および -TSCServerConnectCount オプションに適切な値を指定してください。

KFOT71201 - W

cannot open a new connection for inter-tscd. maintenance code1 : aa....aa code2 : bb....bb (L)

TSC デーモン間の接続の確立に失敗しました。

aa....aa : 保守コード 1

bb....bb : 保守コード 2

(S) 処理を続行しますが、メッセージを送信できません。

(O) 次について確認してください。

1. -TSCMaxTSC, -TSCClientConnectCount, および -TSCServerConnectCount オプションの指定値が不正でないか。
2. メモリが不足していないか。
3. ネットワークインタフェースが異常終了していないか。
4. 複数の osagent の相互連携の設定 (ネットワーク間を含む) に誤りはないか。

【対策】 それぞれ、次のとおり対策してください。

1. -TSCMaxTSC, -TSCClientConnectCount, および -TSCServerConnectCount オプションに適切な値を指定してください。
2. 不要なプロセスを終了させるか、またはメモリを増やしてください。
3. ネットワークインタフェースの設定を通信可能状態にしてください。
4. 複数の osagent の相互連携の設定 (ネットワーク間を含む) を正しく設定してください。詳細については、マニュアル「Borland Enterprise Server VisiBroker デベロッパーズガイド」を参照してください。

KFOT71202 - W

inter-tscd receiver not found : aa....aa (L)

TSC デーモン間のメッセージ受信オブジェクトが見つかりません。

aa....aa : メッセージ受信オブジェクト名称

(S) 処理を続行します。

KFOT71203 - W

can't register inter-tscd receiver : aa....aa (L)

TSC デーモン間のメッセージ受信オブジェクトを登録できません。

aa....aa : メッセージ受信オブジェクト名称

(S) 処理を続行します。

(O) 次について確認してください。

1. メモリが不足していないか。
2. 共用メモリが不足していないか。

【対策】 それぞれ、次のとおり対策してください。

1. 不要なプロセスを終了するか、またはメモリを増やしてください。
2. -TSCMaxTSC オプションの指定値を増やしてください。

KFOT71204 - W

can't remove inter-tscd receiver : aa....aa (L)

6. メッセージ

TSC デーモン間のメッセージ受信オブジェクトの登録を消しません。

aa....aa : メッセージ受信オブジェクト名称

(S) 処理を続行します。

KFOT71250 - W

shared memory allocation error. maintenance code1 : aa....aa code2 : bb....bb (L)

共用メモリが不足していないか。

aa....aa : 保守コード 1

bb....bb : 保守コード 2

(S) 処理を続行しますが、TSC デーモン間の通信に関する情報が不足します。

(O) -TSCMaxTSC オプションの指定値を確認してください。

【対策】 -TSCMaxTSC オプションの指定値を増やしてください。

KFOT71251 - E

UDP initialization error. port : aa....aa (L + E)

UDP の初期化処理に失敗しました。

aa....aa : ポート番号

(S) 処理を続行しますが、TSC デーモン間の通信に関する情報を送信できません。

(O) 次について確認してください。

1. 指定したポート番号がすでに使用されていないか。
2. -TSCMaxTSC, -TSCClientConnectCount, および -TSCServerConnectCount オプションの指定値が不正でないか。
3. メモリが不足していないか。

【対策】 それぞれ、次のとおり対策してください。

1. -TSCPort オプションの指定値を変更してください。
2. -TSCMaxTSC, -TSCClientConnectCount, および -TSCServerConnectCount オプションに適切な値を指定してください。
3. 不要なプロセスを終了させるか、またはメモリを増やしてください。

KFOT71252 - W

UDP receive error. maintenance code : aa....aa (L)

UDP の受信に失敗しました。

aa....aa : 保守コード

(S) 処理を続行しますが、TSC デーモン間の通信に関する情報を受信できません。

(O) メモリが不足していないかを確認してください。

【対策】不要なプロセスを終了させるか、またはメモリを増やしてください。

KFOT71253 - W

UDP send Error. maintenance code : aa....aa address : bb....bb port : cc....cc (L)

UDP の送信に失敗しました。

aa....aa : 保守コード

bb....bb : 送信先 IP アドレス (ブロードキャストアドレス)

cc....cc : 送信先ポート番号

(S) 処理を続行しますが、TSC デーモン間の通信に関する情報を送信できません。

(O) 次について確認してください。

1. メモリが不足していないか。
2. ネットワークインタフェースが異常終了していないか。

【対策】それぞれ、次のとおり対策してください。

1. 不要なプロセスを終了させるか、またはメモリを増やしてください。
2. ネットワークインタフェースの設定を通信可能状態にしてください。

KFOT71255 - W

UDP data processing error : method aa....aa (L)

UDP 通信データの処理に失敗しました。

aa....aa : メソッド名

(S) 処理を続行します。

(O) メモリが不足していないかを確認してください。

【対策】不要なプロセスを終了させるか、またはメモリを増やしてください。

KFOT71256 - W

UDP from other domain : aa....aa (L)

ほかの TSC ドメインの UDP データを受信しました。

aa....aa : TSC ドメイン名称

(S) データを破棄して処理を続行します。

(O) ほかの TSC ドメインの TSC ドメインマネージャが同じポート番号で開始されていないかを確認してください。

【対策】TSC ドメインごとに異なるポート番号で TSC ドメインマネージャを開始してください。

KFOT71300 - W

read/write lock error (L)

リードおよびライトロックが失敗しました。

(S) 処理を続行します。

KFOT71352 - I

usage:tscridinfo [-h] [-TSCDomain domain_name] [-TSCHost host_name]

[-TSCMyHost host_name] (C)

tscridinfo コマンドの使用方法を示します。コマンドの形式が不正な場合、または -h オプションが指定された場合に出力します。

(S) コマンドをエラー終了します。

(O) 指定したオプションに誤りがないか見直してください。

【対策】 指定オプションを修正後、再度コマンドを実行してください。

KFOT71353 - E

domain information not found. maintenance code1 : aa....aa code2 : bb....bb code3 : cc....cc

(E)

指定した TSC ドメインマネージャに TSC ドメイン情報が見つからない場合に表示します。

aa....aa : 保守情報 1

bb....bb : 保守情報 2

cc....cc : 保守情報 3

(S) TSC ドメイン情報取得処理を中断します。

(O) 次について確認してください。

1. オプションの指定値に誤りがないか。
2. メモリが不足していないか。
3. 指定した TSC ドメインマネージャに接続している TSC デーモンがあるか。

【対策】 それぞれ、次のとおり対策してください。

1. オプションの指定値を正しく指定してください。
2. 不要なプロセスを終了させるか、またはメモリを増やしてください。
3. 指定した TSC ドメインマネージャに接続する TSC デーモンを開始してください。

KFOT71354 - E

the specified tscdmd is not support. (E)

指定された TSC ドメインマネージャでは、この機能をサポートしていません。

(S) TSC ドメイン情報取得処理を中断します。

(O) 指定した TSC ドメインマネージャはこのコマンドをサポートしていません。

【対策】 OTM 01-02 以降の TSC ドメインマネージャ名称を指定してください。

KFOT71357 - E

communication manager request failure. maintenance code1 : aa....aa code2 : bb....bb (E)

通信管理オブジェクトへのリクエスト処理が失敗しました。

aa....aa : 保守情報 1

bb....bb : 保守情報 2

(S) TSC ドメイン情報取得処理を中断します。

(O) 次について確認してください。

1. TSC ドメインマネージャが異常終了していないか。
2. メモリが不足していないか。
3. ネットワークインタフェースが異常終了していないか。

【対策】 それぞれ、次のとおり対策してください。

1. TSC ドメインマネージャを開始してください。
2. 不要なプロセスを終了させるか、またはメモリを増やしてください。
3. ネットワークインタフェースの設定を通信可能状態にしてください。

6.2.5 KFOT73000 ~ KFOT73999

KFOT73000 - I

prf tracing service started. ID:aa....aa (S + L + C)

prf トレースサービスを開始しました。

aa....aa : TSC 識別子

KFOT73001 - I

prf tracing service stopped. ID:aa....aa (S + L + C)

prf トレースサービスを終了しました。

aa....aa : TSC 識別子

KFOT73002 - W

prf tracing service cannot be started. ID:aa....aa reason code=bb....bb-cc....cc (S + L + E)

prf トレースサービスの開始で障害が発生しました。

aa....aa : TSC 識別子

bb....bb : 理由コード

6. メッセージ

101 : メモリ不足が発生しました。

102 : メモリ不足が発生しました。

103 : ファイルの検定時, または新規作成時に I/O エラーが発生しました。

104 : msgget システムコールでの, 新しいメッセージ待ち行列 ID が割り当てられませんでした。ipes コマンドで使用状況を確認し, 不要なメッセージキューがある場合は, ipcrm コマンドによって削除する必要があります。

106 : TSCSPOOL 環境変数に示すディレクトリが正しく作成されていない可能性があります。また, TSCSPOOL 環境変数を省略した場合, TSCDIR 環境変数に示すディレクトリを見直す必要があります。

112 : 指定されたトレースサービスは, すでに開始されています。または, トレースサービスは終了しているが, トレースを取得していたプロセスが開始中のため, 新たにトレースサービスを開始できません。

114 : prf トレースサービスで使用する共用メモリが破壊されています。

115 : prf トレースサービスで使用する共用メモリのアドレスが取得できません。

401 : prf トレースサービスの内部制御ファイルのオープン時にエラーが発生しました。

402 : prf トレースサービスの内部制御ファイルが破壊されています。

403 : prf トレースサービスの内部制御ファイルでエラーが発生しました。TSCSPOOL 環境変数に示すディレクトリが正しく作成されていない可能性があります。また, TSCSPOOL 環境変数を省略した場合, TSCDIR 環境変数に示すディレクトリを見直す必要があります。

404 : トレースファイル作成ディレクトリの指定に誤りがあります。

502 : prf トレースサービスの制御に必要な環境を作成するときに障害が発生しました。

1601 : システムコールでエラーが発生しました。

cc....cc : システム内部コード

(S) 該当する TSC デーモンの prf トレースサービスを中止します。

(O) 理由コードから原因を調査してください。

【対策】 理由コードを参照して, 障害の原因を取り除いてください。

KFOT73003 - W

prf tracing service cannot be continued. ID:aa....aa reason code=bb....bb-cc....cc (S + L + E)

prf トレースの取得中に障害が発生しました。

aa....aa : TSC 識別子

bb....bb : 理由コード

101 : メモリ不足が発生しました。

103 : トレースファイルで I/O エラーが発生しました。

104 : msgget システムコールでの、新しいメッセージ待ち行列 ID が割り当てられませんでした。ipcs コマンドで使用状況を確認し、不要なメッセージキューがある場合は、ipcrm コマンドによって削除する必要があります。

1601 : システムコールでエラーが発生しました。

cc....cc : システム内部コード

(S)prf トレースサービスを中止します。

(O)理由コードを確認してください。

【対策】理由コードを参照して、障害の原因を取り除いてください。

KFOT73004 - W

failure to initialize for prf tracing service. ID:aa....aa reason code=bb....bb-cc....cc (S + L + E)

プロセスでの prf トレースサービスの初期化処理で障害が発生しました。この初期化処理はプロセスでの prf トレースを取得するための準備をする処理です。

aa....aa : TSC 識別子

TSC 識別子を特定できない場合、**** を表示します。

bb....bb : 理由コード

101 : メモリ不足が発生しました。

115 : TSC デーモンが制御する共用メモリにアクセスできません。

1601 : システムコールでエラーが発生しました。

cc....cc : システム内部コード

(S)障害の発生したプロセスでの、prf トレース取得を中止します。

(O)理由コードを確認してください。

【対策】理由コードを参照して、障害の原因を取り除いてください。

KFOT73100 - I

now starting statistical information service. ID:aa....aa (S + L + C)

稼働統計情報取得サービスを開始中です。

aa....aa : サービス識別子 (TSC 識別子)

(S)処理を続行します。

KFOT73101 - I

statistical information service started. ID:aa....aa (S + L + C)

6. メッセージ

稼働統計情報取得サービスを開始しました。

aa....aa : サービス識別子 (TSC 識別子)

(S) 処理を続行します。

KFOT73102 - I

statistical information service stopped. ID:aa....aa (S + L + C)

稼働統計情報取得サービスを終了しました。

aa....aa : サービス識別子 (TSC 識別子)

(S) 処理を終了します。

KFOT73104 - W

statistical information service cannot be started. ID:aa....aa reason code=bb....bb-cc....cc (S + L + E)

稼働統計情報取得サービスの初期化処理で障害が発生しました。該当する TSC ノードで稼働統計情報を取得できません。

aa....aa : サービス識別子 (TSC 識別子)

bb....bb : 理由コード

- 1 : メモリ不足が発生しました。メモリサイズを見直してください。
- 2 : 稼働統計で使用する共用メモリのメモリ不足が発生しました。共用メモリのサイズを見直してください。
- 3 : 稼働統計で使用する共用メモリにアクセスできません。保守員に連絡してください。
- 4 : msgget システムコールでの、新しいメッセージ待ち行列 ID が割り当てられませんでした。不要なメッセージ待ち行列を削除するなどしてください。
- 5 : TSCSPool 環境変数の指定に誤りがあります。TSCSPool 環境変数に指定した値を見直してください。
- 6 : タイムスレッドの起動に失敗しました。保守員に連絡してください。
- 8 : TSCSPool 環境変数に指定したパスが長過ぎます。パス長が 1023 バイトを超えていないかどうか見直してください。
- 9 : 指定された稼働統計情報取得サービスは、すでに起動されています。
- 10 : 排他制御に失敗しました。保守員に連絡してください。
- 15 : 内部処理でエラーが発生しました。保守員に連絡してください。

cc....cc : 内部コード

(S) 該当する TSC ノードでは稼働統計情報取得サービスを実行しません。

(O) 理由コードから原因を調査してください。

【対策】理由コードを参照して、障害の原因を取り除いてください。

KFOT73105 - W

failure to create I/O thread for statistical information service. ID:aa....aa reason code=bb....bb-cc....cc (S + L + E)

稼働統計情報取得サービスの I/O 専用スレッドの作成に失敗しました。該当する TSC ノードで稼働統計情報の取得を実行できません。

aa....aa : サービス識別子 (TSC 識別子)

bb....bb : 理由コード

- 1 : メモリ不足が発生しました。メモリサイズを見直してください。
- 3 : 稼働統計で使用する共用メモリにアクセスできません。保守員に連絡してください。
- 5 : TSCSPOOL 環境変数の指定に誤りがあります。TSCSPOOL 環境変数に指定した値を見直してください。
- 8 : TSCSPOOL 環境変数に指定したパスが長過ぎます。パス長が 1023 バイトを超えていないかどうか見直してください。
- 13 : システムコールでエラーが発生しました。直前に出力されたメッセージを確認して、保守員に連絡してください。

cc....cc : 内部コード

(S) 該当する TSC ノードでは稼働統計情報取得サービスを実行しません。

(O) 理由コードから原因を調査してください。

【対策】理由コードを参照して、障害の原因を取り除いてください。

KFOT73106 - W

I/O thread for statistical information service is stopped. ID:aa....aa reason code=bb....bb-cc....cc (S + L + E)

稼働統計情報取得サービスの I/O 専用スレッドを終了しました。該当する TSC ノードで稼働統計情報の取得を継続できません。

aa....aa : サービス識別子 (TSC 識別子)

bb....bb : 理由コード

- 1 : メモリ不足が発生しました。メモリサイズを見直してください。
- 4 : msgget システムコールでの、新しいメッセージ待ち行列 ID が割り当てられませんでした。不要なメッセージ待ち行列を削除するなどしてください。
- 11 : 稼働統計情報ファイルのオープンに失敗しました。保守員に連絡してください。

6. メッセージ

12: 稼働統計情報ファイルの I/O エラーが発生しました。保守員に連絡してください。

cc....cc: 内部コード

(S) 該当する TSC ノードでの稼働統計情報取得サービスを中止します。

(O) 理由コードから原因を調査してください。

【対策】 理由コードを参照して、障害の原因を取り除いてください。

KFOT73107 - W

failure to initialize on process for statistical information service. ID:aa....aa reason code=bb....bb-cc....cc (S + L + E)

稼働統計情報取得サービスのプロセス内の初期化処理に失敗しました。該当するプロセスの稼働統計情報は取得されません。

aa....aa: サービス識別子 (TSC 識別子)

bb....bb: 理由コード

1: メモリ不足が発生しました。メモリサイズを見直してください。

3: 稼働統計で使用する共用メモリにアクセスできません。保守員に連絡してください。

5: TSCSPOOL 環境変数の指定に誤りがあります。TSCSPOOL 環境変数に指定した値を見直してください。

8: TSCSPOOL 環境変数に指定したパスが長過ぎます。パス長が 1023 バイトを超えていないかどうか見直してください。

13: システムコールでエラーが発生しました。直前に出力されたメッセージを確認して、保守員に連絡してください。

cc....cc: 内部コード

(S) 該当するプロセスでは稼働統計情報が取得されません。

(O) 理由コードから原因を調査してください。

【対策】 理由コードを参照して、障害の原因を取り除いてください。

KFOT73200 - E

cannot delete module trace file. file name:aa....aa reason code=bb....bb-cc....cc (S + L + E)

OTM で使用するモジュールトレースファイルの削除ができません。

aa....aa: 削除できないファイルのパス名称

bb....bb: 理由コード

101: メモリ不足

204: モジュールトレースファイルの削除に失敗しました。

1601 : システムコールエラー発生

cc....cc : システム内部コード

(S) モジュールトレースファイルの削除をスキップします。

(O) 理由コードを確認してください。

【対策】理由コードを参照して、障害の原因を取り除いてください。

KFOT73201 - E

module trace cannot be started. reason code=aa....aa-bb....bb (S + L + E)

OTM で使用するモジュールトレース取得の開始中に障害が発生しました。

aa....aa : 理由コード

101 : メモリ不足

106 : TSCSPOOL 環境変数に示すディレクトリが正しく作成されていない可能性があります。また、TSCSPOOL 環境変数を省略した場合、TSCDIR 環境変数に示すディレクトリを見直す必要があります。

201 : \$TSCSPOOL/utt ディレクトリを作成できませんでした。

202 : \$TSCSPOOL/utt/umt ディレクトリを作成できませんでした。

502 : モジュールトレースの制御に必要な環境を作成するときに、障害が発生しました。

504 : タイマ処理の初期化処理に失敗しました。

1601 : システムコールエラー発生

bb....bb : システム内部コード

(S) モジュールトレースを取得しません。

理由コード 504 の場合、システムは続行しモジュールトレースを取得しますが、\$TSCSPOOL/utt/umt ディレクトリ下にできた不要ファイルを削除しません。通常、不要ファイルは一定時間（1 時間）ごとに削除されます。

(O) 理由コードを確認してください。

【対策】理由コードを参照して、障害の原因を取り除いてください。

理由コード 504 については、保守員に連絡してください。

KFOT73202 - E

failure to initialize for module trace. reason code=aa....aa-bb....bb (S + L + E)

OTM で使用するモジュールトレース取得の初期化処理で障害が発生しました。この初期化処理はプロセスでのモジュールトレースを取得するための準備をする処理です。

aa....aa : 理由コード

101 : メモリ不足

6. メッセージ

106 : TSCSPOOL 環境変数に示すディレクトリが正しく作成されていない可能性があります。また、TSCSPOOL 環境変数を省略した場合、TSCDIR 環境変数に示すディレクトリを見直す必要があります。

109 : モジュールトレースファイルの作成時にエラーが発生しました。

201 : \$TSCSPOOL/utt ディレクトリを作成できませんでした。

202 : \$TSCSPOOL/utt/umt ディレクトリを作成できませんでした。

203 : umt 内部の排他制御でエラーが発生しました。

503 : モジュールトレースファイルを作成するディレクトリのオープンに失敗しました。このディレクトリは、TSC ドメインマネージャの開始時に作成します。このメッセージが出力された実行環境で、TSC ドメインマネージャが正常に開始しているか確認してください。

505 : コマンドのオプション引数が不正です。

1601 : システムコールエラー発生

bb...bb : システム内部コード

(S) 障害の発生したプロセスでの、モジュールトレースの取得はしません。

(O) 理由コードを確認してください。

【対策】 理由コードを参照して、障害の原因を取り除いてください。

理由コード 109, 201, 202 については、前に出力されるメッセージ KFOT73351-E または KFOT75500-E の `errno` 値から障害を取り除いてください。

理由コード 203 については、前に出力されるメッセージ KFOT75500-E の `errno` 値を保守員に連絡してください。

理由コード 505 については、前に出力されるメッセージ KFOT73350-E のオプション引数を見直してください。

KFOT73203 - W

error occurred while checking trace file number. reason code=aa...aa-bb...bb (S + L + E)

\$TSCSPOOL/utt/umt ディレクトリ下のファイルの削除中に障害が発生しました。

aa...aa : 理由コード

101 : メモリ不足

203 : umt 内部の排他制御でエラーが発生しました。

bb...bb : システム内部コード

(S) \$TSCSPOOL/utt/umt ディレクトリ下のファイルを削除しませんが、システムの処理は続行します。

(O) 理由コードを確認してください。

【対策】 理由コードによる障害の原因を取り除いてください。理由コード 203 について

は、前に出力されるメッセージ KFOT75500-E の errno 値を保守員に連絡してください。

KFOT73300 - E

cannot execute aa....aa command. reason code=bb....bb-cc....cc (E)

コマンドが実行できませんでした。

aa....aa : 実行できなかったコマンド名称

bb....bb : 理由コード

101 : メモリ不足

103 : ファイルで I/O エラーが発生しました。

106 : 該当するコマンドを実行するのに必要なシステム内部ファイルがありません。

このファイルは、システム開始の延長で作成します。

次の内容を見直してください。

- コマンドを実行した環境での TSCSPOOL 環境変数のディレクトリ
- コマンドに指定した TSC 識別子

107 : ファイルのオープンエラーが発生しました。

108 : 指定されたトレースファイルが壊れています。

116 : 指定されたトレースファイルの数が 256 個を超えています。

804 : ファイルが見つかりません。

1601 : システムコールエラーが発生しました。

cc....cc : システム内部コード

(S) コマンドをエラー終了します。

(O) 理由コードを確認してください。

【対策】 理由コードを参照して障害を取り除いたあと、再度コマンドを実行してください。

KFOT73303 - I

usage:tscstsstart [-h] | [[-TSCDomain domain_name] [-TSCID tscid]

[-TSCInterval interval_time] [-TSCCheckInterval interval_time]] (E)

tscstsstart コマンドの使用方法を示します。コマンドの形式が不正な場合、または -h オプションが指定された場合に出力されます。

(S) コマンドをエラー終了します。

(O) 指定したオプションに誤りがないか見直してください。

【対策】 指定オプションを修正後、再度コマンドを実行してください。

KFOT73304 - I

usage:tscstsstop [-h] | [[-TSCDomain domain_name] [-TSCID tscid]] (E)

tscstsstop コマンドの使用方を示します。コマンドの形式が不正な場合、または -h オプションが指定された場合に出力されます。

(S) コマンドをエラー終了します。

(O) 指定したオプションに誤りがないか見直してください。

【対策】指定オプションを修正後、再度コマンドを実行してください。

KFOT73307 - I

usage:tscstsed [-h] | [[-TSCEdit edit-item[,edit-item]]] [-TSCInterval interval]

[-TSCTime [start][,end]] [-TSCRootAcceptor rootacceptorname[,rootacceptorname]]

[-TSCOperation operationname[,operationname]] [-TSCCsv] [-TSCStdout]

ststrace-file [ststrace-file]] (E)

tscstsed コマンドの使用方を示します。コマンドの形式が不正な場合、または -h オプションが指定された場合に出力されます。

(S) コマンドをエラー終了します。

(O) 指定したオプションに誤りがないか見直してください。

【対策】指定オプションを修正後、再度コマンドを実行してください。

KFOT73308 - E

tscstsstart command cannot be executed. reason code=aa....aa-bb....bb (E)

tscstsstart コマンドを実行できませんでした。

aa....aa : 理由コード

1 : メモリ不足が発生しました。メモリサイズを見直してください。

3 : 稼働統計で使用する共用メモリにアクセスできません。保守員に連絡してください。

5 : TSCSPOOL 環境変数の指定に誤りがあります。TSCSPOOL 環境変数に指定した値を見直してください。また、TSCSPOOL 環境変数を設定していない場合は、TSCDIR 環境変数に指定した値を見直してください。

8 : TSCSPOOL 環境変数に指定したパスが長過ぎます。パス長が 1,023 バイトを超えていないかどうか見直してください。

13 : システムコールでエラーが発生しました。直前に出力されたメッセージを確認して、保守員に連絡してください。

15 : 内部処理でエラーが発生しました。保守員に連絡してください。

16 : IP アドレスの取得に失敗しました。保守員に連絡してください。

17：該当する TSC 識別子の TSC デーモンがオンラインではないか、TSC ノードで稼働統計情報取得サービスが開始されていません。TSC デーモンが開始しているかどうか確認してください。

18：プロセス間の排他制御に失敗しました。保守員に連絡してください。

19：tscstsstart コマンドは実行済みです。必要な場合、tscstsstop コマンドを実行後、再度 tscstsstart コマンドを実行してください。

26：稼働統計情報取得サービスが開始されていません。稼働統計情報を取得する場合は、稼働統計情報サービスを開始してください。

bb...bb：内部コード

(S) コマンドの実行を中止します。

(O) 理由コードから原因を調査してください。

【対策】 理由コードを参照して、障害の原因を取り除いてください。

KFOT73309 - E

tscstsstop command cannot be executed. reason code=aa....aa-bb....bb (E)

tscstsstop コマンドが実行できませんでした。

aa....aa：理由コード

1：メモリ不足が発生しました。メモリサイズを見直してください。

3：稼働統計で使用する共用メモリにアクセスできません。保守員に連絡してください。

5：TSCSPOOL 環境変数の指定に誤りがあります。TSCSPOOL 環境変数に指定した値を見直してください。

8：TSCSPOOL 環境変数に指定したパスが長過ぎます。パス長が 1023 バイトを超えていないかどうか見直してください。

10：排他制御に失敗しました。保守員に連絡してください。

13：システムコールでエラーが発生しました。直前に出力されたメッセージを確認して、保守員に連絡してください。

15：内部処理でエラーが発生しました。保守員に連絡してください。

16：IP アドレスの取得に失敗しました。保守員に連絡してください。

17：該当する TSC 識別子の TSC デーモンがオンラインではないか、または TSC ノードで稼働統計情報取得サービスが開始されていません。TSC デーモンが開始しているかどうか確認してください。開始している場合は、稼働統計情報取得サービスが開始されているかどうか確認してください。

18：プロセス間の排他制御に失敗しました。保守員に連絡してください。

22：tscstsstart コマンドがまだ実行されていません。tscstsstart コマンド実行後

6. メッセージ

に、tscstssstop コマンドを実行してください。

bb....bb : 内部コード

(S) コマンドの実行を中止します。

(O) 理由コードから原因を調査してください。

【対策】 理由コードを参照して、障害の原因を取り除いてください。

KFOT73310 - E

tscstsed command cannot be executed. reason code=aa....aa-bb....bb (E)

tscstsed コマンド実行中にエラーが発生しました。

aa....aa : 理由コード

1 : メモリ不足が発生しました。不要なプロセスを終了するか、またはメモリを増やして再度実行してください。

11 : 編集出力を出力するファイルの作成に失敗しました。ファイルが作成できない理由（権限、OS の設定）を調査し、原因を取り除き再度実行してください。

13 : システムコールでエラーが発生しました。直前に出力されたメッセージを確認して、保守員に連絡してください。

14 : tscstsed コマンドの引数に指定した稼働統計情報ファイル数が 256 個を超えました。コマンドの引数に指定する稼働統計情報ファイル数を 256 以下に指定して実行してください。

15 : 内部処理でエラーになりました。保守員に連絡してください。

23 : 稼働統計情報の編集出力中にエラーが発生しました。保守員に連絡してください。

24 : カレントディレクトリに以前の出力結果ファイルが残っています。統計統計情報ファイルを標準入力から入力しているため、ファイルを上書きするかどうか問い合わせできません。そのため処理を中断しました。稼働統計情報ファイルを標準入力から入力する場合は、実行するディレクトリにある稼働統計出力ファイルを削除または退避してから実行してください。

bb....bb : システム内部コード

(S) 障害の発生したプロセスでの、稼働統計情報の編集出力を中止します。

(O) 理由コードを確認してください。

【対策】 理由コードによる障害の原因を取り除いてください。

KFOT73311 - W

no records found to meet the condition. aborts processing. (C)

稼働統計情報編集の対象となるレコードが見つかりませんでした。処理を中断します。指定した時間間隔ではレコードがないか、指定したファイルの内容がありません。

(S) 処理を中止します。

(O) TSCTime オプションに指定した時間を見直すか、または指定した稼働統計情報ファイルを見直してください。

【対策】 指定オプションを修正してください。または正しい稼働統計情報ファイルを指定して、再度コマンドを実行してください。

KFOT73350 - E

command option is invalid. aa...aa (E)

コマンドのオプションに誤りがあります。

aa...aa : オプション名称

(S) 処理を中止します。

(O) 指定したオプションに誤りがないか見直してください。

【対策】 指定オプションを修正後、再度コマンドを実行してください。

KFOT73351 - E

"aa...aa(bb...bb)" failed. errno=cc...cc (S + L + E)

システムコールでエラーが発生しました。

aa...aa : エラーとなったシステムコール

bb...bb : TSC デーモン内部情報 (システムコール呼び出し元のモジュール名称)

cc...cc : システムコールの errno 値

(S) 処理を中止します。

(O) システム管理者に連絡してください。

【対策】 システムコールがエラーになった原因を調査してください。

KFOT73352 - R

aa...aa file exists. Is it overwritten ? [Y:Yes N:No] (C)

tscstsed コマンド実行時のカレントディレクトリに、以前に実行したときに作成された稼働統計出力ファイルがあります。上書きする場合は "Y" を、上書きしたくない場合は "N" を選択してください。小文字でも指定できます。

aa...aa : ファイル名称

(S) "Y" を指定した場合は、処理を続行し稼働統計出力ファイルを上書きします。"N" を指定した場合は、処理を中断します。"Y" または "N" 以外を指定した場合は、再度問い合わせます。

(O) 指定したオプションに誤りがないか見直してください。

【対策】 メッセージに従って "Y" または "N" を指定してください。

KFOT73353 - I

The execute of aa...aa command was suspended. (E)

実行中のコマンドの処理を中断します。

aa...aa : コマンド名称

(S) コマンドの処理を中断します。

(O) このメッセージの前に出力されているメッセージを確認してください。

【対策】 このメッセージの前にメッセージが出力されている場合は、そのメッセージを基に原因を取り除き、再度実行してください。このメッセージの前にメッセージが出力されていない場合は、保守員に連絡してください。

KFOT73354 - E

the combination of domain name and TSCSPOOL directory is invalid. (E)

TSC ドメイン名称と TSCSPOOL 環境変数の組み合わせが誤っています。

(S) コマンドの処理を中断します。

(O) TSC ドメイン名称と TSCSPOOL 環境変数の組み合わせを確認してください。

【対策】 指定した TSCSPOOL 環境変数がすでにほかの TSC ドメイン名称を持つ TSC ドメインマネージャに使用されていないか確認し、再度コマンドを実行してください。

KFOT73355 - I

buffering of statistical information is started. ID:aa...aa start time:bb...bb output interval:cc...cc connect check interval:dd...dd (L)

稼働統計情報の共用メモリへのバッファリングを開始しました。

aa...aa : TSC 識別子

bb...bb : バッファリングを開始した時刻 (yyyy / MM / DD hh : mm 形式)

cc...cc : ファイルに出力する時間間隔 (分単位)

dd...dd : コネクション数のチェックをする時間間隔 (秒単位)

(S) 稼働統計情報の共用メモリへの取得を開始します。

KFOT73356 - I

buffering of statistical information is stopped. ID:aa...aa (L)

稼働統計情報の共用メモリへのバッファリングを終了しました。

aa...aa : TSC 識別子

(S) 稼働統計情報の共用メモリへの取得を終了します。

KFOT73357 - I

tscstsstart command is accepted. option: aa....aa (L)

tscstsstart コマンドを受け付けました。

aa....aa : 指定されたオプション名およびオプション引数を次の形式で表示します。

オプション名[, オプション引数] [; オプション名[, オプション引数] ...]

(S) コマンドの処理を続行します。

KFOT73501 - E

process aborted.(reason=aa....aa)

処理を続行できないためプロセスを中止します。

aa....aa : 処理を続行できない理由

- スレッド間排他の解除に失敗しました。
- プロセス間排他の解除に失敗しました。

(S) 処理を中止します。

注

このメッセージの出力先は、解除に失敗した排他を取得したときのメッセージと同じです。

KFOT73502 - W

cannot load library. (C)

ライブラリ格納ディレクトリを指定する環境変数、または使用している TPBroker のバージョンが不正です。

(O) ライブラリ格納ディレクトリを指定する環境変数を変更後、または正しい TPBroker のバージョンをインストール後、再度実行してください。

(S) 処理を続行します。ただし、このメッセージが出力されたプロセスからのすべてのメッセージは標準出力、または標準エラー出力に出力され、システムログ、TSC ログ、および TSC ドメインログには出力されません。

KFOT73551 - E

cannot get aa....aa. (E)

aa....aa の情報を取得できません。

aa....aa : 取得できなかった情報の名称

(S) 処理を中止します。

(O) システム管理者に連絡してください。

【対策】情報が取得できる環境になっているか確認してください。

KFOT73552 - E

cannot access directory indicated by aa....aa. "bb....bb" (E)

aa....aa で示されるディレクトリにアクセスできません。

aa....aa : ディレクトリを示す情報の名称

bb....bb : ディレクトリ名称

(S) 処理を中止します。

(O) システム管理者に連絡してください。

【対策】 ディレクトリを示す情報の内容に誤りがないか、またはディレクトリのアクセス権などを確認してください。

KFOT73553 - E

cannot open file "aa....aa". (E)

ファイル aa....aa がオープンできません。

aa....aa : オープンできないファイルの名称

(S) 処理を中止します。

(O) システム管理者に連絡してください。

【対策】 ファイルやディレクトリのアクセス権などを確認してください。

KFOT73554 - E

cannot read file "aa....aa". (E)

ファイル aa....aa の読み込みができません。

aa....aa : 読み込みできないファイルの名称

(S) 処理を中止します。

(O) システム管理者に連絡してください。

【対策】 ファイルやディレクトリのアクセス権などを確認してください。

KFOT73555 - E

cannot close file "aa....aa". (E)

ファイル aa....aa のクローズができません。

aa....aa : クローズできないファイルの名称

(S) 処理を中止します。

(O) システム管理者に連絡してください。

【対策】 ファイルやディレクトリのアクセス権などを確認してください。

KFOT73556 - E

cannot access file "aa...aa". (E)

ファイル aa...aa にアクセスできません。

aa...aa : アクセスできないファイルの名称

(S) 処理を中止します。

(O) システム管理者に連絡してください。

【対策】ファイルやディレクトリのアクセス権などを確認してください。

KFOT73557 - E

aa...aa environment variable undefined. (E)

環境変数 aa...aa が設定されていません。

aa...aa : 設定されていない環境変数の名称

(S) 処理を中止します。

(O) システム管理者に連絡してください。

【対策】環境変数を設定してください。

KFOT73558 - E

invalid option. option=aa...aa value=bb...bb reason=cc...cc. (E)

オプションが不正です。

aa...aa : オプション名称

bb...bb : 不正なオプション引数 (cc...cc が INVALID_VALUE のときだけ表示されます)

cc...cc : 理由コード

NO_OPT_ARG :

オプション引数が指定されていません。

INVALID_VALUE :

オプション引数の値が不正です。

上記以外 :

オプション解析処理中に内部エラーが発生しました。

(S) 処理を中止します。

(O) 理由コードを確認してください。

【対策】理由コードを参照して、それぞれ次のとおり対策してください。

NO_OPT_ARG および INVALID_VALUE :

オプションの指定値を見直し、再度コマンドを実行してください。

6. メッセージ

上記以外：

option , value , および reason の内容を取得して保守員に連絡してください。

KFOT73559 - E

the error occurred while initializing log function. reason=aa....aa detail=bb....bb inter=cc....cc. (E)
TSC ログ機能の初期化中にエラーが発生しました。

aa....aa : エラー発生要因

ENVIRON :

環境変数の取得に失敗しました。

FILE_ACC :

ファイルアクセスに失敗しました。

LOCK :

排他制御に失敗しました。

MEMORY :

メモリが不足しています。

OTHER :

その他のエラーが発生しました。

bb....bb : 詳細情報

cc....cc : 保守情報

(S) 処理を中止します。

(O) 要因を確認してください。

【対策】 要因に応じて、それぞれ次のとおり対策したあと、再度コマンドを実行してください。

ENVIRON :

TSCDIR および TSCSPOOL 環境変数が設定されているかどうか見直してください。

FILE_ACC :

TSCSPOOL ディレクトリが存在し、アクセス権が正しく設定されているか、および TSCSPOOL ディレクトリに十分な空き領域があるかどうか見直してください。

LOCK :

メモリが不足している場合は、不要なプロセスを終了させるか、またはメモリを増やしてください。それ以外の場合は、保守員に連絡してください。

MEMORY :

不要なプロセスを終了させるか、またはメモリを増やしてください。

OTHER :

reason , detail , および inter の内容を取得して保守員に連絡してください。

KFOT73560 - W

cannot get log control information, so using default value for -TSCLogFileSize and -TSCLogFileCount. (L + E)

TSC ログ制御情報を取得できませんでした。

(S)TSC ログファイルの最大サイズ (tscstart コマンドの -TSCLogFileSize オプション指定値) および最大数 (tscstart コマンドの -TSCLogFileCount オプション指定値) にデフォルト値を設定して処理を続行します。

KFOT73561 - E

cannot get log control information. reason=aa....aa detail=bb....bb inter=cc....cc. (E)

ログ制御情報を取得できませんでした。

aa....aa : エラー発生要因

FILE_ACC :

ファイルアクセスに失敗しました。

NO_EXIST :

ファイルがありません。

VERSION :

認識できないバージョンのファイルです。

bb....bb : 詳細情報

cc....cc : 保守情報

(S) コマンドの処理を終了します。

(O) 要因を確認してください。

FILE_ACC :

TSCSPOOL ディレクトリがあるか、および TSCSPOOL ディレクトリ下のファイルを不当に編集したり、またはアクセス権を変更したりしていないか確認してください。

NO_EXIST :

インストール後、tscdmstart コマンドまたは tscstart コマンドを一度でも実行しているか、または TSCSPOOL ディレクトリ下のファイルを不当に削除していないか確認してください。

VERSION :

tslogcat コマンドよりも新しいバージョンの tscdmstart コマンドまたは tscstart コマンドを実行していないか確認してください。

【対策】 tslogcat コマンドと同じバージョンの tscdmstart コマンドまたは tscstart コマ

6. メッセージ

ンドを実行してください。コマンドが正常終了した場合、ログ制御情報が更新されて取得できるようになります。

KFOT73590 - I

usage:tsclogcat [-h] | [-d | -t tscid] (C)

tsclogcat コマンドの使用方法を示します。コマンドの形式が不正な場合、または -h オプションが指定された場合に出力します。

(S) コマンドをエラー終了します。

(O) 指定したオプションに誤りがないか見直してください。

【対策】 指定オプションを修正後、再度コマンドを実行してください。

KFOT73591 - I

no aa...aa log message to output. (C)

出力するログメッセージがありません。

aa...aa : ログメッセージの種別

domain : TSC ドメインログ

tsc : TSC ログ

(S) コマンドを終了します。

KFOT73592 - E

the fatal error occurred, so process terminated. reason=aa...aa detail=bb...bb inter=cc...cc. (E)

エラーが発生したため、処理を終了します。

aa...aa : エラー発生要因

MEMORY :

メモリが不足しています。

OTHER :

その他のエラーが発生しました。

bb...bb : 詳細情報

cc...cc : 保守情報

(S) 処理を終了します。

(O) メモリが不足していないか確認してください。

【対策】 要因に応じて、それぞれ次のとおり対策してください。

MEMORY :

不要なプロセスを終了させるか、またはメモリを増やしてください。

OTHER :

reason , detail , および inter の内容を取得して保守員に連絡してください。

6.2.6 KFOT74000 ~ KFOT74999

KFOT74000 - E

command argument is invalid. option:aa....aa (L + E)

コマンド引数の指定に誤りがあります。

aa....aa : 誤りがあるオプションの名称

(S) コマンドをエラー終了します。

(O) 指定したオプションに誤りがないか見直してください。

【対策】 指定オプションを修正後、再度コマンドを実行してください。

KFOT74001 - E

this processing exceeded the maximum which was specified by the TSCClientConnectCount option. the effective option value at present is aa....aa. (L + E)

-TSCClientConnectCount オプションに指定された最大値を超えました。

aa....aa : 現在有効なオプション値

(S) getTSCClient メソッドの処理を中断します。

(O) -TSCClientConnectCount オプションの指定値を確認してください。

【対策】 動作中のクライアントアプリケーションの終了を待って、再度実行してください。必要であれば、-TSCClientConnectCount オプションの指定値を増やしたあとに、再度実行してください。

KFOT74002 - E

memory allocation error. maintenance code:aa....aa (L + E)

メモリの取得に失敗しました。

aa....aa : 保守コード

(S) 処理を中止します。

(O) システム管理者に連絡してください。

【対策】 不要なプロセスを終了するか、またはメモリを増やしてください。

KFOT74003 - E

shared memory allocation error. maintenance code:aa....aa (L + E)

共用メモリの取得に失敗しました。

aa....aa : 保守コード

6. メッセージ

(S) 処理を中止します。

(O) システム管理者に連絡してください。

【対策】 オプション引数を修正後、再度コマンドを実行してください。

KFOT74004 - E

cannot assign a tscregltd. reason code:aa....aa (L + E)

クライアントアプリケーションに TSC レギュレータを割り当てることができません。

aa....aa : 理由コード

1 : TSC デーモンに TSC レギュレータが接続されていません。

2 : TSC デーモンに接続している TSC レギュレータにコネクションの空きがありません。

(S) 処理を続行します。

(O) 理由コードを確認してください。

【対策】 クライアントアプリケーション数、TSC レギュレータ数などを調整してください。

KFOT74005 - E

unmatched version occurred by regulator. (S + L)

TSC レギュレータの処理でバージョン不一致が発生しました。

(S) 処理を中断します。

(O) 次に示すどれかが発生していますので、TSC レギュレータの機能を利用するアプリケーションプログラムまたは TSC レギュレータのバージョンと TSC デーモンのバージョンを確認してください。

- バージョン 01-00 のアプリケーションプログラムがバージョン 01-01 以降の TSC レギュレータに接続しようとした。
- バージョン 01-01 以降のアプリケーションプログラムがバージョン 01-00 の TSC レギュレータに接続しようとした。
- バージョン 01-00 の TSC レギュレータがバージョン 01-01 以降の TSC デーモンに接続しようとした。
- バージョン 01-01 以降の TSC レギュレータがバージョン 01-00 の TSC デーモンに接続しようとした。

【対策】 アプリケーションプログラムまたは TSC レギュレータのバージョンと TSC デーモンのバージョンが一致するように対策したあと、アプリケーションプログラムまたは TSC レギュレータを再開始してください。

KFOT74550 - E

scheduler initialization error. reason code=aa....aa (L + S + E)

初期化処理に失敗しました。

aa....aa : 理由コード

- 1 : オプション引数解析エラーが発生しました。
- 2 : 共用メモリ不足が発生しました。
- 3 : メモリ不足が発生しました。
- 4 : 通信環境初期化処理エラーが発生しました。

(S)TSC デーモンの開始処理をエラー終了します。

(O)理由コードを確認してください。

【対策】理由コードを参照して障害を取り除いたあと、再度コマンドを実行してください。

KFOT74551 - E

command option is invalid. option:aa....aa (L + E)

コマンドオプション引数に誤りがあります。

aa....aa : オプション名称

(S)処理を中止します。

(O)指定したオプションに誤りがないか確認してください。

【対策】指定オプションを修正後、再度コマンドを実行してください。

KFOT74552 - E

memory allocation error. maintenance code:aa....aa (L + S + E)

メモリ不足が発生しました。

aa....aa : 保守コード

(S)処理を中止します。

(O)システム管理者に連絡してください。

【対策】不要なプロセスを終了させるか、またはメモリを増やしてください。

KFOT74553 - E

shared memory allocation error. maintenance code:aa....aa (L + S + E)

共用メモリの確保に失敗しました。

aa....aa : 保守コード

(S)処理を中止します。

6. メッセージ

(O) システム管理者に連絡してください。

【対策】 オプション引数を修正後、再度コマンドを実行してください。

KFOT74554 - E

cache memory allocation error. allocation size=aa....aa maintenance code:bb....bb (L + S)

サーバキャッシュメモリの確保に失敗しました。

aa....aa : 確保しようとしたサイズ

bb....bb : 保守コード

(S) 処理を中止します。

(O) システム管理者に連絡してください。

【対策】 -TSCServerCacheSize オプション引数を修正後、再度コマンドを実行してください。

KFOT74555 - W

this processing exceeded the maximum request registration. TSCRootAcceptor:aa....aa reason code:bb....bb (L + S)

TSC ルートアクセプタの最大リクエスト登録数を超えたため、リクエストを登録できません。

このメッセージは TSC ルートアクセプタごとに出力されます。このメッセージが出力されたあと、60 秒間は同一 TSC ルートアクセプタに対するメッセージは出力されません。

aa....aa : TSC ルートアクセプタ名称

bb....bb : 理由コード

Max_Request :

-TSCMaxRequestCount オプションに指定した値を超えたため、リクエストを登録できません。

HighPrior_Request :

-TSCMaxRequestCount オプション指定値から、
-TSCHighPriorRequestCount オプション指定値を引いた値を超えたため、プライオリティの低いリクエストを登録できません。

(S) 処理を続行します。

(O) tsclsra コマンドを使用して登録されているリクエスト数を確認してください。

【対策】 サーバアプリケーションに登録したサービス処理が遅延している場合は、その要因を取り除いてください。サーバアプリケーションの処理能力が不足している場合は、同一サーバアプリケーションのプロセス数を増やすか、または -TSCMaxRequestCount オプション指定値もしくは -TSCHighPriorRequestCount オプション指定値を見直して、TSC デーモンを再開始してください。

KFOT74556 - E

cannot return the reply of the request. client ip:aa....aa client pid:bb....bb communication no:cc....cc
(L + S)

リクエストの応答を返せません。

aa....aa : 破棄されたリクエストのクライアントの IP アドレス

bb....bb : 破棄されたリクエストのクライアントのプロセス ID

cc....cc : 保守情報

(S) 処理を続行します。

【対策】 レギュレータのプロセス数を増やすか、または TSC デーモンに直結するクライアントアプリケーションのプロセスのリクエスト送信スレッド数を減らして、直結するクライアントアプリケーションのプロセス数を増やしてください。

KFOT74600 - I

aa....aa is placed in shutdown state. (L)

TSC ルートアクセプタのスケジュールを閉塞しました。

aa....aa : 閉塞した TSC ルートアクセプタ登録名称

KFOT74601 - I

aa....aa restarted. (L)

TSC ルートアクセプタのスケジュールを閉塞解除しました。

aa....aa : 閉塞解除した TSC ルートアクセプタ登録名称

KFOT74700 - E

aa....aa command is used invalidly. (E)

コマンドの使用方法が正しくありません。

aa....aa : コマンド名称

(S) コマンドをエラー終了します。

(O) コマンドに指定したオプションに誤りがないか見直してください。

【対策】 指定オプションを修正後、再度コマンドを実行してください。

KFOT74701 - E

error occurred during aa....aa command processing. reason code=bb....bb (E)

コマンド処理でエラーが発生しました。

aa....aa : コマンド名称

bb....bb : 理由コード

6. メッセージ

- 1 : TSC デーモンが開始していません。
- 2 : 共用メモリの確保に失敗しました。
- 3 : タイムアウトが発生しました。
- 4 : 通信障害が発生しました。
- 5 : プロセス固有メモリの確保に失敗しました。
- 6 : TSC デーモンとのバージョン不一致が発生しました。

(S) コマンドをエラー終了します。

(O) 理由コードを確認してください。

【対策】理由コードを参照して障害を取り除いたあと、再度コマンドを実行してください。

KFOT74702 - E

aa....aa specified with bb....bb command do not match. (E)

コマンドで指定されたオプション引数が正しくありません。

aa....aa : コマンド名称

bb....bb : オプション引数名称

(S) コマンドをエラー終了します。

(O) オプション引数を確認してください。

【対策】指定オプションを修正後、再度コマンドを実行してください。

KFOT74703 - E

aa....aa specified with tscholdra command is already in shutdown state. TSCDomain:bb....bb
TSCID:cc....cc (E)

tscholdra コマンドで指定された TSC ルートアクセプタのスケジュールは閉塞しています。

aa....aa : TSC ルートアクセプタ登録名称

bb....bb : TSC ドメイン名称

cc....cc : TSC 識別子

(S) コマンドをエラー終了します。

(O) オプション引数を確認してください。

【対策】指定オプションを修正後、再度コマンドを実行してください。

KFOT74704 - E

failed in placing all rootacceptor in shutdown state. TSCDomain:aa....aa TSCID:bb....bb (E)

TSC ルートアクセプタの一括閉塞処理でエラーが発生しました。

aa....aa : TSC ドメイン名称

bb....bb : TSC 識別子

(S) コマンドをエラー終了します。

(O) オプション引数を確認してください。

【対策】 指定オプションを修正後，再度コマンドを実行してください。

KFOT74705 - E

aa....aa specified with tscrsra command is not in shutdown state. TSCDomain:bb....bb

TSCID:cc....cc (E)

tscrsra コマンドで指定された TSC ルートアクセプタのスケジュールは閉塞していません。

aa....aa : TSC ルートアクセプタ登録名称

bb....bb : TSC ドメイン名称

cc....cc : TSC 識別子

(S) コマンドをエラー終了します。

(O) オプション引数を確認してください。

【対策】 指定オプションを修正後，再度コマンドを実行してください。

KFOT74706 - E

failed restarting all rootacceptor in shutdown state. TSCDomain:aa....aa TSCID:bb....bb (E)

TSC ルートアクセプタの一括閉塞解除処理でエラーが発生しました。

aa....aa : TSC ドメイン名称

bb....bb : TSC 識別子

(S) コマンドをエラー終了します。

(O) オプション引数を確認してください。

【対策】 指定オプションを修正後，再度コマンドを実行してください。

KFOT74750 - I

usage:tscsra [-TSCDomain domain_name] [-TSCID tscid]

[-TSCRootAcceptor root_acceptor_name] [-TSCMyHost host_name]

[[[-TSCInterface [interface_name]] [-TSCHighPrior] | [-TSCSession]] [-h] (C)

tscsra コマンドの使用方法を示します。コマンドの形式が不正な場合，または -h オプションが指定された場合に出力します。

6. メッセージ

(S) コマンドをエラー終了します。

(O) 指定したオプションに誤りがないか見直してください。

【対策】 指定オプションを修正後、再度コマンドを実行してください。

KFOT74752 - I

usage:tscholdra[-TSCDomain domain_name][-TSCID tscid]

{-TSCRootAcceptor root_acceptor_name | -TSCRootAcceptorALL} [-TSCMyHost host_name]

[[{-TSCRequestLeave} [-TSCTimeOut TimeOut] | [-TSCForceSession]] [-h] (C)

tscholdra コマンドの使用方法を示します。コマンドの形式が不正な場合、または -h オプションが指定された場合に出力します。

(S) コマンドをエラー終了します。

(O) 指定したオプションに誤りがないか見直してください。

【対策】 指定オプションを修正後、再度コマンドを実行してください。

KFOT74753 - I

usage:tsctrlsra [-TSCDomain domain_name] [-TSCID tscid]

{-TSCRootAcceptor root_acceptor_name | -TSCRootAcceptorALL} [-TSCMyHost host_name]

[-h] (C)

tsctrlsra コマンドの使用方法を示します。コマンドの形式が不正な場合、または -h オプションが指定された場合に出力します。

(S) コマンドをエラー終了します。

(O) 指定したオプションに誤りがないか見直してください。

【対策】 指定オプションを修正後、再度コマンドを実行してください。

6.2.7 KFOT75000 ~ KFOT75999

KFOT75000 - E

command argument is invalid. option name:aa....aa (L + E)

コマンド引数の指定に誤りがあります。

aa....aa : 誤りがあるオプションの名称

(S) コマンドをエラー終了します。

(O) 指定したオプションに誤りがないか見直してください。

【対策】 指定オプションを修正後、再度コマンドを実行してください。

KFOT75001 - E

this processing exceeded the maximum which was specified by the TSCClientConnectCount option. the effective option value at present is aa....aa. (S + L)

-TSCClientConnectCount オプションに指定された最大値を超えました。

aa....aa : 現在有効なオプション値

(S) getTSCClient メソッドの処理を中断します。

(O) -TSCClientConnectCount オプションの指定値を確認してください。

【対策】 動作中のクライアントアプリケーションの終了を待って、再度実行してください。必要であれば、-TSCClientConnectCount オプションの指定値を増やしたあとに、再度実行してください。

KFOT75002 - E

this processing exceeded the maximum which was specified by the TSCServerConnectCount option. the effective option value at present is aa....aa. (S + L)

-TSCServerConnectCount オプションに指定された最大値を超えました。

aa....aa : 現在有効なオプション値

(S) getTSCServer メソッドの処理を中断します。

(O) -TSCServerConnectCount オプションの指定値を確認してください。

【対策】 動作中のクライアントアプリケーションの終了を待って、再度実行してください。必要であれば、-TSCServerConnectCount オプションの指定値を増やしたあとに、再度実行してください。

KFOT75003 - E

the error occurred by the initialization processing of TSC-daemon. error code=aa....aa maintenance code1=bb....bb maintenance code2=cc....cc (S + L + E)

TSC デーモンの初期化処理中にエラーが発生しました。

aa....aa : 保守エラーコード

bb....bb : 保守コード 1

cc....cc : 保守コード 2

(S) 処理を中断します。

(O) システム管理者に連絡してください。

【対策】 異常終了要因を調査し、OTM またはアプリケーションプログラムを再開始してください。

KFOT75004 - E

the error occurred by the lock control processing of the shared memory access. (S + L + E)

共用メモリアクセスの排他制御処理中にエラーが発生しました。

(S) 処理を中断します。

(O) システム管理者に連絡してください。

【対策】 オプション引数を修正後、再度コマンドを実行してください。

KFOT75005 - E

the error occurred by the registration processing of a communication object. error code=aa....aa

maintenance code1=bb....bb maintenance code2=cc....cc (S + L + E)

通信オブジェクトの登録処理中にエラーが発生しました。

aa....aa : 保守エラーコード

bb....bb : 保守コード 1

cc....cc : 保守コード 2

(S) 処理を中断します。

(O) 次について確認してください。

1. osagent が異常終了していないか。
2. osagent の負荷が高過ぎないか。
3. メモリが不足していないか。

【対策】 それぞれ、次のとおり対策してください。

1. osagent を再開始したあと、再度コマンドを実行してください。
2. 不要なプロセスを終了してください。
3. 必要であればメモリを増やしてください。

KFOT75050 - E

the error occurred by the lock acquisition processing of the lock resources. error code=aa....aa

maintenance code1=bb....bb maintenance code2=cc....cc (S + L)

排他資源に対する排他取得処理中にエラーが発生しました。

aa....aa : 保守エラーコード

bb....bb : 保守コード 1

cc....cc : 保守コード 2

(S) 処理を中断します。

(O) 次について確認してください。

- メモリが不足していないか。

【対策】 次のとおり対策してください。

- 不要なプロセスを終了するか、またはメモリを増やしてください。

KFOT75051 - E

the error occurred by the lock release processing of the lock resources. error code=aa....aa maintenance code1=bb....bb maintenance code2=cc....cc (S + L)

排他資源に対する排他解放処理中にエラーが発生しました。

aa....aa : 保守エラーコード

bb....bb : 保守コード 1

cc....cc : 保守コード 2

(S) 処理を中断します。

(O) 次について確認してください。

- メモリが不足していないか。

【対策】 次のとおり対策してください。

- 不要なプロセスを終了するか、またはメモリを増やしてください。

KFOT75052 - E

the error occurred by the generation processing of a thread. error code=aa....aa maintenance code1=bb....bb maintenance code2=cc....cc (S + L)

スレッド生成処理中にエラーが発生しました。

aa....aa : 保守エラーコード

bb....bb : 保守コード 1

cc....cc : 保守コード 2

(S) 処理を中断します。

(O) 次について確認してください。

1. スレッド資源が不足していないか。
2. メモリが不足していないか。

【対策】 それぞれ、次のとおり対策してください。

1. スレッド資源の上限の設定値を増やしてください。
2. 不要なプロセスを終了するか、またはメモリを増やしてください。

KFOT75053 - E

the memory allocation failed. maintenance code1=aa....aa maintenance code2=bb....bb (S + L)

メモリ割り当てに失敗しました。

aa....aa : 保守コード 1

bb....bb : 保守コード 2

6. メッセージ

(S) 処理を中断します。

(O) メモリが不足していないかを確認してください。

【対策】 不要なプロセスを終了するか、または必要であればメモリを増やしてください。

KFOT75054 - E

the error occurred by the generation processing of lock resources. error code=aa....aa
maintenance code1=bb....bb maintenance code2=cc....cc (S + L + E)

排他資源生成中にエラーが発生しました。

aa....aa : 保守エラーコード

bb....bb : 保守コード 1

cc....cc : 保守コード 2

(S) 処理を中断します。

(O) メモリが不足していないかを確認してください。

【対策】 不要なプロセスを終了するか、または必要であればメモリを増やしてください。

KFOT75055 - I

sending a request was checked. RootAcceptor name:aa....aa Interface name:bb....bb Operation
name:cc....cc Result=dd....dd reason code=ee....ee-ff....ff-gg....gg (L)

リクエスト送信チェックをしました。

aa....aa : リクエスト送信チェックをしたルートアクセプタ名称

bb....bb : リクエスト送信チェックをしたインタフェース名称

cc....cc : リクエスト送信チェックをしたオペレーション名称

dd....dd : リクエスト送信チェックの終了状態

SUCCESS :

リクエスト送信ができるようになりました。

TSCD_TERMINATING :

リクエスト送信チェック中に、TSC デーモンが終了状態になりました。

COMM_FAILURE :

リクエスト送信チェックで通信障害が発生しました。

INTERNAL_ERROR :

リクエスト送信チェックで内部処理エラーが発生しました。

ee....ee : 内部コード 1

ff....ff : 内部コード 2

gg....gg : 内部コード 3

(S) 処理を続行します。

KFOT75056 - W

the timeout occurred by the request of session. session-ID:aa....aa TSCID:bb....bb TSCAcceptor Name:cc....cc timeout=dd....dd (L)

セッション呼び出しインターバル監視でタイムアウトを検出しました。

aa....aa : タイムアウトしたセッション呼び出しのセッション ID

bb....bb : セッション呼び出し対象サーバが接続する TSC デーモンの TSC 識別子

cc....cc : セッション呼び出し対象の TSC アクセプタ名称

dd....dd : セッション呼び出しインターバル監視時間

(S) TSCRootAcceptor クラスの deactivate() メソッドと同じ処理をして、セッション呼び出し待ち状態だったオブジェクトが使用されないようにします。

(O) 次について確認してください。

- セッション呼び出し中のクライアントが正常にリクエストを要求しているか。
- セッション呼び出しインターバル監視時間が適当な値であるか。

【対策】次のとおり対策してください。

- タイムアウトが発生した要因を調査し、OTM またはアプリケーションプログラムを再開始してください。

KFOT75116 - W

cannot open the specified connection. TSCDomain:aa....aa TSCID:bb....bb ec=cc....cc dc=dd....dd pc=ee....ee cs=ff....ff m1=gg....gg m2=hh....hh m3=ii....ii m4=jj....jj (S + L + C + E)

指定された TSC 識別子への接続の確立に失敗しました。

aa....aa : TSC ドメイン名称

TSC ドメイン名称が特定できない場合、"*****" を表示します。

マルチノードリトライ接続の接続方式に API 優先方式を指定した場合、空白が表示される場合があります。

bb....bb : 接続に失敗した TSC 識別子

TSC 識別子が特定できない場合、"*****" を表示します。

マルチノードリトライ接続の接続方式に API 優先方式を指定した場合、空白が表示される場合があります。

cc....cc : エラーコード

dd....dd : 内容コード

ee....ee : 場所コード

ff....ff : 完了状態

gg....gg : 保守コード 1

6. メッセージ

hh...hh : 保守コード 2

ii...ii : 保守コード 3

jj...jj : 保守コード 4

(S) 処理を続行します。

(O) エラーコード, 内容コード, および場所コードを確認してください。

【対策】エラーコード, 内容コード, および場所コードを確認してください。

KFOT75117 - I

the connection to the TSC-daemon completed. TSCDomain:aa....aa TSCID:bb....bb (S + L + C + E)

TSC 識別子への接続の確立が完了しました。

aa....aa : TSC ドメイン名称

TSC ドメイン名称が特定できない場合, "*****" を表示します。

bb....bb : TSC 識別子

TSC 識別子が特定できない場合, "*****" を表示します。

(S) 処理を続行します。

KFOT75118 - E

the format of exception convert file is invalid. file name: aa....aa line=bb....bb reason:cc....cc. (L)

指定した例外変換テーブルファイルの形式が不正です。処理を終了します。

aa....aa : 例外変換テーブルファイル名称

bb....bb : 解析エラーとなった行

cc....cc : エラー発生要因

INVALID_VERSION :

このバージョンでは, 解析できない形式の例外変換テーブルファイルを指定しました。

INVALID_FILE_FORMAT :

例外変換テーブルファイルの形式が不正です。

INVALID_DATA :

例外変換テーブルファイル中のデータが不正です。

(S) 処理を終了します。

(O) 要因を確認してください。

INVALID_VERSION :

例外変換テーブルファイルに, このバージョンでは解析できない指定が記述されている可能性があります。例外変換テーブルファイルの記述内容を確認して

ください。

INVALID_FILE_FORMAT :

例外変換テーブルファイルが、誤った形式で記述されている可能性があります。
例外変換テーブルファイルの記述内容を確認してください。

INVALID_DATA :

例外変換テーブルファイルに、誤った指定が記述されている可能性があります。
例外変換テーブルファイルの記述内容を確認してください。

KFOT75119 - E

an error occurred during the access of the exception convert file. file name:aa....aa. (L)

例外変換テーブルファイルへのアクセスでエラーが発生しました。

aa....aa : 例外変換テーブルファイル名称

(S) 処理を終了します。

(O) このメッセージの前に出力されているメッセージ (KFOT75500-E) を参照して、障害の原因を取り除いてください。

KFOT75120 - I

the information of the exception changed by exception convert file. file name:AA....AA
BadParam=aa....aa NoMemory=bb....bb CommFailure=cc....cc NoPermission=dd....dd
Internal=ee....ee Marshal=ff....ff Initialize=gg....gg NoImplement=hh....hh BadOperation=ii....ii
NoResources=jj....jj NoResponse=kk....kk BadInvOrder=ll....ll Transient=mm....mm
ObjectNotExist=nn....nn Unknown=oo....oo InvObjref=pp....pp ImpLimit=qq....qq
BadTypecode=rr....rr PersistStore=ss....ss FreeMem=tt....tt InvIdent=uu....uu InvFlag=vv....vv
IntfRepos=ww....ww BadContext=xx....xx ObjAdapter=yy....yy DataConversion=zz....zz (L)

-TSCExceptConvertFile オプションで指定された例外変換テーブルファイルから、デフォルトの例外変換とは異なる例外変換情報を抽出して出力します。

AA....AA : -TSCExceptConvertFile オプションに指定したファイル名称

aa....aa : TSCBadParamException に対応する変換後の CORBA 例外

bb....bb : TSCNoMemoryException に対応する変換後の CORBA 例外

cc....cc : TSCCommFailureException に対応する変換後の CORBA 例外

dd....dd : TSCNoPermissionException に対応する変換後の CORBA 例外

ee....ee : TSCInternalException に対応する変換後の CORBA 例外

ff....ff : TSCMarshalException に対応する変換後の CORBA 例外

gg....gg : TSCInitializeException に対応する変換後の CORBA 例外

hh....hh : TSCNoImplementException に対応する変換後の CORBA 例外

ii....ii : TSCBadOperationException に対応する変換後の CORBA 例外

6. メッセージ

jj....jj : TSCNoResourcesException に対応する変換後の CORBA 例外
kk....kk : TSCNoResponseException に対応する変換後の CORBA 例外
ll....ll : TSCBadInvOrderException に対応する変換後の CORBA 例外
mm....mm : TSCTransientException に対応する変換後の CORBA 例外
nn....nn : TSCObjectNotExistException に対応する変換後の CORBA 例外
oo....oo : TSCUnknownException に対応する変換後の CORBA 例外
pp....pp : TSCInvObjrefException に対応する変換後の CORBA 例外
qq....qq : TSCImpLimitException に対応する変換後の CORBA 例外
rr....rr : TSCBadTypecodeException に対応する変換後の CORBA 例外
ss....ss : TSCPersistStoreException に対応する変換後の CORBA 例外
tt....tt : TSCFreeMemException に対応する変換後の CORBA 例外
uu....uu : TSCInvIdentException に対応する変換後の CORBA 例外
vv....vv : TSCInvFlagException に対応する変換後の CORBA 例外
ww....ww : TSCIntfReposException に対応する変換後の CORBA 例外
xx....xx : TSCBadContextException に対応する変換後の CORBA 例外
yy....yy : TSCObjAdapterException に対応する変換後の CORBA 例外
zz....zz : TSCDataConversionException に対応する変換後の CORBA 例外
(S) 処理を続行します。

KFOT75122 - E

the error occurred by the deactivation processing of TSCRootAcceptor. TSCDomain:aa....aa
TSCID:bb....bb RootAcceptor name:cc....cc exception:dd....dd,ee....ee,ff....ff,gg....gg maintenance
code=hh....hh (L)

TSC ルートアクセプタの非活性処理でエラーが発生しました。

aa....aa : TSC ドメイン名称

bb....bb : TSC 識別子

cc....cc : TSC ルートアクセプタ名称

dd....dd : エラーコード

ee....ee : 内容コード

ff....ff : 場所コード

gg....gg : 完了状態

hh....hh : 保守コード

(S) 処理を続行します。

(O) TSC ドメイン名称, TSC 識別子, TSC ルートアクセプタ名称, 内容コード, 場所コード, および完了状態を参照し, 障害の原因を取り除いてください。障害の原因が不明な場合は, 保守コードを取得し, 保守員に連絡してください。

KFOT75125 - W

the timeout occurred in the TSCThreadFactory class. method: aa....aa rootacceptor name: bb....bb thread-ID=cc....cc watch_time=dd....dd (L)

TSCThreadFactory クラスの呼び出し中にタイムアウトを検出しました。

aa....aa : 呼び出し中のメソッド名称

bb....bb : TSC ルートアクセプタ名称

TSC ルートアクセプタ名称が指定されていない場合, "default" を表示します。

cc....cc : タイムアウトしたスレッド ID

dd....dd : タイムアウト監視時間 (秒)

(S) 処理を続行します。

(O) システム管理者に連絡してください。

【対策】 タイムアウトが発生した要因を調査し, 必要に応じて OTM またはアプリケーションプログラムを再開始してください。

KFOT75126 - W

the timeout occurred in the TSCObjectFactory class. method: aa....aa acceptor name:bb....bb interface name:cc....cc thread-ID=dd....dd watch_time=ee....ee (L)

TSCObjectFactory クラスの呼び出し中にタイムアウトを検出しました。

aa....aa : 呼び出し中のメソッド名称

bb....bb : TSC アクセプタ名称

TSC アクセプタ名称が指定されていない場合, "*****" を表示します。

cc....cc : タイムアウトしたインタフェース名称

dd....dd : タイムアウトしたスレッド ID

ee....ee : タイムアウト監視時間 (秒)

(S) 処理を続行します。

(O) システム管理者に連絡してください。

【対策】 タイムアウトが発生した要因を調査し, 必要に応じて OTM またはアプリケーションプログラムを再開始してください。

KFOT75306 - E

the connection failure between TSC-regulator and TSC-daemon. maintenance code1=aa....aa
maintenance code2=bb....bb (L)

TSC レギュレータで、TSC デーモンとのコネクション切断を検知しました。

aa....aa : 保守情報 1

bb....bb : 保守情報 2

(S) 処理を中断します。

(O) システム管理者に連絡してください。

KFOT75410 - I

usage : tsclscon { [-h] | [-TSCDomain domain_name] [-TSCID tscid] [-TSCMyHost host_name]
[-TSCList] [-TSCReg pid[,pid]] [-TSCOrb APID[,APID]]} (C + E)

tsclscon コマンドの使用方法を示します。コマンドの形式が不正な場合、または -h オプションが指定された場合に出力します。

(S) コマンドを終了します。

(O) 指定したオプションに誤りがないか見直してください。

【対策】 指定オプションを修正後、再度コマンドを実行してください

KFOT75411 - E

cannot get connection information.reason code=aa....aa-bb....bb-cc....cc (E)

コネクション情報取得中にエラーが発生しました。

aa....aa : 理由コード

101 : メモリ不足が発生しました。

102 : 対象となる TSC デーモンの情報が取得できませんでした。コマンド実行時の -TSCDomain オプション、-TSCID オプション、-TSCMyHost オプション、および TSCSPOOL 環境変数の指定値を見直してください。

または、対象となる TSC デーモンが開始していません。

106 : 解読できないバージョンの情報が含まれています。情報取得対象となる TSC デーモンと同じバージョンのコマンドで、再度情報を取得してください。

上記以外 : その他内部エラー

bb....bb : 内部コード 1

cc....cc : 内部コード 2

(S) コマンドを終了します。

(O) 理由コードを確認してください。

【対策】理由コードを参照して、障害の原因を取り除いてください。

上記以外の理由コードについては、内部コード 1 および 2 を取得し保守員に連絡してください。

KFOT75492-W

the CORBA exception was thrown in user application. class name:aa...aa minor=bb...bb completed=cc...cc (L)

アプリケーションプログラムの内部で CORBA 例外が発生しました。

aa...aa : CORBA 例外クラス名

CORBA 例外クラス名を取得できない場合、"****" を表示します。

bb...bb : CORBA 例外のマイナーコード

cc...cc : CORBA 例外処理の完了状態

(S) 例外情報を出力後、処理を続行します。

(O) CORBA 例外クラス名、CORBA 例外のマイナーコード、および CORBA 例外処理の完了状態を取得し、原因を調査してください。

【対策】アプリケーションプログラムのユーザ実装部分で、CORBA 例外を通知する可能性がある個所を見直してください。

KFOT75493 - E

the stack information of aa...aa:No.bb...bb:cc...cc (L)

アプリケーションプログラム内での java.lang.RuntimeException または java.lang.Error の発生時のコールスタック情報です。

aa...aa : 例外クラスまたはエラークラス名称

bb...bb : コールスタックナンバー

cc...cc : スタック情報

(S) 例外またはエラー発生時のコールスタック情報を出力後、処理を続行します。

(O) アプリケーションプログラムの実行環境が正しいか確認してください。

KFOT75494 - E

the java.lang.RuntimeException was thrown in the user application.the exception class name :aa...aa (L)

アプリケーションプログラム内で java.lang.RuntimeException が発生しました。

aa...aa : 例外クラス名称

(S) 処理を続行します。

(O) アプリケーションプログラムの実行環境が正しいか確認してください。

KFOT75495 - E

the java.lang.Error was thrown in the user application.the error class name :aa....aa (L)

アプリケーションプログラム内で java.lang.Error が発生しました。

aa....aa : エラークラス名称

(S) 処理を続行します。

(O) アプリケーションプログラムの実行環境が正しいか確認してください。

KFOT75496 - E

the java.lang.Error was thrown in the system or user application.the error class name :aa....aa

(L)

システム内、またはアプリケーションプログラム内で java.lang.Error が発生しました。

aa....aa : エラークラス名称

(S) 処理を続行します。

(O) アプリケーションプログラムの実行環境が正しいか確認して、システム管理者に連絡してください。

KFOT75497 - E

the java.lang.RuntimeException was thrown in the system. the exception class name:aa....aa (L)

システム内で java.lang.RuntimeException が発生しました。

aa....aa : 例外クラス名称

(S) 処理を続行します。

(O) アプリケーションプログラムの実行環境が正しいか確認して、システム管理者に連絡してください。

KFOT75500 - E

aa....aa(bb....bb) is failed.errno=cc....cc:dd....dd (S + L + C + E)

システムコールでエラーが発生しました。

aa....aa : エラーとなったシステムコール

bb....bb : TSC 内部情報 (システムコール呼び出し元モジュール名称)

cc....cc : システムコールの errno 値

dd....dd : システムコールエラーの内容

(S) 障害の重要度によって、次のどれかの処置をとります。

- 処理を打ち切り、プロセスを異常終了させます。
- 処理を打ち切り、実行中サービスの呼び出し元へリターンします。
- そのまま処理を続行します。

(O) システムコール名称と errno 値を基に、該当するマニュアルで原因を調査してください。

【対策】 errno 値を参照して、原因を確認して対策してください。

KFOT75800 - E

An exception occurred. code=aa...aa ExpAddr:bb...bb (S + L + C)

コード aa...aa の構造化例外が bb...bb の命令を実行中に発生しました。

aa...aa : 例外コード

bb...bb : 例外が発生したアドレス

(S) スタックトレースを取得して、処理を中断します。

(O) システム管理者に連絡してください。

KFOT75999 - E

aa...aa(bb...bb) killed by code=cc...cc (S + L + C + E)

異常が発生したため、OTM のサービスを終了しました。

aa...aa : エラーを検出した内部処理コード

bb...bb : 終了したプロセス ID

cc...cc : アボートコード (異常終了要因コード)

(S) 障害の重要度によって、次のどれかの処置をとります。

- TSC ドメインマネージャプロセスを異常終了させます。
- TSC デーモンプロセスを異常終了させます。
- TSC レギュレータプロセスを異常終了させます。
- PRF デーモンプロセスを異常終了させます。
- クライアントアプリケーションプロセスを異常終了させます。
- サーバアプリケーションプロセスを異常終了させます。

(O) システム管理者に連絡してください。

【対策】 異常終了要因を調査し、OTM またはアプリケーションプログラムを再開始してください。アボートコードについて、「付録 A アボートコード一覧」を参照してください。

6.2.8 KFOT76000 ~ KFOT76999

KFOT76000 - I

usage:tscver -h | file_path (C + E)

tscver コマンドの使用方法を示します。コマンドの形式が不正な場合、または -h オプションが指定された場合に出力します。

6. メッセージ

(S) コマンドをエラー終了します。

(O) 指定したオプションに誤りがないか見直してください。

【対策】 指定オプションを修正後、再度コマンドを実行してください。

KFOT76001 - I

usage:tscjver -h | file_path (C + E)

tscjver コマンドの使用方法を示します。コマンドの形式が不正な場合、または -h オプションが指定された場合に出力します。

(S) コマンドをエラー終了します。

(O) 指定したオプションに誤りがないか見直してください。

【対策】 指定オプションを修正後、再度コマンドを実行してください。

KFOT76002 - E

the memory allocation failed. (E)

メモリ割り当てに失敗しました。

(S) 処理を中断します。

(O) メモリが不足していないかを確認してください。

【対策】 不要なプロセスを終了するか、または必要であればメモリを増やしてください。

KFOT76003 - E

the setting of an environment variable is unreasonable. environment variable name:aa...aa (E)

環境変数の指定に誤りがあります。

aa...aa : 誤りがある環境変数の名称

(S) 処理を中断します。

(O) 環境変数の指定に誤りがないか見直してください。

【対策】 環境変数の指定を修正後、再度コマンドを実行してください。

KFOT76004 - E

the specified file path is unreasonable. file path:aa...aa (E)

ファイルパスの指定に誤りがあります。

aa...aa : 誤りがあるファイルパスの名称

(S) 処理を中断します。

(O) 指定したファイルパスに誤りがないか見直してください。

【対策】 ファイルパスの指定を修正後、再度コマンドを実行してください。

KFOT76005 - E

the specified file path is too long. (E)

指定されたファイルパスが長過ぎます。

(S) 処理を中断します。

(O) 指定したファイルパスの文字列長が 255 バイトを超えていないか見直してください。

【対策】ファイルパスの指定を修正後再度コマンドを実行してください。

KFOT76006 - E

command argument is invalid. (E)

コマンド引数の指定に誤りがあります。

(S) 処理を中断します。

(O) 指定したオプションに誤りがないか見直してください。

【対策】指定オプションを修正後、再度コマンドを実行してください。

KFOT76007 - E

the specified file path doesn't exist. file path:aa....aa (E)

指定されたファイルパスはありません。

aa....aa：指定されたファイルパスの名称

(S) コマンドをエラー終了します。

(O) 指定したファイルパスに誤りがないか見直してください。

【対策】ファイルパスの指定を修正後、再度コマンドを実行してください。

KFOT76008 - E

the version information doesn't exist in the specified file path. (E)

指定されたファイルパスにバージョン情報はありません。

(S) コマンドをエラー終了します。

(O) 指定したファイルパスに誤りがないか見直してください。

【対策】ファイルパスの指定を修正後、再度コマンドを実行してください。

KFOT76009 - E

permission denied. file path:aa....aa (E)

指定されたファイルへのアクセス権限はありません。

aa....aa：指定されたファイルパスの名称

(S) コマンドをエラー終了します。

6. メッセージ

(O) 指定したファイルパスのアクセス権を見直してください。

【対策】ファイルパスのアクセス権限を chmod コマンドで修正後、再度コマンドを実行してください。

KFOT76010 - E

cannot get version information. reason code=aa....aa (E)

バージョン情報取得中にエラーが発生しました。

aa....aa : 要因コード

- 1 : TSCDIR 環境変数の設定に誤りがあります。
- 2 : インストールに失敗しています。
- 3 : TSCDIR 環境変数の設定に誤りがあるか、またはインストールに失敗しています。

(S) コマンドを終了します。

(O) 要因コードを参照して原因を調査してください。

【対策】それぞれ、次のとおり対策してください。

1. TSCDIR 環境変数を適当なものに修正してください。
2. 再インストールしてください。
3. TSCDIR 環境変数を適当なものに修正してください。または再インストールしてください。

KFOT76014 - E

the wrong library was loaded. Name:aa....aa PP-No:bb....bb Version:cc....cc (S + L + C)

不正なライブラリがロードされました。

aa....aa : ライブラリファイル名

bb....bb : 製品形名

cc....cc : 製品のバージョン

(S) デーモンおよびコマンドの処理を中断します。

(O) インストール環境や環境変数の設定内容を見直してください。設定内容を修正後、再度コマンドを実行してください。

6.2.9 KFOT77000 ~ KFOT77999

KFOT77000 - E

invalid option for the connector for ORB.option=aa....aa value=bb....bb reason=cc....cc. (L+E)

TSCORB コネクタを開始する tscstartgw コマンドのオプションの指定に誤りがありま

す。

aa....aa : 誤りがあるオプションの名称

bb....bb : aa....aa で示されるオプションに指定した値
指定値がない場合は何も表示されません。

cc....cc : エラー発生要因

NO_OPTION :

指定を省略できないオプションが指定されていません。

NO_OPT_ARG :

必要なオプション引数がありません。

OVER_LIMIT :

オプション引数の値が指定できる範囲を超えています。または、オプション引数の文字列長が制限を超えています。

INV_VAL :

オプション引数の値が不正です。

COMBINATION :

オプションの組み合わせが不正です。

上記以外 :

オプションの解析中に内部エラーが発生しました。

(S) 処理を中断します。

(O)tscstartgw コマンドに指定したオプションに誤りがないかどうかを見直してください。

【対策】 指定オプションを修正後、再度コマンドを実行してください。内部エラーの場合は保守員に連絡してください。

KFOT77001 - E

the error occurred while initializing the connector for ORB. kind=aa....aa proc=bb....bb inter=cc....cc detail=dd....dd. (L+E)

TSCORB コネクタの初期化中にエラーが発生しました。

aa....aa : エラー発生要因

OTM_EXCEPT :

OTM 例外が発生しました。

ORB_EXCEPT :

CORBA 例外が発生しました。

MEMORY :

メモリ不足が発生しました。

上記以外 :

6. メッセージ

内部エラーが発生しました。

bb....bb : エラー発生個所

OPTION : オプション解析処理

CREATE_OBJ : CORBA オブジェクトの作成・登録処理

OTM_CON : OTM コネクションの初期化処理

ORB_CON : ORB コネクションの初期化処理

上記以外 : その他の内部処理

cc....cc : 保守情報

dd....dd : エラー詳細情報

(S) 処理を中断します。

(O) エラー発生要因, エラー発生個所, 保守情報, およびエラー詳細情報を取得し, システム管理者に連絡してください。

【対策】 次のとおり対策してください。

- aa....aa が MEMORY の場合は, 不要なプロセスを終了させるか, またはメモリを増やしてください。
- aa....aa が OTM_EXCEPT の場合は, TSCORB コネクタを管理する TSC デーモンに障害が発生している可能性があります。ほかのメッセージに従って TSC デーモンの障害を取り除いてから, 再度 TSCORB コネクタを起動してください。または, TSCORB コネクタを管理する TSC デーモンのクライアント接続数 (TSCORB コネクタを開始する tscorbstart コマンドの -TSCClientConnectCount オプションの指定値) が不足している可能性があります。クライアント接続数を見直して, 再度 TSCORB コネクタを開始してください。
- aa....aa が ORB_EXCEPT の場合は, TPBroker の環境設定およびオプションが正しく設定されているかどうかを見直し, 再度 TSCORB コネクタを開始してください。問題がない場合は保守員に連絡してください。
- 上記以外の場合は, エラー発生要因, エラー発生個所, 保守情報, およびエラー詳細情報を取得し, 保守員に連絡してください。

KFOT77002 - I

create CORBA Object. rep_id=aa....aa op=bb....bb connect_way=cc....cc. (L+C)

CORBA オブジェクトを作成しました。

aa....aa : リポジトリ ID

bb....bb : オペレーション名称

複数の場合は, コンマ(,) で区切って出力されます。

cc....cc : TPBroker の osagent を使用した接続の可否

該当するオブジェクトに対して, TPBroker の osagent を使用した接続ができる場

合は "0001" , IOR 文字列を使用する接続ができる場合は "0010" , 両方ともできる場合は "0011" が出力されます。

(S) 処理を続行します。

KFOT77004 - E

cannot get process informations of the connector for ORB. kind=aa....aa inter=bb....bb. (E)
TSCORB コネクタのプロセス情報取得に失敗しました。

aa....aa : エラー発生要因

SHM_ACC :

共用メモリのアクセス時にエラーが発生しました。

MEMORY :

メモリ不足が発生しました。

上記以外 :

内部エラーが発生しました。

bb....bb : 保守情報

(S) 処理を中断します。

(O) エラー発生要因および保守情報を取得し、システム管理者に連絡してください。

【対策】次のとおり対策してください。

- aa....aa が SHM_ACC の場合は、TSCSPOOL 環境変数が正しく設定されているかどうか、該当する TSC デーモンが正しく起動されているかどうか、TSCSPOOL 下のファイルを不当に削除・変更していないかどうかを確認してください。それらに問題がない場合は保守員に連絡してください。
- aa....aa が MEMORY の場合は、不要なプロセスを終了させるか、またはメモリを増やしてください。
- 上記以外の場合は、保守員に連絡してください。

付録

付録 A	アボートコード一覧
付録 B	tscgetref コマンドの出力例
付録 C	tscstsed コマンドの出力例
付録 D	例外変換テーブルファイルの作成方法

付録 A アポートコード一覧

OTM のメッセージが出力するアポートコードを、コード順に次の表に示します。次の表にないアポートコードが出力された場合は core ファイルを取得し、保守員に連絡してください。

表 A-1 アポートコードの一覧

アポートコード	原因	対策
aWTO001	C++ のサーバアプリケーションで時間監視機能のタイムアウトが発生しました。	タイムアウトが発生した要因を調査し、OTM またはアプリケーションプログラムを再開始してください。
bCo003 bCo011 bCo023 bCo024 bSh001 bSh003 bSh004 bSh007	メモリ不足が発生しました。	メモリを再度、見積もり直してください。
eCEV001 ~ eCEV999	コネクション切断を検知しました。	動作中のメモリ不足などが考えられます。動作環境のメモリを増やしてください。また、TSC デーモンが起動されているかどうかを確認してください。引き続き多発する場合は、保守員に連絡してください。
eEVT001 ~ eEVT999	コネクション切断処理中に同期制御でエラーが発生しました。	動作中のメモリ不足が考えられます。動作環境のメモリを増やしてください。引き続き多発する場合は、保守員に連絡してください。
eFLK001 ~ eFLK999	コネクション切断処理中に排他確保に失敗しました。	動作中のメモリ不足が考えられます。動作環境のメモリを増やしてください。引き続き多発する場合は、保守員に連絡してください。
eFPM001 ~ eFPM999	TSC デーモン内でメモリ不足が発生しました。	メモリを見積もり直して、再度実行してください。
eNPM001 ~ eNPM999	通信処理中にメモリ不足が発生しました。	メモリを見積もり直して、再度実行してください。
eULK001 ~ eULK999	TSC デーモン内での排他解放に失敗しました。	動作中のメモリ不足が考えられます。動作環境のメモリを増やしてください。引き続き多発する場合は、保守員に連絡してください。
eWST001	C++ の TSCRootAcceptor クラスの destroy() メソッドでタイムアウトが発生しました。	ABC_TSCfactimpl クラスの destroy() メソッドまたは TSCThreadFactory クラスの destroy() メソッドの処理からリターンしていない可能性があります。要因を調査し、アプリケーションプログラムを再開始してください。

アボートコード	原因	対策
r000000 r000001 r000002 r000003 r000004 r000008 r000009 r000010 r000011 r000012 r000013	排他確保処理または解放処理でエラーが発生しました。	プロセス固有メモリ不足が発生している可能性があります。メモリ不足が発生している場合は、メモリサイズを見直してください。繰り返し発生する場合には、保守員に連絡してください。
r001000	メモリ不足が発生しました。	メモリサイズを見直してください。繰り返し発生する場合には、保守員に連絡してください。
r001001 r001002 r001003 r001004 r001005 r001006 r001007 r001008 r001009 r001010 r001011 r001012 r001013 r001014 r001015 r001016	排他確保処理または解放処理でエラーが発生しました。	プロセス固有メモリ不足が発生している可能性があります。メモリ不足が発生している場合は、メモリサイズを見直してください。繰り返し発生する場合には、保守員に連絡してください。
s002000 s002001 s003000 s003001 s004000 s004001	排他確保処理または解放処理でエラーが発生しました。	プロセス固有メモリ不足が発生している可能性があります。メモリ不足が発生している場合は、再度メモリを見直してください。繰り返し発生する場合には、保守員に連絡してください。
s005000	メモリ不足が発生しました。	メモリサイズを見直してください。繰り返し発生する場合には、保守員に連絡してください。
uFPM001 ~ uFPM999	メモリ不足が発生しました。	メモリを見積もり直して、再度実行してください。
uLCK001 ~ uLCK999	排他確保に失敗しました。	動作中のメモリ不足が考えられます。動作環境のメモリを増やしてください。引き続き多発する場合は、保守員に連絡してください。
uNPM001 ~ uNPM999	メモリ不足が発生しました。	メモリを見積もり直して、再度実行してください。

付録 B tscgetref コマンドの出力例

tscgetref コマンドを使用して接続先情報を出力する場合、コマンドオプションを組み合わせることで次の表に示す処理ができます。

表 B-1 tscgetref コマンドの処理

処理	説明
接続先情報の抽出	オンライン中の TSC ノードの最新情報、または接続先情報ファイルからコマンドオプションで指定された値と合致するものを抽出します。
接続先情報のソート	オンライン中の TSC ノードの最新情報、または接続先情報ファイルの内容をコマンドオプションで指定された優先順位に従ってソートします。
接続先情報のマージ	オンライン中の TSC ノードの最新情報、または接続先情報ファイルの内容を一つにまとめます。

ここでは、次の図に示す環境のホスト 1 上で tscgetref コマンドを実行する場合のコマンドオプションの指定方法、および出力される接続先情報ファイルの内容を示します。

図 B-1 tscgetref コマンドを使用する環境



付録 B.1 一つの TSC ノードの接続先情報を出力する例

ここでは、次に示す場合のコマンドオプションの指定方法と出力例を示します。

- TSC ドメイン "HITACHI" の TSC ノード "sales" の情報だけを抽出します。
- 出力先の接続先情報ファイルとして、"/tmp/ref_file" を指定します。

TSC ドメインと TSC ノードは、-TSCDomain オプションと -TSCID オプションで指定します。また、出力先情報ファイルの指定には、-TSCOutFile オプションを使用します。

(1) コマンドオプションの指定方法

```
tscgetref -TSCDomain HITACHI -TSCID sales
          -TSCOutFile /tmp/ref_file
```

(2) 接続先情報ファイル (/tmp/ref_file) に出力される内容

```
TSCREFINFO:01:#
HITACHI#sales#0#0#111.11.111.11#15000##
HITACHI#sales#0#1#111.11.111.11#15000##IOR:002020200000000c49401002...0fa
HITACHI#sales#1#0#111.11.111.11#0##
```

付録 B.2 複数の TSC ドメインと TSC ノードの接続先情報をソートして出力する例

ここでは、次に示す場合のコマンドオプションの指定方法と出力例を示します。

- TSC ドメイン名称を "SOFT", "HITACHI", "SYSTEM" の順序でソートします。
- TSC ノード (TSC 識別子) を "payroll", "purchase", "sales" の順序でソートします。
- TSC レギュレータを経由する接続に関する情報だけを抽出します。
- 出力先の接続先情報ファイルとして、"/tmp/ref_file" を指定します。

TSC ドメイン名称と TSC 識別子のソートの順序は、-TSCDomain オプションと -TSCID オプションでの指定順序に従います。また、接続経路による情報の限定には、-TSCRequestWay オプションを使用します。

(1) コマンドオプションの指定方法

```
tscgetref -TSCDomain SOFT,HITACHI,SYSTEM
          -TSCID payroll,purchase,sales -TSCRequestWay 1
          -TSCOutFile /tmp/ref_file
```

(2) 接続先情報ファイル (/tmp/ref_file) に出力される内容

```
TSCREFINFO:01:#
SOFT#purchase#1#0#222.22.222.222#0##
SOFT#purchase#1#1#222.22.222.222#16001##IOR:002020200000000c4940100200...0fca
SOFT#sales#1#0#111.11.111.11#0##
HITACHI#payroll#1#0#222.22.222.222#0##
HITACHI#payroll#1#1#222.22.222.222#15001##IOR:002020200000000c4940100...00c
HITACHI#purchase#1#0#111.11.111.11#0##
HITACHI#sales#1#0#111.11.111.11#0##
SYSTEM#payroll#1#0#222.22.222.222#0##
SYSTEM#payroll#1#1#222.22.222.222#17001##IOR:002020200000000c4940100...1ba
SYSTEM#sales#1#0#111.11.111.11#0##
```

付録 B.3 すべての TSC ドメインから TSC ノードを選択して接続先情報を出力する例

ここでは、次に示す場合のコマンドオプションの指定方法と出力例を示します。

- オンライン中のすべての TSC ドメインから、TSC ノード "sales" の情報だけを抽出します。
- 出力先の接続先情報ファイルとして、"/tmp/ref_file" を指定します。

すべての TSC ドメインの情報を出力するには、-TSCDomainAll オプションを使用します。ただし、-TSCDomainAll オプションを使用した場合は、TSC ドメインごとのソートはできません。

(1) コマンドオプションの指定方法

```
tscgetref -TSCDomainAll -TSCID sales
          -TSCOutFile /tmp/ref_file
```

(2) 接続先情報ファイル (/tmp/ref_file) に出力される内容

```
TSCREFINFO:01:#
HITACHI#sales#0#0#111.11.111.11#15000##
HITACHI#sales#0#1#111.11.111.11#15000##IOR:002020200000000c4940100200000...3e1
HITACHI#sales#1#0#111.11.111.11#0##
SOFT#sales#0#0#111.11.111.11#17000##
SOFT#sales#0#1#111.11.111.11#17000##IOR:002020200000000c494010020000ca91032...c0a
SOFT#sales#1#0#111.11.111.11#0##
SYSTEM#sales#0#0#111.11.111.11#18000##
SYSTEM#sales#0#1#111.11.111.11#18000##IOR:002020200000000c49401002000c0e1...0ca
SYSTEM#sales#1#0#111.11.111.11#0##
```

付録 B.4 複数の接続先情報ファイルをマージして出力する例

ここでは、次に示す場合のコマンドオプションの指定方法と出力例を示します。

- 複数の接続先情報ファイル "/tmp/ref_file1" , および "/tmp/ref_file2" を入力元として接続先情報をマージします。
- 入力元の接続先情報ファイルから , 属する TSC ドメインに関係なく TSC ノード "payroll" と "sales" の情報だけを抽出して , "payroll" , "sales" の順序でソートします。
- 出力先の接続先情報ファイルとして , "/tmp/ref_file3" を指定します。

マージする入力元の接続先情報ファイルの指定には , -TSCInFile オプションを使用します。なお , 重複するレコードは削除されます。

(1) コマンドオプションの指定方法

```
tscgetref -TSCDomainAll -TSCID payroll,sales
          -TSCInFile /tmp/ref_file2,/tmp/ref_file1
          -TSCOutFile /tmp/ref_file3
```

(2) 入力元の接続先情報ファイル (/tmp/ref_file1) の内容

```
TSCREFINFO:01:#
HITACHI#sales#0#0#111.11.111.11#15000##
HITACHI#sales#0#1#111.11.111.11#15000##IOR:002020200000000c4940100200000...3e1
HITACHI#sales#1#0#111.11.111.11#0##
SOFT#sales#0#0#111.11.111.11#17000##
SOFT#sales#0#1#111.11.111.11#17000##IOR:002020200000000c494010020000a09cb...c0a
SOFT#sales#1#0#111.11.111.11#0##
SYSTEM#sales#0#0#111.11.111.11#18000##
SYSTEM#sales#0#1#111.11.111.11#18000##IOR:002020200000000c49401002000ca1...0ca
SYSTEM#sales#1#0#111.11.111.11#0##
```

(3) 入力元の接続先情報ファイル (/tmp/ref_file2) の内容

```
TSCREFINFO:01:#
SYSTEM#sales#0#0#111.11.111.11#18000##
SYSTEM#sales#0#1#111.11.111.11#18000##IOR:002020200000000c49401002000ca1...0ca
SYSTEM#sales#1#0#111.11.111.11#0##
SYSTEM#payroll#0#0#222.22.222.222#17000##
SYSTEM#payroll#0#1#222.22.222.222#17000##IOR:002020200000000c4940100200ca01...0bc
SYSTEM#payroll#1#0#222.22.222.222#0##
SYSTEM#payroll#1#1#222.22.222.222#17001##IOR:002020200000000c4940100200ea0...1ba
```

(4) 接続先情報ファイル (/tmp/ref_file3) に出力される内容

```
TSCREFINFO:01.#
SYSTEM#payroll#0#0#222.22.222.222#17000##
SYSTEM#payroll#0#1#222.22.222.222#17000#IOR:002020200000000c494010020000ca1...0bc
SYSTEM#payroll#1#0#222.22.222.222#0##
SYSTEM#payroll#1#1#222.22.222.222#17001##IOR:002020200000000c494010020ecb0...1ba
SYSTEM#sales#0#0#111.11.111.11#18000##
SYSTEM#sales#0#1#111.11.111.11#18000##IOR:002020200000000c49401002000cf1...0ca
SYSTEM#sales#1#0#111.11.111.11#0##
HITACHI#sales#0#0#111.11.111.11#15000##
HITACHI#sales#0#1#111.11.111.11#15000##IOR:002020200000000c4940100200000...3e1HITACHI#sales#1#0#111.11.111.11#0##
SOFT#sales#0#0#111.11.111.11#17000##
SOFT#sales#0#1#111.11.111.11#17000##IOR:002020200000000c49401002000de9810...c0a
SOFT#sales#1#0#111.11.111.11#0##
```

付録 C tscstsed コマンドの出力例

tscstsed コマンドを使用して稼働統計情報を編集・出力する場合、コマンドオプションの指定の組み合わせによって、出力結果が異なります。また、この出力結果は、tscststart コマンドと tscststop コマンドの実行のタイミングに依存します。

ここでは、次の表に示す場合の tscstsed コマンドの出力例を示します。

表 C-1 tscstsed コマンドのオプション指定

オプション		出力例の説明	参照先
-TSCedit	-TSCCsv		
nod	指定しない	TSC ノード全体でまとめた情報が出力されます。	付録 C.1
nod	指定する	TSC ノード全体でまとめた情報が CSV 形式で出力されます。	付録 C.2
rac	指定しない	TSC ルートアクセプタ単位でスケジュールキュー情報が出力されます。	付録 C.3
rac	指定する	TSC ルートアクセプタ単位のスケジュールキュー情報が CSV 形式で出力されます。	付録 C.4
opr	指定しない	サーバアプリケーションのオペレーション単位でリクエスト情報が出力されます。	付録 C.5
opr	指定する	サーバアプリケーションのオペレーション単位のリクエスト情報が CSV 形式で出力されます。	付録 C.6

さらに、次に示す場合の出力例を示します。

- 稼働統計情報ファイルに、ある時間帯の対象レコードがない場合
- tscststart コマンドと tscststop コマンドの実行を何度か繰り返して稼働統計情報を出力した場合

付録 C.1 TSC ノード全体でまとめた情報の出力例

*** Statistics Information by TSCNode ***

TSCDomain :TSCDOMAIN

TSCID :172.17.13.186

** Edit time 2002/03/25 16:52 - 2002/03/25 17:02 **

statistical data : 2002/03/25 16:52:00 - 2002/03/25 16:54:52

Event	Count	Maximum	Minimum	Average (unit)
Request Response Time	503	88227	4071	16046(usec)
Request Execution Time	890	62292	63	498(usec)
Request Time Out	2	-	-	-(count)
Request Error	0	-	-	-(count)
Request ResponseTime(ORB)	386	82432	4155	12765(usec)
Request Time Out(ORB)	0	-	-	-(count)
Request Error(ORB)	0	-	-	-(count)
Schedule Wait Count	892	1	1	1(count)
Schedule Wait Count(HPR)	892	1	1	1(count)
Schedule Wait Time	892	41241	73	962(usec)
Request Overflow	0	-	-	-(count)
Request Overflow(LPR)	0	-	-	-(count)
AP down	2	-	-	-(count)
Regulator down	0	-	-	-(count)
ORBConnector down	0	-	-	-(count)
Regulator-AP	-	0	0	0(count)
TSCDaemon-ClientAP	-	1	1	1(count)
TSCDaemon-ServerAP(Ctr)	-	1	1	1(count)
TSCDaemon-ServerAP(Req)	-	1	1	1(count)
TSCDaemon-Regulator	-	0	0	0(count)
TSCDaemon-ORBConnector	-	1	1	1(count)

付録 C.2 TSC ノード全体でまとめた情報の CSV 形式での出力例

```

*** Statistics Information by TSCNode ***
TSCDomain, TSCID, start, end, Event, Count, Maximum, Minimum, Average, unit
OTM_SI_ZZZ.2002/01/07 09:37.2002/01/07 09:47,Request Response Time, 0, 0, 0, 0(usec)
OTM_SI_ZZZ.2002/01/07 09:37.2002/01/07 09:47,Request Execution Time, 1, 2003763, 2003763, 2003763(usec)
OTM_SI_ZZZ.2002/01/07 09:37.2002/01/07 09:47,Request Time Out, 7...(count)
OTM_SI_ZZZ.2002/01/07 09:37.2002/01/07 09:47,Request Error, 12...(count)
OTM_SI_ZZZ.2002/01/07 09:37.2002/01/07 09:47,Request Response Time(ORB), 0, 0, 0, 0(usec)
OTM_SI_ZZZ.2002/01/07 09:37.2002/01/07 09:47,Request Time Out(ORB), 0...(count)
OTM_SI_ZZZ.2002/01/07 09:37.2002/01/07 09:47,Request Error(ORB), 0...(count)
OTM_SI_ZZZ.2002/01/07 09:37.2002/01/07 09:47,Schedule Wait Count, 8, 7, 1, 4(count)
OTM_SI_ZZZ.2002/01/07 09:37.2002/01/07 09:47,Schedule Wait Count(HPR), 8, 7, 1, 4(count)
OTM_SI_ZZZ.2002/01/07 09:37.2002/01/07 09:47,Schedule Wait Time, 8, 60005165, 659, 49858628(usec)
OTM_SI_ZZZ.2002/01/07 09:37.2002/01/07 09:47,Request Overflow, 0...(count)
OTM_SI_ZZZ.2002/01/07 09:37.2002/01/07 09:47,Request Overflow(LPR), 0...(count)
OTM_SI_ZZZ.2002/01/07 09:37.2002/01/07 09:47,AP down, 1...(count)
OTM_SI_ZZZ.2002/01/07 09:37.2002/01/07 09:47,Regulator down, 0...(count)
OTM_SI_ZZZ.2002/01/07 09:37.2002/01/07 09:47,ORBConnector down, 0...(count)
OTM_SI_ZZZ.2002/01/07 09:37.2002/01/07 09:47,Regulator-AP, 0, 0, 0(0(count)
OTM_SI_ZZZ.2002/01/07 09:37.2002/01/07 09:47,TSCDaemon-ClientAP, 0, 0, 0(0(count)
OTM_SI_ZZZ.2002/01/07 09:37.2002/01/07 09:47,TSCDaemon-ServerAP(Ctr), 1, 1, 1(1(count)
OTM_SI_ZZZ.2002/01/07 09:37.2002/01/07 09:47,TSCDaemon-ServerAP(Req), 1, 1, 1(1(count)
OTM_SI_ZZZ.2002/01/07 09:37.2002/01/07 09:47,TSCDaemon-Regulator, 0, 0, 0(0(count)
OTM_SI_ZZZ.2002/01/07 09:37.2002/01/07 09:47,TSCDaemon-ORBConnector, 0, 0, 0(0(count)
OTM_SI_ZZZ.2002/01/07 09:47.2002/01/07 09:57,Request Response Time, 0, 0, 0, 0(0(usec)
OTM_SI_ZZZ.2002/01/07 09:47.2002/01/07 09:57,Request Execution Time, 0, 0, 0, 0(0(usec)
OTM_SI_ZZZ.2002/01/07 09:47.2002/01/07 09:57,Request Time Out, 0...(count)
OTM_SI_ZZZ.2002/01/07 09:47.2002/01/07 09:57,Request Error, 0...(count)
OTM_SI_ZZZ.2002/01/07 09:47.2002/01/07 09:57,Request Response Time(ORB), 0, 0, 0, 0(0(usec)
OTM_SI_ZZZ.2002/01/07 09:47.2002/01/07 09:57,Request Time Out(ORB), 0...(count)
OTM_SI_ZZZ.2002/01/07 09:47.2002/01/07 09:57,Request Error(ORB), 0...(count)
OTM_SI_ZZZ.2002/01/07 09:47.2002/01/07 09:57,Schedule Wait Count, 0, 0, 0, 0(0(count)
OTM_SI_ZZZ.2002/01/07 09:47.2002/01/07 09:57,Schedule Wait Count(HPR), 0, 0, 0, 0(0(count)
OTM_SI_ZZZ.2002/01/07 09:47.2002/01/07 09:57,Schedule Wait Time, 0, 0, 0, 0(0(usec)
OTM_SI_ZZZ.2002/01/07 09:47.2002/01/07 09:57,Request Overflow, 0...(count)
OTM_SI_ZZZ.2002/01/07 09:47.2002/01/07 09:57,Request Overflow(LPR), 0...(count)
OTM_SI_ZZZ.2002/01/07 09:47.2002/01/07 09:57,AP down, 1...(count)
OTM_SI_ZZZ.2002/01/07 09:47.2002/01/07 09:57,Regulator down, 0...(count)
OTM_SI_ZZZ.2002/01/07 09:47.2002/01/07 09:57,ORBConnector down, 0...(count)
OTM_SI_ZZZ.2002/01/07 09:47.2002/01/07 09:57,Regulator-AP, 0, 0, 0(0(count)
OTM_SI_ZZZ.2002/01/07 09:47.2002/01/07 09:57,TSCDaemon-ClientAP, 0, 0, 0(0(count)
OTM_SI_ZZZ.2002/01/07 09:47.2002/01/07 09:57,TSCDaemon-ServerAP(Ctr), 1, 0, 1(1(count)
OTM_SI_ZZZ.2002/01/07 09:47.2002/01/07 09:57,TSCDaemon-ServerAP(Req), 1, 0, 1(1(count)
OTM_SI_ZZZ.2002/01/07 09:47.2002/01/07 09:57,TSCDaemon-Regulator, 0, 0, 0(0(count)
OTM_SI_ZZZ.2002/01/07 09:47.2002/01/07 09:57,TSCDaemon-ORBConnector, 0, 0, 0(0(count)

```

付録 C.3 TSC ルートアクセプタ単位の情報出力例

*** Statistics Information by RootAcceptor ***

TSCDomain :TSCDOMAIN
TSCID :172.17.13.186

** Edit time 2002/03/25 16:52 - 2002/03/25 17:02 **
statistical data : 2002/03/25 16:52:00 - 2002/03/25 16:54:52

RootAcceptor:serviceX

Event	Count	Maximum	Minimum	Average (unit)
Schedule Wait Count	892	1	1	1(count)
Schedule Wait Count(HPR)	892	1	1	1(count)
Schedule Wait Time	892	41241	73	962(usec)
Request Overflow	0	-	-	-(count)
Request Overflow(LPR)	0	-	-	-(count)
TSCDaemon-ServerAP(Req)	-	1	1	1(count)

付録 C.4 TSC ルートアクセプタ単位の情報の CSV 形式での出力例

```

*** Statistics Information by RootAcceptor ***
TSCDomain, TSCID, RootAcceptor, start, end, Event, Count, Maximum, Minimum, Average, unit
TSCDOMAIN.172.17.13.186.serviceX.2002/03/25 16:52.2002/03/25 17:02.Schedule Wait Count(HPR), 892, 1, 1, 1,(count)
TSCDOMAIN.172.17.13.186.serviceX.2002/03/25 16:52.2002/03/25 17:02.Schedule Wait Time, 892, 41241, 73, 962,(usec)
TSCDOMAIN.172.17.13.186.serviceX.2002/03/25 16:52.2002/03/25 17:02.Request Overflow, 0,...(count)
TSCDOMAIN.172.17.13.186.serviceX.2002/03/25 16:52.2002/03/25 17:02.Request Overflow(LPR), 0,...(count)
TSCDOMAIN.172.17.13.186.serviceX.2002/03/25 16:52.2002/03/25 17:02.TSCDaemon-ServerAP(Req), 1, 1, 1,(count)

```

付録 C.5 サーバアプリケーションのオペレーション単位の出力例

```

*** Statistics Information by Operation ***

```

```
TSCDomain :TSCDOMAIN
```

```
TSCID :172.17.13.186
```

```

-----
** Edit time 2002/03/25 16:52 - 2002/03/25 17:02 **

```

```

statistical data : 2002/03/25 16:53:16 - 2002/03/25 16:53:57
-----

```

```
Interface:ABC
```

```
Operation:call
```

```

-----
Event          Count  Maximum  Minimum  Average (unit)
Kind of Request
  Synchronized    890      -      -      -(count)
  Oneway           0        -      -      -(count)
Request Response Time    503    88227    4071    16046(usec)
Request Execution Time   890    62292     63     498(usec)
Request Time Out         2        -      -      -(count)
Request Error            0        -      -      -(count)
Request ResponseTime(ORB) 386    82432    4155    12765(usec)
Request Time Out(ORB)     0        -      -      -(count)
Request Error(ORB)       0        -      -      -(count)

```

付録 C.6 サーバアプリケーションのオペレーション単位の CSV 形式での出力例

```

*** Statistics Information by Operation ***
TSCDomain, TSCID, Interface, Operation, start, end, Event, Count, Maximum, Minimum, Average, unit
TSCDOMAIN.172.17.13.186.ABC.call.2002/03/25 16:52.2002/03/25 17:02.Kind of Request(Synchronized), 890...(count)
TSCDOMAIN.172.17.13.186.ABC.call.2002/03/25 16:52.2002/03/25 17:02.Kind of Request(Oneway), 0...(count)
TSCDOMAIN.172.17.13.186.ABC.call.2002/03/25 16:52.2002/03/25 17:02.Request Response Time, 503, 88227, 4071, 16046.(usec)
TSCDOMAIN.172.17.13.186.ABC.call.2002/03/25 16:52.2002/03/25 17:02.Request Execution Time, 890, 62292, 63, 498.(usec)
TSCDOMAIN.172.17.13.186.ABC.call.2002/03/25 16:52.2002/03/25 17:02.Request Time Out, 2...(count)
TSCDOMAIN.172.17.13.186.ABC.call.2002/03/25 16:52.2002/03/25 17:02.Request Error, 0...(count)
TSCDOMAIN.172.17.13.186.ABC.call.2002/03/25 16:52.2002/03/25 17:02.Request Response Time(ORB), 386, 82432, 4155, 12765.(usec)
TSCDOMAIN.172.17.13.186.ABC.call.2002/03/25 16:52.2002/03/25 17:02.Request Time Out(ORB), 0...(count)
TSCDOMAIN.172.17.13.186.ABC.call.2002/03/25 16:52.2002/03/25 17:02.Request Error(ORB), 0...(count)

```

付録 C.7 稼働統計情報ファイルに対象レコードがない場合の出力例

ここでは、ある時間帯の対象レコードが稼働統計情報ファイルにない場合の出力例を示します。次に示す場合に、レコードがない状態になります。

- tscststart コマンドを実行していない場合
- tscststart コマンドの実行後、tscststop コマンドを実行して稼働統計情報の出力を終了したが、情報が取得されなかった場合
- tscststart コマンドを実行して稼働統計情報を取得したが、サーバアプリケーションおよびクライアントアプリケーションを実行していないために、情報が取得されなかった場合（ただし、TSC ルートアクセプタ単位、またはサーバアプリケーションのオペレーション単位に取得した場合に限ります）

レコードがない時間帯には、`*** No Applicable data ***` が出力されます。

次に示す例では、17時25分から17時26分までの時間帯と、17時29分から17時30分までの時間帯にはレコードがあり、17時26分から17時29分までの時間帯にはレコードがない稼働統計情報ファイルを使用するものとします。この稼働統計情報ファイルを1分間隔で編集した場合の出力例です。

*** Statistics Information by RootAcceptor ***

TSCDomain :TSCDOMAIN

TSCID :172.17.13.186

** Edit time 2002/03/25 17:25 - 2002/03/25 17:26 **

statistical data : 2002/03/25 17:25:00 - 2002/03/25 17:25:59

RootAcceptor:RootAcceptor1

Event	Count	Maximum	Minimum	Average (unit)
Schedule Wait Count	1066	3	1	1(count)
Schedule Wait Count(HPR)	59	1	1	1(count)
Schedule Wait Time	1066	404553	131	14884(usec)
Request Overflow	0	-	-	-(count)
Request Overflow(LPR)	0	-	-	-(count)
TSCDaemon-ServerAP(Req)	-	1	1	1(count)

RootAcceptor:RootAcceptor2

Event	Count	Maximum	Minimum	Average (unit)
Schedule Wait Count	1067	3	1	1(count)
Schedule Wait Count(HPR)	59	1	1	1(count)
Schedule Wait Time	1067	299783	122	16637(usec)
Request Overflow	0	-	-	-(count)
Request Overflow(LPR)	0	-	-	-(count)
TSCDaemon-ServerAP(Req)	-	1	1	1(count)

RootAcceptor:RootAcceptor12

Event	Count	Maximum	Minimum	Average (unit)
Schedule Wait Count	1068	4	1	1(count)
Schedule Wait Count(HPR)	59	1	1	1(count)
Schedule Wait Time	1068	249915	124	17752(usec)
Request Overflow	0	-	-	-(count)
Request Overflow(LPR)	0	-	-	-(count)
TSCDaemon-ServerAP(Req)	-	1	1	1(count)

** Edit time 2002/03/25 17:26 - 2002/03/25 17:27 **

*** No Applicable data ***

** Edit time 2002/03/25 17:27 - 2002/03/25 17:28 **

*** No Applicable data ***

** Edit time 2002/03/25 17:28 - 2002/03/25 17:29 **

*** No Applicable data ***

付録 C.8 稼働統計情報の出力を何度か繰り返した場合の出力例

ここでは、次に示す条件で稼働統計情報ファイルを編集する場合の出力例を示します。

- 次の三つの時間帯で、tscststart コマンドと tscststop コマンドを繰り返し実行します。

1. 2002/3/25 17:21:00 ~ 2002/3/25 17:25:59

2. 2002/3/25 17:29:00 ~ 2002/3/25 17:31:59
3. 2002/3/25 17:38:00 ~ 2002/3/25 17:39:59

- tscstsed コマンドの -TSCTime オプションで、編集開始時刻と終了時刻を 2002 年 3 月 25 日の 17 時 21 分から翌日の 17 時 21 分までと指定します。このときの -TSCTime オプションの指定方法は次のとおりです。

```
-TSCTime 172103252002 172103252002
```

- tscstsed コマンドの -TSCInterval オプションで、編集・出力の間隔を 24 時間と指定します。このときの -TSCInterval オプションの指定方法は次のとおりです。

```
-TSCInterval 1440
```

*** Statistics Information by TSCNode ***

TSCDomain :TSCDOMAIN

TSCID :172.17.13.186

** Edit time 2002/03/25 17:21 - 2002/03/26 17:21 **

statistical data : 2002/03/25 17:21:00 - 2002/03/25 17:25:59

 : 2002/03/25 17:29:00 - 2002/03/25 17:31:59

 : 2002/03/25 17:38:00 - 2002/03/25 17:39:59

no data : 2002/03/25 17:25:59 - 2002/03/25 17:29:00

 : 2002/03/25 17:31:59 - 2002/03/25 17:38:00

 } ※1
 } ※2

Event	Count	Maximum	Minimum	Average (unit)
Request Response Time	38900	2345686	12216	47026(usec)
Request Execution Time	38895	259371	467	1529(usec)
Request Time Out	0	-	-	-(count)
Request Error	0	-	-	-(count)
Request ResponseTime(ORB)	29800	2191614	12468	42765(usec)
Request Time Out(ORB)	0	-	-	-(count)
Request Error(ORB)	0	-	-	-(count)
Schedule Wait Count	38905	4	1	1(count)
Schedule Wait Count(HPR)	59	1	1	1(count)
Schedule Wait Time	38903	404553	116	6842(usec)
Request Overflow	0	-	-	-(count)
Request Overflow(LPR)	0	-	-	-(count)
AP down	0	-	-	-(count)
Regulator down	0	-	-	-(count)
ORBConnector down	0	-	-	-(count)
Regulator-AP	-	2	0	1(count)
TSCDaemon-ClientAP	-	2	1	2(count)
TSCDaemon-ServerAP(Ctr)	-	11	11	11(count)
TSCDaemon-ServerAP(Req)	-	3	3	3(count)
TSCDaemon-Regulator	-	1	1	1(count)
TSCDaemon-ORBConnector	-	1	1	1(count)

注 1

編集レコードのある時間帯です。

注 2

編集レコードのない時間帯です。

付録 D 例外変換テーブルファイルの作成方法

TSCORB コネクタ経由で接続している ORB クライアントに、OTM のシステム例外を通知するときは、OTM のシステム例外を CORBA 例外に変換して通知します。

ここでは、TSCORB コネクタの開始時に `tscstartgw` コマンドの `-TSCExceptConvertFile` オプションに指定する例外変換テーブルファイルの作成方法を示します。ただし、例外変換テーブルファイルは、デフォルトの例外変換以外の方法で変換する場合にだけ使用します。

(1) 例外変換テーブルファイルの形式

```
<OTMt○ORB>
変換前の例外コード=変換後の例外コード
</OTMt○ORB>
```

(2) 例外変換テーブルファイルの記述項目

OTMt○ORB

OTM の `TSCSystemException` を CORBA 例外に変換する方式であることを示します。例外変換テーブルファイルでは、開始タグ (`<OTMt○ORB>`) と終了タグ (`</OTMt○ORB>`) で、変換する例外コードを挟んで記述します。

変換前の例外コード

~ < 符号なし整数, アスタリスク (*), およびコンマ (,) > ((1 ~ 26))

変換前の例外 (OTM の `TSCSystemException`) に対応する例外コードを指定します。すべての例外コードを同一の例外コードに変換する場合は、アスタリスク (*) を指定します。変換前の例外コードを複数指定する場合は、区切り文字としてコンマ (,) を使用します。各例外に対応する例外コードを次の表に示します。

表 D-1 各例外に対応する例外コード

例外	対応する例外コード	例外	対応する例外コード
BAD_PARAM	1	OBJECT_NOT_EXIST	14
NO_MEMORY	2	UNKNOWN	15
COMM_FAILURE	3	INV_OBJREF	16
NO_PERMISSION	4	IMP_LIMIT	17
INTERNAL	5	BAD_TYPECODE	18
MARSHAL	6	PERSIST_STORE	19
INITIALIZE	7	FREE_MEM	20
NO_IMPLEMENT	8	INV_IDENT	21
BAD_OPERATION	9	INV_FLAG	22
NO_RESOURCES	10	INTF_REPOS	23

例外	対応する例外コード	例外	対応する例外コード
NO_RESPONSE	11	BAD_CONTEXT	24
BAD_INV_ORDER	12	OBJ_ADAPTER	25
TRANSIENT	13	DATA_CONVERSION	26

変換後の例外コード

～ < 符号なし整数 > ((1 ~ 26))

変換後の例外 (CORBA 例外) に対応する例外コードを指定します。各例外に対応する例外コードは、表 D-1 に示した値と同じです。

(3) 例外変換テーブルファイルの注意事項

複数の例外に対する変換方法を指定する場合は、" 変換前の例外コード = 変換後の例外コード " の行を、改行して繰り返し指定します。

コメント行は、<!-- --> のタグで囲んで記述してください。

変換前の例外コードと等号 (=) の間、および等号 (=) と変換後の例外コードの間に空白やタブは記述できません。

例外変換の指定に誤りがあった場合は、例外変換テーブルファイルは使用できません。

同じ値の変換前の例外コードを複数回指定すると、最後に記述された指定内容が有効となります。

変換前の例外コードにアスタリスク (*) を記述すると、すべての OTM 例外が同一の CORBA 例外に変換されます。

同じ値の変換前の例外コードを複数回指定すると、あとに記述された指定内容が有効となるため、アスタリスク (*) を記述する前の指定は無効となってしまいます。したがって、アスタリスク (*) を使用する場合は、例外変換テーブルファイルの最初の行に記述することをお勧めします。

コンマ (,) を使用して複数の変換前の例外コードを指定した場合、その中にアスタリスク (*) が含まれていると、アスタリスク (*) だけを指定した場合と同様の結果となります。

< > で囲んだタグの中に空白やタブを記述すると、空白も含めてタグ名称として認識されます。つまり、< OTMtoORB> (: 空白) などと記述すると、例外変換のタグとして認識されません。また、コメントの開始タグ (<!--) や終了タグ (</) の途中にも空白やタブなどを記述しないでください。その場合は、コメントの開始・終了タグと認識されません。

開始タグ (<OTMtoORB>) と終了タグ (</OTMtoORB>) の間に記述できるタグは、コメントのタグだけです。また、<OTMtoORB> と </OTMtoORB> の範囲外には、空行 (空白、タブだけの行も含む) とコメントだけを記述できます。

ただし、OTMtoORB 以外のタグを記述しても、開始タグと終了タグが正しく記述してある場合はエラーにはならないで、無視されます。この場合、開始タグに対応する終了タグが記述されていない場合などはエラーになります。

タグを記述する行では、タグ開始文字 (<) の前、およびタグ終了文字 (>) のあとには、空白とタブだけを記述できます。

ASCII 文字だけを使用できます。ただし、コメントの中では、その他の文字も使用できます。

例外変換テーブルファイルから作成した例外変換テーブルは、TSCORB コネクタのプロセス内で使用され、例外変換処理が行われます。なお、例外変換テーブルは、一つの TSCORB コネクタのプロセスに対して、一つだけ使用できます。

(4) 例外変換テーブルファイルの記述例

ここでは、次に示すように OTM の TSCSystemException を CORBA 例外に変換する場合の例外変換テーブルファイルの記述例を示します。

- COMM_FAILURE, INTERNAL, および INITIALIZE の例外を, OBJ_ADAPTER に変換します。
- TRANSIENT の例外を INITIALIZE に変換します。
- COMM_FAILURE, INTERNAL, INITIALIZE, および TRANSIENT 以外の例外を, すべて UNKNOWN に変換します。

```
<!-- OTM例外をCORBA例外に変換する -->
<!-- すべての例外をUNKNOWNに変換する -->
<OTMtoORB>
*=15

<!-- COMM_FAILURE, INTERNAL, およびINITIALIZEの例外を -->
<!-- OBJ_ADAPTERに変換する -->
3, 5, 7=25

<!-- TRANSIENTの例外をINITIALIZEに変換する -->
13=7
</OTMtoORB>
```

索引

記号

-TSCAPName 180
-TSCCheckInterval 78, 201
-TSCClientConnectCount 158
-TSCCsv 74, 190
-TSCDebugInfo 88, 115, 188, 211
-TSCDispatchPolicy 34, 168
-TSCEdit 74, 189
-TSCHighPriorBorder 35, 170
-TSCHighPriorRequestCount 35, 169
-TSCIDLFile 69, 175
-TSCIORConnect 23, 70, 158, 176
-TSCLogFileCount 167
-TSCLogFileSize 167
-TSCMaxRequestCount 34, 169
-TSCParallelCount 57, 214
-TSCRebindInterval 16, 209
-TSCRebindTimes 16, 209, 216
-TSCRetryReference 23, 28, 210, 217
-TSCRetryWay 24, 26, 28, 210, 217
-TSCSendHost 48, 112
-TSCSendHostInterval 48, 112
-TSCSendInterval 46, 111
-TSCServerConnectCount 167
-TSCSessionInterval 212, 218
-TSCStatsFileCount 73, 170
-TSCStatsUse 72, 170
-TSCTimeOutRebind 209, 216
-TSCTimeOutRebind オプション指定時に
オートリバインド機能が動作する例外 17
-TSCWatchMethod 67, 216

A

API 優先方式 23, 218
API 優先方式の場合の接続対象 25

C

CLASSPATH 85, 93
CORBA オブジェクトの名称 176

CORBA に準拠するクライアントとの接続 5

I

IDL ファイル 175
IOR 文字列 15, 22, 27, 70, 127
IOR 文字列の取得 117
IOR 文字列ファイルの出力先ディレクトリ
119
IOR 文字列ファイルの名称 119
IP アドレス 27, 127, 128, 140

J

jar ファイルのバージョン情報の出力 133

O

ORB クライアント 9
ORB クライアントから OTM に接続する仕
組み 70
ORB クライアントへの例外の通知 71
osagent 9, 15, 48, 70
OTM 9
OTM - Client 9
OTM - Client で取得できる障害情報 88
OTM - Client の運用 83
OTM - Client の開始 86
OTM - Client の開始と終了 86
OTM - Client の環境設定 84
OTM - Client の環境設定の流れ 84
OTM - Client の環境変数 84
OTM - Client の終了 86
OTM - Client の障害対策 88
OTM - Connector for ORB 11, 69
OTM - Connector for ORB の運用 100
OTM 実行環境ディレクトリ 85, 93
OTM で取得できる障害情報 104
OTM の運用 91
OTM の開始 97
OTM の開始と終了 97
OTM の環境設定 92

OTM の環境設定の流れ 92

OTM の環境変数 92

OTM の終了 98

OTM の障害対策 104

P

PATH 85, 93

PP 形名 206

T

TPBroker 9, 11

TPBroker OTM の概要 2

TPBroker との連携 102

TPBroker についてのトラブルシューティング情報 155

TPBroker のスマートエージェント 15, 48, 70

TSCAdm クラスの get_status() メソッド 21, 67

TSCAdm クラスの getTSCClient() メソッド 23, 28

TSCClient オブジェクトでの接続の保持 29

TSCDIR 85, 93

tscdminfo 108

tscdmstart 110

tscdmstop 115

tscgetior 71, 117

tscgetpid 121

tscgetref 123

tscgetref コマンドの出力例 350

tscholdra 130

tscjver 133

tscllogcat 135

tsclscon 137

tsclsra 144

TSCORB コネクタ 11

TSCORB コネクタ異常終了の件数 77, 194

TSCORB コネクタ経由の接続の仕組み 69

TSCORB コネクタ数 140

TSCORB コネクタと TSC デーモンの対応 71

TSCORB コネクタの異常終了時の再起動 101

TSCORB コネクタの開始 100, 172

TSCORB コネクタの開始処理の待ち時間 174

TSCORB コネクタの開始と終了 100

TSCORB コネクタの機能 69

TSCORB コネクタの終了 101, 185

TSCORB コネクタの状態の検出 101

TSCORB コネクタのリクエスト障害 76, 193

TSCORB コネクタのリクエストタイムアウト 76, 193

TSCORB コネクタのレスポンス時間 76, 193

tscprcls 101, 151

tsclasget 155

tscregld 157

tsclidinfo 160

tsclsr 162

TSCRootAcceptor クラスの activate() メソッド 57

TSCRootAcceptor クラスの

setParallelCount() メソッド 57

TSCSPOOL 85, 93, 135

tsctest 164

tsctestgw 101, 172

tsctestpre 54, 179

tsctest 183

tsctestgw 101, 185

tsctestpre 54, 187

tsctestsed 189

tsctestsed コマンドのコマンドオプションの組み合わせ 191

tsctestsed コマンドの出力例 356

tscteststart 72, 201

tscteststop 72, 203

tsctest 205

TSCWatchTime クラス 67

TSC アクセプタ名称 177, 215

TSC アクセプタ名称を指定しないリクエストの振り分け 62

- TSC アクセプタ名称を指定するリクエストの振り分け 63
- TSC 識別子 10, 27, 127, 128
- TSC システムプロセス 41
- TSC スプールディレクトリ 155
- TSC デモン 10
- TSC デモンが開始した日時 140
- TSC デモンが管理するプロセス 41
- TSC デモン管理最大数 110
- TSC デモンに登録できる TSC ルートアクセプタ数 168
- TSC デモンによるサーバアプリケーションの登録と管理の流れ 43
- TSC デモンの TSC 識別子 140
- TSC デモンの TSC ドメイン名称 140
- TSC デモンの開始 164
- TSC デモンの開始時刻 154
- TSC デモンの開始処理の待ち時間 167
- TSC デモンの機能 34
- TSC デモンの終了 183
- TSC デモンを強制終了 183
- TSC ドメイン 10
- TSC ドメイン管理ポート番号 110
- TSC ドメイン情報の出力 160
- TSC ドメイン情報の表示と削除 108
- TSC ドメイン情報を削除 108
- TSC ドメインマネージャ 10
- TSC ドメインマネージャの開始 110
- TSC ドメインマネージャの開始形態 46
- TSC ドメインマネージャの機能 46
- TSC ドメインマネージャの終了 115
- TSC ドメインマネージャを強制終了 115
- TSC ドメインマネージャを強制的に正常開始 113
- TSC ドメイン名称 10, 27, 127, 128
- TSC ドメインログ 135
- TSC ノード 10
- TSC ノード間での通信 46
- TSC ノード情報 46
- TSC ノード情報の送信間隔 48, 111, 112
- TSC ノード情報の配布 48
- TSC ノード全体でまとめた稼働統計出力情報 78
- TSC ノード全体でまとめた情報の CSV 形式での出力例 358
- TSC ノード全体でまとめた情報の出力例 357
- TSC ノードとの接続 15
- TSC ノードとのリバインド 16
- TSC ユーザアクセプタ 56
- TSC ユーザオブジェクト 55
- TSC ユーザオブジェクトの呼び出し 35
- TSC ユーザプロキシ 62
- TSC ルートアクセプタ 56
- TSC ルートアクセプタキュー保留時間 169
- TSC ルートアクセプタ情報の出力 144
- TSC ルートアクセプタ数 168
- TSC ルートアクセプタ単位の稼働統計出力情報 80
- TSC ルートアクセプタ単位の情報の CSV 形式での出力例 359
- TSC ルートアクセプタ単位の情報の出力例 358
- TSC ルートアクセプタ登録名称 214
- TSC ルートアクセプタの状態 146
- TSC ルートアクセプタの閉塞 130
- TSC ルートアクセプタの閉塞解除 162
- TSC ルートアクセプタの閉塞処理の待ち時間 131
- TSC ルートアクセプタを一括して閉塞 131
- TSC ルートアクセプタを一括して閉塞解除 162
- TSC レギュレータ 10
- TSC レギュレータ異常終了の件数 77, 194
- TSC レギュレータ数 140
- TSC レギュレータと TSC デモンの対応 32
- TSC レギュレータの IOR 接続 158
- TSC レギュレータの IOR 文字列 158
- TSC レギュレータの開始 157
- TSC レギュレータの機能 31
- TSC レギュレータの動作モード 32
- TSC レギュレータのプロセス数 166
- TSC ログ 135
- TSC ログファイル最大サイズ 167
- TSC ログファイル最大数 167

あ

アプリケーション識別子
122, 138, 141, 153, 172, 179, 185, 187, 213
アプリケーションプログラム異常終了の件数
77, 194
アプリケーションプログラムの管理 42
アポートコード一覧 348

い

異常検知 5
インストール先ディレクトリ 85, 93

う

運用コマンドの一覧 106
運用支援機能のプロセス監視 121

お

応答時間の監視 5
オートリバインド機能 16
オートリバインド機能が動作する例外 16
オートリバインド機能の設定 16

か

稼働統計出力の出力形式 74
稼働統計出力の出力情報 75
稼働統計出力ファイル 72
稼働統計情報 72
稼働統計情報が取得されていない時間帯 193
稼働統計情報が取得されている時間帯 193
稼働統計情報取得時のオペレーション名の取り扱い 81
稼働統計情報の概要 72
稼働統計情報の出力形式 192
稼働統計情報の出力を何度か繰り返した場合の出力例 361
稼働統計情報の取得の開始 201
稼働統計情報の取得の終了 203
稼働統計情報のファイル運用 73
稼働統計情報の編集と出力 189
稼働統計情報ファイル 72

稼働統計情報ファイルに対象レコードがない場合の出力例 360

稼働統計情報ファイルの格納ディレクトリ
73

稼働統計情報ファイルのサイズ 170

稼働統計情報ファイルの世代切り替え 74

稼働統計情報ファイルの世代数 170

稼働統計情報ファイルのバックアップファイル名 74

稼働統計情報ファイルのファイル名 73

稼働統計情報を取得するかどうかを指定 170

稼働統計情報を編集出力する時間の間隔 189

カレントディレクトリ ID 152

環境変数による TSC ドメインの管理 93

管理情報を引き継いで TSC ドメインマネージャを自動強制開始 113

き

キューを共有しないスケジュール方法 61

キューを共有するスケジュール方法 59

共通メッセージの取得を抑止 155

く

クライアントアプリケーション 9

クライアントアプリケーション数 140

クライアントアプリケーションの開始 207

クライアントアプリケーションの機能 14

クライアントアプリケーションの状態の検出
21

クライアント通信用バッファ数 208, 215

クライアントの構成 9

クラス監視時間 216

こ

高プライオリティしきい値 170

高プライオリティリクエスト数 169

コネクション情報 78

コネクション情報の出力 137

コネクション情報を取得する時間の間隔 202

コネクション数 140, 158, 177

コネクションの集約 4

コネクションのレギュレート 31

コピーライト 206
 コマンドオプションファイル名称 166

さ

サーバアプリケーション 11
 サーバアプリケーション異常終了時のリクエスト保持 39
 サーバアプリケーションからのリクエストの要求 55
 サーバアプリケーション数 140
 サーバアプリケーションのオペレーション単位の CSV 形式での出力例 359
 サーバアプリケーションのオペレーション単位の稼働統計出力情報 80
 サーバアプリケーションのオペレーション単位の出力例 359
 サーバアプリケーションの開始
 54, 179, 213
 サーバアプリケーションの開始処理の待ち時間 180
 サーバアプリケーションの開始と終了 54
 サーバアプリケーションの機能 54
 サーバアプリケーションの時間監視 66
 サーバアプリケーションの実行形式ファイル名称 180
 サーバアプリケーションの終了 54, 187
 サーバアプリケーションの状態の検出 67
 サーバアプリケーションの閉塞制御 38
 サーバアプリケーションを強制終了 187
 サーバオブジェクトの常駐化 3
 サーバキャッシュサイズ 168
 サーバの構成 9
 サーバの動的なメンテナンス 5
 サービス閉塞 38
 サブネットマスク 112

し

システム構成 7
 システム構成例 49
 システムの構成 8
 実行形式ファイルおよびライブラリのバージョン情報を出力 205

実行形式ファイル名称 153, 180
 順次方式 23, 218
 障害の局所化 5
 常駐スレッド数 214

す

スケラブルな負荷分散 4
 スケジュール情報 76
 スケジュールできなかったリクエスト数
 77, 194
 スケジュールできなかったリクエスト数（低
 プライオリティ）77
 スケジュールポリシー 34, 168
 スケジュール待ち件数 76, 193
 スケジュール待ち件数（高プライオリティ）
 77
 スケジュール待ち時間 77, 194
 スケジュール用キューの閉塞制御 38
 スケジュール用キューを共有しない方法 59
 スケジュール用キューを共有する方法 59
 スマートエージェントの設定 48
 スレッド 55
 スレッド ID 222
 スレッドと TSC ユーザオブジェクトの構成
 56
 スレッドの最大値 169
 スレッドの並行実行 57
 スレッドパーセション（ThreadSession）
 158
 スレッドプーリング（ThreadPool）158

せ

製品名称 206
 セッション開始メソッド発行数 76, 193
 セッション型呼び出し回数 199
 セッション情報 76
 セッション情報の削除 45
 セッション数 193
 セッション呼び出しインターバル監視時間
 212, 218
 セッション呼び出しインターバル時間監視 44
 セッション呼び出し機能 19

セッション呼び出し時のリクエストの振り分け
65

接続および再接続の実行 28

接続クライアントの最大数 140, 167

接続経路 15, 27, 127, 128, 208, 215

接続サーバアプリケーションの最大数
140, 167

接続先情報の取得と編集 123

接続先情報ファイル 27

接続先情報ファイルの URL 210

接続先情報ファイルの形式 27

接続先情報ファイルのソートキー 128

接続先情報ファイル名称 126, 210, 217

接続先情報ファイルを使用する接続 22

接続先情報ファイルをマージして出力する例
353

接続先情報を出力する例 352

接続先情報をソートして出力する例 352

接続する TSC ノード 16

接続日時 141

接続ピーク値 140

接続ピーク値が記録された日時 140

接続方法 15, 27, 127, 128

た

多様なプログラミング言語への対応 5

と

同期型呼び出し回数 199

トラブルシューティング情報の取得と出力 155

トレース情報 211

は

バージョンおよびリビジョン 206

バージョン情報の出力 205

パラレルカウント 57, 214

ひ

非応答型呼び出し回数 199

ビルド OS 206

ビルドコンパイラ 206

ビルド日付 206

ふ

ファイル検索方式 23, 218

ファイル検索方式の場合の接続対象 24

負荷分散 35

複数のネットワークセグメントで構成する

TSC ドメイン 48

プライオリティ 35

プライオリティ値 174, 208, 215

プロセス ID 138, 140, 152, 222

プロセス ID の取得 121

プロセス監視定義に記述する項目 102

プロセス種別 141, 152

プロセス情報 77

プロセス情報の出力 151

プロセス数の範囲 12

プロセスステータス 153

プロセスの開始時刻 152

プロセスの管理 41

プロセスの配置 40

分散処理システムの仕組み 2

へ

閉塞制御 38

ほ

ポート番号 27, 127, 128, 141

ま

マルチスレッド環境での処理 57

マルチノードリトライ順序 23, 218

マルチノードリトライ接続 22

マルチノードリトライ接続での接続方式の組
み合わせ 23

マルチノードリトライ接続の接続対象 24

マルチノードリトライ接続の接続方式
210, 217

マルチノードリトライ接続の動作 22

マルチノードリトライ方式 23, 218

め

メソッド監視時間 216
 メソッドの実行時間の監視 5
 メッセージ ID 記号の意味 223
 メッセージ一覧 224
 メッセージの記述形式 222
 メッセージの形式 222
 メッセージの出力形式 222
 メッセージの出力先種別 223
 メッセージの種類 223
 メッセージの編集と出力 135

ゆ

ユーザ環境変数定義ファイル 94
 ユーザ環境変数定義ファイルの記述形式 95
 ユーザ環境変数定義ファイル名称
 166, 174, 180
 ユーザ定義 IDL インタフェースの例 56

ら

ライブラリ格納ディレクトリを指定する環境
 変数 85, 93
 ランダム方式 23, 218

り

リクエスト実行時間 75, 193
 リクエスト種別 75
 リクエスト障害 76, 193
 リクエスト情報 75
 リクエスト数 169
 リクエストタイムアウト 76, 193
 リクエストの形態 14
 リクエストの処理 34
 リクエストのスケジューリング 2
 リクエストのスケジュール 59
 リクエストの登録 34
 リクエストの負荷分散 35
 リクエストの振り分け 34, 62
 リクエストの閉塞制御 38
 リクエストの流量制御 37
 リバインド回数 16, 209, 216

リバインド間隔 16, 209, 216
 流量制御 37

れ

例外変換テーブルファイル 71, 174
 例外変換テーブルファイルの作成方法 363
 レスポンス時間 75, 193

ろ

ログメッセージ 211

(株) 日和 出版センター 行き

FAX 番号 0120-210-454 (フリーダイヤル)

日立マニュアル注文書

ご注文日	年 月 日
送付先ご住所	〒
お客様名 (団体名, 又は法人名など)	
お名前	
電話番号	()
FAX 番号	()

資料番号	マニュアル名	数量
合計		

マニュアルのご注文について、ご不明な点は
(株) 日和 出版センター (☎03-5281-5054) へお問い合わせください。