

# FUJITSU Software

## Interstage Application Server/ Interstage Web Server Express

### トラブルシューティング集

Windows/Solaris/Linux

B1WS-1040-04Z0(00)  
2014年2月

# まえがき

---

## 本書の目的

本書は、Interstage Application Server製品を使用中に発生しうる既知のトラブルについて、事例・現象と解決方法を説明しています。

## 前提知識

本書を読む場合、以下の知識が必要です。

- ・ インターネットに関する基本的な知識
- ・ オブジェクト指向技術に関する基本的な知識
- ・ 分散オブジェクト技術(CORBA)に関する基本的な知識
- ・ リレーショナルデータベースに関する基本的な知識
- ・ 使用するOSに関する基本的な知識

## 本書の構成

本書は以下の構成になっています。

### 第1章 障害調査資料の採取

障害調査のために採取される資料について説明しています。

### 第2章 インストール・アンインストール時の異常

インストール・アンインストール中に発生しうる問題の現象と対処方法を説明しています。

### 第3章 Interstage初期化時の異常

Interstage Application Serverの初期化処理中に発生しうる問題の現象と対処方法を説明しています。

### 第4章 Interstage管理コンソール操作時の異常

Interstage管理コンソールの操作中に発生しうる問題の現象と対処方法を説明しています。

### 第5章 Interstage運用時の異常

Interstage運用中に発生しうる問題の現象と対処方法を説明しています。

### 第6章 マルチサーバ管理の異常 **Windows32/64** **Solaris32** **Linux32/64**

マルチサーバ運用中に発生しうる問題の現象と対処方法を説明しています。

### 第7章 Interstageリストア／移入時の異常

Interstageのリストア／移入時に発生しうる問題の現象と対処方法を説明しています。

### 第8章 Interstage HTTP Server運用時の異常

Interstage HTTP Serverの運用中に発生しうる問題の現象と対処方法を説明しています。

### 第9章 J2EEアプリケーション開発・運用時の異常

J2EEアプリケーションの開発や運用中に発生しうる問題の現象と対処方法を説明しています。

### 第10章 Java EE使用時の異常

Java EEアプリケーションの運用時に発生するトラブルの解決方法について説明しています。

### 第11章 CORBAサービス使用時の異常

CORBAサービスの使用に際して、発生しうる問題の現象と対処方法を説明しています。

### 第12章 コンポーネントトランザクションサービス使用時の異常

コンポーネントトランザクションサービスの使用に際して、発生しうる問題の現象と対処方法を説明しています。

### 第13章 データベース連携サービス使用時の異常 **Windows32/64** **Solaris32** **Linux32/64**

データベース連携サービスの使用に際して、発生しうる問題の現象と対処方法を説明しています。

### 第14章 イベントサービス運用時の異常

イベントサービスの運用中に発生しうる問題の現象と対処方法を説明しています。

#### 第15章 クラスタサービス機能使用時の異常

クラスタサービス機能の使用に際して、発生しうる問題の現象と対処方法を説明しています。

#### 第16章 Interstage シングル・サインオン運用時の異常

Interstage シングル・サインオンのトラブル発生時の対処方法について説明しています。

#### 第17章 MessageQueueDirector運用時の異常

MessageQueueDirector運用時に発生するトラブルの対処方法について説明しています。

#### 第18章 Interstage ディレクトリサービス使用時の異常

Interstage ディレクトリサービス使用時に発生しうる問題の現象と対処方法を説明しています。

#### 第19章 Java実行環境運用時の異常

Java実行環境において運用時に発生する例外と対処方法を説明しています。

#### 第20章 Interstage data store使用時の異常

Interstage data store使用時にトラブルが発生した場合の対処方法について説明します。

#### 付録A Javaツール機能

Javaツール機能について説明しています。

## 製品の表記について

本マニュアルでの以下の表記については、それぞれの基本ソフトウェアに対応した製品を示しています。

表記	説明
RHEL5(x86)	Red Hat Enterprise Linux 5 (for x86)を前提基本ソフトウェアとした本製品
RHEL5(Intel64)	Red Hat Enterprise Linux 5 (for Intel64)を前提基本ソフトウェアとした本製品
RHEL6(x86)	Red Hat Enterprise Linux 6 (for x86)を前提基本ソフトウェアとした本製品
RHEL6(Intel64)	Red Hat Enterprise Linux 6 (for Intel64)を前提基本ソフトウェアとした本製品

## 輸出許可

本ドキュメントには、「外国為替および外国貿易管理法」に基づく特定技術が含まれています。したがって本ドキュメントを非居住者に提供する場合には、経済産業大臣の許可が必要となる場合がありますので、ご注意ください。

## 著作権

Copyright 2002-2014 FUJITSU LIMITED

2014年2月 第4版  
2012年8月 初版

# 目次

第1章 障害調査資料の採取	1
1.1 一括情報採取ツール	1
1.2 グローバルトランザクション連携時のダンプの採取	2
1.3 CORBAサービスのトレース情報の採取	4
1.3.1 トレース情報の内容と採取コマンド	4
1.3.2 イベント種別と採取タイミング	5
1.3.3 トレース情報採取の前準備	10
1.3.4 採取手順	12
1.3.5 トレース情報のテキスト出力	12
1.3.6 トレース情報の分析	14
1.4 CORBAサービスのログ情報の採取	15
1.4.1 アクセスログのデータ	16
1.4.2 プロセスログのデータ	17
1.4.3 エラーログ	19
1.4.4 インフォメーションログ	20
1.4.5 アクセスログ採取レベル(access_log_levelパラメタ)	21
1.4.6 ログ採取のためのアプリケーション開発	26
1.4.7 ログ採取環境(configファイルの設定)	27
1.4.8 ログ採取環境の動的変更(odcntllogコマンド)	29
1.4.9 出力結果の見方	30
1.5 CORBAサービスのスナップショットの採取	30
1.5.1 機能概要	30
1.5.2 環境設定	30
1.5.3 運用	31
1.5.3.1 スナップ情報の採取	32
1.5.3.2 スナップ情報の出力形式	33
1.5.3.3 スナップ情報の分析	34
1.5.3.4 スナップ情報採取状況の確認	36
1.6 CORBAサービスのIPCログの採取	36
1.7 CORBAサービスのネーミングサービスのユーザ例外ログの採取	39
1.7.1 ログのデータ	39
1.7.2 ログ採取環境(nsconfigファイルの設定)	41
1.8 クラッシュダンプ・コアダンプの採取	41
1.8.1 Windows (R) におけるクラッシュダンプの採取	41
1.8.2 Solarisにおけるコアダンプの採取	43
1.8.3 Linuxにおけるコアダンプの採取	43
1.9 Interstage/ワークユニット停止時の調査資料の自動採取	44
第2章 インストール・アンインストール時の異常	46
2.1 共通事項	46
2.2 アンインストールと管理(ミドルウェア)	48
2.3 機能の追加と削除	49
2.4 サーバマシン起動時の異常	50
2.5 アンインストール直後の起動時の異常	51
第3章 Interstage初期化時の異常	52
3.1 Interstage初期化時	52
第4章 Interstage管理コンソール操作中の異常	54
4.1 Interstage管理コンソール起動時の異常	54
4.2 ログイン時の異常	57
4.3 Interstage管理コンソールの統計情報の異常	59
4.4 画面の異常	59
4.5 エラーログ/エラーコードの出力	63
4.6 その他の異常	63

第5章 Interstage運用時の異常.....	72
5.1 共通事項.....	72
5.2 Interstageの起動/停止時の異常.....	75
5.2.1 Java EE機能/Webサーバの使用時.....	75
5.2.2 マルチ言語サービス/J2EE互換機能の使用時.....	75
5.2.3 Windows(R)マシンの再起動/シャットダウン時.....	78
5.3 業務システム運用時の異常.....	78
5.4 Visual Studioで作成したアプリケーションが動作しない場合.....	79
5.5 デスクトップヒープが枯渇した場合.....	80
第6章 マルチサーバ管理の異常.....	81
6.1 トラブル解決のための調査方法.....	81
6.2 メッセージの見方.....	81
6.3 環境構築編.....	82
6.4 運用編.....	83
6.5 管理サーバに異常が発生した場合.....	84
第7章 Interstageリストア/移入時の異常.....	86
7.1 CORBAサービスのリストア/移入.....	86
7.2 コンポーネントトランザクションサービスのリストア/移入.....	87
7.3 Interstage シングル・サインオンのリストア/移入.....	88
7.4 JJSERVERクラスタ資源のリストア/移入.....	89
第8章 Interstage HTTP Server運用時の異常.....	92
8.1 Webサーバの起動/停止時の異常.....	92
8.2 Webサーバの運用時の異常.....	94
8.3 よくある質問とその対処方法.....	98
第9章 J2EEアプリケーション開発・運用時の異常.....	101
第10章 Java EE使用時の異常.....	102
10.1 コマンド実行時の異常.....	102
10.1.1 初期化コマンドに関する異常.....	103
10.2 運用中の異常.....	103
10.2.1 運用環境に関する異常.....	103
10.2.2 Java EEアプリケーションで文字化けが発生する場合の対処.....	111
10.2.3 アプリケーションの運用に関する異常.....	113
10.2.4 Dependency Injection失敗時の異常.....	114
10.2.5 Windows Server(R) 2003でnonpaged poolがリークする.....	115
10.2.6 大規模ファイル使用時の異常.....	116
10.2.7 SSL通信ができない.....	116
10.2.8 Interstageディレクトリサービスへ接続できない.....	116
10.2.9 認証に失敗する旨のメッセージが出力される場合.....	117
10.2.10 追加プロパティに関する異常.....	118
10.2.11 jersey利用時の異常.....	119
10.3 Webアプリケーションの開発・運用時の異常.....	120
10.4 EJBアプリケーションの開発・運用時の異常.....	123
10.5 EJBディスパッチ機能の異常.....	123
10.6 Interstage Java EE管理コンソール操作時の異常.....	123
10.6.1 Interstage Java EE管理コンソール起動時の異常.....	123
10.6.2 ログイン時の異常.....	125
10.6.3 画面の異常.....	125
10.6.4 その他の異常.....	129
10.7 Webサービス・Webサービスクライアントの開発・運用時の異常.....	131
10.8 JMSの開発・運用時の異常.....	131
10.9 スレッドダンプが出力された場合の対処.....	134
10.10 Java EEセキュリティの異常.....	138
10.11 ログの異常.....	138

10.12 アンインストール時の異常	138
<b>第11章 CORBAサービス使用時の異常</b>	<b>140</b>
11.1 CORBAサービス運用中のシステム例外発生	140
11.2 クライアントへのレスポンスが悪い	143
11.3 アプリケーションが停止しない	143
11.4 アプリケーションがエラーとなる	144
11.5 CORBAワークユニットの標準出力/標準エラー出力がファイルに出力されない	145
11.6 インスタンス保持機能が正常に動作しない	145
11.7 アプリケーションコンパイル時の異常	146
<b>第12章 コンポーネントランザクションサービス使用時の異常</b>	<b>147</b>
12.1 アプリケーション処理要求時の異常発生時の対処	147
12.2 アプリケーション作成時の異常発生時の対処	149
12.3 ワークユニット使用時の異常発生時の対処	149
12.3.1 ワークユニット起動失敗	149
12.3.2 ワークユニット起動コマンド無応答	151
12.3.3 ワークユニット停止コマンド無応答	152
12.3.4 ワークユニットが異常終了	152
12.3.5 サーバアプリケーション無応答	154
12.3.6 コンソールからログインできない	155
12.3.7 コマンドの実行が失敗する	155
12.3.8 Interstageに対する操作が無応答となる	156
12.4 性能監視ツール使用時の異常発生時の対処	156
12.4.1 性能情報が採取できない	156
12.4.2 性能情報のリアルタイム監視ができない	158
12.5 Interstage運用APIの異常発生時の対処	158
12.6 EXTPのアンインストールが失敗する	159
12.7 セッション情報管理機能を使用した場合の異常への対処	159
<b>第13章 データベース連携サービス使用時の異常</b>	<b>160</b>
13.1 アプリケーション運用中の異常	160
13.2 アプリケーション作成時の異常	162
13.3 マシンのシステムダウン	162
13.4 OTSシステムの異常	163
13.5 リソース管理プログラムの異常	164
13.6 OTSシステムとリソース管理プログラムの異常	166
13.7 Oracle使用時の異常	166
13.8 OTS動作環境の異常	167
13.9 Interstage管理コンソール使用時の異常	168
<b>第14章 イベントサービス運用時の異常</b>	<b>170</b>
14.1 コマンド実行時の異常	170
14.2 アプリケーション運用中の異常	170
14.3 アプリケーション作成時の異常	172
14.4 クラスタサービス運用中の異常	172
<b>第15章 クラスタサービス機能使用時の異常</b>	<b>174</b>
15.1 クラスタサービスの起動が失敗する	174
15.2 クラスタサービスの切り替えが失敗する	174
<b>第16章 Interstage シングル・サインオン運用時の異常</b>	<b>175</b>
16.1 トラブル発生時の対処	175
16.1.1 トラブル発生から対処までの流れ	175
16.1.2 ログの出力先	176
16.2 トラブル事例	177
16.2.1 認証に関するトラブル	178
16.2.2 POSTリクエストに対する認証に関するトラブル	180
16.2.3 統合Windows認証に関するトラブル	182

16.2.4	業務サーバの認可に関するトラブル	183
16.2.5	リポジトリサーバに関するトラブル	184
16.2.6	ユーザ情報を登録するディレクトリサービスにActive Directoryを使用する場合のトラブル	185
16.2.7	Microsoft(R) Internet Information Servicesに関するトラブル	187
16.2.8	Webブラウザに関するトラブル	187
16.2.9	認証サーバ間連携サービスに関するトラブル	187
16.2.10	Interstage管理コンソールに関するトラブル	190
16.2.11	Interstage ディレクトリサービスのレプリケーション機能使用時のトラブル	191
16.2.12	旧バージョンからの移行に関するトラブル	191
<b>第17章</b>	<b>MessageQueueDirector運用時の異常</b>	<b>193</b>
17.1	イベントチャンネル連携サービス運用時の異常	193
17.2	アプリケーション実行時の異常	196
17.3	コマンド実行時の異常	198
<b>第18章</b>	<b>Interstage ディレクトリサービス使用時の異常</b>	<b>199</b>
18.1	リポジトリを起動できない	199
18.2	LDAPコマンドが正常に動作しない	199
18.3	リポジトリの応答が遅い	200
18.4	リポジトリの応答がない	201
18.5	トップエントリを操作できない	201
18.6	JNDIアプリケーションで例外が発生する	201
18.7	登録データが文字化けしている	202
18.8	アクセスログを読めない	203
18.9	データファイルを読めない	203
18.10	クラスタ運用時の異常	203
18.11	アプリケーションの移行時の異常	203
18.12	テーブル作成コマンド実行時にエラーが発生する	204
18.13	テーブル作成コマンド実行時にデータベースのエラーが発生する	204
18.14	テーブル作成コマンドの応答がない	206
18.15	エントリ管理ツールのショートカットキーが使用できない	206
18.16	エントリ管理ツールの文字入力フィールドで、文字を入力できない	207
18.17	アクセス制御が正しく機能しない	207
18.18	スキーマ拡張に関する異常	207
18.19	Active Directoryとの通信に関するトラブル	208
<b>第19章</b>	<b>Java実行環境運用時の異常</b>	<b>209</b>
19.1	FJVMのトラブルシューティング機能	209
<b>第20章</b>	<b>Interstage data store使用時の異常</b>	<b>210</b>
20.1	データストアのイベントログにエラーメッセージが出力される	210
<b>付録A</b>	<b>Javaツール機能</b>	<b>211</b>
A.1	メソッドトレース機能	211
A.1.1	動作原理	211
A.1.2	使用手順	212
A.1.3	格納先	212
A.1.4	制御ファイルの作成方法	213
A.1.5	使用方法	224
A.1.6	トレース情報出力の形式	227
A.1.7	トレース情報の分析	230
A.1.8	トレース不可のクラス	232
A.1.9	メッセージ	242
A.1.10	注意事項	246
A.2	jheap	247
A.3	スレッドダンプツール	250
A.3.1	thdumpコマンドの使用方法	251
A.3.2	thdumpSVCコマンドの使用方法	252

A.3.3 オプション.....	254
A.3.4 サンプルプログラムと出力例.....	255
A.3.5 注意事項.....	257
A.4 JDKに含まれるトラブルシューティングに役立つツール.....	258
A.4.1 注意事項.....	258
A.4.1.1 jinfo使用時の注意事項.....	258
A.4.1.2 jmap使用時の注意事項.....	259
A.4.1.3 jstack使用時の注意事項.....	259
A.4.1.4 jstat使用時の注意事項.....	259
A.4.1.5 jvisualvm使用時の注意事項.....	260
A.4.1.6 jconsole使用時の注意事項.....	263
A.4.1.7 jps使用時の注意事項.....	264
A.4.1.8 jcmd使用時の注意事項.....	264
A.4.1.9 jhat使用時の注意事項.....	264
A.5 Java監視機能.....	265
A.5.1 パフォーマンスデータの収集.....	266
A.5.2 コンソール機能.....	268
A.5.3 jconsoleの起動.....	268
A.5.4 監視対象アプリケーションへの接続.....	269
A.5.5 Java監視機能のタブ.....	271
A.5.6 メソッドサンプリング機能.....	272
A.5.7 ヒープ分析機能.....	281
A.5.8 VMオプションの変更.....	288
A.5.9 接続の解除.....	293
A.5.10 jconsoleの終了.....	295
A.5.11 データ説明.....	295
A.5.12 メッセージ.....	303
A.5.13 IJServer(J2EE)を使用する場合.....	317
A.6 チュートリアル.....	320
A.6.1 前提となる知識.....	321
A.6.2 注意事項.....	321
A.6.3 ガーベジ・コレクション(GC)について分析する.....	321
A.6.3.1 GCログの採取.....	321
A.6.3.2 GCログ分析のポイント.....	322
A.6.3.3 jstatコマンドを使った調査方法.....	323
A.6.4 アプリケーションの性能ボトルネックを調べる.....	324
A.6.4.1 Java VisualVMを使った調査.....	329
A.6.5 メモリリークを分析する.....	330
A.6.5.1 リークしているクラスの特典.....	330
A.6.5.2 jmapコマンドを使った調査方法.....	335
A.6.5.3 ヒープダンプの採取と分析.....	336
A.6.5.4 Java VisualVMを使った調査.....	344
A.6.6 ハングアップやスローダウンの原因を調べる.....	344
A.6.6.1 デッドロックの検出.....	345
A.6.6.2 ハングアップやスローダウンの調査.....	346
索引.....	348

# 第1章 障害調査資料の採取

障害調査のために採取される資料について、以下の内容で説明します。

- ・ 一括情報採取ツール
- ・ グローバルトランザクション連携時のダンプの採取 **Windows32/64** **Solaris32** **Linux32/64**
- ・ CORBAサービスのトレース情報の採取
- ・ CORBAサービスのログ情報の採取
- ・ CORBAサービスのスナップショットの採取
- ・ CORBAサービスのIPCログの採取
- ・ CORBAサービスのネーミングサービスのユーザ例外ログの採取
- ・ クラッシュダンプ・コアダンプの採取
- ・ Interstage/ワークユニット停止時の調査資料の自動採取

## 1.1 一括情報採取ツール

一括情報採取ツールとはInterstage運用中のトラブル発生時に調査資料採取を行うコマンド(iscollectinfo)です。トラブル発生時、技術員連絡を行う前に本コマンドを使用して調査資料の採取を行ってください。iscollectinfoは以下の場所に格納されています。

**Windows32/64**

<Interstageインストールフォルダ>%bin

**Solaris32/64** **Linux32/64**

/opt/FJSVisas/bin

コマンドの仕様については“リファレンスマニュアル(コマンド編)”の“保守情報採取コマンド”にある“iscollectinfo”を参照してください。

### ポイント

FJQSS(資料採取ツール)により、iscollectinfoコマンドと同じ情報を採取できます。

**Windows32/64**

FJQSSについては、以下で表示されるマニュアルを参照してください。

- － 「スタート」メニューの「FJQSS(資料採取ツール)」－「FJQSS ユーザーズガイド」

**Solaris32/64** **Linux32/64**

FJQSSについては、以下に格納されているマニュアルを参照してください。

- － マニュアルパッケージの“FJQSS”フォルダ配下

### 注意

**Windows32/64**

- － Windowsではパス名の長さの制限のために、資料採取に失敗する場合があります。一括情報採取ツールの“調査資料の格納先ディレクトリ”、または、FJQSSの採取資料の“作業フォルダ”には、絶対パス名長で20バイト程度

を目安に、なるべく短いパス名を指定してください。指定方法については、それぞれ以下のマニュアルを参照してください。

- 一括情報採取ツール  
“リファレンスマニュアル(コマンド編)”の“保守情報採取コマンド”にある“iscollectinfo”
- FJQSS  
“FJQSS (資料採取ツール) ユーザーズガイド”

**Solaris32/64** **Linux32/64**

- 採取資料を複写する際は、必ず `cp -pR` コマンドによって複写を実施してください。
- UpdateAdvisorがインストールされた環境で一括情報採取ツールを使って調査資料採取を行う場合、ツールを実行する前に、UpdateAdvisorの利用許諾への同意を行ってください。  
利用許諾への同意を行っていない場合、一括情報採取ツールが停止し、資料が採取できません。  
UpdateAdvisorの利用許諾は、UpdateAdvisorをインストール後、初めて `uam` コマンドを実行したときに表示されます。

**EE** **SJE**

## 1.2 グローバルトランザクション連携時のダンプの採取 **Windows32/64**

**Solaris32** **Linux32/64**

グローバルトランザクション連携時のダンプの採取には、以下の3つの方法があります。

- 自動採取
- 手動一時採取
- 手動常時採取

以下に、それぞれの採取方法について説明します。

### ■自動採取

OTSシステムおよびリソース管理プログラムが異常終了した場合、OTSシステムによって、以下のフォルダ配下に自動的にダンプファイルが採取され、ダンプファイル名が標準エラーに出力されます。

**Windows32/64**

C:\¥Interstage¥ots¥var

**Solaris32**

/opt/FSUNots/var

**Linux32/64**

/opt/FJSVots/var

### ■手動一時採取

`otsgetdump` コマンドを使用して、異常が発生した時点のOTSシステムの障害調査ダンプファイルを採取できます。以下の場合に、手動一時採取を行ってください。

- システムエラーや内部矛盾など、システム管理者に通知する必要があるエラーメッセージが表示された場合
- OTSシステムのプログラムやアプリケーションで異常が発生した場合

本コマンドは、ダンプを出力するマシン上で実行してください。

```
otsgetdump [-all] [-on|-off] [-c] [-s] [-rリソース定義名] [-w] [-e] [-t]
```

## 注意

### Windows32 Solaris32

V3以前のバージョンで使用していたリソース定義ファイルはそのままでは使用することができません。移行方法については、“移行ガイド”の“データベース連携サービスの移行方法”で説明されています。

otsgetdumpコマンドで採取するダンプファイルは、以下のフォルダ配下に格納されます。

### Windows32/64

```
C:¥Interstage¥var
```

### Solaris32

```
/opt/FSUNots/var
```

### Linux32/64

```
/opt/FJSVots/var
```

## ■ 手動常時採取

otsgetdumpコマンドにより、ダンプファイル採取の開始や停止を任意のタイミングで指示できます。ダンプファイルの採取が指示されると、停止要求があるまでダンプファイルを採取します。以下のような場合に、手動常時採取を行います。

## ◆ 再現性のある現象で、確実に障害を調査したい場合

otsgetdumpコマンドの-onオプションを指定して、ダンプファイルを採取します。採取を停止させる場合は、-offオプションを指定してください。

以下に、使用例を示します。

## 例

ダンプの採取を開始します。

```
otsgetdump -on
```

全ダンプを採取します。

```
otsgetdump -all
```

OTSシステムトレースおよびリソース2つのトレースを採取します。

```
otsgetdump -s -r resourcedef1, resourcedef2
```

ダンプの採取を終了します。

```
otsgetdump -off
```



注意

Windows32 Solaris32

V3以前のバージョンで使用していたリソース定義ファイルは、そのまま使用できません。移行方法については、“移行ガイド”の“データベース連携サービスの移行方法”で説明されています。

## 1.3 CORBAサービスのトレース情報の採取

CORBAサービスでは、CORBAアプリケーション運用におけるトラブル発生時など、原因調査のための情報として、動作トレースを採取しています。

以下にCORBAサービスのトレース機能の概要と操作手順について説明します。

### 1.3.1 トレース情報の内容と採取コマンド

CORBAサービスのトレース機能は、アプリケーションの入出力情報をロギングします。ロギングされたトレース情報は、通常運用中は共用メモリ上に採取され、コマンド操作によりファイルに出力することができます。採取したトレース情報は、アプリケーション運用時のトラブルの原因究明に役立てることができます。

採取されるトレース情報と、トレース情報採取コマンドについて説明します。

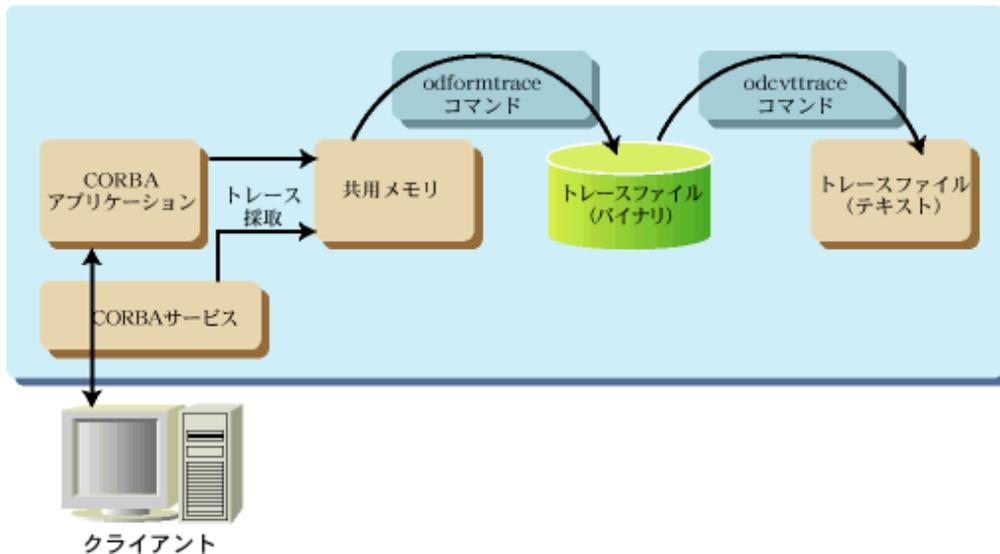
#### ■採取されるトレース情報

CORBAアプリケーションのトレース情報として、以下の動作に対する情報が採取されます。

- ・ サーバアプリケーション活性化時
- ・ コネクション確立時
- ・ リクエストの送信から応答の返信まで
- ・ コネクション切断時
- ・ サーバアプリケーション非活性化時
- ・ エラー発生時(エラー詳細情報)

#### ■トレース情報採取コマンドと生成物

トレース情報採取に使用するコマンドと生成物の関係を以下に示します。



## ■トレース情報採取コマンド

### odformtraceコマンド

odformtraceコマンドは、共用メモリ上に採取されたトレース情報をトレースファイル(バイナリ)に出力します。トレース情報は、プロセスごとに別々のトレースファイルに出力されます。古いファイルはサフィックスが“log”から“old”に変更されて保存されます。

### odcvtraceコマンド

odcvtraceコマンドは、odformtraceコマンドによって出力されたトレースファイルを可読な形式にするため、テキストファイルに変換します。

### odprthdrtraceコマンド

odprthdrtraceコマンドは、トレース情報のプロセスIDとアプリケーションのコマンド実行形式をテキスト形式で出力します。プロセスIDとコマンド実行形式だけを参照する場合は、odcvtraceコマンドを使用せずにodprthdrtraceコマンドを使用します。

## 1.3.2 イベント種別と採取タイミング

トレース情報が採取されるアプリケーションの動作とイベント種別について説明します。

### ■アプリケーション動作と採取場所

アプリケーションの動作と採取場所について以下に示します。

アプリケーションの動作	採取場所	備考
[サーバアプリケーション活性化]	サーバ側(ライブラリ)	BOA_impl_is_readyメソッド(他)の発行
[コネクション確立]	クライアント側(ライブラリ)	
[リクエストの送信から応答の返信まで]	クライアント側(ライブラリ、スタブ) サーバ側(CORBAサービス、ライブラリ、スタブ)	
[コネクション切断]	クライアント側(ライブラリ)	
[サーバアプリケーション非活性化]	サーバ側(ライブラリ)	BOA_deactivate_implメソッド(他)の発行

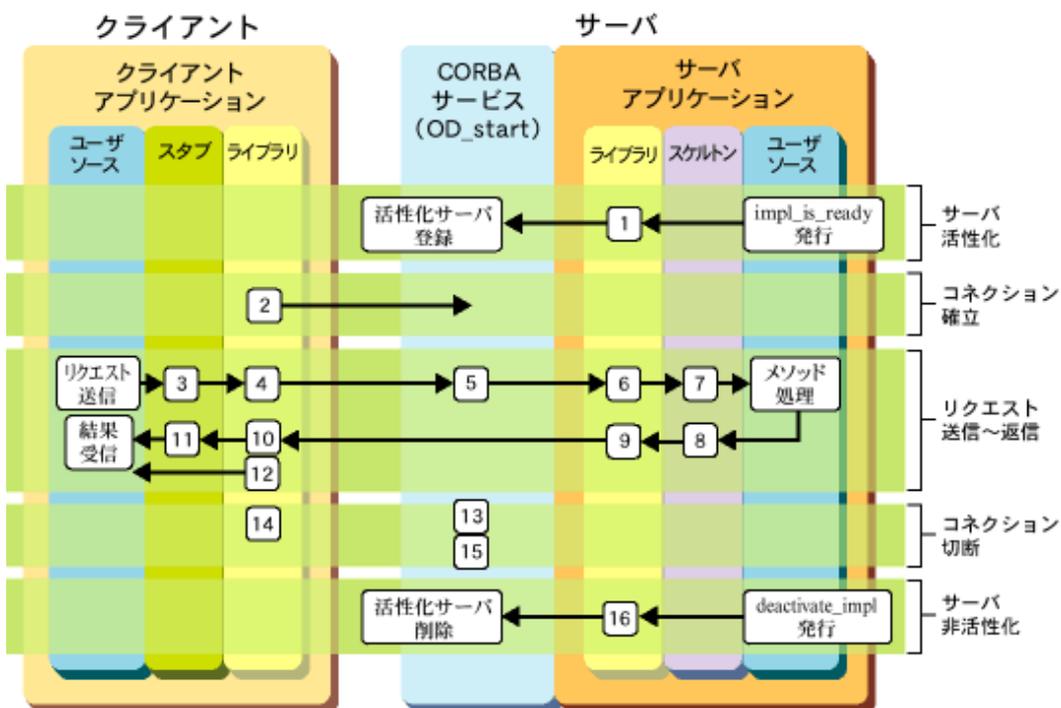
アプリケーションの動作	採取場所	備考
[その他の情報]	クライアント側、サーバ側	

## 注意

スタブ・スケルトンでトレース情報を採取するためには、プログラミング時に静的起動インタフェース・静的スケルトンインタフェースを使用する必要があります。

## ■トレース採取の流れ

アプリケーションの動作と採取場所により、トレースのイベント種別が異なります。イベント種別と採取場所の関係を下図に示します。図中の番号と“■イベント種別”の番号は、一致しています。



## ■イベント種別

### [サーバアプリケーション活性化]

(1) implementation is ready (implementation = %s1)

[可変情報]

%s1:インプリメンテーションリポジトリID

[意味]

インプリメンテーションリポジトリID%s1のサーバアプリケーションが活性状態になりました。

### [コネクション確立]

(2) connected to server (host = %s1, port = %s2)

[可変情報]

%s1:サーバマシン名

%s2:ポート番号

[意味]

クライアントアプリケーションがサーバマシン%s1のポート番号%s2に接続しました。

#### [リクエストの送信から応答の返信まで]

(3) STUB: server method call start (request\_id = %s1, intf\_id = %s2, operation = %s3)

[可変情報]

%s1:リクエストID

%s2:サーバアプリケーションのインタフェースリポジトリID

%s3:サーバメソッド名

[意味]

スタブのサーバメソッド呼出しが開始しました。

**注)**

- 静的起動インタフェースのアプリケーションで出力されます。
- Javaアプリケーションでは出力されません。

(4) send request to server (request\_id = %s1, intf\_id = %s2, host = %s3, operation = %s4)

[可変情報]

%s1:リクエストID

%s2:サーバアプリケーションのインタフェースリポジトリID

%s3:サーバマシン名

%s4:サーバメソッド名

[意味]

リクエストが送信されました。

(4)-1 client: The number of requests on a process exceeded the maximum limit

[意味]

クライアントアプリケーションが同時に送信できるリクエスト数が環境設定で設定された数を超えました(od10969メッセージと同等の情報)。

(4)-2 The number of connections to a server exceeded the maximum limit

[意味]

クライアントアプリケーションからサーバに接続するコネクション数が環境設定で設定した数を超えて接続しようとしてしまいました(od10917メッセージと同等の情報)。

(5) queue request (request\_id = %s1, server pid = %s2, intf\_id = %s3)

[可変情報]

%s1:リクエストID

%s2:リクエストのディスパッチ先サーバプロセスID

%s3:リクエストのディスパッチ先サーバのインタフェースリポジトリID

[意味]

リクエストID%s1のリクエストがCORBAサービス(OD\_startサービス)によって受け付けられ、キューイングされました。  
%s2が0の場合は、空き状態のサーバが存在しなかったためキューイング時にディスパッチ先サーバプロセスが決定されなかったことを示します。

(5)-1 server: The number of requests exceeded the maximum limit

[意味]

サーバマシンにおいて同時に処理できるリクエスト数が環境設定で設定された数を超えました(od10968メッセージと同等の情報)。

(5)-2 The number of connections from clients exceeded the maximum limit

[意味]

サーバでクライアントから受け付けるコネクション数が環境設定で設定した数を超えて、リクエストを受け付けました(od10918メッセージと同等の情報)。

(6) receive request from client (request\_id = %s1, intf\_id = %s2, operation = %s3)

[可変情報]

%s1:リクエストID  
%s2:サーバアプリケーションのインタフェースリポジトリID  
%s3:サーバメソッド名

[意味]

リクエストがサーバアプリケーションに受け付けられました。

(7) SKEL: server method invoke start (request\_id = %s1, intf\_id = %s2, operation = %s3)

[可変情報]

%s1:リクエストID  
%s2:サーバアプリケーションのインタフェースリポジトリID  
%s3:サーバメソッド名

[意味]

スケルトンのサーバメソッド呼出しが開始しました。

注)

- 静的スケルトンインタフェースのアプリケーションで出力されます。
- Javaアプリケーションでは出力されません。

(8) SKEL: server method invoke end (request\_id = %s1, intf\_id = %s2, operation = %s3)

[可変情報]

%s1:リクエストID  
%s2:サーバアプリケーションのインタフェースリポジトリID  
%s3:サーバメソッド名

[意味]

スケルトンのサーバメソッド呼出しが終了しました。

注)

- 静的スケルトンインタフェースのアプリケーションで出力されます。
- Javaアプリケーションでは出力されません。

(9) send reply to client (request\_id = %s1)

[可変情報]

%s1:リクエストID

[意味]

リクエストからの返信がサーバアプリケーションから送り出されました。

(9)-1 send standard exception to client (request\_id = %s1, excep\_id = %s2, minor = 0x%s3)

[可変情報]

%s1:リクエストID  
%s2:標準例外のID  
%s3:マイナーコード

[意味]

標準例外がサーバアプリケーションから送り出されました。

(9)-2 send user exception to client (request\_id = %s1, excep\_id = %s2)

[可変情報]

%s1:リクエストID  
%s2:ユーザ例外のID

[意味]

ユーザ例外がサーバアプリケーションから送り出されました。

(9)-3 fail to send reply to client (request\_id = %s1, from = %s2, intf\_id = %s3, operation = %s4)

[可変情報]

%s1:リクエストID  
%s2:クライアントマシンのIPアドレス  
%s3:サーバアプリケーションのインタフェースリポジトリID  
%s4:サーバメソッド名

[意味]

応答の送信に失敗しました(od10605メッセージと同等の情報)。

(10) receive reply from server (request\_id = %s1)

[可変情報]

%s1:リクエストID

[意味]

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からの返信を受け取りました。

(10)-1 receive standard exception from server (request\_id = %s1, except\_id = %s2, minor = 0x%s3)

[可変情報]

%s1:リクエストID  
%s2:標準例外のID  
%s3:マイナーコード

[意味]

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から送り出された標準例外を受け取りました。

(10)-2 receive user exception from server (request\_id = %s1, except\_id = %s2)

[可変情報]

%s1:リクエストID  
%s2:ユーザ例外のID

[意味]

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から送り出されたユーザ例外を受け取りました。

(11) STUB: server method call end (request\_id = %s1, intf\_id = %s2, operation = %s3)

[可変情報]

%s1:リクエストID  
%s2:サーバアプリケーションのインタフェースリポジトリID  
%s3:サーバメソッド名

[意味]

スタブのサーバメソッド呼出しが終了しました。

**注)**

- 静的起動インタフェースのアプリケーションで出力されます。
- Javaアプリケーションでは出力されません。

(12) connection to server (%s1) time out

[可変情報]

%s1:サーバマシン名

[意味]

クライアントからのリクエストが待機時間(configファイルのperiod\_receive\_timeout)を超えても復帰しなかったため、クライアントでタイムアウトが発生しました(od10925メッセージと同等の情報)。

## [コネクション切断]

(13) connection to client (%s1) was closed

[可変情報]

%s1:クライアントマシンのIPアドレス

[意味]  
クライアントマシンとの接続が閉塞されました。

(14) connection to server (%s1) was closed

[可変情報]  
%s1:サーバホスト名(またはIPアドレス)

[意味]  
サーバマシンとの接続が閉塞されました。

(15) connection to client (%s1) time out

[可変情報]  
%s1:クライアントマシンのIPアドレス

[意味]  
クライアントからのリクエスト送信が無通信監視時間(configファイルのperiod\_idle\_con\_timeout)を超えてもなかったため、サーバでタイムアウトが発生しました。

#### [サーバアプリケーション非活性化]

(16) implementation is deactivated (implementation = %s1)

[可変情報]  
%s1:インプリメンテーションリポジトリID

[意味]  
サーバが非活性状態になりました。

#### [その他の情報]

information : (%s1, %s2): %s3, %s4

[可変情報]  
%s1:ファイル名  
%s2:行番号  
%s3:エラー詳細  
%s4:エラー番号

[意味]  
CORBAサービス(ObjectDirector)の情報メッセージです(od10924メッセージと同等の情報)。

### 1.3.3 トレース情報採取の前準備

---

トレース情報採取のための前準備について説明します。

#### ■インストール時

トレース情報を採取するためには、本製品の“サーバ機能”のCORBAサービスをインストールする必要があります。“クライアント機能”(CORBAサービスクライアント)をインストールしても、トレース機能が有効になりませんので注意してください。

#### ■アプリケーション開発時

##### ◆IDLコンパイル

IDLファイルから生成されるスタブ・スケルトンにロギング機能を組み込むため、IDLコンパイルを行います。

#### 注意

- 本機能はV4.0(以降)のIDLコンパイラで使用できます。
- Javaマッピングではロギング機能は組み込まれません。

- IDLコンパイラ実行時に-nologオプションを指定するとロギング処理は組み込まれなくなりますが、トラブル発生時の原因究明のために-nologオプションを指定しないことを推奨します。

#### ◆プログラミング

静的起動インタフェース・静的スケルトンインタフェースを使用してプログラミングすることにより、スタブ・スケルトンでもトレース情報を採取できます。

静的起動インタフェース・静的スケルトンインタフェースを使用しない場合は、CORBAサービス、およびライブラリのみでトレース情報が採取されます。

#### ◆ライブラリのリンク

トレース情報を採取するためには、アプリケーションにCORBAサービスのサーバ用ライブラリをリンクする必要があります。以下に必要なサーバ用ライブラリを示します。

##### Windows32/64

開発言語	ライブラリ名
C・C++	ODSV.LIB
COBOL (スレッドモード)	ODCOBCBLMTSV.LIB または ODCOBCBLSVUC.LIB
COBOL (プロセスモード)	ODCOBCBLSV.LIB または ODCOBCBLSVUC.LIB
OOCOBOL	ODOOCOBSV.LIB または ODOOCOBSVUC.LIB

##### Solaris32/64 Linux32/64

開発言語	ライブラリ名
C・C++	libOM.so
COBOL (スレッドモード)	libOMcblMT.so または libOMcblUC.so
COBOL (プロセスモード)	libOMcbl.so
OOCOBOL	libOMoocob.so または libOMoocobUC.so

### ■アプリケーション運用時

#### ◆configファイルの設定

トレース情報を採取するためのconfigファイルのパラメタについて説明します。

trace\_use(初期値:yes)

トレース情報を採取するかどうかを指定します。

trace\_size\_per\_process(初期値:10000/バイト)

トレース情報を採取する最大メモリサイズ(バイト:プロセス単位)を指定します。この値は1プロセスごとの最大サイズのため、プロセス数分のメモリ容量を確保する必要があります。

詳細は“◆共用メモリ容量”を参照してください。

trace\_file\_synch\_level(初期値:stop)

トレースファイルへの出力タイミングを指定します。指定可能な値を以下に示します。セパレータとして“&”を使用することで、複数指定することができます。

設定値	意味
none	odformtraceコマンド実行時のみトレースファイルに出力します。
exit	アプリケーション正常終了時に、終了したアプリケーションのトレース情報をトレースファイルに出力します。

設定値	意味
vanish	アプリケーション異常終了時に、終了したアプリケーションのトレース情報をトレースファイルに出力します。
stop	CORBAサービス終了時にすべてのアプリケーションのトレース情報をトレースファイルに出力します。
loop	メモリ上のトレース情報のサイズがtrace_size_per_processを超えた場合にトレースファイルに出力します。

なお、本パラメタにどの値が設定されていても、odformtraceコマンドを実行するとトレース情報が出力されます。

#### ◆共用メモリ容量

トレース情報は、共用メモリに採取されるため、以下のメモリサイズを確保しておく必要があります。  
 なお、トレース情報のサイズがメモリサイズの上限值に達した場合、メモリ上の古い情報に上書きされます。

trace\_size\_per\_process(最大メモリサイズ)(注) × max\_processes(最大プロセス数)(注) + 20Kバイト

(注) configファイルのパラメタです。

### 1.3.4 採取手順

トレース情報を採取する手順を以下に示します。

1. configファイルのtrace\_useパラメタに“yes”が設定されていることを確認します。  
 なお、configファイルを変更した場合は、CORBAサービスを再起動してください。
2. CORBAアプリケーションを起動します。
3. トレース情報は、以下のタイミングでトレースファイルに出力されます。
  - odformtraceコマンド起動時、すべて、または指定されたプロセスのトレース情報を出力します。
  - "trace\_file\_synch\_level = exit"の場合は、アプリケーション正常終了時に終了したアプリケーションのトレース情報を出力します。
  - "trace\_file\_synch\_level = vanish"の場合は、アプリケーション異常終了時に終了したアプリケーションのトレース情報を出力します。
  - "trace\_file\_synch\_level = stop"の場合は、CORBAサービス終了時にすべてのアプリケーションのトレース情報を出力します。
  - "trace\_file\_synch\_level = loop"の場合は、メモリ上のトレース情報のサイズがtrace\_size\_per\_processを超えたときに出力します。
4. トレースファイルを可読な形式にするため、odcvtraceコマンドでテキストファイルに変換します。

### 1.3.5 トレース情報のテキスト出力

トレースファイルをodcvtraceコマンドでテキストファイルに変換した出力形式について説明します。

#### ■出力形式

```

ProcessID   : 362
CommandLine : simple_s

thread      time          event
-----
000000A0 16:27:08.651 implementation is ready (implementation = IDL:test1/intf1:1.0)

```

## ProcessID

トレース情報が採取されたプロセスのプロセスID。

## CommandLine

アプリケーションのコマンド実行形式(引数を含む)。空白を含め最大127文字の文字列が表示されます。

## thread

スレッドID。

## time

トレース情報が採取された時間。「時:分:秒.ミリ秒」で表示されます。

## event

トレース情報に記録されたイベント(“[■ イベント種別](#)”を参照)。

## 備考

出力文字列が127文字で切られてしまい、正しいコマンド実行文が取得できなかった際、プロセスが以下に該当する場合は、記載の方法を使用することでコマンド実行文を特定できます。



### ● プロセスがJServerとして運用しているアプリケーションの場合

Interstage管理コンソールの“ワークユニット:[ワークユニット名]:ログ参照”の“起動情報”にコマンド実行文全てが出力されます。また、以下のログファイルを参照することでも、コマンド実行文全てを確認できます。

#### Windows32/64

J2EE共通ディレクトリ\ijserver\IJServer名\log\[%プロセス通番%\]info.log (デフォルト)

#### Solaris32/64 Linux32/64

J2EE共通ディレクトリ/ijserver/IJServer名/log/[プロセス通番]/info.log (デフォルト)

上記のログの詳細については、“J2EE ユーザーズガイド(旧版互換)”の“J2EEアプリケーションが運用される環境(IJServer)” - “IJServerのファイル構成”を参照してください。

### ● プロセスがCORBAワークユニットとして運用しているアプリケーションの場合

Interstage管理コンソールの“ワークユニット:[ワークユニット名]:[インプリメンテーションリポジトリID]:環境設定”において、対象のワークユニットの“起動パラメタ”を確認してください。対象プロセスのワークユニット名、インプリメンテーションリポジトリIDが分からない場合は、islistapprocコマンドを実行し、出力結果からワークユニット名とインプリメンテーションリポジトリIDを特定してください。

または、ワークユニット定義の“Param for Executable File”(アプリケーション起動時に渡すパラメタ)を参照してください。ワークユニット定義を特定するには、islistapprocコマンドを実行し、出力結果より対象のプロセスのワークユニット名を確認してください。このワークユニット名をワークユニット定義の“Name”と比較し、一致した定義が対象プロセスのワークユニット定義になります。

islistapprocコマンドの詳細については、“リファレンスマニュアル(コマンド編)”を参照してください。

### ● プロセスがJavaアプレットを使用した運用を行っている場合

HTMLファイルの“PARAM NAME”の設定値を確認することで、パラメタの特定が可能です。

### ● プロセスをバッチファイル、コマンドプロンプト、シェルなどから起動した場合

バッチファイル、コマンドプロンプト、シェルを確認してください。



#### Solaris32/64 Linux32/64

CORBA\_ORB\_init()(C言語の場合)の引数に誤りがある場合、コマンド実行形式(CommandLine)は正しく表示されません。CORBA\_ORB\_init()への引数の渡し方は、“リファレンスマニュアル(API編)”の各言語のインタフェースを参照してください。ただし、Java言語の場合は、org.omg.CORBA.ORB.init()の引数に関係なく、正しく表示されない場合があります。

す。表示されたプロセスID(ProcessID)から、psコマンド、または備考に記載した方法を使用して正しいコマンド実行形式を取得してください。

### 1.3.6 トレース情報の分析

ここではクライアントアプリケーション、サーバアプリケーション(ネーミングサービス)、OD\_startサービスがすべて同一マシン“hostABC”にある場合のトレース情報について説明します。クライアントマシンのIPアドレスは「10.34.111.222」と仮定します。

#### ■クライアントアプリケーションのトレース情報

```
ProcessID   : 176
CommandLine : simple_c

thread      time          event
-----
00000140 16:27:10.684 connect to server (host = hostABC, port = 8002)
00000140 16:27:10.684 send request to serverd_request (request_id = 1,
                               intf_id = IDL:CosNaming/NamingContextExt:1.0, host = hostABC, operation = resolve)
000000A5 16:27:10.694 receive reply from server (request_id = 1)
00000140 16:27:10.714 connection to server (hostABC) was closed
```

クライアントアプリケーションのプロセスIDは176で、コマンドラインは“simple\_c”です。

1. 時刻「16:27:10.684」にサーバマシン“hostABC”のポート8002番に接続しています。
2. 時刻「16:27:10.684」にリクエストID1のリクエストを送信しています。サーバアプリケーションのインタフェースリポジトリIDは“IDL:CosNaming/Naming Context:1.0”で、サーバメソッド名は“resolve”です。
3. 時刻「16:27:10.646」にクライアントアプリケーションはリクエストID1のリクエストに対する返信をサーバアプリケーションから受け取っています。
4. 時刻「16:27:10.714」にクライアントアプリケーションはサーバマシン“hostABC”に対する接続を閉塞しています。

#### ■OD\_startサービスのトレース情報

Windows32/64

```
ProcessID   : 339
CommandLine : odstart.exe

thread      time          event
-----
00000094 16:26:57.925 implementation is ready (implementation = IDL:OM_ORB/admin:1.0)
0000008C 16:27:10.684 queue request (request_id = 1, server pid = 111,
                               intf_id = IDL:CosNaming/NamingContextExt:1.0)
00000097 16:27:12.947 connection to client (010.034.111.222) was closed
```

Solaris32/64 Linux32/64

```
ProcessID   : 339
CommandLine : OD_start

thread      time          event
-----
00000094 16:26:57.925 implementation is ready (implementation = IDL:OM_ORB/admin:1.0)
0000008C 16:27:10.684 queue request (request_id = 1, server pid = 111,
```

```
intf_id = IDL:CosNaming/NamingContextExt:1.0)
00000097 16:27:12.947 connection to client (010.034.111.222) was closed
```

OD\_startサービスのプロセスIDは339です。

1. 時刻「16:26:57.925」にインプリメンテーションリポジトリID“IDL:OM\_ORB/admin:1.0”で活性化しています。
2. 時刻「16:27:10.684」にOD\_startサービスは、クライアントアプリケーションからのリクエスト(リクエストIDは1)を受け付けてキューイングしています。このリクエストは、プロセスID1、インタフェースリポジトリID“IDL:/CosNaming/NamingContextExt:1.0”(ネーミングサービス)にディスパッチされています。
3. クライアント(10.34.111.222)に対する接続が閉塞されました。

## ■ネーミングサービスのトレース情報

Windows32/64

```
ProcessID : 111
CommandLine : D:\WINNT\system32\Naming.exe

thread      time          event
-----
0000009A 16:26:59.277 implementation is ready (
                    implementation = IDL:CosNaming/NamingContext:1.0)
00000169 16:27:10.684 receive request from client (request_id = 1,
                    intf_id = IDL:CosNaming/NamingContextExt:1.0, operation = resolve)
00000169 16:27:10.684 send reply to client (request_id = 1)
```

Solaris32/64 Linux32/64

```
ProcessID : 111
CommandLine : CosNaming_s

thread      time          event
-----
0000009A 16:26:59.277 implementation is ready (
                    implementation = IDL:CosNaming/NamingContext:1.0)
00000169 16:27:10.684 receive request from client (request_id = 1,
                    intf_id = IDL:CosNaming/NamingContextExt:1.0, operation = resolve)
00000169 16:27:10.684 send reply to client (request_id = 1)
```

ネーミングサービスのプロセスIDは111です。

1. 時刻「16:26:59.277」にインプリメンテーションリポジトリID“IDL:CosNaming/NamingContext”で活性化しています。
2. 時刻「16:27:10.684」にサーバアプリケーションは、リクエストID1のリクエストを受け付けています。インタフェースリポジトリID“IDL:CosNaming/NamingContextExt:1.0”のメソッド“resolve”が呼び出されています。
3. 時刻「16:27:10.684」にサーバアプリケーションは、リクエストID1のリクエストに対する返信をクライアントアプリケーションに返しています。

## 1.4 CORBAサービスのログ情報の採取

CORBAサービスは、CORBAアプリケーションの動作状態をログ採取することができます。

CORBAサービスのログには「アクセスログ」、「プロセスログ」、「エラーログ」、「インフォメーションログ」の4種類があり、それぞれ、CORBAアプリケーションのメソッド発行によるアクセスログ、プロセスの起動・停止状態、エラー情報、正常系の情報が採取されます。採取されたデータはログファイルに出力されます。

採取されたログにより、運用中のトラブル発生時の原因究明に役立てることができます。

CORBAサービスの4種類のログについて、以下に説明します。

## アクセスログ

CORBAアプリケーションのメソッド動作中のログ(送受信処理のトレース)がサーバ側で採取されます。何らかの原因によりクライアント/サーバ間のメソッド送受信処理が中断・停止してしまった場合に、どの処理まで正常に行われているか確認することができます。

## プロセスログ

CORBAアプリケーションの起動時・終了時に、そのプロセスIDとコマンド実行文が採取されます。エラーなどが発生した場合に、プロセスの特定が容易になります。

## エラーログ

CORBAアプリケーションの技術員向け内部情報が採取されます。

## インフォメーションログ

CORBAアプリケーションの技術員向け内部情報が採取されます。



ログ情報資料を採取する際は、cpコマンド等によって複製してください。mvコマンド等によって移動したりファイル名を変更した場合、CORBAサービスを再起動するまでログ情報が出力されなくなる場合があります。

## 1.4.1 アクセスログのデータ

### ■ログファイル

アクセスログは、以下のファイルに出力されます。

#### ◆ファイル名

**Windows32/64**

log\_file\_path(configファイル)未指定時(デフォルト)

```
<CORBAサービスインストールパス>%var%accesslog、accesslog.old
```

log\_file\_path(configファイル)指定時

```
<log_file_pathで指定したパス>%accesslog、accesslog.old
```



Windows(R)クライアント用ライブラリ(ODWIN.DLL)をリンクしたアプリケーションは、アクセスログ採取の対象外となります。

**Solaris32/64** **Linux32/64**

log\_file\_path(configファイル)未指定時(デフォルト)

```
<CORBAサービスインストールパス>/var/accesslog、accesslog.old
```

log\_file\_path(configファイル)指定時

```
<log_file_pathで指定したパス>/accesslog、accesslog.old
```

#### ◆ファイルサイズ

ログファイルが生成されると、最大で以下のディスク領域が必要となります。ログ採取を行う場合には、十分な領域を確保してください。

access\_log\_size値(configファイル) × 2 [バイト]

configファイルの各パラメタについては、“[1.4.7 ログ採取環境\(configファイルの設定\)](#)”を参照してください。

#### ■データ形式

アクセスログのデータは以下の形式で出力されます。

プロセスID/スレッドID 時刻 メッセージ(可変情報)

##### プロセスID/スレッドID

ログ採取の対象となるメソッドのプロセス・スレッドのID

##### 時刻

ログが採取された時刻

##### メッセージ(可変情報)

アクセスログ採取レベルに対応するメッセージが表示されます。詳細は“[1.4.5 アクセスログ採取レベル\(access\\_log\\_levelパラメタ\)](#)”を参照してください。

#### 出力例

アクセスログの出力例を以下に示します。

```
764/00000360 Wed Oct 31 16:18:12.132 ObjectDirector started
764/00000578 Thu Nov 01 21:09:24.033 send standard exception to client
(request_id = 2, excep_id = IDL:CORBA/StExcep/NO_IMPLEMENT:1.0, minor = 0x464a0080)
1840/000007A4 Thu Nov 01 21:09:24.033 receive standard exception from server
(request_id = 2, excep_id = IDL:CORBA/StExcep/NO_IMPLEMENT:1.0, minor = 0x464a0880)
```

## 1.4.2 プロセスログのデータ

#### ■ログファイル

プロセスログは以下のファイルに出力されます。

#### ◆ファイル名

 Windows32/64

log\_file\_path(configファイル)未指定時(デフォルト)

```
サーバ用ライブラリ(ODSV.DLL)をリンクしたアプリケーションの場合
<CORBAサービスインストールパス>%var%proclog、proclog.old
クライアント用ライブラリ(ODWIN.DLL)をリンクしたアプリケーションの場合
<CORBAサービスインストールパス>%var%proclogcl、proclogcl.old
```

log\_file\_path(configファイル)指定時

```
サーバ用ライブラリ(ODSV.DLL)をリンクしたアプリケーションの場合
<log_file_pathで指定したパス>%proclog、proclog.old
```

クライアント用ライブラリ(ODWIN.DLL)をリンクしたアプリケーションの場合  
<log\_file\_pathで指定したパス>%proclogcl、proclogcl.old

**Solaris32/64** **Linux32/64**

log\_file\_path(configファイル)未指定時(デフォルト)

<CORBAサービスインストールパス>/var/proclog、proclog.old

log\_file\_path(configファイル)指定時

<log\_file\_pathで指定したパス>/proclog、proclog.old

#### ◆ファイルサイズ

ログファイルが生成されると、最大で以下のディスク領域が必要となります。ログ採取を行う場合には、十分な領域を確保してください。

**Windows32/64**

process\_log\_size値(configファイル) × 4 [バイト]

**Solaris32/64** **Linux32/64**

process\_log\_size値(configファイル) × 2 [バイト]

configファイルの各パラメタについては、“[1.4.7 ログ採取環境\(configファイルの設定\)](#)”を参照してください。

#### ■データ形式

プロセスログのデータは以下の形式で出力されます。

**CORBAサービス(OD\_startプロセス)の場合**

時刻 ObjectDirector started (プロセスID)

**CORBAサービス以外のプロセスの場合**

時刻 起動/停止 [プロセスID] コマンド実行文

##### 時刻

ログ採取された時刻

##### 起動/停止

“A”はコマンド(プロセス)の起動時、“D”は停止時のログであることを示します。

##### プロセスID

ログ採取されたコマンドのプロセスID

##### コマンド実行文

コマンドの実行形式(引数を含む)

出力される文字列長の上限值は、以下のとおりです。

- Windows(R) : 128バイト
- Solaris : 127バイト

•Linux : 127バイト  
コマンド実行文の文字列長が上記の上限値を越えている場合は、出力文字列は上限値までで切られます。

**注)** Solaris、Linuxの場合、上限値未満で切られる場合があります。

## 出力例

プロセスログの出力例を以下に示します。

```
Fri Oct 05 20:23:21.203 ObjectDirector started (1156)
Fri Oct 05 20:23:26.310 A [ 1848] simple_s
Fri Oct 05 20:23:29.705 A [ 1820] simple_c
Fri Oct 05 20:23:34.532 D [ 1820] simple_c
Fri Oct 05 20:23:40.921 D [ 1848] simple_s
Fri Oct 05 20:23:46.299 A [ 2084] OD_impl_inst -ax def
Fri Oct 05 20:23:46.339 D [ 2084] OD_impl_inst -ax def
```

## 備考

出力文字列が上限値で切られてしまい、正しいコマンド実行文が取得できなかった際、プロセスが以下に該当する場合は、記載の方法を使用することでコマンド実行文を特定できます。



### ●プロセスがJServerとして運用しているアプリケーションの場合

Interstage管理コンソールの“ワークユニット:[ワークユニット名]:ログ参照”の“起動情報”にコマンド実行文全てが出力されます。また、以下のログファイルを参照することでも、コマンド実行文全てを確認できます。

**Windows32/64**

J2EE共通ディレクトリ¥ijserver¥IJServer名¥log¥[プロセス通番]¥info.log (デフォルト)

**Solaris32/64** **Linux32/64**

J2EE共通ディレクトリ/ijserver/IJServer名/log/[プロセス通番]/info.log (デフォルト)

上記のログの詳細については、“J2EE ユーザーズガイド(旧版互換)”の“J2EEアプリケーションが運用される環境(IJServer)” - “IJServerのファイル構成”を参照してください。

### ●プロセスがCORBAワークユニットとして運用しているアプリケーションの場合

Interstage管理コンソールの“ワークユニット:[ワークユニット名]:[インプリメンテーションリポジトリID]:環境設定”において、対象のワークユニットの“起動パラメタ”を確認してください。対象プロセスのワークユニット名、インプリメンテーションリポジトリIDが分からない場合は、islistaplprocコマンドを実行し、出力結果からワークユニット名とインプリメンテーションリポジトリIDを特定してください。

または、ワークユニット定義の“Param for Executable File”(アプリケーション起動時に渡すパラメタ)を参照してください。ワークユニット定義を特定するには、islistaplprocコマンドを実行し、出力結果より対象のプロセスのワークユニット名を確認してください。このワークユニット名をワークユニット定義の“Name”と比較し、一致した定義が対象プロセスのワークユニット定義になります。

islistaplprocコマンドの詳細については、“リファレンスマニュアル(コマンド編)”を参照してください。

### ●プロセスがJavaアプレットを使用した運用を行っている場合

HTMLファイルの“PARAM NAME”の設定値を確認することで、パラメタの特定が可能です。

### ●プロセスをバッチファイル、コマンドプロンプト、シェルなどから起動した場合

バッチファイル、コマンドプロンプト、シェルを確認してください。

## 1.4.3 エラーログ

## ■ログファイル

エラーログは以下のファイルに出力されます。

### ◆ファイル名

Windows32/64

log\_file\_path(configファイル)未指定時(デフォルト)

サーバ用ライブラリ(ODSV.DLL)をリンクしたアプリケーションの場合  
<CORBAサービスインストールパス>%var%errlog、errlog.old  
クライアント用ライブラリ(ODWIN.DLL)をリンクしたアプリケーションの場合  
<CORBAサービスインストールパス>%var%errlogcl、errlogcl.old

log\_file\_path(configファイル)指定時

サーバ用ライブラリ(ODSV.DLL)をリンクしたアプリケーションの場合  
<log\_file\_pathで指定したパス>%errlog、errlog.old  
クライアント用ライブラリ(ODWIN.DLL)をリンクしたアプリケーションの場合  
<log\_file\_pathで指定したパス>%errlogcl、errlogcl.old

Solaris32/64 Linux32/64

log\_file\_path(configファイル)未指定時(デフォルト)

<CORBAサービスインストールパス>/var/errlog、errlog.old

log\_file\_path(configファイル)指定時

<log\_file\_pathで指定したパス>/errlog、errlog.old

### ◆ファイルサイズ

ログファイルが生成されると、最大で以下のディスク領域が必要となります。ログ採取を行う場合には、十分な領域を確保してください。

Windows32/64

error\_log\_size値(configファイル) × 4 [バイト]

Solaris32/64 Linux32/64

error\_log\_size値(configファイル) × 2 [バイト]

configファイルの各パラメタについては、“[1.4.7 ログ採取環境\(configファイルの設定\)](#)”を参照してください。

## ■エラーログのデータ

エラーログのデータとして、技術員向け内部情報が出力されます。トラブル発生時に調査を行う場合は、アクセスログ・プロセスログ・インフォメーションログとともに技術員に提供してください。

## 1.4.4 インフォメーションログ

### ■ログファイル

インフォメーションログは以下のファイルに出力されます。

### ◆ファイル名

Windows32/64

#### log\_file\_path(configファイル)未指定時(デフォルト)

サーバ用ライブラリ(ODSV.DLL)をリンクしたアプリケーションの場合  
<CORBAサービスインストールパス>%var%infolog、infolog.old  
クライアント用ライブラリ(ODWIN.DLL)をリンクしたアプリケーションの場合  
<CORBAサービスインストールパス>%var%infologcl、infologcl.old

#### log\_file\_path(configファイル)指定時

サーバ用ライブラリ(ODSV.DLL)をリンクしたアプリケーションの場合  
<log\_file\_pathで指定したパス>%infolog、infolog.old  
クライアント用ライブラリ(ODWIN.DLL)をリンクしたアプリケーションの場合  
<log\_file\_pathで指定したパス>%infologcl、infologcl.old

**Solaris32/64** **Linux32/64**

#### log\_file\_path(configファイル)未指定時(デフォルト)

<CORBAサービスインストールパス>/var/infolog、infolog.old

#### log\_file\_path(configファイル)指定時

<log\_file\_pathで指定したパス>/infolog、infolog.old

### ◆ファイルサイズ

ログファイルが生成されると、最大で以下のディスク領域が必要となります。ログ採取を行う場合には、十分な領域を確保してください。

**Windows32/64**

info\_log\_size値(configファイル) × 4 [バイト]

**Solaris32/64** **Linux32/64**

info\_log\_size値(configファイル) × 2 [バイト]

configファイルの各パラメタについては、“[1.4.7 ログ採取環境\(configファイルの設定\)](#)”を参照してください。

### ■インフォメーションログのデータ

インフォメーションログのデータとして、技術員向け内部情報が出力されます。トラブル発生時に調査を行う場合は、アクセスログ・プロセスログ・エラーログとともに技術員に提供してください。

## 1.4.5 アクセスログ採取レベル(access\_log\_levelパラメタ)

アクセスログの採取レベルとそのキーワード、およびメッセージについて説明します。

### ポイント

.....  
アクセスログ採取レベルのデフォルト設定 (“access\_log\_level=send\_stex:rcv\_stex:send\_userex:rcv\_userex:close\_resp\_info”)では、例外の送受信ログと接続閉塞時のリクエスト情報が採取されるため、例外を送受信したプロセスを特定することができます。  
.....

## ■アクセスログ採取レベルのキーワード

アクセスログの採取レベル(ログ種別)とその意味・採取箇所は以下のとおりです。

No.	キーワード	採取箇所	意味
1	connect	クライアント側ライブラリ	クライアントアプリケーションからサーバマシンに接続
2	stub_begin	スタブ	スタブでサーバメソッド呼出しが開始
3	send_req	クライアント側ライブラリ	リクエストをサーバアプリケーションに送信
4	queue_in	CORBAサービス(サーバ側)	リクエストをCORBAサービスで受け付け、キューイング
5	recv_req	サーバ側ライブラリ	リクエストをサーバアプリケーションで受け付け
6	skel_begin	スケルトン	スケルトンでサーバメソッド呼出しが開始
7	skel_end	スケルトン	スケルトンでサーバメソッド呼出しが終了
8	send_reply	サーバ側ライブラリ	リクエスト返信をクライアントアプリケーションに送信
9	send_stex	サーバ側ライブラリ	標準例外をクライアントアプリケーションに送信
10	send_userex	サーバ側ライブラリ	ユーザ例外をクライアントアプリケーションに送信
11	recv_reply	クライアント側ライブラリ	サーバアプリケーションからのリクエスト返信を受信
12	recv_stex	クライアント側ライブラリ	サーバアプリケーションからの標準例外を受信
13	recv_userex	クライアント側ライブラリ	サーバアプリケーションからのユーザ例外を受信
14	stub_end	スタブ	スタブでサーバメソッド呼出しが終了
15	close_resp	CORBAサービス(サーバ側)	クライアントマシンとの接続が閉塞
16	close_init	クライアント側ライブラリ	サーバマシンとの接続が閉塞
17	close_resp_info	CORBAサービス(サーバ側)	クライアントマシンとの接続が閉塞された際における、サーバで処理中のリクエスト情報

それぞれの採取レベルのメッセージについて以下で説明します。

### (1)connect

[メッセージ]

**connected to server (host = %s1, port = %s2)**

[可変情報]

%s1:サーバマシン名

%s2:ポート番号

[意味]

クライアントアプリケーションがサーバマシン%s1のポート番号%s2に接続しました。

### (2)stub\_begin

[メッセージ]

**STUB: server method call start (request\_id = %s1, intf\_id = %s2, operation = %s3)**

[可変情報]

%s1:リクエストID

%s2:サーバアプリケーションのインタフェースリポジトリID

%s3:サーバメソッド名

[意味]

スタブでサーバメソッド呼出しが開始しました。

**注)**

静的起動インタフェースのアプリケーションでのみ出力されます。また、Javaアプリケーションでは出力されません。

**(3)send\_req**

[メッセージ]

**send request to server (request\_id = %s1, intf\_id = %s2, host = %s3, operation = %s4)**

[可変情報]

%s1:リクエストID  
%s2:サーバアプリケーションのインタフェースリポジトリID  
%s3:サーバマシン名  
%s4:サーバメソッド名

[意味]

リクエストがサーバアプリケーションに送信されました。

**(4)queue\_in**

[メッセージ]

**queue request (request\_id = %s1, server pid = %s2, intf\_id = %s3)**

[可変情報]

%s1:リクエストID  
%s2:リクエストのディスパッチ先サーバプロセスID  
%s3:リクエストのディスパッチ先サーバのインタフェースリポジトリID

[意味]

リクエスト(リクエストID%s1)がCORBAサービス(OD\_startサービス)によって受け付けられ、キューイングされました。

**(5)recv\_req**

[メッセージ]

**receive request from client (request\_id = %s1, intf\_id = %s2, operation = %s3)**

[可変情報]

%s1:リクエストID  
%s2:サーバアプリケーションのインタフェースリポジトリID  
%s3:サーバメソッド名

[意味]

リクエストがサーバアプリケーションに受け付けられました。

**(6)skel\_begin**

[メッセージ]

**SKEL: server method invoke start (request\_id = %s1, intf\_id = %s2, operation = %s3)**

[可変情報]

%s1:リクエストID  
%s2:サーバアプリケーションのインタフェースリポジトリID  
%s3:サーバメソッド名

[意味]

スケルトンでのサーバメソッド呼出しが開始しました。

**注)**

静的スケルトンインタフェースのアプリケーションでのみ出力されます。また、Javaアプリケーションでは出力されません。

### (7)skel\_end

[メッセージ]

**SKEL: server method invoke end (request\_id = %s1, intf\_id = %s2, operation = %s3)**

[可変情報]

%s1:リクエストID  
%s2:サーバアプリケーションのインタフェースリポジトリID  
%s3:サーバメソッド名

[意味]

スケルトンでのサーバメソッド呼出しが終了しました。

**注)**

静的スケルトンインタフェースのアプリケーションでのみ出力されます。また、Javaアプリケーションでは出力されません。

### (8)send\_reply

[メッセージ]

**send reply to client (request\_id = %s1)**

[可変情報]

%s1:リクエストID

[意味]

リクエストに対する返信がサーバアプリケーションから送り出されました。

### (9)send\_stex

[メッセージ]

**send standard exception to client (request\_id = %s1, excep\_id = %s2, minor = 0x%s3)**

[可変情報]

%s1:リクエストID  
%s2:標準例外のID  
%s3:マイナーコード

[意味]

標準例外がサーバアプリケーションから送り出されました。

### (10)send\_userex

[メッセージ]

**send user exception to client (request\_id = %s1, excep\_id = %s2)**

[可変情報]

%s1:リクエストID  
%s2:ユーザ例外のID

[意味]

ユーザ例外がサーバアプリケーションから送り出されました。

### (11)recv\_reply

[メッセージ]

**receive reply from server (request\_id = %s1)**

[可変情報]

%s1:リクエストID

[意味]

サーバアプリケーションからのリクエスト返信をクライアントアプリケーションで受け取りました。

**(12)rcv\_stex**

[メッセージ]

**receive standard exception from server (request\_id = %s1, excep\_id = %s2, minor = 0x%s3)**

[可変情報]

%s1:リクエストID  
%s2:標準例外のID  
%s3:マイナーコード

[意味]

サーバアプリケーションからの標準例外をクライアントアプリケーションで受け取りました。

**(13)rcv\_userex**

[メッセージ]

**receive user exception from server (request\_id = %s1, excep\_id = %s2)**

[可変情報]

%s1:リクエストID  
%s2:ユーザ例外のID

[意味]

サーバアプリケーションからのユーザ例外をクライアントアプリケーションで受け取りました。

**(14)stub\_end**

[メッセージ]

**STUB: server method call end (request\_id = %s1, intf\_id = %s2, operation = %s3)**

[可変情報]

%s1:リクエストID  
%s2:サーバアプリケーションのインタフェースリポジトリID  
%s3:サーバメソッド名

[意味]

スタブでサーバメソッド呼出しが終了しました。

**注)**

静的スケルトンインタフェースのアプリケーションでのみ出力されます。また、Javaアプリケーションでは出力されません。

**(15)close\_resp**

[メッセージ]

**connection to client (%s1) was closed**

[可変情報]

%s1:クライアントマシンのIPアドレス

[意味]

クライアントマシンとの接続が閉塞されました。

**(16)close\_init**

[メッセージ]

**connection to server (%s1) was closed**

[可変情報]

%s1:サーバホスト名(またはIPアドレス)

[意味]

サーバマシンとの接続が閉塞されました。

(17)close\_resp\_info

[メッセージ]

request on the closed connection (request\_id = %s1, server pid = %s2, intf\_id = %s3, operation = %s4, oneway = %s5)

[可変情報]

%s1:リクエストID

%s2:リクエストのディスパッチ先サーバプロセスID

%s3:サーバアプリケーションのインタフェースリポジトリID

%s4:サーバメソッド名

%s5:onewayリクエストの場合は“yes”、それ以外の場合は“no”

[意味]

クライアントマシンとの接続が閉塞された際における、サーバで処理中のリクエスト情報です。

## 1.4.6 ログ採取のためのアプリケーション開発

ログ採取を行うため、アプリケーション開発時に以下に注意する必要があります。

### ■IDLコンパイル

IDLファイルから生成されるスタブ・スケルトンにログ採取機能を組み込むため、IDLコンパイルを行います。



- ログ採取機能は、V4.0L20以降のIDLコンパイラを使用することにより有効になります。
- Javaマッピングではログ採取機能は組み込まれません。
- IDLコンパイラ実行時に-nologオプションを指定するとログ採取処理は組み込まれなくなりますが、トラブル発生時の原因究明のために-nologオプションを指定しないことを推奨します。

### ■プログラミング

静的起動インタフェース・静的スケルトンインタフェースを使用してプログラミングすることにより、スタブ・スケルトンでもログ採取できます。

静的起動インタフェース・静的スケルトンインタフェースを使用しない場合は、CORBAサービス、およびライブラリのみでログ採取されます。

### ■ライブラリのリンク

ログ採取するためには、アプリケーションにサーバ用ライブラリ(“CORBAサービス”パッケージに含まれる)をリンクする必要があります。以下に必要なサーバ用ライブラリを示します。

Windows32/64

開発言語	ライブラリ名
C・C++	ODSV.LIB

開発言語	ライブラリ名
COBOL (スレッドモード)	ODCOBCBLMTSV.LIB または ODCOBCBLSVUC.LIB
COBOL (プロセスモード)	ODCOBCBLSV.LIB または ODCOBCBLSVUC.LIB
OOCOBOL	ODOOCOBSV.LIB または ODOOCOBSVUC.LIB

**Solaris32/64** **Linux32/64**

開発言語	ライブラリ名
C・C++	libOM.so
COBOL (スレッドモード)	libOMcblMT.so または libOMcblUC.so
COBOL (プロセスモード)	libOMcbl.so
OOCOBOL	libOMoocob.so または libOMoocobUC.so

## 注意

サーバ用ライブラリは、本製品の“CORBA サービス”(サーバ機能)に含まれています。インストール時には、必ず、“CORBA サービス”パッケージを選択してください。“CORBA サービスクライアント”(クライアント機能)をインストールしても、ログ採取機能は有効になりませんので注意してください。

## 1.4.7 ログ採取環境(configファイルの設定)

ログ採取環境(ログ採取状態、ログファイルサイズなど)は、CORBAサービスのconfigファイルに設定しておくことができます。ログ採取に関するconfigファイルのパラメタ定義について説明します。

## ポイント

- ・ configファイルについては、“チューニングガイド”の“config”を参照してください。
- ・ 以下に説明するログ採取環境のパラメタ定義は、odcntllogコマンドを使用することにより、CORBAサービスを停止・再起動しなくても、一時的に変更することができます。ただし、“log\_file\_path”については、odcntllogコマンドでは変更できません。詳細については“1.4.8 ログ採取環境の動的変更(odcntllogコマンド)”を参照してください。

### ■ログ採取環境の共通パラメタ

ログ採取環境の共通パラメタとして、以下があります。

#### log\_file\_path

ログファイル出力先のパス。絶対パスで指定します。128バイトより長いパスは指定できません。本パラメタを指定しなかった場合(デフォルト)、ログファイルは以下の場所に出力されます。

**Windows32/64**

<CORBAサービスインストールパス>%var 配下

**Solaris32/64** **Linux32/64**

<CORBAサービスインストールパス>/var 配下

### ■アクセスログのパラメタ

アクセスログには、以下の3つのパラメタがあります。

### access\_log\_policy

アクセスログのCORBAサービス起動時の採取/非採取の状態。

start : 起動時からアクセスログ採取を行う。(デフォルト)  
standby : 起動時にはアクセスログ採取を行わない。

### access\_log\_size

アクセスログファイルの最大サイズ(単位:バイト。デフォルトは3000000)



アクセスログは、accesslog、accesslog.oldの2ファイルに採取されます。このため、最大でaccess\_log\_size × 2 のディスク領域が必要となります。

### access\_log\_level

アクセスログの採取レベル。採取レベルを示すキーワードを区切り文字“:”で連結して指定します。“all”を指定すると、すべての採取レベルを指定したものとみなされます。

(デフォルト:send\_stex:rcv\_stex:send\_userex:rcv\_userex:close\_resp\_info)

採取レベルのキーワードは“[1.4.5 アクセスログ採取レベル\(access\\_log\\_levelパラメタ\)](#)”を参照してください。

## ■プロセスログ

プロセスログには、以下の2つのパラメタがあります。

### process\_log\_policy

プロセスログのCORBAサービス起動時の採取/非採取の状態。

start : 起動時からプロセスログ採取を行う。(デフォルト)  
standby : 起動時にはプロセスログ採取を行わない。

### process\_log\_size

プロセスログファイルの最大サイズ(単位:バイト。デフォルトは3000000)



#### Windows32/64

プロセスログは、以下のファイルに採取されます。

- サーバ用ライブラリ(ODSV.DLL)をリンクしたアプリケーションの場合、proclog、proclog.old(2ファイル)
- クライアント用ライブラリ(ODWIN.DLL)をリンクしたアプリケーションの場合、proclogcl、proclogcl.old(2ファイル)

このため、最大でprocess\_log\_size × 4 のディスク領域が必要となります。

#### Solaris32/64 Linux32/64

プロセスログは、proclog、proclog.oldの2ファイルに採取されます。このため、最大でprocess\_log\_size × 2 のディスク領域が必要となります。

## ■エラーログ

エラーログには、以下の2つのパラメタがあります。

### error\_log\_policy

エラーログのCORBAサービス起動時の採取/非採取の状態。

start : 起動時からエラーログ採取を行う。(デフォルト)  
standby : 起動時にはエラーログ採取を行わない。

#### error\_log\_size

エラーログファイルの最大サイズ(単位:バイト。デフォルトは3000000)



#### Windows32/64

エラーログは、以下のファイルに採取されます。

- サーバ用ライブラリ(ODSV.DLL)をリンクしたアプリケーションの場合、errlog、errlog.old(2ファイル)
  - クライアント用ライブラリ(ODWIN.DLL)をリンクしたアプリケーションの場合、errlogcl、errlogcl.old(2ファイル)
- このため、最大でerror\_log\_size × 4 のディスク領域が必要となります。

#### Solaris32/64 Linux32/64

エラーログは、errlog、errlog.oldの2ファイルに採取されます。このため、最大でerror\_log\_size × 2 のディスク領域が必要となります。

### ■ インフォメーションログ

インフォメーションログには、以下の2つのパラメタがあります。

#### info\_log\_policy

インフォメーションログのCORBAサービス起動時の採取/非採取の状態。

start : 起動時からエラーログ採取を行う。(デフォルト)  
standby : 起動時にはエラーログ採取を行わない。

#### info\_log\_size

インフォメーションログファイルの最大サイズ(単位:バイト。デフォルトは3000000)



#### Windows32/64

インフォメーションログは、以下のファイルに採取されます。

- サーバ用ライブラリ(ODSV.DLL)をリンクしたアプリケーションの場合、infolog、infolog.old(2ファイル)
  - クライアント用ライブラリ(ODWIN.DLL)をリンクしたアプリケーションの場合、infologcl、infologcl.old(2ファイル)
- このため、最大でinfo\_log\_size × 4 のディスク領域が必要となります。

#### Solaris32/64 Linux32/64

インフォメーションログは、infolog、infolog.oldの2ファイルに採取されます。このため、最大でinfo\_log\_size × 2 のディスク領域が必要となります。

## 1.4.8 ログ採取環境の動的変更(odcntllogコマンド)

ログ採取環境は、CORBAサービス起動時にはconfigファイルのパラメタ定義が有効となりますが、odcntllogコマンドを実行することにより、その定義を一時的に変更することができます。以下のログ採取環境の定義が、odcntllogコマンドで変更できます。

- ログファイルの最大サイズ
- アクセスログの採取レベル

なお、odcntllogコマンドで変更した内容は、CORBAサービス停止時まで有効となり、configファイルには反映されません。CORBAサービス再起動後も有効としたい場合はconfigファイルを更新する必要があります。



#### ポイント

odcntllogコマンドは、ログの採取開始/停止、ログ採取環境の状態表示も行います。詳細については、“リファレンスマニュアル(コマンド編)”を参照してください。

## 1.4.9 出力結果の見方

アクセスログとプロセスログの出力とその見方について説明します。

### アクセスログの出力例

```
768/000003F0 Tue Nov 06 14:39:57.889 ObjectDirector started
784/000006BC Tue Nov 06 14:40:35.323 send user exception to client
      (request_id = 5, excep_id = IDL:CosNaming/NamingContext/NotFound:1.0)
752/000005F0 Tue Nov 06 14:40:35.323 receive user exception from server
      (request_id = 5, excep_id = IDL:CosNaming/NamingContext/NotFound:1.0)
```

### プロセスログの出力例 Windows32/64

```
Tue Nov 06 14:39:57.889 ObjectDirector started (768)
Tue Nov 06 14:39:59.401 A [ 784] D:¥WINNT¥system32¥Naming.exe
Tue Nov 06 14:40:35.293 A [ 752] STRING_C
```

### プロセスログの出力例 Solaris32/64 Linux32/64

```
Tue Nov 06 14:39:57.889 ObjectDirector started (768)
Tue Nov 06 14:39:59.401 A [ 784] CosNaming_s
Tue Nov 06 14:40:35.293 A [ 752] STRING_C
```

### ログの見方

アクセスログから、プロセス784がNotFound例外を出し(“send user exception to client”)、プロセス752がNotFound例外を受け取っています(“receive user exception from server”)。

また、プロセスログから、プロセス784がネーミングサービス(Naming.exe / CosNaming\_s)、プロセス752がアプリケーション“STRING\_C”であることがわかります。

STRING\_Cからネーミングサービスへのリクエスト(resolveメソッドなど)に対して、指定されたオブジェクトがネーミングサービスに存在しないなどの原因によりNotFound例外が発生し、STRING\_Cがその例外を受け取ったと考えられます。

## 1.5 CORBAサービスのスナップショットの採取

スナップショット機能の概要と操作手順について説明します。

### 1.5.1 機能概要

スナップショット機能は、アプリケーションの入出力情報をプロセス単位またはインプリメンテーションリポジトリID単位にロギングします。ロギングされたスナップ情報は、メモリ上に採取され、ファイルへ出力することができます。

本機能を使用することにより、開発初期時のアプリケーションのデバッグ、運用時のトラブル原因究明に役立てることができます。

### 1.5.2 環境設定

スナップショットの運用を行う前に設定する内容について説明します。

## ■スナップショットの採取の設定

スナップショットを採取するかどうかをconfigファイルのsnap\_useパラメタで設定します。初期値は、“yes”です。変更する場合のみ設定が必要です。

## ■スナップショットサイズの設定

スナップショットで採取するスナップ情報のサイズをconfigファイルのsnap\_sizeパラメタで設定します。このサイズはスナップ情報を格納するメモリサイズに相当します。初期値は、40000バイトです。変更する場合のみ設定が必要です。スナップ情報のサイズがメモリサイズの上限值に達した場合は、メモリ上の古い情報から上書きされます。

**注)** 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がメッセージを1回送受信すると、Action(“1.5.3.2 スナップ情報の出力形式”参照)ごとに、以下のサイズ(概算値)のスナップ情報を採取します。スナップショットサイズは、これらの点と必要な情報量/マシンのメモリ容量を考慮して設定してください。

### request send

60+インプリメンテーションリポジトリID長+ホスト名長+オペレーション名長(バイト)

### request queue-in

40+インプリメンテーションリポジトリID長(バイト)

### request rcv/reply send

60+インプリメンテーションリポジトリID長+オペレーション名長(バイト)

### reply rcv

50+インプリメンテーションリポジトリID長+ホスト名長(バイト)

## 1.5.3 運用

スナップショットの運用について説明します。

### ポイント

スナップショットを採取するためには、アプリケーションにサーバ用ライブラリ(“CORBAサービス”パッケージに含まれる)をリンクする必要があります。以下に必要なサーバ用ライブラリを示します。

#### Windows32/64

開発言語	ライブラリ名
C・C++	ODSV.LIB
Java	ODjava4.jar
COBOL (スレッドモード)	ODCOBCBLMTSV.LIB または ODCOBCBLSVUC.LIB
COBOL (プロセスモード)	ODCOBCBLSV.LIB または ODCOBCBLSVUC.LIB
OOCOBOL	ODOCOBSV.LIB または ODOCOBSVUC.LIB

#### Solaris32/64

開発言語	ライブラリ名
C・C++	libOM.so
Java	ODjava4.jar
COBOL (スレッドモード)	libOMcblMT.so
COBOL (プロセスモード)	libOMcbl.so
OOCOBOL	libOMoocob.so

開発言語	ライブラリ名
C・C++	libOM.so
Java	ODjava4.jar

## 注意

サーバ用ライブラリは、本製品の“CORBAサービス”(サーバ機能)に含まれています。インストール時には、必ず、“CORBAサービス”パッケージを選択してください。“CORBAサービスクライアント”(クライアント機能)をインストールしても、ログ採取機能は有効になりませんので注意してください。

### 1.5.3.1 スナップ情報の採取

#### ■手順

スナップ情報を採取する手順を以下に示します。

- スナップ情報の採取開始  
サーバアプリケーションが動作していることを確認後、`odstartsnap`コマンドを使用してスナップ情報の採取を開始します。
- スナップ情報採取終了  
目的のスナップ情報を採取した後、`odstopsnap`コマンドを使用してスナップ情報の採取を終了します。スナップ情報の採取を終了する必要がある場合は、そのまま3.に進んでください。
- スナップショットのフォーマット出力  
`odformsnap`コマンドを使用して、採取したスナップ情報をファイルに出力します。このスナップ情報により、アプリケーションの送受信情報を取得することができます。

必要に応じて、`odfreesnap`コマンドを使用してメモリ上のスナップ情報をクリアしてください。

#### ■コマンド

スナップ情報の採取は、以下に示す4つのコマンドで操作を行います。

##### **odstartsnap**コマンド

スナップ情報の採取を開始します。このコマンドを使用する場合、スナップ情報を採取するプロセスが起動されている必要があります。

##### **odstopsnap**コマンド

スナップ情報の採取を終了します。

##### **odformsnap**コマンド

メモリ上に採取されたスナップ情報をファイルへ出力します。作成されるディレクトリは、コマンド使用時のカレントディレクトリです。

##### **odfreesnap**コマンド

メモリ上に採取したスナップ情報をクリアします。

## 1.5.3.2 スナップ情報の出力形式

odformsnapコマンドが出力するスナップ情報の出力形式を以下に示します。

Time	ProcessID	Action	Type	requestID	QueueID	implID	operation	host
22:33:39.180	359	request send	client	1	-----	IDL:CosNaming/NamingContext:1.0	resolve	hostABC
22:33:39.191	202	request queue-in	manager	1	1b3f664	IDL:CosNaming/NamingContext:1.0	----	---
22:33:39.191	248	request rcv	server	1	1b3f664	IDL:CosNaming/NamingContext:1.0	resolve	---
22:33:39.831	248	reply send	server	1	1b3f664	IDL:CosNaming/NamingContext:1.0	resolve	---
22:33:39.831	359	reply rcv	client	1	-----	IDL:CosNaming/NamingContext:1.0	----	hostABC

### Time

スナップ情報が採取された時間。「時:分:秒.ミリ秒」で表示されます。これはスナップ情報が記録された時間であるため、実際に送受信された時刻と完全には一致しません。

### ProcessID

送受信を行ったプロセスのプロセスID。

### Action

送受信の動作種別を示します。動作種別は以下の5つです。

#### — request send

リクエストが送信されたことを示します。

#### — request queue-in

リクエストがOD\_startサービスによって受け付けられ、キューイングされたことを示します。

#### — request rcv

リクエストがサーバアプリケーションに受け付けられたことを示します。

#### — reply send

リクエストからの返信がサーバアプリケーションから送り出されたことを示します。

#### — reply rcv

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からの返信を受け取ったことを示します。

### Type

送受信を行ったプロセス種別を示します。プロセス種別は以下の3つです。

#### — client

プロセスがクライアントアプリケーションとして動作していることを示します。

#### — server

プロセスがサーバアプリケーションとして動作していることを示します。

#### — manager

プロセスがOD\_startサービスであることを示します。

### requestID

送受信を行っているリクエストのIDを示します。

### QueueID

送受信を行っているリクエストがキューイングされたキューのIDを示します。これは動作種別が“request queue-in”、“request rcv”、“reply send”の3つの場合に表示されます。

### implID

相手先サーバアプリケーションのインプリメンテーションリポジトリIDを示します。これは動作種別によって意味が異なります。

— **request send**

相手先サーバアプリケーションのインプリメンテーションリポジトリIDを示します。

— **request queue-in**

OD\_startサービスが受け付けたリクエストのキューイング先のサーバアプリケーションのインプリメンテーションリポジトリIDを示します。

— **request rcv**

リクエストを受け付けたサーバアプリケーションのインプリメンテーションリポジトリIDを示します。

— **reply send**

“request rcv”と同じ意味です。

— **reply rcv**

“request send”と同じ意味です。

**operation**

相手先サーバメソッドのオペレーション名を示します。これは動作種別が“request send”、“request rcv”、“reply send”の3つの場合に表示されます。

**host**

相手先サーバマシンのホスト名を示します。これは、動作種別が“request send”、“reply rcv”の場合に表示されません。

### 1.5.3.3 スナップ情報の分析

以下の4つの例をもとにスナップ情報を分析する方法について説明します。

#### 例1:サーバアプリケーションとクライアントアプリケーションが同一マシン上で動作している場合

ここではクライアントアプリケーション、サーバアプリケーション(ネーミングサービス)、OD\_startサービスがすべて同一マシン“hostABC”にある場合のスナップ情報について説明します。

Time	ProcessID	Action	Type	requestID	QueueID	implID	operation	host
22:33:39.180	359	request send	client	1	-----	IDL:CosNaming/NamingContext:1.0	resolve	hostABC
22:33:39.191	202	request queue-in	manager	1	1b3f664	IDL:CosNaming/NamingContext:1.0	---	---
22:33:39.191	248	request rcv	server	1	1b3f664	IDL:CosNaming/NamingContext:1.0	resolve	---
22:33:39.831	248	reply send	server	1	1b3f664	IDL:CosNaming/NamingContext:1.0	resolve	---
22:33:39.831	359	reply rcv	client	1	-----	IDL:CosNaming/NamingContext:1.0	---	hostABC

1. まず、時刻「22:33:39.180」にクライアントアプリケーションはリクエストID1のリクエストを送信しています。相手先サーバアプリケーションのインプリメンテーションリポジトリIDは“IDL:CosNaming/NamingContextExt:1.0”で、相手先サーバアプリケーションはネーミングサービスです。メソッド名は“resolve”であり、相手先のホストは“hostABC”であることがわかります。
2. 時刻「22:33:39.191」にOD\_startサービスはクライアントアプリケーションからのリクエスト(リクエストIDは1)を受け付けてキューイングしています。キューイングしたキューIDは、“1b3f664”です。
3. 時刻「22:33:39.191」にサーバアプリケーションはキューID“1b3f664”のキューを取り出し、リクエストID1のリクエストを受け付けています。メソッド“resolve”が呼び出されています。

- 時刻「22:33:39.831」にサーバアプリケーションはリクエストID1のリクエストに対する返信をクライアントアプリケーションに返しています。このリクエストに対応していたキューIDは“1b3f664”であり、メソッド名は“resolve”であることを示します。
- 時刻「22:33:39.831」にクライアントアプリケーションはリクエストID1のリクエストに対する返信をサーバアプリケーションから受け取っています。

### 注意

送受信のタイミングとスナップ情報の書き込みのタイミングは完全に一致しないため、“reply send”と“reply rcv”の時刻が逆転する場合があります。

## 例2:サーバアプリケーションとクライアントアプリケーションが別マシン上で動作している場合

ここではサーバアプリケーション(ネーミングサービス)とOD\_startサービスが“hostXYZ”にあり、クライアントアプリケーションは別マシンで動作している場合のスナップ情報について説明します。

### [クライアントマシンでの出力例]

Time	ProcessID	Action	Type	requestID	QueueID	implID	operation	host
22:35:39.150	407	request send	client	3	-----	IDL:CosNaming/NamingContext:1.0	resolve	hostXYZ
22:35:40.534	407	reply rcv	client	3	-----	IDL:CosNaming/NamingContext:1.0	---	hostXYZ

### [サーバマシンでの出力例]

Time	ProcessID	Action	Type	requestID	QueueID	implID	operation	host
22:35:39.182	328	request queue-in	manager	3	2b3f656	IDL:CosNaming/NamingContext:1.0	---	---
22:35:40.012	258	request rcv	server	3	2b3f656	IDL:CosNaming/NamingContext:1.0	resolve	---
22:35:40.442	258	reply send	server	3	2b3f656	IDL:CosNaming/NamingContext:1.0	resolve	---

クライアントアプリケーション側で採取されるスナップ情報は、動作種別が“request send”のものと“reply rcv”のものです。クライアントアプリケーション側のスナップ情報を見ると、リクエストID3のリクエストを時刻「22:35:39.150」に送信し、時刻「22:35:40.534」に返信を受け取っていることがわかります。

サーバアプリケーション側で採取されるスナップ情報は、動作種別が“request queue-in”、“request rcv”、“reply send”のものです。サーバアプリケーション側のスナップ情報を見ると時刻「22:35:39.182」にOD\_startサービスがリクエストID3のリクエストをキューイングし、サーバアプリケーションが時刻「22:35:40.442」に返信をクライアントアプリケーションに送っていることがわかります。

## 例3:サーバアプリケーションが異常状態にある場合

ここではサーバアプリケーション(ネーミングサービス)が異常状態にある場合のスナップ情報について説明します。

Time	ProcessID	Action	Type	requestID	QueueID	implID	operation	host
------	-----------	--------	------	-----------	---------	--------	-----------	------

```

-----
22:41:59.730      237 request send   client      10 ----- IDL:CosNaming/NamingContext:1.0 resolve
hostABC
22:41:59.730      202 request rcv    manager     10 3bff4b8 IDL:CosNaming/NamingContext:1.0 resolve ---
22:41:59.730      202 reply send     manager     10 3bff4b8 IDL:CosNaming/NamingContext:1.0 resolve ---
22:41:59.730      237 reply rcv     client      10 ----- IDL:CosNaming/NamingContext:1.0 ---
hostABC

```

リクエストの送信先サーバプロセスが起動されていない等の異常状態にある場合は、OD\_startサービスがサーバアプリケーションの代わりにリクエストを受け取り、例外情報をクライアントアプリケーションに返すことがあります。

例外情報は、スナップ情報の出力からはわかりません。また、OD\_startサービスがリクエストを処理している場合はリクエストがキューイングされません。そのため、動作種別が“request queue-in”であるスナップ情報はなく、キューID“3bff4b8”は意味を持ちません。

#### 例4:OD\_startサービスに対してリクエストが発行された場合

ここではOD\_startサービスに対してリクエストが発行された場合のスナップ情報について説明します。

Time	ProcessID	Action	Type	requestID	QueueID	implID	operation	host
-----								
22:35:51.541	359	request send	client	6	-----	IDL:OM_ORB/admin:1.0	synch	hostABC
22:35:51.551	202	request rcv	manager	6	0	IDL:OM_ORB/admin:1.0	synch	---
22:35:51.581	202	reply send	manager	6	0	IDL:OM_ORB/admin:1.0	synch	---
22:35:51.581	359	reply rcv	client	6	-----	IDL:OM_ORB/admin:1.0	---	hostABC

管理用コマンドの中には、OD\_startサービスに対してリクエストを送るものがあります(例:OD\_impl\_inst)。OD\_startサービスに送られたリクエストは、キューイングされずにOD\_startサービスによって直接処理されます。“request rcv”に対応するプロセス種別が“server”ではなく“manager”である場合、OD\_startサービスがリクエストを処理していることを示します。その場合、リクエストがキューイングされないため、動作種別が“request queue-in”であるスナップ情報はありません。キューIDは“0”になっていますが、OD\_startサービスによって直接処理されるリクエストに対応するキューIDは意味を持ちません。

### 1.5.3.4 スナップ情報採取状況の確認

スナップ情報の採取状況を確認するには、odlistsnapコマンドを使用します。

odlistsnapコマンドは、インプリメンテーションリポジトリIDごとのスナップ情報の採取状況を以下のように出力します。

- ・ 採取している場合 : “getting”
- ・ 採取していない場合 : “non getting”

odlistsnapコマンドの実行例を以下に示します。

```

> odlistsnap
PID      ImplementationRepositoryID      status
1085     IDL:OM_ORB/admin:1.0             non getting
1121     IDL:CosNaming/NamingContext:1.0  non getting

```

## 1.6 CORBAサービスのIPCログの採取 Solaris32/64 Linux32/64

CORBAサービスでは、OD\_startプロセス起動処理時にIPCログを出力します。IPCログでは、以下のコマンドの実行結果が採取されます。

- ・ ipcs -a
- ・ ps -ef

**Linux32/64**

以下のコマンドの実行結果も採取されます。

- ipcs -p
- ipcs -t

IPCログに出力された情報を参照することで、OD\_startプロセス起動成功／失敗時のIPC資源の状態を確認できます。

## ■ ログファイル

IPCログは以下のファイルに出力されます。

```
<GORBAサービスインストールパス>/var/ipclog、ipclog.old
```

IPCログファイルのサイズは、ipclog、ipclog.oldともに最大で100Kバイトです。

## ■ データ形式

IPCログのデータは以下の形式で出力されます。

### Solaris32/64

```
*****
State : Success
Date  : Sat Oct 23 16:55:47 2004
=====
“ipcs -a” 出力結果
=====
“ps -ef” 出力結果
```

### Linux32/64

```
*****
State : Success
Date  : Sat Oct 23 16:55:47 2004
=====
“ipcs -a” 出力結果
=====
“ipcs -p” 出力結果
=====
“ipcs -t” 出力結果
=====
“ps -ef” 出力結果
```

### State

OD\_startプロセスの起動に成功した場合は“Success”、失敗した場合は“Failure”が出力されます。

### Date

ログ採取された時刻です。

### “ipcs -a”出力結果

OD\_startプロセス起動処理後の“ipcs -a”コマンドの出力結果です。出力結果はOSのLANG環境変数に依存します。

### “ipcs -p”出力結果 Linux32/64

OD\_startプロセス起動処理後の“ipcs -p”コマンドの出力結果です。出力結果はOSのLANG環境変数に依存します。

## “ipcs -t”出力結果 Linux32/64

OD\_startプロセス起動処理後の“ipcs -t”コマンドの出力結果です。出力結果はOSのLANG環境変数に依存します。

## “ps -ef”出力結果

OD\_startプロセス起動処理後の“ps -ef”コマンドの出力結果です。出力結果はOSのLANG環境変数に依存します。

### 出力例

IPCログの出力例を以下に示します。

#### Solaris32/64

```
*****
State : Success
Date : Sat Oct 23 19:03:28 2004
=====
IPC status from <running system> as of Sat Oct 23 19:03:28 JST 2004
T      ID      KEY      MODE      OWNER      GROUP      CREATOR      CGROUP      CBYTES      QNUM      QBYTES      LSPID      LRPID
STIME  RTIME      CTIME
Message Queues:
q      959488    0xd11fb30 -Rrw-----   root      other      root      other      0      0      32768      8465      8464
17:09:27 17:09:27 17:08:20
q      924673    0xb11fb30 -Rrw-----   root      other      root      other      0      0      32768      0      0 no-
entry no-entry 17:08:20
T      ID      KEY      MODE      OWNER      GROUP      CREATOR      CGROUP      NATTCH      SEGSZ      CPID      LPID
ATIME  DTIME      CTIME
Shared Memory:
m      1026048    0x211fb30 --rw-----   root      other      root      other      1      12      8465      8467
17:08:20 17:08:20 17:08:20
m      1003521    0x111fb30 --rw-----   root      other      root      other      1      131112    8466      8467
18:16:29 18:16:29 17:08:20
T      ID      KEY      MODE      OWNER      GROUP      CREATOR      CGROUP      NSEMS      OTIME      CTIME
Semaphores:
s      63635456   0x1044eee --ra-ra-ra-   root      other      root      other      1 no-entry 17:10:09
s      61341697   0x2044eee --ra-ra-ra-   root      other      root      other      1 no-entry 17:10:09
=====
      UID  PID  PPID  C   STIME TTY      TIME CMD
      root  0   0  0   Jul 05 ?       0:17 sched
      root  1   0  0   Jul 05 ?       0:29 /etc/init -
      root  2   0  0   Jul 05 ?       0:00 pageout
      root  3   0  1   Jul 05 ?      5474:21 fsflush
      root 348  1  0   Jul 05 ?       0:00 /usr/lib/saf/sac -t 300
```

#### Linux32/64

```
*****
State : Success
Date : Sat Oct 23 19:03:28 2004
=====
----- Shared Memory Segments -----
key      shmid      owner      perms      bytes      nattch      status
0x051f00f6 66453504   root      777        1324644    2
0x011f00f6 66486273   root      777        7269528    1
----- Semaphore Arrays -----
key      semid      owner      perms      nsems
0x00000000 1058832384 root      777        1
0x0200068d 32769      root      666        1
```

```

----- Message Queues -----
key      msqid    owner    perms    used-bytes  messages
0x0d37051d 0      root     600      0           0
0x0b37051d 32769   root     600      0           0

=====

----- Shared Memory Creator/Last-op -----
shmids   owner    cpid     lpid
66453504 root     29262    29401
66486273 root     29262    29271

----- Message Queues PIDs -----
msqid    owner    lspd     lrpid
0        root     2528     2527
32769   root     0        0

=====

----- Shared Memory Attach/Detach/Change Times -----
shmids   owner    attached          detached          changed
66453504 root     Oct 23 19:03:28   Oct 23 19:03:28   Oct 23 19:03:26
66486273 root     Oct 23 19:03:26   Oct 23 19:03:26   Oct 23 19:03:26

----- Shared Memory Operation/Change Times -----
shmids   owner    last-op          last-changed
1058832384 root     Sat Oct 23 19:03:26 2004   Sat Oct 23 19:03:26 2004
32769   root     Not set          Fri Oct 1 17:46:25 2004

----- Message Queues Send/Recv/Change Times -----
msqid    owner    send          recv          change
0        root     Oct 1 17:46:59   Oct 1 17:46:59   Oct 1 17:46:35
32769   root     Not set          Not set          Oct 1 17:46:35

=====

UID      PID  PPID  C  STIME TTY      TIME CMD
root     1    0    0  0 Oct01 ?      00:00:03 init [5]

root     2    1    0  0 Oct01 ?      00:00:00 [migration/0]
root     3    1    0  0 Oct01 ?      00:00:00 [ksoftirqd/0]
root     4    1    0  0 Oct01 ?      00:00:00 [events/0]
root     5    4    0  0 Oct01 ?      00:00:00 [khelper]
root     6    4    0  0 Oct01 ?      00:00:00 [kacpid]

```

## 1.7 CORBAサービスのネーミングサービスのユーザ例外ログの採取

CORBAサービスのネーミングサービスのログ情報は、ネーミングサービスインタフェースを使用した場合に、ネーミングサービスのユーザ定義例外の発生情報がロギングされます。

ネーミングサービスインタフェースでユーザ定義例外が発生した場合、発生APIおよびパラメタの指定不備を特定できるようになります。

### 1.7.1 ログのデータ

#### ■ログファイル

ネーミングサービスのユーザ例外ログは、以下のファイルに出力されます。

## ◆ファイル名

Windows32/64

<CORBAサービスインストールパス>%var%cn\_userexception\_log、cn\_userexception\_log.old

Solaris32/64 Linux32/64

<CORBAサービスインストールパス>/var/cn\_userexception\_log、cn\_userexception\_log.old

## ◆ファイルサイズ

ログファイルが作成されると、最大で以下のディスク領域が必要となります。ログ採取を行う場合は、十分な領域を確保してください。

cn\_userexception\_log\_sizeの値 (nsconfigファイル) × 2 [バイト]

nsconfigファイルの各パラメタについては、“[1.7.2 ログ採取環境 \(nsconfigファイルの設定\)](#)”を参照してください。

## ■データ形式

ユーザ例外ログのデータは、以下の形式で出力されます。

年-月-日 時:分:秒.ミリ秒 プロセスID ユーザ例外名 指定した名前 例外発生API クライアントのIPアドレス

### 年-月-日

ログが採取された年月日。

### 時:分:秒.ミリ秒

ログが採取された時刻。

### プロセスID

ネーミングサービスのプロセスID。

### ユーザ例外名

発生したユーザ例外名。

以下のユーザ例外が採取対象になります。

- NotFound:IDL:CosNaming/NamingContext/NotFound:1.0
- CannotProceed:IDL:CosNaming/NamingContext/CannotProceed:1.0
- InvalidName:IDL:CosNaming/NamingContext/InvalidName:1.0
- AlreadyBound:IDL:CosNaming/NamingContext/AlreadyBound:1.0
- NotEmpty:IDL:CosNaming/NamingContext/NotEmpty:1.0
- InvalidAddress:IDL:CosNaming/NamingContextExt/InvalidAddress:1.0

### 指定した名前

ネーミングサービスAPIに指定されたネーミングサービス登録オブジェクトの名前。

ネーミングコンテキスト拡張インタフェースを使用している場合は、文字列表記のバインディング名が出力されます。

### 例外発生API

例外が発生したネーミングサービスAPI名。

### クライアントのIPアドレス

ネーミングサービスAPIを呼び出したクライアントホストのIPアドレス。

### 出力例

ユーザ例外ログの出力例を以下に示します。

```
2003-05-07 02:04:06.123+09:00 1234 NotFound NAME_TEST
CosNaming_NamingContext_resolve() 10.1.1.1
2003-05-07 02:04:07.123+09:00 1234 AlreadyBound CONTEXT-A/CONTEXT-B/NAME_TEST1
CosNaming_NamingContext_bind() 10.1.1.1
2003-05-07 02:04:08.123+09:00 1234 NotFound CONTEXT-A/NAME_TEST2 CosNaming_NamingContext_resolve() 10.1.1.1
```

## 注意

“指定した名前”が手動で作成されたコンテキスト配下にある場合は、コンテキストも出力されます。

## 1.7.2 ログ採取環境 (nsconfigファイルの設定)

ユーザ例外ログ採取環境 (ログ採取の有無、ログファイルサイズなど) は、ネーミングサービスの nsconfig ファイルで設定します。

ログ採取に関する nsconfig ファイルのパラメタ定義について説明します。

### cn\_userexception\_log\_use

ユーザ例外ログの採取 / 非採取の状態です。  
以下のように指定します。

- yes: ユーザ例外ログを採取します (初期値)。
- no: ユーザ例外ログを採取しません。

### cn\_userexception\_log\_size

ユーザ例外ログファイルの最大サイズ (バイト) です (初期値: 2000000)。

## 注意

ユーザ例外ログファイルは、cn\_userexception\_log、cn\_userexception\_log.old の 2 ファイルに採取されます。このため、最大で cn\_userexception\_log\_size × 2 のディスク領域が必要となります。

## 参照

nsconfig ファイルについては、“チューニングガイド”の“nsconfig”を参照してください。

## 1.8 クラッシュダンプ・コアダンプの採取

プロセスが消滅したときに、クラッシュダンプ・コアダンプを採取することにより、異常終了の原因を調査することができる場合があります。

### 1.8.1 Windows (R) におけるクラッシュダンプの採取

ここでは、Windows (R) におけるワトソン博士を使用したクラッシュダンプの採取方法を説明します。

#### ■ワトソン博士について

ワトソン博士は Microsoft Corporation のソフトウェアで、プログラムエラーのためのデバッガです。  
プログラムエラーが発生すると、ワトソン博士が自動的にログファイルにデバッグ情報を出力します。なお、ログファイル名は、「drwtsn32.log」です。また、ログファイルの出力先は、ワトソン博士を起動して、設定することができます。  
ワトソン博士の詳細は、Microsoft Corporation の Web ページを参照してください。

## ■ワトソン博士の設定

クラッシュダンプの採取には、Windows(R)に同梱されている「ワトソン博士」を使用します。  
次の例を参考にして、「ワトソン博士」を設定してください。この設定を行うことにより、異常終了時に、自動的にクラッシュダンプが出力されるようになります。



例

### ワトソン博士の設定例 (Windows Server(R) 2003、Windows(R) XPの場合)

1. MS-DOSコマンドプロンプトなどで“drwtsn32 -i”コマンドを投入します。[ワトソン博士が既定のアプリケーション デバッグとしてインストールされました。]のメッセージが表示されます。
2. 更に、MS-DOSコマンドプロンプトなどで、“drwtsn32”コマンドを実行します。[Windows ワトソン博士]の設定画面が表示されますので、以下を確認してください。
  - [ログファイルパス(L)]、[クラッシュダンプ(P)]が正しく指定されているか
  - [クラッシュダンプの種類(Y)]を[完全]と設定しているか
  - [すべてのスレッド コンテキストをダンプ(A)]のチェックボックスがチェックされているか
  - [既定のログ ファイルに追加(E)]のチェックボックスがチェックされているか
  - [メッセージ ボックスによる通知(U)]のチェックボックスがチェックされているか
  - [クラッシュ ダンプ ファイルの作成(T)]のチェックボックスがチェックされているか



注意

### Windows Server(R) 2003 x64 Editionの場合

32ビットモード版のワトソン博士の環境設定を行う場合は、MS-DOSコマンドプロンプトなどで“%SystemRoot%\SysWow64\drwtsn32”コマンドを実行し、上述と同様の設定をしておく必要があります。



参考

### Windows(R) XP/Windows Server(R) 2003 以外の場合

ワトソン博士の機能が提供されていません。  
ワトソン博士の代わりにWindowsエラー 報告 (Windows Error Reporting (WER)) の機能を使用します。  
次の例を参考にして、WERを設定してください。



例

### WERの設定例

1. MS-DOSコマンドプロンプトなどで“regedit”コマンドを投入し、レジストリエディタを起動します。
2. 「HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Windows\Windows Error Reporting\LocalDumps」キーを作成します。
3. LocalDumpsキーにREG\_DWORD型でDumpTypeという値を作成し、「2」を設定します。

WERに関する設定の方法については、以下の情報も参照ください。

- ・ 富士通製サーバ製品の場合：  
<http://primeserver.fujitsu.com/primergy/soft/ssupportguide/>
- ・ 上記製品以外の場合：  
<http://msdn.microsoft.com/en-us/library/bb787181.aspx>

## ■注意事項

Windows Server(R) 2003の初期版においては、ユーザダンプが出力されない問題をはじめとして、その他にもJavaの実行動作に影響を及ぼす問題などがあります。

たとえば、次のような問題があります。

- <http://support.microsoft.com/kb/836080/en-us>
- <http://support.microsoft.com/kb/837018/en-us>
- <http://support.microsoft.com/kb/841176/en-us>

Windows Server(R) 2003を使用する場合は、Service Pack 1またはHotfixを適用してください。

## 1.8.2 Solarisにおけるコアダンプの採取 Solaris32/64

ここでは、Solarisにおけるコアダンプ採取のための注意事項を説明します。

### ■コアダンプが出力されない場合の確認

コアダンプが出力されない場合の原因として、システムリソース等の問題がまず考えられます。カレントディレクトリの書き込み権、ディスク容量、limit(1)コマンド結果を確認してください。

## 1.8.3 Linuxにおけるコアダンプの採取 Linux32/64

ここでは、Linuxにおけるコアダンプ採取のための注意事項を説明します。

### ■コアダンプが出力されない場合の確認

コアダンプが出力されない場合の原因として、システムリソース等の問題がまず考えられます。カレントディレクトリの書き込み権、ディスク容量、limit(1)コマンド結果を確認してください。

また、Linuxではハード/OSの出荷時もしくはOSのUpdate適用により、デフォルトでコアダンプの出力が設定されていない場合があります。以下を実施してコアダンプが出力されるようにしてください。

《コアダンプ出力設定方法》

- isstartコマンドでInterstageを起動させる場合

sh(bash)で“ulimit -c unlimited”コマンド実行後、Interstageを起動させます。ワークユニット起動ユーザがInterstage起動ユーザと違う場合は、ワークユニット起動前に“ulimit -c unlimited”コマンドを実行してから、ワークユニットを起動させます。

- RCプロシジャでOS起動時に自動的にInterstageが起動するように設定されている場合

以下の方法を実施することで、OS再起動後にcoreが出力されるようになります。

/etc/init.d/functionsファイルに、

```
# make sure it doesn't core dump anywhere; while this could mask
# problems with the daemon, it also closes some security problems
```

```
ulimit -S -c 0 >/dev/null 2>&1
```

または、

```
ulimit -S -c ${DAEMON_COREFILE_LIMIT:-0} >/dev/null 2>1
```

と記述されていますので、上記の設定で“0”を、“unlimited”に変更してください。

```
ulimit -S -c unlimited >/dev/null 2>&1
```

/etc/rc2.d/S99startisに、以下の<---の記述を追加してください。

```
#!/bin/sh
# Interstage Application Server
# S99starttis : Interstage Application Server start procedure
```

```
OD_HOME=/opt/FJSVod
export OD_HOME
```

```
ulimit -c unlimited <---  
  
/opt/FJSVod/bin/odalive > /dev/null  
while [ "$?" != "0" ]  
do  
    sleep 1  
    /opt/FJSVod/bin/odalive > /dev/null  
done  
  
/opt/FJSVtd/bin/isstart
```

## 1.9 Interstage／ワークユニット停止時の調査資料の自動採取

Interstageおよびワークユニットの停止処理時間が一定の時間(停止監視時間)を超過した場合に、停止処理が無応答と判断し、技術員が原因調査を行うための調査資料を自動的に採取します。本機能を「調査資料自動採取機能」と呼びます。

調査資料自動採取機能について以下に説明します。

### ■調査資料の採取契機

Interstageまたはワークユニットの停止操作を行った際に、停止処理にかかった時間が停止監視時間を超過した場合に調査資料を採取します。

停止監視時間に指定可能な値は、300～3600(単位:秒)です。初期値は、300秒です。

停止監視時間を設定する場合は、以下のどちらかの方法で設定してください。環境変数とコマンドが共に設定された場合は、コマンドの設定値が有効となります。

#### 環境変数で設定する場合

環境変数IS\_ISSTOP\_MONITOR\_TIMERに、停止監視時間を設定します。本環境変数に停止監視時間を設定した状態でInterstage／Interstage管理コンソールを起動すると、設定値が有効となります。

環境変数の詳細については、「チューニングガイド」の「Interstageのチューニング」-「環境変数について」を参照してください。

#### コマンドで設定する場合

isstop/isstopwuコマンド実行時に、-tオプションの引数として停止監視時間を指定して設定します。

isstop/isstopwuコマンドの詳細については、「リファレンスマニュアル(コマンド編)」を参照してください。

### ポイント

調査資料の自動採取が行われた場合でも、Interstage／ワークユニットの停止操作は、継続して実行されます。したがって、資料の採取後に停止処理が終了した場合、停止操作は正常に終了します。

### ■調査資料の採取有無の確認方法

調査資料の自動採取が行われると、Windows(R)の場合はイベントログファイル、Solaris/Linuxの場合はシステムログファイルに、以下のメッセージが出力されます。

本メッセージの出力有無を確認することにより、資料採取の実行有無を判断できます。

操作内容	出力メッセージ(メッセージID)
Interstage停止操作時	is20270
ワークユニット停止操作時	is20271

## ■調査資料自動採取時の対処方法

停止処理が無応答となった場合、まず停止処理に時間がかかっている原因を調査します。例えば、以下の原因が考えられます。

- ・ ワークユニットで運用しているアプリケーションの停止処理に時間がかかっている。
- ・ ワークユニットで運用しているアプリケーションの停止処理がハングしている。
- ・ サーバが高負荷状態となったため、停止に時間がかかっている。

停止処理が無応答となった原因が特定できた場合、その対処を行ってください。

停止処理が無応答となった原因を調査しても問題を解決できない場合は、`iscollectinfo`コマンドで調査資料を採取し、技術員に連絡してください。調査資料は、`iscollectinfo`コマンドで採取されます。

## 注意

本調査資料が不要の場合は、必要に応じて以下のディレクトリ配下に出力されたファイル(サブディレクトリ配下を含む)を削除してください。

**Windows32/64** (インストールパスはデフォルト)

```
C:¥Interstage¥td¥trc¥manage¥isstop_hung¥
C:¥Interstage¥td¥trc¥manage¥isstop_hung_old¥
C:¥Interstage¥td¥trc¥manage¥isstopwu_hung¥
C:¥Interstage¥td¥trc¥manage¥isstopwu_hung_old¥
```

**Solaris32/64** (インストールパスはデフォルト)

```
/var/opt/FSUNtd/trc/manage/isstop_hung/
/var/opt/FSUNtd/trc/manage/isstop_hung_old/
/var/opt/FSUNtd/trc/manage/isstopwu_hung/
/var/opt/FSUNtd/trc/manage/isstopwu_hung_old/
```

**Linux32/64**

```
/var/opt/FJSVtd/trc/manage/isstop_hung/
/var/opt/FJSVtd/trc/manage/isstop_hung_old/
/var/opt/FJSVtd/trc/manage/isstopwu_hung/
/var/opt/FJSVtd/trc/manage/isstopwu_hung_old/
```

## 第2章 インストール・アンインストール時の異常

### 2.1 共通事項

#### Windows32/64

ここでは、Interstageのインストール時に発生したトラブルの事例およびその対処について説明します。

#### Interstageのインストールが失敗する

##### ターミナルサービスのモードの状態に誤りがある場合の対処

以下のいずれかの場合、ターミナルサービスがアプリケーションサーバモードの状態、Interstageをインストールした可能性があります。

- ・ インストールが失敗し、「エラー番号:0x80040702 詳細:dllのロードに失敗しました。:odautosetup」のポップアップが出力される。
- ・ インストールが失敗し、イベントビューアにメッセージihs00012が出力される。
- ・ インストールが失敗し、「エラー: is20102:INTERSTAGEの起動に失敗しました 理由コード(10)」のポップアップが出力される。

上記に合致する場合、以下の作業を実施してください。

1. 「アプリケーションの追加と削除」から“Interstage Application Server Enterprise Edition※”を実行し、Interstageのアンインストールを行う。  
※) Interstage Application Server Standard-J Editionの場合は“Interstage Application Server Standard-J Edition”, Interstage Web Server Expressの場合は“Interstage Web Server Express”が表示されます。
2. 「CHANGE USER /INSTALL」を実行し、ターミナルサービスをインストールモードに変更する。
3. Interstageを再度インストールする。
4. 「CHANGE USER /EXECUTE」を実行し、ターミナルサービスを実行モードに変更する。

#### インストール先の入力ができない

Windows(R) XPまたはWindows Server(R) 2003上でインストールを行う場合、「インストール先の選択」画面において、「参照」ボタンを押下し、「ディレクトリの選択」画面で以下の操作を実施すると、「ディレクトリの選択」画面からカーソルがはずれ、ディレクトリのパスを入力できない事象が発生する場合があります。

- ・ 不当なフォルダの入力(ドライブ名が記載されていないなど)を行い、再入力をうながすダイアログで「OK」ボタンを押下する
- ・ 「キャンセル」ボタンを押下し、再度「インストール先の選択」画面から素早く「参照」ボタンを押下する
- ・ 「ディレクトリの選択」画面から別の画面をアクティブな状態にする

上記の現象が発生した場合は、「キャンセル」ボタンを押下し、「ディレクトリの選択」画面を終了させ、再度「インストール先の選択」画面から「参照」ボタンを押下し、「ディレクトリの選択」画面から正しいフォルダを入力してください。

#### 機能の追加と削除が行えない

Interstageを導入済みの環境において、機能の追加と削除を実行すると以下のメッセージが表示され実行できない場合があります。

「インストールサポートファイルをインストールできませんでした。致命的なエラーです。」

本現象が発生した場合、機能の追加と削除、およびアンインストールができない状態となっております。

この状態でInterstageの機能構成を変更するには、一旦、手動によりInterstageの資材を削除したのちに改めてInterstageをインストールしなおす必要があります。なお、手動で削除する場合、レジストリなどの資材を削除しなければならないた

め、十分注意を行う必要があります。

削除手順につきましては、InterstageのインストールDVD-ROMドライブの直下(ドライブEの場合、E:\HowToDel.txtになります)の“HowToDel.txt”を参照して行ってください。

## アンインストールを行うとインストール画面が表示される

インストール後、すぐにアンインストールを行った場合、アンインストール画面ではなく、インストール画面が表示される場合があります。この現象はインストール完了後、タスクバーにインストーラのタスクが残った状態のままアンインストールを実行することで発生する可能性があります。この場合、インストール画面を終了し、再度アンインストールを実行してください。

## アンインストールが行えない

アンインストールを実行すると以下のメッセージが表示されアンインストールできない場合があります。「インストールサポートファイルをインストールできませんでした。致命的なエラーです。」

本現象が発生した場合、通常のアンインストールができない状態となっております。

本現象が発生した場合、手動によりInterstageの資材を削除する必要があります。手動で削除する場合、レジストリなどの資材を削除しなければならないため、十分注意を行う必要があります。

削除手順につきましては、InterstageのインストールDVD-ROMドライブの直下(ドライブEの場合、E:\HowToDel.txtになります)の“HowToDel.txt”を参照して行ってください。

## アンインストール後のコンピュータ再起動が行えない

アンインストール後に「InstallShield Wizardの完了」画面において、「はい、今すぐコンピュータを再起動します。」を選択後、コンピュータの再起動が実行されず、タスクバーに「Interstage Application Server」が残存する場合があります。このとき、タスクマネージャによりInterstage Application Serverのタスクが「応答せず」となっている場合は、タスクの終了を行い、再度コンピュータの再起動を行ってください。

## Windows Defenderの警告について

Windows Defenderが有効になっている場合、Interstageのサーバ機能をインストールすると、Windows Defenderの履歴およびイベントビューアのシステムログに、以下のWindows Defenderからのメッセージが出力されることがあります。

これは、Windows Defenderのリアルタイム保護エージェントがサービスのソフトウェアの登録を監視しているため、そのまま使用して問題ありません。

また、インストール中、Windows Defenderのアイコンがタスクバーの通知領域に表示されることがあります。この場合、Windows Defenderを開き、“コンピュータの設定に対する変更を確認する”の画面で、[操作を適用する]をクリックしてください。

### ■Windows Defenderの履歴のメッセージ

日本語

このプログラムは、望ましくない動作をする可能性があります。

英語

This program has potentially unwanted behavior.

### ■イベントビューアのシステムログ

日本語

Windows Defender リアルタイム保護エージェントで、変更が検出されました。これらの変更を行ったソフトウェアに潜在的リスクがないか分析することをお勧めします。これらのプログラムの動作方法に関する情報を使用して、これらのプログラムの実行を許可するか、コンピュータから削除するかを選択できます。プログラムまたはソフトウェア発行者を信頼できる場合のみ、変更を許可してください。Windows Defender は許可された変更を元に戻せません。

Windows Defender Real-Time Protection agent has detected changes. Microsoft recommends you analyze the software that made these changes for potential risks. You can use information about how these programs operate to choose whether to allow them to run or remove them from your computer. Allow changes only if you trust the program or the software publisher. Windows Defender can't undo changes that you allow.

Windows32/64

## インストール後のセットアップ時にエラーダイアログが出力される

インストール後のセットアップ時に以下のエラーダイアログが出力される場合がありますが、インストールには問題はありません。

### ■エラーダイアログ内容

以下のサービスの起動処理でエラーが発生しました。  
Interstage Operation Tool

### ■対処方法

インストール後、手動で以下のサービスを起動してください。

- ・ Interstage Operation Tool

## 2.2 アンインストールと管理(ミドルウェア)

ここでは、“アンインストールと管理(ミドルウェア)”に関連したトラブル発生時の対処について説明します。

Windows32/64

### Interstageのアンインストールが実行できない

Interstageのインストール資材を削除してしまったり、なんらかの異常により、“アンインストールと管理(ミドルウェア)”からInterstageを削除できない状態になった場合、次の手順でアンインストールを実行してください。

#### 1. Interstageのアンインストール

該当するエディションのサーバパッケージDVDを用意して、以下のコマンドを実行することでInterstageをアンインストールすることができます。ただし、“アンインストールと管理(ミドルウェア)”の製品情報とFJQSSの情報は削除されませんので、次に示す手順にて削除してください。また、この手順で失敗する場合は、サーバパッケージDVDドライブの直下(ドライブEの場合、E:\HowToDel.txtになります)の“HowToDel.txt”を参照してインストール環境を削除してください。

```
<サーバパッケージDVD>:\$ISTAGE¥Disk1¥Setup.exe UninstallMode
```

#### 2. FJQSS情報の削除

レジストリ、スタートメニュー、ファイルの削除が必要となります。詳しくは、サーバパッケージDVDドライブの直下の“HowToDel.txt”を参照してください。

#### 3. “アンインストールと管理(ミドルウェア)”の管理DBからソフトウェア情報の削除

削除用コマンド(ciope.exe)が用意されています。詳しくは、サーバパッケージDVDドライブの直下の“HowToDel.txt”を参照してください。

Solaris32/64 Linux32/64

### Interstageのアンインストールが実行できない

FJSVisasの資材を削除してしまったり、なんらかの異常により、“アンインストールと管理(ミドルウェア)”からInterstageを削除できない状態になった場合、次の手順でアンインストールを実行してください。

### 1. Interstageのアンインストール

該当するエディションのサーバパッケージDVDを用意して、以下のコマンドを実行してアンインストールしてください。

```
<サーバパッケージDVD>/installer/uninstall.sh
```

### 2. “アンインストールと管理(ミドルウェア)”の管理DBからソフトウェア情報の削除

該当するエディションのサーバパッケージDVDを用意して、以下のコマンドを実行してください。

```
<サーバパッケージDVD>/installer/ConfigFiles/delete_product.sh
```

## 2.3 機能の追加と削除

ここでは、Interstageの機能の追加と削除時に発生したトラブルの事例およびその対処について説明します。

Windows32/64

### Interstageの機能の追加と削除が失敗する

Interstageの機能の追加と削除の実行前に、以下のいずれかの作業を行わなかった可能性が考えられます。

- SNMPサービスへ性能監視ツールを登録している場合は、機能の追加と削除の実行前に、SNMPサービスを停止する必要があります。
- 機能の追加と削除を行う前に、同一サーバ上で動作している以下の製品を停止してください。
  - Systemwalker Centric Manager
  - Systemwalker Operation Manager
  - IPCOMサーバ負荷計測エージェント
- 機能の追加と削除を行う前に、同一サーバ上で動作しているInterstage運用APIを使用するアプリケーションプログラムを停止してください。

上記の作業を行った上で、再度、機能の追加と削除を実施してください。

### Interstageの機能の追加と削除時にエラーが発生する

以下のエラーメッセージダイアログが発生する場合、イベントビューアが起動されている可能性があります。

```
isjmxwrpd.exe - エントリポイントが見つかりません
```

```
プロシジャエントリポイント otm_msggetがダイナミックリンクライブラリlibextpmsg.dllから見つかりません
```

インストール作業を行う場合は、すべてのWindows(R)上のアプリケーションを終了させてからインストールを実施してください。

なお、本現象が発生しても、エラーメッセージダイアログを終了し、その後コンピュータの再起動を実施することで、運用を継続することができます。

Intersgageを再インストールする必要はありません。

### Interstageの機能の追加が失敗する

以下のエラーメッセージダイアログが発生する場合、機能追加対象のサーバが管理対象サーバとなっている可能性があります。

管理対象サーバへは機能の追加と削除インストールができません。  
サイトより削除してから再度インストールしてください。

インストール作業を行う場合は、機能追加対象のサーバをサイトから削除してからインストールを実施してください。

**Solaris32** **Linux32/64**

### Interstageの機能の追加が失敗する

以下のエラーメッセージが出力される場合、機能追加対象のサーバが管理対象サーバとなっている可能性があります。

管理対象サーバへはカスタムインストールによる機能の追加ができません。  
サイトから削除してから再度インストールしてください。

インストール作業を行う場合は、機能追加対象のサーバをサイトから削除してからインストールを実施してください。

## 2.4 サーバマシン起動時の異常

ここでは、Interstageのインストール後、サーバマシン起動時に発生したトラブルの事例およびその対処について説明します。

### サーバマシン起動時に、イベントログまたはsyslogにメッセージが出力される

マシン起動時に、イベントログまたはsyslogに以下のメッセージが出力された場合、次に示す2つの原因が考えられます。

**Windows32/64**

F3FMjs2su: ERROR: 2003: Servlet service failed in the start.

**Solaris32/64** **Linux32/64**

FJSVjs2su: ERROR: 2003: Servlet service failed in the start.

- Interstage管理コンソール用Servletサービスで使用するJServlet環境定義ファイルのJDK/JREのパスに誤りがあります。
- Interstage管理コンソール用ServletサービスのServletコンテナで使用するポート番号が使用中です。

以下の対処を行ってください。

#### Interstage管理コンソール用Servletサービスで使用するJServlet環境定義ファイルのJDK/JREのパスに誤りがある場合の対処

1. Interstage管理コンソールを使用するために必要なサービスを停止します。詳細については、“運用ガイド(基本編)”の“Interstage管理コンソールの起動・停止”を参照してください。
2. JServlet環境定義ファイルの定義項目opmanager.bin、opmanager.envを、JDK/JREをインストールしたパスに従って修正してください。また、JREの場合には、opmanager.envでJDKの“tools.jar”を定義しないでください。
3. Interstage管理コンソールを使用するために必要なサービスを起動します。詳細については、“運用ガイド(基本編)”の“Interstage管理コンソールの起動・停止”を参照してください。

#### Interstage管理コンソール用ServletサービスのServletコンテナで使用するポート番号が使用中である場合の対処

“運用ガイド(基本編)”の“Interstage管理コンソール環境のカスタマイズ”を参照して、Interstage管理コンソール用Servletサービスのポート番号を未使用のポート番号に変更してください。

#### JDKまたはJREの入れ替えを行う場合の対処

Interstageのインストール後、JDKまたはJREの入れ替えを行った場合には、“運用ガイド(基本編)”の“Interstage管理コンソール環境のカスタマイズ”の“JDKまたはJREの入れ替えを行う場合の対処”を参照して対処してください。

## 2.5 アンインストール直後の起動時の異常

---

### アンインストール後のOS再起動に失敗する場合の対処 [Linux32/64](#)

/etc/profileにInterstageのコマンドが設定されている場合、OSの起動に失敗する場合があります。

/etc/profileにInterstageのコマンドが設定されているかどうかを確認し、設定されている場合は、該当のコマンドを/etc/profileから削除してください。

## 第3章 Interstage初期化時の異常

### 3.1 Interstage初期化時

ここでは、Interstageの初期化時に発生したトラブルの事例およびその対処について説明します。

#### 参照

Interstageの初期化方法については、“運用ガイド(基本編)”を参照してください。

#### isinitコマンドが異常終了する

isinitコマンドが、od10940のメッセージを出力して異常終了する場合は、CORBAサービスが使用するポート番号が、他のプログラムですでに使用中であることが考えられます。

CORBAサービスが使用するポート番号に、未使用のポート番号を指定し、再度、Interstageの初期化を行ってください。CORBAサービスが使用するポート番号は、以下のどちらかの方法で指定します。

- Interstage動作環境定義のCorba Port Number
- config(CORBAサービスの動作環境ファイル)のIIOP\_port

#### CORBA通信に使用するポート番号の定義値が有効とならない Solaris32/64

CORBA通信に使用するポート番号の定義値が有効とならない場合は、CORBA通信に使用するポート番号の設定内容を見直す必要があります。

CORBA通信に使用するポート番号には、以下の3種類の定義方法があります。

1. Interstage動作環境定義の定義項目「Corba Port Number」に対する設定
2. CORBAサービスの動作環境ファイル(config)の定義項目「IIOP\_port」に対する設定
3. /etc/servicesに定義したodserverに対する設定

上記3種類の定義は、1.の優先度が最も高く、2.、3.の順序で続きます。

したがって、いずれかの定義方法でポート番号を定義しても、より優先度の高い定義項目に定義値が入力されていた場合、そちらが有効となりますので注意が必要です。

CORBA通信に使用するポート番号の定義値が有効とならない場合には、より優先度の高い定義項目に、定義値が設定されていないか確認してください。

#### Interstage初期化時にis30141のメッセージが出力される異常 Solaris32/64

isinitコマンドの実行画面に以下のメッセージが出力されている場合は、再度、isinitコマンドを実行してください。

```
Daemon start fails
```

#### Interstage初期化時にis30365のメッセージが出力される異常 Solaris32

マルチシステム機能を使用している場合、本コマンドと同時にisdeletesysコマンドまたはiscreatesysコマンドを実行した可能性があります。この場合、isdeletesysコマンドでシステムを削除してください。

### Interstage初期化時にots9020のメッセージが出力される異常 Solaris32

OTSの初期化時にシステムログファイルの削除に失敗しました。/etc/opt/FSUNots/sysconfigファイルが何らかの理由で削除、または破壊されている可能性があります。Interstage動作環境定義ファイルを参照して、「OTS」から始まる項目を参照し、手動でsysconfigファイルを再作成してください。セットアップ情報ファイルと同じ形式で作成できます。

### Interstage初期化時にots9019のメッセージが出力される異常 Solaris32

Interstageをtype2で初期化する際、OTSの初期化が行われていないにも関わらず、/etc/opt/FSUNots/sysconfigファイルが存在しています。/etc/opt/FSUNots/sysconfigファイルをリネームした後に、再度Interstageの初期化を行ってください。再度Interstageの初期化を行っても正しく完了しない場合は、以下のコマンドを実行してください。

```
OD_start
CosNaming_s &
otssetup -f リネームしたsysconfigファイル名
OD_stop
isinit -f type2
```

### Interstage初期化時にots9061のメッセージが出力される異常 Solaris32

システムログファイルだけが残った状態になっています。以下のコマンドを実行して復旧してください。

```
isinit -w type2
```

## メモリ不足の異常

Interstageを運用中に、「メモリ不足が発生しました」というメッセージが表示されることがあります。しばらくしてから再度処理を実行してください。

このメッセージが、頻繁に出力される場合は、「チューニングガイド」を参考にして、Interstageを運用するために必要なメモリ量を再見積もりし、メモリが十分に用意されているかを確認してください。

Interstageの必要とするメモリ量に対して、メモリが十分に用意されている場合は、他のアプリケーションの使用メモリ量が不足している可能性があります。同一マシン上で運用している他のアプリケーションについても再見積もりを行い、メモリ量が適当であるかを確認してください。

再見積もりの結果、メモリ量が不足していた場合は、システム管理者に連絡してください。システム管理者は、メインメモリの増設、またはWindows(R)の場合は仮想メモリのページファイルを、Solaris/Linuxの場合はスワップ領域を拡張してください。

### 再インストール直後の異常 Solaris32/64 Linux32/64

Interstageの各パッケージの再インストールを行った直後に、Interstageの動作が不安定な場合、前回のアンインストール時に削除されなかったファイルが悪影響をおよぼしている可能性があります。

再度アンインストールを行い、以下のディレクトリが残留している場合は、削除してから、再インストールを行って操作を続行してください。

```
/etc/opt/PKGNAME
/var/opt/PKGNAME
/opt/PKGNAME
```

PKGNAME: アンインストールしたパッケージ名

## 第4章 Interstage管理コンソール操作中の異常

### 4.1 Interstage管理コンソール起動時の異常

ここでは、Interstage管理コンソールを使用した場合のトラブル事例およびその対処について説明します。Interstage管理コンソールで出力されたエラーの対処が、“メッセージ集”を参照しても、対処ができない場合や、本事例に該当しない事象となった場合は、技術員に連絡してください。

#### スタートメニューまたはショートカットからInterstage管理コンソールを起動すると、Interstage管理コンソールの起動に失敗する

本現象が発生した場合は、Interstage管理コンソールが使用していたWebブラウザのプロセスが残存している可能性があります。以下のいずれかの対処を行うか、または以下の回避方法により、Interstage管理コンソールを起動してください。

##### ■対処方法

- ・ 本現象が発生したシステムを再起動します。
- ・ 以下の手順で、残存しているWebブラウザのプロセスを終了します。
  1. Windows(R)の「タスクマネージャ」を起動します。
  2. [プロセス]タブをクリックします。
  3. イメージ名「iexplore.exe」をクリックします。
  4. [プロセスの終了]をクリックします。



##### 注意

Webブラウザ以外のプロセスを終了させることがないように、十分に注意してください。

##### ■回避方法

以下の手順で、Interstage管理コンソールを起動します。

##### Internet Explorer 8/9/10の場合

1. Webブラウザを起動します。
2. 1.で起動したWebブラウザの[ファイル]メニューの[新規セッション]をクリックして、新しくWebブラウザを起動します。
3. 2.で起動したWebブラウザで、Interstage管理コンソールのURLを指定します。

##### Internet Explorer 7の場合

1. Webブラウザを起動します。
2. Interstage管理コンソールのURLを指定します。

#### Interstage管理コンソールの起動時に、以下のエラーメッセージが出力されてログイン画面が表示できない

##### ■Included servlet error: 500/Internal Servlet Error

本現象が発生した場合は、Webブラウザの設定に誤りがある可能性があります。以下の対処を行ってください。

1. Internet Explorerの[ツール]メニューの[インターネット オプション]で[言語]ボタンをクリックします。

2. 日本語[ja]または英語[en]が存在するかを確認します。
3. 2.で、日本語[ja]および英語[en]がどちらも存在しない場合は、[追加]ボタンをクリックし、日本語[ja]または英語[en]を選択して追加します。
4. Webブラウザを再起動し、Interstage管理コンソールに再接続します。

**■無効な証明書を受信しました。あなたの証明書は、認証局によって発行された別の証明書と同じシリアル番号を持っています**

本現象が発生した場合は、WebブラウザにInterstage管理コンソール用の証明書が登録されている可能性があります。以下の対処を行ってください。

1. Internet Explorerの[ツール]メニューの[インターネット オプション]で[コンテンツ]タブを選択します。
2. [証明書]欄の[証明書]ボタンをクリックします。
3. [信頼されたルート証明期間]タブを選択し、以下と一致する証明書がないかを確認します。  
発行先: Interstage Application Server  
発行者: Interstage Application Server
4. 3.で、該当する証明書がある場合は、その証明書を選択し、[削除]ボタンをクリックします。
5. Webブラウザを再起動し、Interstage管理コンソールに再接続します。

**■Internet Explorerではこのページは表示できません**

**■Internal Server Error**

本現象が発生した場合は、以下の原因が考えられます。必要に応じた対処を行ってください。

- 接続先サーバのネットワークが過負荷の状態である可能性があります。netstatコマンドなどでサーバのネットワーク状態を確認してください。
- 指定したURLに誤りがある可能性があります。指定するURLは、本製品のインストール時に設定したInterstage管理コンソールにおけるSSL暗号化通信の使用の有無により異なります。SSL通信の使用の有無により以下のURLを指定してください。

	URL	備考
SSL暗号化通信を使用しない場合	http://ホスト名: ポート番号/ IsAdmin/	[ホスト名] 本製品をインストールしたサーバのホスト名/IPアドレス [ポート番号] 本製品のインストール時に設定したInterstage管理コンソール用の
SSL暗号化通信を使用する場合	https://ホスト名: ポート番号/ IsAdmin/	Interstage HTTP Serverのポート番号(デフォルト:12000)

- Interstage管理コンソールが動作するために必要なサービスが起動されていない可能性があります。

**Windows32/64**

Windows(R)のサービスから、以下のサービスが起動しているかを確認し、起動されていないサービスがある場合は、そのサービスを起動してください。

- Interstage JServlet (OperationManagement)
- Interstage Operation Tool
- Interstage Operation Tool (FJapache)

上記のすべてのサービスが起動されていない場合は、「Interstage Operation Tool」を起動してください。「Interstage Operation Tool (FJapache)」および「Interstage JServlet (OperationManagement)」は、「Interstage Operation Tool」と依存関係があるため、「Interstage Operation Tool」を起動することによりすべて起動できます。

**Solaris32/64 Linux32/64**

以下のサービスが起動しているかを確認してください。

- Interstage JMXサービス
- Interstage管理コンソール用Servletサービス
- Interstage HTTP Server (Interstage管理コンソール用) (注)

起動していない場合は、以下のコマンドを実行してください。

1. Interstage JMXサービスを起動します。

```
# /opt/FJSvisjmx/bin/isjmxstart
```

2. Interstage管理コンソール用Servletサービスを起動します。

```
# /opt/FJSVjs2su/bin/jssvstart
```

3. Interstage HTTP Server (Interstage管理コンソール用)を起動します。

```
# /opt/FJSVihb/bin/httpd -f /etc/opt/FJSVisgui/httpd.conf -s "#ISCONSOLE" -K
```

**注)** Interstage HTTP Server (Interstage管理コンソール用)の起動状態は、システムログ上に以下の情報メッセージが出力されているにより確認します。

以下の情報メッセージが出力されている場合、Interstage HTTP Server (Interstage管理コンソール用)は正常に起動しています。

以下の情報メッセージが出力されていない場合、Interstage HTTP Server (Interstage管理コンソール用)は起動していません。

```
UX:IHS: INFO: ihs00001: HTTP Server started. [#ISCONSOLE]
```

Interstage HTTP Server (Interstage管理コンソール用)が起動していない場合は、システムログ上にエラーメッセージが出力されていないかを確認してください。

エラーメッセージが出力されている場合は、出力されたエラーメッセージおよびInterstage HTTP Serverのエラーログを参照し、失敗の原因を取り除いてください。エラーメッセージについては“メッセージ集”の“メッセージ番号がihsで始まるメッセージ”を、エラーログについては“メッセージ集”の“Interstage HTTP Serverのエラーログ”を参照してください。エラーメッセージが出力されていない場合は、Systemwalker Centric Managerにおいて、サーバ操作制御機能の録画機能を利用してswexecコマンドの実行を録画していないかを確認してください。サーバ操作制御の録画機能を利用してswexecコマンドの実行を録画している場合は、swexecコマンドの実行を録画しないように設定した後、Interstage HTTP Server (Interstage管理コンソール用)を起動してください。サーバ操作制御機能の録画機能およびswexecコマンドについては、Systemwalker Centric Managerのマニュアルを参照してください。

## OS/サービス起動時、イベントビューアにエラーメッセージが出力されることがある Windows32/64

OS/サービス起動時、イベントビューアのシステムログに以下のメッセージが出力された場合、OS/サービス起動時のシステム負荷が一時的に高かった可能性があります。

```
xxx (注) サービスは起動時に停止しました。
```

```
xxx (注) サービスは開始時にハングしました。
```

```
xxx (注) サービスは次のエラーのため開始できませんでした。
指定されたサービスは無効であるか、または有効なデバイスが関連付けられていないため、開始できません。
```

```
xxx (注) サービスへの接続中にタイムアウト (30000 ミリ秒) になりました。
```

**注)** xxxは、以下のいずれかのサービスです。

- Interstage Operation Tool
- Interstage Operation Tool(FJapache)
- Interstage JServlet(OperationManagement)

イベントビューアのアプリケーションログを参照して、エラーメッセージに表示されたサービスが正常に起動されているかを確認してください。

サービスが正常に起動されている場合は、対処する必要はありません。本メッセージは無視して、操作を継続してください。

サービスが起動されていない場合は、手動でサービスを起動してください。サービスの起動に失敗した場合は、出力されたメッセージを確認し、“メッセージ集”を参照して対処してください。

### Interstage管理コンソールで[ヘルプ]ボタンを押下すると、エラーが発生する

以下のエラーメッセージダイアログが発生する場合、対象外のWebブラウザを使用している可能性があります。利用可能なWebブラウザについては、“インストールガイド”を参照してください。

```
ランタイムエラーが発生しました。
デバッグしますか？

行 : xx
エラー : 'undefined' は宣言されていません。
```

### Interstage Java EE管理コンソールで[コンソール切替]ボタンを押下すると、Interstage管理コンソールでログインエラーis20760が発生する

本現象が発生した場合、使用しているアカウントでは、GUI統合運用が使用できない可能性があります。Interstage管理コンソールにログイン可能なアカウントであるかを確認してください。ログイン可能なアカウントではない場合、以下の対処／回避を行ってから、Interstage管理コンソールを使用してください。

#### ■ 対処方法

Interstage Java EE管理コンソールおよびInterstage管理コンソールにログイン可能なアカウントを作成します。異なるアカウントでは、GUI統合運用を使用できません。

#### ■ 回避方法

異なるアカウントでInterstage Java EE管理コンソールおよびInterstage管理コンソールを操作する場合は、[コンソール切替]ボタンを使用しないで、それぞれ別のWebブラウザで管理コンソールを起動してください。

#### 注意

Interstage Java EE管理コンソールで[コンソール切替]ボタンを押下すると、Interstage管理コンソールでログインエラーis20760以外のエラーが発生する場合は、「[4.2 ログイン時の異常](#)」を参照して対処してください。

## 4.2 ログイン時の異常

### Interstage管理コンソールにログインすることができない、またはボタンを押下しても画面が遷移しない

本現象が発生した場合は、Webブラウザの設定に誤りがある可能性があります。以下の事項を確認してください。

- Interstage管理コンソールにログインするには、対象のサービシステムのユーザ名とパスワードを指定する必要があります。正しいユーザ名とパスワードが指定されているかを確認してください。
- Interstage管理コンソールにログインするには、利用するWebブラウザの設定において以下を設定してください。
  - プロキシサーバを利用しない
  - Javaスクリプトを有効にする
  - スタイルシートを有効にする

ー 一時ファイルのバージョンの確認を行う

## ログイン時に、以下のエラーメッセージが出力されてInterstage管理コンソールのログインに失敗する

### ■IS: エラー: is40010: 内部論理矛盾が発生しました(エラー情報1.1314)

本現象が発生した場合は、Interstage Operation ToolサービスおよびInterstage JServlet (OperationManagement) サービスのログインアカウントを変更していないかを確認してください。デフォルトの「ローカルシステムアカウント」が設定されている場合に、本現象が発生することがあります。

確認方法は、[コントロールパネル]ー[管理ツール]ー[サービス]においてサービスのプロパティを確認し、ログオンタブで「ローカルシステムアカウント」が設定されているかを確認してください。

### ■IS:エラー:is40010: 内部論理矛盾が発生しました。(エラー情報=1, 1385) Windows32/64

本現象が発生した場合、Interstageがインストールされているオペレーティングシステムへのアクセスが許可されていない可能性があります。オペレーティングシステムにログイン可能なアカウントを使用して、Interstage管理コンソールにログインしてください。

### ■IS: エラー: is40003: Interstage JMX サービスに接続できませんでした

Interstage管理コンソールにログインする際にLANが切断されている場合は、ログイン時にエラーメッセージis40003が出力されます。再度ログイン画面を表示し、ログインしてください。

### ■IS: エラー: is20760: ユーザ認証に失敗しました KIND=OS DETAIL=IS: エラー:is40022: ユーザアカウントに対するセキュリティ設定によりログインが拒否されました。

本現象が発生した場合は、パスワードが設定されていない可能性があります。ユーザアカウントにパスワードが設定されているかを確認してください。

#### Solaris32/64 Linux32/64

セキュリティモードが強化セキュリティモードの場合は、管理者権限をもつユーザ、またはInterstage運用グループに所属するユーザでログインしてください。

### ■内部矛盾が発生しました。処理を中止します。iscollectinfoコマンドを使用して調査情報を採取した後、エラー情報を控えて技術員に連絡してください。

本現象が発生した場合は、セッションタイムアウトが発生した後に、WebブラウザのURLフィールドにて直接ログイン画面のURLを指定した可能性があります。エラーメッセージが出力されたWebブラウザを閉じて、新しいWebブラウザを起動して実行してください。

## ログイン後にブランクページが表示され、そのあとの操作ができない

本現象が発生した場合は、Interstage管理コンソールを動作させるために必要なWebブラウザの設定が無効になっている可能性があります。

以下の手順で、Interstage管理コンソールが動作しているゾーンのWebブラウザの設定を有効にしてください。

1. Interstage管理コンソールを表示した状態で、Webブラウザの[ツール]メニューの[インターネット オプション]で、[セキュリティ]タブをクリックします。  
この時点で選択されているゾーンを変更しないでください。
2. [レベルのカスタマイズ]をクリックします。
3. [スクリプト]セクションで、[アクティブスクリプト]を「有効にする」に変更します。
4. Webブラウザを再起動し、Interstage管理コンソールにログインします。

## 4.3 Interstage管理コンソールの統計情報の異常

Interstage管理コンソールに表示される以下の統計情報が、正しく表示されない場合があります。

- IJServer: モニタ
- Webアプリケーション: モニタ
- EJBアプリケーション: モニタ

統計情報が正しく表示されない場合の原因と対処を説明します。

- Oracleのコネクションプーリングを使用するデータソースを使用し、データソース情報が正しく出力されない場合、サポートされていないJDBCドライバを使用している可能性があります。JDBCドライバのバージョンを確認してください。JDK/JRE6を使用する場合は、Oracle Database 11g Release 1、またはOracle Database 11g Release 2のJDBCドライバを使用してください。
- IJServerのモニタにおいて、Servletコンテナ情報のスレッドプールの値に異常な値(負の値)が出力された場合、IJServerで資源の枯渇などの異常が発生している可能性があります。コンテナログおよび起動情報 (info.log)を参照し、出力されているメッセージに従って対処してください。

## 4.4 画面の異常

### Interstageのセットアップが失敗したあとにセットアップログを表示したウィンドウを閉じないで操作を続けると、正常に画面が遷移できない

本現象が発生した場合は、セットアップログが表示されているウィンドウを閉じてから操作を続行するようにしてください。

### Interstage管理コンソールの一部の文字が稀に文字化けすることがある

本現象が発生した場合は、以下のどちらかの対処を行ってください。

- 画面を再度読み込みます。
- Webブラウザのエンコード指定を「日本語(自動選択)」に指定して、再度読み込みます。

### Interstage管理コンソールの操作中に、以下のエラーメッセージが出力されて、そのあとの操作ができない

■Internet Explorerではこのページは表示できません

■Internal Server Error

本現象が発生した場合は、Interstage管理コンソールが動作するために必要なサービスが起動されていない可能性があります。確認方法を参照し、必要なサービスが起動されているかを確認してください。

### 注意

- [コンソール切替]ボタンの押下時、または[Interstage Java EE管理コンソール]のツリーノードのクリック時に、本現象が発生した場合は、起動状態にかかわらず、サービスを再起動してください。
- 本現象は、ウィンドウ全体ではなく、フレーム単位に発生することがあります。

### ■確認方法

 Windows32/64

Windows(R)のサービスから、以下のサービスが起動しているかを確認し、起動されていないサービスがある場合は、そのサービスを起動してください。

- Interstage JServlet (OperationManagement)
- Interstage Operation Tool
- Interstage Operation Tool (FJapache)

上記のすべてのサービスが起動されていない場合は、「Interstage Operation Tool」を起動してください。  
「Interstage Operation Tool (FJapache)」および「Interstage JServlet (OperationManagement)」は、「Interstage Operation Tool」と依存関係があるため、「Interstage Operation Tool」を起動することによりすべて起動できます。

**Solaris32/64 Linux32/64**

以下のサービスが起動しているかを確認してください。

- Interstage JMXサービス
- Interstage管理コンソール用Servletサービス
- Interstage HTTP Server (Interstage管理コンソール用)

起動していない場合は、以下のコマンドを実行してください。

1. Interstage JMXサービスを起動します。

```
# /opt/FJSVisjmx/bin/isjmxstart
```

2. Interstage管理コンソール用Servletサービスを起動します。

```
# /opt/FJSVjs2su/bin/jssvstart
```

3. Interstage HTTP Server (Interstage管理コンソール用)を起動します。

```
# /opt/FJSVihs/bin/httpd -f /etc/opt/FJSVisgui/httpd.conf -s "#ISCONSOLE" -K
```

再起動する場合は、以下のコマンドでサービスを停止してから、起動してください。

1. Interstage JMXサービスを停止します。

```
# /opt/FJSVisjmx/bin/isjmxstop
```

2. Interstage管理コンソール用Servletサービスを停止します。

```
# /opt/FJSVjs2su/bin/jssvstop
```

3. Interstage HTTP Server (Interstage管理コンソール用)を停止します。

```
# kill `cat /var/opt/FJSVisgui/tmp/httpd.pid`
```

また、該当するサービスのログファイルなどを参照し、問題が発生していないかを確認してください。  
ログファイルなどを参照しても問題が解決できない場合は、iscollectinfoコマンドを使用して調査情報を採取した後、技術員に連絡してください。

### Interstage管理コンソールの操作中に、無応答となる または、ブランクページが表示される

本現象が発生した場合は、以下の原因が考えられます。

- 画面の描画が完了する前に、次の操作を行った可能性があります。Interstage管理コンソールの操作は、画面の描画が完了してから次の操作を行う必要があります。
- サーバのシステム時間を大幅に戻した可能性があります。

以下の対処を行ってから操作してください。

1. 現在作業中のWebブラウザを閉じます。
2. 新しくWebブラウザを起動します。

3. 再度Interstage管理コンソールにログインします。

### Interstage管理コンソールの操作中、セッションタイムアウト時間に到達する前にセッションタイムアウトが発生する

本現象が発生した場合は、サーバのシステム時間を大幅に進めた可能性があります。  
再度Interstage管理コンソールにログイン後、操作を行ってください。

なお、Webブラウザには、以下のエラーが発生します。

Internet Explorerではこのページは表示できません。
-----------------------------------

### Interstage管理コンソールから操作を行うと、画面が遷移しない

本現象が発生した場合は、Interstage管理コンソールを動作させるために必要なWebブラウザの設定が無効になっている可能性があります。

以下の手順で、Interstage管理コンソールが動作しているゾーンのWebブラウザの設定を有効にしてください。

1. Interstage 管理コンソールを表示した状態で、Webブラウザの[ツール]メニューの[インターネット オプション]で、[セキュリティ]タブをクリックします。  
この時点で選択されているゾーンを変更しないでください。
2. [レベルのカスタマイズ]をクリックします。
3. [スクリプト]セクションで、[アクティブスクリプト]を「有効にする」に変更します。
4. Webブラウザを再起動します。
5. Interstage管理コンソールにログインします。

### Interstage管理コンソールのツリー部だけが英語で表示される

本現象が発生した場合は、エンコードに日本語以外の言語を指定したWebブラウザを使用してInterstage管理コンソールにアクセスした可能性があります。

以下の手順で回避してください。

1. Interstage管理コンソールのサービスを再起動します。
2. エンコードに日本語を指定したWebブラウザを使用してInterstage管理コンソールにアクセスします。

### 以下のエラーメッセージが出力されて、画面が更新されない

#### ■IS: エラー: is40003: Interstage JMX サービスに接続できませんでした

Interstage管理コンソールにログイン後、Interstage Operation Toolを再起動した場合、またはLANを切断した場合、Interstage管理コンソールのその後の処理中にエラーメッセージis40003が出力されることがあります。  
再度ログイン画面を表示し、ログインしてください。

### ステータスバーに文字列が表示されない

本現象が発生した場合は、Interstage管理コンソールを動作させるために必要なWebブラウザの設定が無効になっている可能性があります。

以下の手順で、Interstage管理コンソールが動作しているゾーンのWebブラウザの設定を有効にしてください。

1. Interstage 管理コンソールを表示した状態で、Webブラウザの[ツール]メニューの[インターネット オプション]で、[セキュリティ]タブをクリックします。  
この時点で選択されているゾーンを変更しないでください。
2. [レベルのカスタマイズ]をクリックします。

3. [スクリプト]セクションで、[スクリプトでのステータスバーの更新を許可する]を「有効にする」に変更します。

## 注意

セキュリティ設定を変更するゾーンに信頼できないWebサイトが含まれる場合は、本設定を変更しないでください。

## ズーム機能を使用した場合、画面表示が崩れる

ズーム機能を使用した場合、画面表示が崩れる場合があります。画面表示が崩れる場合は、標準倍率で使用してください。

## Interstage管理コンソールの画面遷移および操作に異常が発生する

### ■Internet Explorer 8/9/10において、複数のWebブラウザから操作している場合

1つの端末から複数のInterstage管理コンソールを操作することはできません。本操作を行った場合、Interstage管理コンソールの画面の遷移、および操作に異常が発生する可能性があります。

1つの端末から複数のInterstage管理コンソールを操作する場合は、以下の対処を行ってください。

- Internet Explorerの[ファイル]メニューの[新規セッション]をクリックして、新しくWebブラウザを起動します。
- Internet Explorer 7を使用して、Interstage管理コンソールを操作します。

### ■タブブラウズ機能を使用している場合

1つのWebブラウザ上で、タブブラウズ機能を使用して、複数のタブからInterstage管理コンソールを起動することはできません。本操作を行った場合、Interstage管理コンソールの画面の遷移、および操作に異常が発生する可能性があります。Interstage管理コンソールを使用する場合は、複数のタブを使用しないでください。

また、複数のInterstage管理コンソールを起動する場合は、セッションが異なる複数のWebブラウザを起動してInterstage管理コンソールを操作してください。

### ■タブブラウズ機能を使用していない場合

Internet Explorerにおいて、ショートカットの起動時にウィンドウを再使用する設定が無効になっている可能性があります。以下の手順で、ショートカットの起動時にウィンドウを再使用する設定を有効にしてください。

1. Internet Explorerの[メニュー]—[ツール]—[インターネットオプション]—[詳細設定]で、[ショートカットの起動時にウィンドウを再使用する(タブブラウズが無効である場合)]のチェックボックスが選択されているかを確認します。
2. 1.で、[ショートカットの起動時にウィンドウを再使用する(タブブラウズが無効である場合)]のチェックボックスが選択されていない場合は、チェックボックスを選択します。

## Interstage管理コンソールの画面表示中に、画面が崩れて表示される

本現象は、特に問題ありません。使用中のWebブラウザまたはオペレーティングシステムの状況により、画面レンダリングに時間がかかることがあります。この場合、表示中の画面が画面表示完了後の画面と異なったり、一時的に崩れたり、存在しない項目が表示されたりすることがあります。



## Interstage管理コンソールのツリー部に、ワークユニットのアイコン「+」が表示されない

アプリケーションのJavaヒープ領域の不足により、Interstage管理コンソールの描画処理に失敗した可能性があります。アプリケーションが使用しているJavaヒープ領域が不足していないかを確認し、必要に応じて、Javaヒープ領域の設定値をチューニングしてください。

## Interstage管理コンソールに[Interstage Java EE管理コンソール]のツリーノードが表示されない

本現象が発生した際に、[コンソール切替]ボタンが表示されていない場合は、Java EEがインストールされていない可能性があります。Java EEがインストールされているかを確認し、Java EEがインストールされていない場合は、Java EEをインストールしてください。

本現象が発生した際に、[コンソール切替]ボタンが表示されている場合は、GUI統合運用を使用しているためであり、正常な動作です。必要に応じて、以下の対処/回避を行ってください。

### ■対処方法

Interstage Java EE管理コンソールを操作する場合は、[コンソール切替]ボタンを押下して、管理コンソール画面を切り替えて操作してください。

### ■回避方法

表示しているInterstage管理コンソールをそのままの状態、Interstage Java EE管理コンソールを使用する場合は、新しくWebブラウザを起動して、Interstage Java EE管理コンソールにログインして操作してください。

EE SJE

## 4.5 エラーログ/エラーコードの出力

Interstage管理コンソールの[システム] > [ワークユニット] > [ワークユニット名] > [ログ参照]で、IJServerワークユニットのコンテナログのメッセージが文字化けする **Solaris32/64** **Linux32/64**

本現象が発生した場合は、コンテナログに出力されているメッセージの文字コードがInterstage JMXサービスの環境変数LANGの設定と適合していません。以下のどちらかの対処を行ってください。

- ・ワークユニットの環境変数LANGの値と、Interstage JMXサービスの設定値を合わせてください。
- ・アプリケーションでメッセージを出力している場合は、アプリケーションで出力している文字コードと、Interstage JMXサービスの環境変数LANGの設定を合わせてください。

ワークユニットの環境変数については、“運用ガイド(基本編)”の“ワークユニットの設計”を、Interstage JMXサービスの起動については、“運用ガイド(基本編)”の“Interstage管理コンソールの起動・停止”を参照してください。

## 4.6 その他の異常

EE SJE

Interstageの構成変更を実行した際に、ワークユニットの起動/停止メッセージが出力される

Interstage管理コンソールの[システム] > [環境設定]画面においてトランザクションサービス(OTS)の設定を変更した場合、構成変更の処理中に一時的にInterstageが起動されます。この際、自動起動を設定したワークユニットが一時的に起動されるためにメッセージが出力されています。

構成変更時に起動されたInterstageおよびワークユニットは、構成変更の完了時に停止します。

EE SJE

Interstage管理コンソールの[ワークユニット] > [ワークユニット名] > [配備]で、[配備]ボタンを押下後、配備が完了する前にタイムアウトが発生し、以下のメッセージが表示される

```
Gateway Time-out
The proxy server did not receive a timely response from the upstream server.
```

本現象が発生した場合は、Interstage管理コンソールのタイムアウト時間の設定、およびセッションタイムアウト時間の設定を変更後、再度実施してください。

Interstage管理コンソールのタイムアウト時間、およびセッションタイムアウト時間は、以下の手順で変更します。

**Windows32/64**

1. 以下のファイルで、Interstage管理コンソールのタイムアウト時間を設定します。

```
[Interstageインストールフォルダ]¥gui¥etc¥httpd.conf
```

2. 上記ファイルのTimeoutディレクティブに設定した値で設定します。単位は秒、指定可能な範囲は0～65535です。省略した場合は、「300」が設定されます。

```
Timeout 1800
```

3. 以下のファイルで、Interstage管理コンソールのセッションタイムアウト時間を設定します。

```
[Interstageインストールフォルダ]¥gui¥isAdmin¥WEB-INF¥web.xml
```

4. 上記ファイルの<session-timeout>タグに設定した値で設定します。単位は分、指定可能な範囲は1～35791394です。タグを省略した場合は、「30」が設定されます。

-35791394～-2の負の値を指定した場合、タイムアウトしません。

タイムアウトしない設定にした場合、Interstage管理者以外の第三者により不正操作される可能性が高くなり、セキュリティ上の問題が発生することが考えられるため、タイムアウトしない設定を使用する場合は注意してください。

```
<session-config>
  <session-timeout>30</session-timeout>
</session-config>
```

5. 以下のサービスを再起動します。
  - Interstage Operation Tool(FJapache)
  - Interstage JServlet(OperationManagement)

#### Solaris32/64 Linux32/64

1. 以下のファイルで、Interstage管理コンソールのタイムアウト時間を設定します。

```
/etc/opt/FJSVisgui/httpd.conf
```

2. 上記ファイルのTimeoutディレクティブに設定した値で設定します。単位は秒、指定可能な範囲は0～65535です。省略した場合は、「300」が設定されます。

```
Timeout 1800
```

3. 以下のファイルで、Interstage管理コンソールのセッションタイムアウト時間を設定します。

```
/opt/FJSVisgui/isadmin/WEB-INF/web.xml
```

4. 上記ファイルの<session-timeout>タグに設定した値で設定します。単位は分、指定可能な範囲は1～35791394です。タグを省略した場合は、「30」が設定されます。

-35791394～-2の負の値を指定した場合、タイムアウトしません。

タイムアウトしない設定にした場合、Interstage管理者以外の第三者により不正操作される可能性が高くなり、セキュリティ上の問題が発生することが考えられるため、タイムアウトしない設定を使用する場合は注意してください。

```
<session-config>
  <session-timeout>30</session-timeout>
</session-config>
```

5. Interstage HTTP Server (Interstage管理コンソール用)、およびInterstage管理コンソール用Servletサービスを、以下の手順で再起動します。

1. killコマンドで、Interstage HTTP Server (Interstage管理コンソール用)のプロセスを停止します。

```
# kill `cat /var/opt/FJSVisgui/tmp/httpd.pid`
```

2. Interstage HTTP Server (Interstage管理コンソール用) を起動します。

```
# /opt/FJSVihs/bin/httpd -f /etc/opt/FJSVisgui/httpd.conf -s "#ISCONSOLE" -K
```

3. 以下のコマンドで、Interstage管理コンソール用Servletサービスを再起動します。

```
# /opt/FJSVjs2su/bin/jssvstop  
# /opt/FJSVjs2su/bin/jssvstart
```

## 注意

- Interstage管理コンソールのタイムアウト時間の設定単位と、Interstage管理コンソールのセッションタイムアウト時間の設定単位は異なりますので、注意してください。
  - Interstage管理コンソールのタイムアウト時間: 秒単位
  - Interstage管理コンソールのセッションタイムアウト時間: 分単位
- Interstage管理コンソールのセッションタイムアウト時間には、Interstage管理コンソールのタイムアウト時間以上の値を設定してください。

EE SJE

## Interstage管理コンソールの[ワークユニット] > [ワークユニット名] > [配備]で[配備]ボタンを押下後、配備が完了する前にセッションタイムアウトが発生する

本現象が発生した場合は、セッションタイムアウト時間の設定を変更後、再度実施してください。  
Interstage管理コンソールのセッションタイムアウト時間は、以下の手順で変更します。

### Windows32/64

1. 以下のファイルで、Interstage管理コンソールのセッションタイムアウト時間を設定します。

```
[Interstageインストールフォルダ]¥gui¥isAdmin¥WEB-INF¥web.xml
```

2. 上記ファイルの<session-timeout>タグに設定した値で設定します。単位は分、指定可能な範囲は0~35791394です。タグを省略した場合は、「30」が設定されます。0または負の値を指定した場合、タイムアウトしません。

```
<session-config>  
  <session-timeout>30</session-timeout>  
</session-config>
```

3. Interstage JServlet(OperationManagement)サービスを再起動します。

### Solaris32/64 Linux32/64

1. 以下のファイルで、Interstage管理コンソールのセッションタイムアウト時間を設定します。

```
/opt/FJSVisgui/isadmin/WEB-INF/web.xml
```

2. 上記ファイルの<session-timeout>タグに設定した値で設定します。単位は分、指定可能な範囲は0~35791394です。タグを省略した場合は、「30」が設定されます。0または負の値を指定した場合、タイムアウトしません。

```
<session-config>  
  <session-timeout>30</session-timeout>  
</session-config>
```

3. 以下のコマンドで、Interstage JServlet(OperationManagement)を再起動します。

```
/opt/FJSVjs2su/bin/jssvstop  
/opt/FJSVjs2su/bin/jssvstart
```

**EE SJE** Interstage管理コンソールの[ワークユニット] > [ワークユニット名] > [配備]で[配備]ボタンを押下後、Interstage管理コンソール上にメッセージ「IS: 情報: is40201: 配備処理が完了しました(ファイル名=%s)」が表示されているが、Webブラウザのステータスバーに「配備処理中です。しばらくお待ちください。」の文字が表示された状態となる場合がある

本現象が発生した場合は、配備処理は完了しているため、次操作を行っても影響ありません。

**EE** Interstage管理コンソールの[ワークユニット] > [ワークユニット名] > [配備]で、ネットワークドライブ上のファイルを指定して[配備]ボタンを押下すると、配備処理に失敗する **Windows32/64 Solaris32 Linux32/64**

1. エラーメッセージ「IS: エラー: is40210: 指定されたファイルが見つかりません(ファイル名=%s)」が出力された場合は、配備対象ファイルの存在先として管理サーバを選択し、ネットワークドライブ上のファイルを指定した可能性があります。管理サーバ上にあるファイルを指定して再度実施してください。
2. エラーメッセージ「IS: エラー: is21106: 指定された配備ファイルが存在しません: 配備ファイルの正しいパスを指定してください(ファイル=%s)」が出力された場合は、配備対象ファイルの存在先として管理対象サーバを選択し、ネットワークドライブ上のファイルを指定した可能性があります。管理対象サーバ上にあるファイルを指定して再度実施してください。

**EE** Interstage管理コンソールが無応答状態になるか、画面右下に以下のいずれかのメッセージが出力された場合、操作している画面が正しく表示されないことがある **Windows32/64 Solaris32 Linux32/64**

- IS: エラー: is20711: 通信エラーが発生しました エラー情報=%s
- IS: エラー: is20725: サーバとの接続に失敗しました。エラー情報=%s
- IS: エラー: is20726: Interstage JMXサービスとの接続に失敗しました。エラー情報=%s
- IS: エラー: is20727: 通信エラーが発生しました 例外情報=%s1 例外メッセージ=%s2

管理対象サーバでメモリ不足が発生している可能性があります。しばらくしてから再度処理を実行してください。本現象が頻繁に出力される場合は、“チューニングガイド”を参照して、Interstageを運用するために必要となるメモリ量を再見積りし、メモリが十分に用意されているかを確認してください。Interstageの必要とするメモリ量に対して、メモリが十分に用意されている場合は、同一マシン上で運用している他のアプリケーションについても再見積りを実施し、メモリ量が適当であるかを調査してください。再見積りの結果、メモリ量が不足していることが確認された場合は、メインメモリの増設またはスワップ領域の拡張を行ってください。

**EE SJE** Interstage管理コンソールの[ワークユニット] > [ワークユニット名] > [配備]、および[リソース] > [connector] > [配備]で、[配備]ボタンを押下後、Interstage管理コンソール上にisAdmin Errorが表示される

Interstageをインストールしているサーバのテンポラリディレクトリのあるディスク容量が不足している可能性があります。不要なファイルを削除してディスクの空き容量を増やした後、再度処理を実行してください。

**EE SJE** Interstage管理コンソールの[ワークユニット] > [ワークユニット名] > [配備]で、配備ファイル入力後に[配備]ボタン、[リセット]ボタン、詳細設定[表示]を押下すると、以下の現象が発生する

- Webページに問題がある旨のダイアログが表示される
- ブラウザのステータスバーにページでエラーが発生した旨が表示される
- 何も動作しない

本現象が発生した場合は、Webブラウザにより、配備ファイルに指定した値が不正と判断された可能性があります。以下のいずれかの対処を行ってください。

- ・ 配備ファイルに実際に存在するファイルを指定します。

- Interstage管理コンソールがサポートする他のブラウザで操作します。
- 他のオペレーティングシステム、またはサービスパッケレベルの異なるマシン上で動作するWebブラウザを使用してInterstage管理コンソールを操作します。

### Interstage管理コンソールのリンク画面からツールが起動できない

以下の条件の場合、Interstage管理コンソールのツールノードからリンクされているツールが起動されない。

#### 【条件】

1. Interstage管理コンソールの操作端末にWindows(R) XPを利用している。かつ
2. Windows(R) XPにSP2以降を適用している。かつ
3. Internet Explorerを利用している。かつ
4. Internet Explorerのポップアップブロック機能を有効にしている。かつ
5. [システム]> [ツール]> [リンク] で表示されているツール名をクリックした場合

#### 【対処】

本現象が発生した場合、Internet Explorerの[ツール]の[ポップアップブロック]で[ポップアップブロックを無効にする]を選択し、一時的にポップアップ画面を有効にしてツールを起動してください。

### Interstage管理コンソールから操作を行うと、エラーが発生する

Interstage管理コンソールを使用して操作した際に、その操作が想定外のエラーとなった場合は、以下の操作を行った後、Interstage JMXサービス(Windows(R)版ではInterstage Operation Toolサービス)などのサービスを再起動していない可能性があります。

- Interstage資源のリストア／移入操作
- isinitコマンド、ismodifyserviceコマンド、isiniterviceコマンドを使用した初期化操作

上記の条件に該当する場合は、以下のサービスを再起動してから、再度、同様の操作を行ってください。

#### Windows32/64

- Interstage Operation Toolサービス
- Interstage JServlet(OperationManagement)サービス

#### Solaris32/64 Linux32/64

- Interstage管理コンソール用Servletサービス

### Interstage管理コンソールに日本語と英語のメッセージが混在して表示される

本現象が発生した場合は、Interstage Application Serverの起動時に指定している環境変数LANGの設定が、サービスごとに統一されていない可能性があります。

以下の起動コマンドの実行時に指定している環境変数LANGが、Interstage管理コンソールに表示したいメッセージの言語(日本語／英語)と一致しているかを確認してください。一致していない場合は、環境変数LANGを一致させた上で、そのサービスを再起動してください。

- isjmxstart
- isstart

## Interstage管理コンソールで、ファイルのダウンロードができない

本現象が発生した場合は、Interstage管理コンソールを動作させるために必要なWebブラウザの設定が無効になっている可能性があります。

以下の手順で、Interstage管理コンソールが動作しているゾーンのWebブラウザの設定を有効にしてください。

### Internet Explorer 9/10の場合

1. Interstage管理コンソールを表示した状態で、Webブラウザの[ツール]メニューの[インターネット オプション]で、[セキュリティ]タブをクリックします。  
この時点で選択されているゾーンを変更しないでください。
2. [レベルのカスタマイズ]ボタンをクリックします。
3. [ダウンロード]セクションで、[ファイルのダウンロード]を「有効にする」に変更します。

### Internet Explorer 7/8の場合

1. Interstage管理コンソールを表示した状態で、Webブラウザの[ツール]メニューの[インターネット オプション]で、[セキュリティ]タブをクリックします。  
この時点で選択されているゾーンを変更しないでください。
2. [レベルのカスタマイズ]ボタンをクリックします。
3. [ダウンロード]セクションで、[ファイルのダウンロード]および[ファイルのダウンロード時に自動的にダイアログを表示]を「有効にする」に変更します。



上記の対処後にダウンロードを行った際、ブラウザの上部にセキュリティに関する情報バー表示された場合は、情報バーに表示されている対処を行い、再度ダウンロードしてください。

## SSL暗号化通信中のファイルのダウンロード時に、ダウンロード先のファイル名と異なるファイル名でダウンロードされる

Internet Explorer 9使用時に本現象が発生した場合は、以下の手順で、Webブラウザの設定を変更して回避してください。

1. Internet Explorerの[ツール]メニューの[インターネット オプション]で、[詳細設定]タブをクリックします。
2. [セキュリティ]セクションで、[暗号化されたページをディスクに保存しない]のチェックボックスを選択しない設定に変更します。

## ファイルまたはログのダウンロード時に、以下の現象が発生し、ダウンロードができない

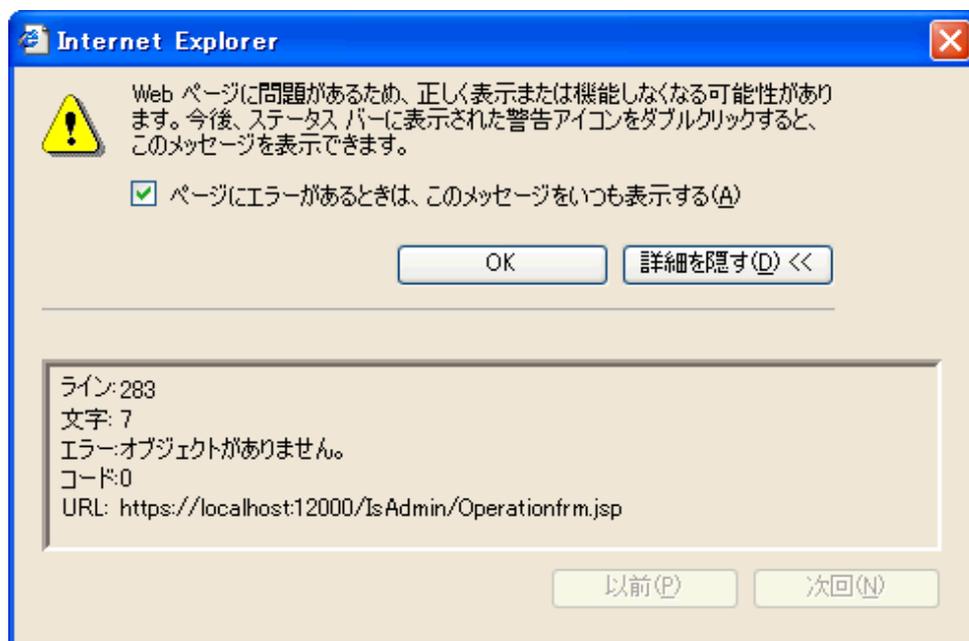
- 「xxxは移動または削除された可能性があります。」というダイアログが表示される
- 「xxxが見つかりません。名前を正しく入力したかどうかを確認してから、やり直してください。」というダイアログが表示される
- 何も動作しない

Internet Explorer 9使用時に、ファイルまたはログのダウンロード時のダイアログで[ファイルを開く]ボタンを押下した際に本現象が発生した場合は、以下の方法で対処してください。

1. ファイルまたはログのダウンロード時のダイアログで[保存]ボタンを押下し、いったんファイルを保存します。
2. 保存したファイルを参照します。

## ファイルまたはログのダウンロード時に、スクリプトエラーが発生する

本現象が発生した場合は、使用しているWebブラウザ端末の拡張子(.log)の設定に問題がある可能性があります。以下のようなスクリプトエラーを示すポップアップが表示されます。



以下の手順で回避してください。

1. レジストリエディタを起動します。
2. 以下のキーの値を変更します。  
[キー]  
HKEY\_CLASSES\_ROOT¥.log  
[値]  
Content Typeに「octet-stream」を設定します。

上記の設定を変更後、Interstage管理コンソールのログインより処理を再度実施し、ファイルまたはログのダウンロードを実施してください。

### 注意

ファイルまたはログのダウンロードを実施したあとは、変更した値を変更前の値に戻してください。本設定を使用している他製品の動作に影響が発生する可能性があるため、本設定を変更する場合は十分に注意してください。

EE SJE

## クライアントスタブJARファイルのダウンロード時に拡張子が「.zip」に変更される

クライアントスタブJARファイルの拡張子を「.zip」から「.jar」に変更してください。

## Interstage管理コンソールの操作中に、以下のエラーメッセージが出力される

■内部矛盾が発生しました。処理を中止します。iscollectinfoコマンドを使用して調査情報を採取した後、エラー情報を控えて技術員に連絡してください。

本現象が発生した場合は、以下の操作により、予期しないリクエストがサーバに通知された可能性があります。

- Interstage管理コンソール上でボタンの押下などの操作を行い、次の画面に遷移する前に他の操作を行った。
- Webブラウザが提供している以下の操作を行った。
  - 戻る
  - 進む
  - 中止
  - 更新
  - 移動

エラーメッセージが出力されたWebブラウザを閉じて、新しいWebブラウザを起動して操作してください。

EE SJE

## Interstage管理コンソールのツリーで[ワークユニット] > [ワークユニット名]をクリックした場合、V8.0互換モードまたはV8以前に作成されたIJSERVERでなくても以下のメッセージが出力されることがある

■このIJSERVERは、V8.0互換モードまたはV8以前に作成されたIJSERVERです。

このIJSERVER (IJSERVER配下を含む)の機能範囲は、V8.0相当となります。

他のユーザが該当のIJSERVERに対して操作を行っているため、IJSERVERの情報取得に失敗した可能性があります。

画面右下に原因となるメッセージが出力されています。“メッセージ集”を参照し、そのメッセージの対処を行ってください。

## アプリケーションログ(Windows(R)の場合)／システムログ(Solaris/Linuxの場合)に、以下のエラーメッセージが出力される

```
IS: エラー: is20760:ユーザ認証に失敗しました KIND=OS DETAIL=IS: エラー: is40021: ユーザ名もしくはパスワードが誤っています
```

上記のメッセージは、Interstage管理コンソールのログインに失敗したことを通知するエラーメッセージです。出力を抑止することはできません。

## [コンソール切替]ボタン押下時に、セッションタイムアウトが発生した旨のダイアログが表示される

本現象が発生した場合は、切替え先のInterstage Java EE管理コンソールでセッション切れ(セッションタイムアウト／サービスの再起動)が発生した可能性があります。

セッション切れが発生する前に、Interstage Java EE管理コンソールに切り替えてください。また、頻繁にセッションタイムアウトが発生する場合は、セッションタイムアウト時間に問題がないかを確認してください。

## ログアウト時に、セキュリティ情報に関する警告ダイアログが表示される

本現象が発生した場合は、GUI統合運用を行っているInterstage Java EE管理コンソールとInterstage管理コンソールの間で、SSL暗号化通信の使用有無の設定が異なっている可能性があります。

Interstage Java EE管理コンソールとInterstage管理コンソールの間で、SSL暗号化通信の使用有無の設定を合わせてください。

**[Interstage Java EE管理コンソール]ツリーノードからInterstage Java EE管理コンソールを起動した場合、証明書に問題があることを示すダイアログ／警告ページが表示される**

本現象が発生した場合は、製品インストール時に、SSL暗号化通信用に生成された証明書を利用している可能性があります。「使用上の注意」の「注意事項」－「Java EEの注意事項」－「Interstage Java EE管理コンソールの操作上の注意事項」を参照して、対処してください。

## 第5章 Interstage運用時の異常

Interstageのシステム異常発生時の対処について、以下の内容で説明します。

- ・ Interstageの起動/停止時の異常
- ・ 業務システム運用時の異常

### 5.1 共通事項

#### メモリ不足の異常

Interstageを運用中に、「メモリ不足が発生しました」というメッセージが表示されることがあります。通常は、しばらくしてから再度処理を実行してください。

このようなメッセージが、頻繁に出力される場合は、“チューニングガイド”を参考に、Interstageを運用するために必要なメモリ容量を再見積もりし、メモリが十分に用意されているかを確認してください。

#### Linux32/64

/var/log/messagesに以下のメッセージが出力されている場合は、システムのメモリ不足により、プロセスが強制終了された可能性があります。

```
Out of Memory: Killed process %s1(%s2).  
[可変情報]  
%s1 : プロセスID  
%s2 : プロセス名
```

Interstageの必要とするメモリ容量に対して、メモリが十分に用意されている場合は、他のアプリケーションの使用メモリ容量が不足していることが考えられます。同一マシン上で運用している他のアプリケーションについても再見積もりを実施し、メモリ容量に問題がないかを調査してください。

再見積もりの結果、メモリ容量が不足していることが確認された場合は、システム管理者に連絡してください。システム管理者は、メインメモリの増設、またはスワップ領域の拡張を行ってください。

#### ディスク容量不足が発生した場合

Interstageを運用中に、以下などのメッセージが表示された場合、ディスク容量の不足が発生しています。

ディスク容量が不足しています

ディスクが一杯になりました

ディスクの領域不足

空き領域がありません

No space left on device

この場合、“チューニングガイド”を参考に、Interstageを運用するために必要なディスク容量が用意されているかを確認してください。

また、Interstageの各サービスなどを再起動する場合、再起動前にディスク容量不足を解決してください。

#### リモートでサービスにアクセスできない

以下を確認してください。

- システムのネットワークの設定

システムファイアウォールの状態および、“使用上の注意”の“使用するポート番号の設定”で示しているサービスのファイアウォール設定

## 再インストール直後の異常 Solaris32/64 Linux32/64

Interstageの各パッケージの再インストールを行った直後に、Interstageの動作が不安定な場合、前回のアンインストール時に削除されなかったファイルが悪影響を及ぼしている場合があります。この場合、再度アンインストールを行い、以下のディレクトリが残留していた場合は削除してください。削除が終了したら、再度インストールを行って、操作を続行してください。

```
/etc/opt/PKGNAME
```

```
/var/opt/PKGNAME
```

```
/opt/PKGNAME
```

PKGNAME:アンインストールしたパッケージ名

## スタートメニューから実行した機能がすぐに異常終了する

Interstageをインストールしたマシンのディスプレイで、モニタの画面の色が適切に設定されていない場合、スタートメニューから機能を選択し実行しても、起動直後に異常終了する可能性があります。詳細については、“使用上の注意”の“モニタの画面の色について”を参照してください。

## Interstage管理コンソールに業務構成管理機能が管理するリポジトリの容量不足エラーが表示される

Interstageを停止後、Interstage管理コンソールのリポジトリの環境設定画面で、リポジトリの最大使用量を増加するかまたは無制限に変更してください。

## バージョン6以前で使用していた運用操作のバッチファイルを使用すると、運用操作が異常となる

### バージョン6以前で動作していたSystemwalker OperationMGRなどの運用管理製品から起動していたコマンドが異常となる Windows32/64

バージョン7から各サービスが提供しているコマンドおよびDLLファイルの格納先が変更されたため、旧バージョンで使用していたバッチファイルを使用すると、バッチで記載しているフォルダ配下に該当のコマンドが存在せず異常となる場合があります。

この場合、変更された格納先に応じて、バッチファイルを修正してください。

また、issetcompatiblepathコマンドにより、旧バージョンでのコマンドおよびDLLファイルの格納先を復元することができます。

コマンドおよびDLLファイルの格納先については、“移行ガイド”を参照してください。

issetcompatiblepathコマンドについては、“リファレンスマニュアル(コマンド編)”を参照してください。

## Interstageのコマンドが復帰しない、またはInterstage管理コンソールなどが応答しない場合 Solaris32/64 Linux32/64

Interstageのコマンド実行中に「Ctrl+Z」キーを押下、または「kill -s SIGSTOP プロセスID」を実行した場合、そのコマンドが停止します。コマンドの排他処理で停止している場合、コマンドが再開または終了するまで同じ排他資源を使用しているコマンドを実行できません。以下のコマンドを実行して、コマンドが停止しているかを確認してください。

#### Solaris32/64

```
# jobs
[1] + Suspended                OD_or_adm
```

#### Linux32/64

```
# jobs
[1]+  Stopped                  OD_or_adm
```

```
# ps -elf
F S UID          PID  PPID  C PRI  NI ADDR SZ WCHAN  STIME TTY          TIME CMD
:
x T root         xxxxx xxxxx  x  xx   x -  xxxx finish xx:xx pts/x   00:00:00 OD_or_adm
```

jobsコマンド／psコマンドのプロセスの状態コードが“T”で、Interstageのコマンド名が表示された場合は、停止状態であるため、bg／fgコマンドを実行してコマンドを再開します。コマンドの詳細については、オペレーティングシステムのマニュアルを参照してください。

## リストア直後の異常

Interstageの資源のリストア直後に、Interstageの動作が不安定となった場合、リストアが正しい手順で行われなかった可能性があります。

リストアは、Interstageのインストールディレクトリ、インストールドライブ(Windows (R)の場合)やリストア対象とならないすべての資源(Interstageの運用時に使用するユーザ資源、ユーザアプリケーションなど)が、バックアップ前と同じディレクトリ構成に配置されているなどの注意が必要です。

バックアップとリストアの手順については、“運用ガイド(基本編)”を参照してください。

## ファイルディスクリプタ不足の異常

Interstageを運用中にオープンしているファイル数が多すぎるため、使用しているサービス／アプリケーションで異常が発生する場合があります。ファイルディスクリプタは、一般的に以下の運用時に使用されます。

- ・ ファイルにアクセスした場合
- ・ ソケット通信を行う場合

ファイルディスクリプタ不足で異常が発生した場合は、不要なアプリケーションまたはInterstageに対する操作を終了し、システムで使用できるファイルディスクリプタ数を増やしてください。

#### Solaris32/64 Linux32/64

ulimitコマンドまたは/etc/systemファイルの定義値で利用できるファイルディスクリプタを増加させることもできます。ファイルディスクリプタのチューニング方法については、使用しているオペレーティングシステムのマニュアルを参照してください。

## 「<Windowsフォルダ>%hsperfdata\_SYSTEM%NNNN(NNNNは数字)」というファイルが原因でバックアップできない Windows32/64

このファイルは、Java VMがJavaアプリケーション動作時に作成する一時ファイルです。バックアップ対象からはずしてください。

また、Javaアプリケーション動作中に削除／移動を行うと、Javaを使用したシステム動作に影響を与えます。Javaアプリケーション動作中は、削除／移動を行わないでください。

## 5.2 Interstageの起動/停止時の異常

Interstageの起動/停止時に異常が発生した場合の対処方法について、以下に説明します。

### 注意

標準インストール/カスタムインストールでJava EE機能/ Webサーバをインストールし、以下の機能を使用しないでInterstageを運用している場合は、「5.2.1 Java EE機能/ Webサーバの使用時」を参照して対処してください。カスタムインストールで以下の機能をインストールしてInterstageを運用している場合は、「5.2.2 マルチ言語サービス/ J2EE互換機能の使用時」を参照して対処してください。

EE

- マルチ言語サービス

EE SJE

- J2EE互換

### 5.2.1 Java EE機能/ Webサーバの使用時

#### Interstage起動時の異常

isstartコマンドによるInterstageの起動処理中、またはマシンプート時のInterstageの自動起動処理中に異常が発生した場合、起動が完了しているサービスをすべて停止した後、起動処理を中止します。

ただし、停止処理中に異常が発生した場合、処理は中断されます。

以下のいずれかに出力されたInterstage起動に失敗した原因を示すメッセージを確認し、そのメッセージに対する対処を行ってください。

- isstartコマンドの投入画面
- **Windows32/64**  
イベントビューア (アプリケーションのログ)
- **Solaris32/64**  
/var/adm/messages
- **Linux32/64**  
/var/log/messages

### 5.2.2 マルチ言語サービス/ J2EE互換機能の使用時

#### 注意

**Solaris32/64 Linux32/64**

Interstage起動時に異常が発生した場合は、まずオペレーティングシステムのチューニングが正しく行われているかを確認してください。詳細については、“チューニングガイド”を参照してください。

#### Interstage起動時の異常

isstartコマンドによるInterstageの起動処理中、またはマシンプート時のInterstageの自動起動処理中に異常が発生した場合、起動の完了しているサービスをすべて停止した後、起動処理を中止します。

ただし、停止処理中に異常が発生した場合、処理は中断されます。

以下のいずれかに出力されたInterstage起動に失敗した原因を示すメッセージを確認し、そのメッセージに対する対処を行ってください。

- isstartコマンドの投入画面
- **Windows32/64**  
イベントビューア (アプリケーションのログおよびシステムのログ)

- **Solaris32/64**  
/var/adm/messages
- **Linux32/64**  
/var/log/messages

#### **Windows32/64**

イベントビューア(システムのログ)に以下のメッセージが出力されている場合、ネットワーク環境が正常に設定されていない可能性があります。ネットワークが使用可能であるか、ホスト名/IPアドレスが正しく設定されているかを確認してください。

```
OD_start サービスは予期せぬ原因により終了しました
```

#### **Solaris32/64 Linux32/64**

インタフェースリポジトリへ登録したインタフェース情報が多い場合、またはCORBAサービスの動作環境ファイル(config)に「logging = yes」を定義した場合、インタフェースリポジトリ/ネーミングサービスの起動に時間がかかります。インタフェースリポジトリやEJB用インタフェースリポジトリおよびネーミングサービスの起動に1分以上かかった場合、Interstageの起動に失敗するため、注意してください。

本条件にあてはまる場合は、環境変数IS\_ISV\_SVCCTRL\_TIMERを設定してからisstartコマンドを実行してください。環境変数IS\_ISV\_SVCCTRL\_TIMERには、エラーメッセージis20102の出力から情報メッセージod20001/エラーメッセージod30102が出力されるまでの秒数+90の値を設定してください。

例えば、エラーメッセージis20102が、10時00分00秒に出力され、情報メッセージod20001/エラーメッセージod30102が、10時00分10秒に出力された場合、環境変数IS\_ISV\_SVCCTRL\_TIMERには、100を設定します。

```
IS_ISV_SVCCTRL_TIMER=100
export IS_ISV_SVCCTRL_TIMER
```

なお、本現象は、エラーメッセージis20102の出力内容により特定できます。エラーメッセージis20102が出力されている場合は、「メッセージ集」を参照し、原因を特定して、その対処を行ってください。

### Interstage起動時にis30141のメッセージが出力される異常 **Solaris32/64**

isstartコマンドの実行画面に、以下のメッセージが出力されている場合は、再度isstartコマンドを実行してください。

```
Daemon start fails
```

### Interstage起動時にis30151のメッセージが出力される異常 **Solaris32**

マルチシステム機能を使用している場合、本コマンドと同時にisdeletesysコマンド/iscreatesysコマンドを実行した可能性があります。この場合、isdeletesysコマンドによりシステムを削除してください。

### Interstageの起動時に時間がかかる

Interstageの起動時に時間がかかる場合、以下の原因が考えられます。

- Interstage シングル・サインオンにおいてSSOリポジトリにロール定義およびサイト定義を多量に登録してリポジトリサーバを運用し、かつInterstage HTTP Serverと連動設定を行っている場合、Interstageの起動に時間がかかることがあります。本現象の詳細および回避方法については、「[16.2.5 リポジトリサーバに関するトラブル](#)」を参照してください。
- **Windows32/64**  
ネーミングサービスに対してネーミングコンテキストの登録・削除を繰り返し行ったあとにInterstageの起動を行った場合、Interstageの起動に時間がかかることがあります。本現象の原因は、ネーミングサービスの起動処理に時間がかかるためであり、特に問題はありません。

## Interstage停止時の異常

isstopコマンドで停止中に異常が発生した場合、異常の発生したサービス名を含むエラーメッセージを表示して終了します。停止時の異常の対処方法については、“メッセージ集”を参照してください。対処後、再度isstopコマンドを実行してください。

### Solaris32/64 Linux32/64

動作中のCORBAのアプリケーションが多い場合など、CORBAサービスの停止に時間がかかることがあります。CORBAサービスの停止に1分以上かかった場合は、エラーメッセージis20110が出力されて、Interstageの停止に失敗する可能性があります。注意してください(CORBAサービスの停止処理は、継続されます)。

エラーメッセージis20110が理由コード「15」で出力され、次に情報メッセージod10005のメッセージが出力された場合は、環境変数IS\_ISV\_SVCCTRL\_TIMERを設定してからisstopコマンドを実行してください。環境変数IS\_ISV\_SVCCTRL\_TIMERには、情報メッセージod10003の出力から情報メッセージod10005が出力されるまでの秒数+60の値を設定してください。

例えば、情報メッセージod10003が10時00分00秒に出力され、情報メッセージod10005が10時02分00秒に出力された場合、環境変数IS\_ISV\_SVCCTRL\_TIMERには、180を設定します。

```
IS_ISV_SVCCTRL_TIMER=180
export IS_ISV_SVCCTRL_TIMER
```

## Interstage停止処理が無応答

ワークユニットを停止しないでアプリケーションを入れ替えた場合、isstopコマンドでInterstage停止時に、isstopコマンドがハングアップすることがあります。アプリケーションを入れ替える場合は、必ずワークユニットを停止してから入れ替えてください。

## Interstage停止時にis30168のメッセージが出力される異常 Solaris32

マルチシステム機能を使用している場合、本コマンドと同時にisdeletesysコマンド/iscreatesysコマンドを実行した可能性があります。この場合、isdeletesysコマンドでシステムを削除してください。

## Interstage起動/停止時にアプリケーションポップアップが発生する場合 Windows32/64

isstartコマンド/isstopコマンドを実行時に、以下のアプリケーションポップアップが発生した場合、デスクトップヒープが使い果たされている可能性があります。本現象が発生した場合は、デスクトップヒープを拡張する必要があります。デスクトップヒープを拡張する手順については、「5.5 デスクトップヒープが枯渇した場合」を参照してください。

```
Apache.exe - アプリケーション エラー :
アプリケーションを正しく初期化できませんでした (0xc0000142)。
[OK] をクリックしてアプリケーションを終了してください。
```

## RCプロシジャからInterstageおよびワークユニットを起動するとコンソールログインができなくなった場合

### Solaris32/64

RCプロシジャにInterstageを起動するスクリプトを登録して、サーバ・ブート時に起動スクリプトからInterstageを起動すると、コンソール端末からのログイン時に以下のメッセージが出力され、ログインできなくなることがあります。

```
console login: root
Warning -- ttymon cannot allocate controlling tty on "/dev/console",
there may be another session active on this port.
```

本現象が発生した場合、Interstage起動スクリプトに記述したInterstageの起動処理の前に、コンソールメッセージを抑制するための環境変数を設定して回避してください。

```
IS_ISV_CONSOLE_OUTPUT=OFF
```

## 注意

すべてのコンソールメッセージが抑止されるわけではありません。

## 例

```
#!/sbin/sh
# INTERSTAGE start procedure
...

# Set Environment Variable
IS_ISV_CONSOLE_OUTPUT=OFF
export IS_ISV_CONSOLE_OUTPUT
...

# Execute Start Command
isstart
...
```

上記の対処が有効であるかを確認するには、情報メッセージtd11028がシステムログ(/var/adm/messages)に出力され、コンソールに出力されていない場合に有効と判断できます。

## 5.2.3 Windows(R)マシンの再起動/シャットダウン時

### Windows(R)マシンの再起動・シャットダウン時の異常 Windows32/64

イベントビューア(システムのログ)に以下のメッセージが出力されている場合、Interstageを停止せずにシャットダウンした可能性があります。

```
INTERSTAGE サービスは、サービス固有エラー 指定されたリソース名がイメージ ファイルに見つかりません。 で終了しました。
```

または

```
INTERSTAGEは次のサービス固有のエラーで終了しました。 1814 (0x716)
```

本現象が発生した場合、シャットダウン前にInterstageを停止しているかを確認してください。Interstageを停止していない場合は、Interstageを停止してからシャットダウンしてください。

## 5.3 業務システム運用時の異常

業務システムの運用時に、異常が発生した場合の対処方法について、以下の内容で説明します。

- Interstageシステムの異常
- エラーメッセージが通知された場合

### Interstageシステムの異常

業務システム運用中に、Interstageが異常終了した場合は、出力されたエラーメッセージに対する対処を行ってください。対処完了後は、isstatコマンドを実行し、動作中のサービスが存在しないかを確認してください。動作中のサービスが存在する場合は、以下のコマンドを実行してInterstageを停止してください。

```
isstop -f
```

ただし、動作中のサービスを使用している他製品が動作している場合は、その製品の運用にも影響するため、注意してください。

## エラーメッセージが通知された場合

業務システム運用中に、エラーメッセージが通知された場合は、“メッセージ集”を参照して対処方法を確認し、対処してください。ただし、通知されたメッセージの内容がシステムエラーの場合は、以下の手順で対処してください。

### Windows32/64

1. イベントビューアを起動して、以下のログをファイルに退避してください。
  - アプリケーション ログ
  - セキュリティ ログ
  - システム ログ
2. 技術員に連絡してください。

### Solaris32/64 Linux32/64

1. syslog.confファイル内で設定したログファイルを、別のファイルに退避します。
2. 技術員に連絡してください。

## マシンシャットダウン時にtd11045やEXTP4002のメッセージが出力される異常 Solaris32

拡張システムのInterstage動作中にマシンのシャットダウンが行われた場合、メッセージtd11045/EXTP4002、およびcoreファイルが出力される可能性があります。

マシンのシャットダウンを行う場合は、シャットダウン前に、動作中の拡張システムのInterstageを停止する必要があります。RCプロシジャに、拡張システムのInterstageを停止するスクリプトを登録するなど、マシンのシャットダウン前にInterstageを停止するようにしてください。

## OD\_startプロセスの異常終了後、Interstageの操作で無応答となった場合

CORBAワークユニットの運用中にOD\_startプロセスが異常終了した場合は、その後のInterstageの操作で無応答となることがあります。

OD\_startプロセスが異常終了した場合は、オペレーティングシステムを再起動してください。

## 5.4 Visual Studioで作成したアプリケーションが動作しない場合

Visual Studioがインストールされていない環境でVisual Studioで作成したアプリケーションを動作させる場合は、再頒布用ランタイムがシステムにインストールされている必要があります。再頒布用ランタイムの詳細については、Visual Studioのマニュアルを参照してください。

例えば、Visual Studio 2005 Service Pack 1で作成したアプリケーションを動作させる場合は、Visual Studio 2005 Service Pack 1のインストールフォルダ配下に格納されている“Microsoft Visual C++ 2005 再頒布可能パッケージ”をアプリケーション運用先のシステムに適用してください。

Interstageをインストールすると、再頒布可能パッケージがインストールされる場合がありますが、この場合もVisual Studio 2005 Service Pack 1内の再頒布可能パッケージをインストールしてください。

なお、再頒布可能パッケージがインストールされていない環境でアプリケーションを実行した場合、以下のようなメッセージが出力されることがあります。

SideBySide “C:¥ap.exe” のアクティブ化コンテキストの生成に失敗しました。  
従属アセンブリ Microsoft.VC80.CRT.processorArchitecture=“x86”, publicKeyToken=“xxxxxxxxxxxxxxxx”,  
type=“win32”, version=“8.0.50727.762” が見つかりませんでした。  
詳細な診断を行うには sxstrace.exe を実行してください。

## 5.5 デスクトップヒープが枯渇した場合

Interstage運用時にデスクトップヒープが枯渇した場合、イベントログに以下のメッセージが出力されて、アプリケーションの起動に失敗することがあります。

```
アプリケーションを正しく初期化できませんでした (0xc0000142)。
```

本現象が発生した場合、デスクトップヒープを拡張する必要があります。

Windows Server(R) 2003でのデスクトップヒープの拡張手順を以下に示します。

### Windows Server(R) 2003の場合

1. レジストリエディタを起動します。  
[スタート]―[ファイル名を指定して実行]で、「regedit」と入力して[OK]ボタンを押下します。
2. SubSystemsキーに移動します。  
HKEY\_LOCAL\_MACHINEサブツリーから次のキーに移動します。

```
¥System¥CurrentControlSet¥Control¥Session Manager¥SubSystems
```

3. [Windows]の値を選択します。
4. [編集]メニューで、[文字列]を選択します。
5. SharedSectionパラメタの値を変更して、デスクトップヒープを増加します。  
3番目の値「zzzz」を増加(256KB/512KBずつ)してください。

```
SharedSection=xxxx, yyyy, zzzz
```

**注1)** 1番目の値「xxxx」および2番目の値「yyyy」を変更する必要はありません。

**注2)** 3番目の値「zzzz」が省略されている場合、省略値は2番目の値「yyyy」と同じ値となります。3番目の値「zzzz」には、2番目の値「yyyy」よりも大きな値を設定してください(単位:KB)。



例

変更前

```
SharedSection=1024, 3072, 512
```

変更後

```
SharedSection=1024, 3072, 1024
```

6. システムを再起動します。

その他のオペレーティングシステムでの拡張方法については、マイクロソフトサポート技術情報を参照してください。デスクトップヒープについては、「マイクロソフト サポート技術情報 - 184802」を参照してください。



注意

レジストリエディタまたは別の方法を使用してレジストリを誤って変更すると、深刻な問題が発生することがあります。最悪の場合、オペレーティングシステムの再インストールが必要となることがあります。

システムのバックアップをするなど、十分に注意して変更してください。

なお、デスクトップヒープの適正値を見積もる方法はありません。値を変更する場合は、徐々に拡張してください。

## 第6章 マルチサーバ管理の異常 Windows32/64 Solaris32 Linux32/64

マルチサーバ管理でトラブルが発生した場合の対処方法について説明します。

### 6.1 トラブル解決のための調査方法 Windows32/64 Solaris32 Linux32/64

マルチサーバ管理のトラブル調査は、以下の手順で行ってください。

1. どの管理対象サーバで発生したトラブルかを判別します。  
管理サーバに出力されているエラーログを参照することによって、どこの管理対象サーバでエラーが発生したかを知ることができます。
2. エラーが発生した管理対象サーバで、トラブルを解決します。  
エラーが発生した管理対象サーバに出力されているエラーログを参照することにより、実際に発生したエラー内容が判別し、問題を解決する糸口となります。

### 6.2 メッセージの見方 Windows32/64 Solaris32 Linux32/64

マルチサーバ管理機能が出力するメッセージの見方について説明します。

#### 通常メッセージ形式

ラベル： エラー種別： メッセージ番号： メッセージ内容

#### マルチサーバ管理機能が出力するメッセージ形式

管理サーバのInterstage管理コンソールにログインし、Interstageの操作を行った場合、Interstage管理コンソールには、以下のいずれかの形式のメッセージが出力されます。

##### 形式

[発生日時 発生管理対象サーバ名] ラベル： エラー種別： メッセージ番号： メッセージ内容

##### 意味

事象が発生した日時、および発生した管理対象サーバの名前が付加されて出力されます。それ以外は、通常のメッセージと同じです。

##### 形式

[発生日時 発生サーバグループ名] ラベル： エラー種別： メッセージ番号： メッセージ内容

##### 意味

事象が発生した日時と、発生したサーバグループの名前が付加されて出力されます。それ以外は、通常のメッセージと同じです。

##### 形式

[発生日時] ラベル： エラー種別： メッセージ番号： メッセージ内容

##### 意味

事象が発生した日時が付加されて出力されます。本メッセージは、管理サーバ上で発生した事象に対するメッセージです。それ以外は、通常のメッセージと同じです。

管理サーバのInterstage管理コンソール上に出力される上記メッセージを参照することにより、異常が発生したサーバグループや管理対象サーバを容易に特定することができます。

## 例

サーバグループsvrgrp002に配置されたIJSERVER WU001の起動時に、サーバグループsvrgrp002に所属する管理対象サーバserver01上でのアプリケーションの起動に失敗した場合、Interstage管理コンソール上に以下のようにメッセージが出力されます。

(3)のメッセージより、IJSERVERの起動操作において、一部の管理対象サーバ上でアプリケーションの起動に失敗したことがわかります。

(2)のメッセージより、起動に失敗したサーバが、サーバグループsvrgrp002に所属している管理対象サーバであることがわかります。

(1)のメッセージより、起動に失敗した管理対象サーバが、サーバserver01であることが分かります。異常が発生した管理対象サーバの出力メッセージを参照し、異常原因を特定します。

```
(1) [2005/01/05 21:43:29 server01] IS: エラー: is20450:ワークユニット (WU001) の処理に失敗しました 理由コード (518)
(2) [2005/01/05 21:43:29 svrgrp002] IS: エラー: is21117:IJSERVER起動処理の一部に失敗しました
IJSERVER=WU001 (svrgrp002)
(3) [2005/01/05 21:43:29] IS: エラー: is21117:IJSERVER起動処理の一部に失敗しました IJSERVER=WU001 (svrgrp002)
```

## 6.3 環境構築編 Windows32/64 Solaris32 Linux32/64

マルチサーバ管理の環境構築時に発生するトラブルへの対処方法について説明します。

### ■ 管理対象サーバをサーバグループに追加した場合にInterstageが停止する

他管理対象サーバのネーミングサービスやインタフェースリポジトリをライン構成で参照している管理対象サーバを追加する場合、Interstageが停止することがあります。

本現象が発生した場合、Interstage管理コンソールの[一括操作] > [Interstage管理コンソール] > [Interstage Application Server] > [システム] > [状態]タブから対象のサーバグループを選択し、[起動]ボタンよりInterstageを再起動してください。

### ■ 管理対象サーバを追加できない

マルチサーバ管理機能において、サポート対象外のバージョンのInterstageを運用しているサーバをサイトに追加した可能性があります。

サポート対象外のバージョンのInterstageを運用しているサーバは、サポート対象のバージョンのInterstageに移行してから、サイトにサーバを追加してください。サポート対象のバージョンについては、「マルチサーバ運用ガイド」の「マルチサーバ管理機能」を参照してください。

なお、サイトにサポート対象外のバージョンのInterstageを運用しているサーバを追加すると、エラーメッセージis20727が出力されて、サイトへの追加に失敗する場合があります。

### ■ シングル・サインオンが正しく動作しない

管理対象サーバのInterstage管理コンソールを使用して、シングル・サインオンのリポジトリサーバ、認証サーバ、またはWebサーバ(Interstage HTTP Server)上で動作する業務サーバの環境構築を行った後に、管理対象サーバ上のInterstage HTTP Serverの環境定義の内容を管理サーバ上の業務構成管理のリポジトリに反映していない可能性があります。

以下の操作を行ってください。

1. 管理サーバのInterstage管理コンソールにログインします。
2. Interstage管理コンソールの  
[一括操作] > [Interstage管理コンソール] > [Interstage Application Server] > [サービス] > [Webサーバ] > [サーバグループ名またはサーバ名] > [定義の取得]タブで、

リポジトリサーバ、認証サーバ、またはInterstage HTTP Server上で動作する業務サーバの環境構築を行った管理対象サーバを選択して[選択]ボタンをクリックし、[適用]ボタンをクリックします。

### 3. Interstage管理コンソールの

[一括操作] > [Interstage管理コンソール] > [Interstage Application Server] > [サービス] > [Webサーバ]で、操作対象のサーバグループまたは管理対象サーバを選択し、[起動]をクリックします。

または、

[一括操作] > [Interstage管理コンソール] > [Interstage Application Server] > [サービス] > [Webサーバ] > [サーバグループ名またはサーバ名]で、[起動]をクリックします。

管理対象サーバがサーバグループに含まれる場合は、サーバグループに含まれるすべての管理対象サーバに対して環境構築を行ったあと、上記の手順を行ってください。

## ■ Interstage管理コンソールでシステムの状態が「不明」と表示される Windows32/64

Interstage管理コンソールの[一括操作] > [Interstage管理コンソール] > [Interstage Application Server] > [システム]においてシステムの状態が“不明”と表示され、かつ、Interstage管理コンソールの操作時にエラーメッセージis20726が出力される場合、isjmx.xmlにローカルホストが設定されていない可能性があります。

“マルチサーバ運用ガイド”の“サーバグループ環境の初期構築”を参照し、isjmx.xmlにローカルホストを設定してください。

## ■ サーバグループ内の管理対象サーバにインストールされている機能について管理サーバから使用できない

サーバグループに最初に追加された管理対象サーバにおいてインストールされていなかった機能は、最初に追加されたサーバがサーバグループから削除された後であっても、そのサーバグループでは使用できなくなります。

以下の操作を行ってください。

1. サーバグループに所属しているサーバをすべて削除します。  
手順の詳細については、“マルチサーバ運用ガイド”の“サーバグループからのサーバ削除”を参照してください。
2. 削除したサーバを再度サーバグループへ追加します。  
手順の詳細については、“マルチサーバ運用ガイド”の“サーバグループへのサーバ追加”を参照してください。

## 6.4 運用編 Windows32/64 Solaris32 Linux32/64

マルチサーバ管理の運用時に発生するトラブルへの対処方法(マルチサーバ特有の事象)について説明します。各機能のトラブル詳細については、各機能のトラブルシューティングも参照してください。

### ■ CORBAの通信エラーが返ってくる

本現象が発生した場合は、以下の原因が考えられます。

#### 単体運用の管理対象サーバの場合

- ・「参照先ホスト」がサイトから削除されています。
- ・ Interstageが運用されていません。

#### サーバグループに所属している管理対象サーバの場合

- ・「参照先ホスト」がサーバグループから削除されています。
- ・ Interstageが運用されていません。

以下の対処を行ってください。

参照先サーバのInterstage状態を確認してください。参照先サーバが削除されている場合は、サイトの構成を確認してください。問題がない場合は、他の問題の可能性もあります。“11.1 CORBAサービス運用中のシステム例外発生”を参照して対処してください。

## ■ サーバグループの一部分のサービスやワークユニットが運用操作をしていないのに起動・停止する

本現象が発生した場合、以下の原因が考えられます。それぞれの対処を行ってください。

- ・ そのサーバに固有の異常が発生している(停止している)場合は、メッセージを参照して対処してください。
- ・ サーバグループの一部サーバがクラスタシステムで運用されている場合は、サーバグループに所属している管理対象サーバをクラスタシステムで運用しないでください。サーバグループに所属している管理対象サーバをクラスタシステムで運用すると、Interstage管理コンソールからの一括操作とクラスタシステムの運用操作のバッティングにより、Interstageのサービスやワークユニットなどがユーザの意図しないタイミングで起動/停止される可能性があります。

## ■ 管理対象サーバから応答が返ってこない

管理対象サーバで、処理が完了しているのに管理対象サーバから応答が返らない場合は、以下の原因が考えられます。

- ・ Interstage JMXサービスでは、管理対象サーバの処理中は処理結果を受信するまで待機しますが、管理対象サーバの操作中にLANが切断されて管理対象サーバからの処理結果を受信できない状態となった可能性があります。

LANを接続した後に、再度Interstage管理コンソールにログインして、処理結果を確認してください。また、処理結果の受信時間は時間監視しているため、タイムアウト時間を超過すると通信エラーが発生します。

なお、時間監視時間は、カスタマイズできます。カスタマイズ方法については、“運用ガイド(基本編)”の“Interstage管理コンソール環境のカスタマイズ”を参照してください。

## ■ Interstage管理コンソールでシステムの状態が“不明”と表示される

Interstage管理コンソールの[一括操作] > [Interstage管理コンソール] > [Interstage Application Server] > [システム]においてシステムの状態が“不明”と表示され、かつ、Interstage管理コンソールの操作時にエラーメッセージis20726が出力される場合、isjmx.xmlにローカルホストが設定されていない可能性があります。

“マルチサーバ運用ガイド”の“サーバグループ環境の初期構築”を参照し、isjmx.xmlにローカルホストを設定してください。

## ■ サイト移行後の環境においてアプリケーションがエラーとなる

サイト移行後の環境においてアプリケーションがエラーとなる場合、以下の原因が考えられます。

- ・ サイト情報の変更失敗しています。
- ・ 管理対象サーバの移入に失敗しています。

サイト情報定義ファイルの定義内容に誤りがないかを確認してください。また、移入する一括実行バッチファイル(シェルスクリプト)のカスタマイズに誤りがないかを確認してください。

バックアップ資源の移入先サーバとサイト情報定義ファイルが一致しているかを確認してください。

確認後、サイトの移行を最初から行ってください。サイトの移行手順については、“マルチサーバ運用ガイド”の“開発環境から運用環境へのサイト移行”を参照してください。

## 6.5 管理サーバに異常が発生した場合

管理サーバに異常が発生した場合の対処方法について説明します。

### ■ 管理サーバダウン時の管理対象サーバの運用

サーバ故障などの理由により管理サーバが使用できない場合、以下の対処を行ってください。

- ・ Interstage管理コンソールによる管理対象サーバの直接操作  
Interstage管理コンソールで管理対象サーバに直接ログインし、個々のサーバを個別に操作します。この場合、

Interstage／アプリケーションの起動などの運用操作だけが可能となります。アプリケーションの定義変更／配備操作はできません。アプリケーションの定義変更／配備操作は、管理サーバを復旧させた後に行ってください。

- 管理サーバの他サーバへの切り替え  
問題が発生した管理サーバを復旧させることができない場合は、管理サーバ環境を他サーバへ移行することで管理サーバ機能を復旧します。詳細については、“マルチサーバ運用ガイド”の“管理サーバのバックアップ／リストア”を参照してください。
- 管理対象サーバをサイトから削除してスタンドアロンサーバとして運用する  
スタンドアロンサーバとして運用する管理対象サーバにおいて、isleaveiteコマンドでスタンドアロンサーバとします。

#### ■ 管理サーバのInterstage ディレクトリサービスが故障した場合

管理サーバのユーザリポジトリとして使用しているInterstage ディレクトリサービスが故障した場合、管理サーバ上でisresetuserrepを実行して、ログイン認証方法を一時的にOS認証に切り換えて運用してください。

Interstage ディレクトリサービスを復旧させた後、ログイン認証方法をInterstage ディレクトリサービスによる認証に戻すには、Interstage管理コンソールの

[Interstage管理コンソール] > [統合管理] > [Interstage Application Server] から[セキュリティ] > [運用セキュリティ]で行ってください。

## 第7章 Interstageリストア／移入時の異常

Interstageのリストア／移入時に発生する異常について、以下の内容で説明します。

- CORBAサービスのリストア／移入
- コンポーネントトランザクションサービスのリストア／移入
- Interstage シングル・サインオンのリストア／移入
- IIServerクラスタ資源のリストア／移入

### 7.1 CORBAサービスのリストア／移入

ここでは、CORBAサービスのリストア／移入時に発生したトラブルの事例およびその対処について説明します。



CORBAサービスのリストア／移入方法については、「運用ガイド(基本編)」の「メンテナンス(資源のバックアップ/他サーバへの資源移行/ホスト情報の変更)」を参照してください。

#### リストア／移入時にエラーメッセージod16212を出力して失敗する Windows32/64

odrestoresysコマンドを実行してCORBAサービスのリストア／移入を行う際に、エラーメッセージod16212が出力されて異常終了する場合は、リストア／移入環境の8.3形式(ショートネーム)のインストールディレクトリが、バックアップ／移出資源の8.3形式(ショートネーム)のインストールディレクトリと一致していない可能性があります。

以下の手順で、8.3形式(ショートネーム)のインストールディレクトリを一致させてください。

1. エラーメッセージod16212に含まれるバックアップ／移出資源の8.3形式(ショートネーム)のインストールディレクトリを確認します。
2. リストア／移入環境の8.3形式(ショートネーム)のインストールディレクトリを確認します。



本製品のインストールディレクトリが「C:¥Interstage」の場合

1. Interstageのインストールディレクトリに移動します。

```
cd C:¥Interstage
cd ..
```

2. dirコマンドを使用して、8.3形式(ショートネーム)のディレクトリ名を確認します。

```
dir /x
XXXX/XX/XX XX:XX <DIR> INTERS~1 Interstage
```

上記の例の場合、Interstageのインストールディレクトリに設定されている8.3形式(ショートネーム)のディレクトリ名は、dirコマンドの実行結果から「C:¥INTER~1」となります。

3. バックアップ／移出資源の8.3形式(ショートネーム)のインストールディレクトリとリストア／移入環境の8.3形式(ショートネーム)のインストールディレクトリが一致していない場合は、8.3形式(ショートネーム)のインストールディレクトリが一致するようにInterstageを再インストールします。



エラーメッセージod16212に含まれていたバックアップ／移出資源の8.3形式(ショートネーム)のインストールディレクトリが「C:¥INTER~2」であった場合

1. Interstageをインストールするトップディレクトリに移動します。

```
cd C:¥
```

2. 8.3形式(ショートネーム)が「INTER~1」となるようにディレクトリを作成します。

```
mkdir Interstage1
```

3. dirコマンドを使用して、8.3形式(ショートネーム)のディレクトリ名を確認します。

```
dir /x
XXX/XX/XX XX:XX <DIR> INTER~1 Interstage1
```

4. Interstageをインストールします。

5. dirコマンドを使用して、「C:¥Interstage」の8.3形式(ショートネーム)のディレクトリ名を確認します。

```
dir /x
XXX/XX/XX XX:XX <DIR> INTER~2 Interstage
XXX/XX/XX XX:XX <DIR> INTER~1 Interstage1
```

.....

上記の例の場合、先に「Interstage1」というダミーディレクトリを作成することにより、「Interstage」の8.3形式(ショートネーム)のディレクトリ名を「INTER~2」にしています。

## 7.2 コンポーネントランザクションサービスのリストア/移入

ここでは、コンポーネントランザクションサービスのリストア/移入時に発生したトラブルの事例およびその対処について説明します。



.....

コンポーネントランザクションサービスのリストア/移入方法については、「運用ガイド(基本編)」の「メンテナンス(資源のバックアップ/他サーバへの資源移行/ホスト情報の変更)」を参照してください。

.....

### リストア/移入時にエラーメッセージEXTP0910を出力して失敗する Windows32/64

tdrestoresysコマンドを実行してコンポーネントランザクションサービスのリストア/移入を行う際に、エラーメッセージEXTP0910が出力されて異常終了する場合は、リストア/移入環境の8.3形式(ショートネーム)のインストールディレクトリが、バックアップ/移出資源の8.3形式(ショートネーム)のインストールディレクトリと一致していない可能性があります。以下の手順で、8.3形式(ショートネーム)のインストールディレクトリを一致させてください。

1. エラーメッセージEXTP0910に含まれるバックアップ/移出資源の8.3形式(ショートネーム)のインストールディレクトリを確認します。
2. リストア/移入環境の8.3形式(ショートネーム)のインストールディレクトリを確認します。



.....

本製品のインストールディレクトリが「C:¥Interstage」の場合

1. Interstageのインストールディレクトリに移動します。

```
cd C:¥Interstage
cd ..
```

2. dirコマンドを使用して、8.3形式(ショートネーム)のディレクトリ名を確認します。

```
dir /x
XXX/XX/XX XX:XX <DIR> INTER~1 Interstage
```

.....

上記の例の場合、Interstageのインストールディレクトリに設定されている8.3形式(ショートネーム)のディレクトリ名は、dirコマンドの実行結果から「C:¥INTER~1」となります。

3. バックアップ/移出資源の8.3形式(ショートネーム)のインストールディレクトリとリストア/移入環境の8.3形式(ショートネーム)のインストールディレクトリが一致していない場合は、8.3形式(ショートネーム)のインストールディレクトリが一致するようにInterstageを再インストールします。



#### 例

エラーメッセージEXTP0910に含まれていたバックアップ/移出資源の8.3形式(ショートネーム)のインストールディレクトリが「C:¥INTER~2」であった場合

1. Interstageをインストールするトップディレクトリに移動します。

```
cd C:¥
```

2. 8.3形式(ショートネーム)が「INTER~1」となるようにディレクトリを作成します。

```
mkdir Interstage1
```

3. dirコマンドを使用して、8.3形式(ショートネーム)のディレクトリ名を確認します。

```
dir /x
XXXX/XX/XX XX:XX <DIR> INTER~1 Interstage1
```

4. Interstageをインストールします。

5. dirコマンドを使用して、「C:¥Interstage」の8.3形式(ショートネーム)のディレクトリ名を確認します。

```
dir /x
XXXX/XX/XX XX:XX <DIR> INTER~2 Interstage
XXXX/XX/XX XX:XX <DIR> INTER~1 Interstage1
```

上記の例の場合、先に「Interstage1」というダミーディレクトリを作成することにより、「Interstage」の8.3形式(ショートネーム)のディレクトリ名を「INTER~2」にしています。

## 7.3 Interstage シングル・サインオンのリストア/移入

ここでは、Interstageシングル・サインオンのリストア/移入時に発生するトラブルの事例およびその対処について説明します。



#### 参照

Interstageシングル・サインオンのリストア/移入方法については、「運用ガイド(基本編)」の「メンテナンス(資源のバックアップ/他サーバへの資源移行/ホスト情報の変更)」を参照してください。

### リストア/移入時にエラーメッセージ[0134]を出力して失敗する Windows32/64

ssorestoreコマンドを実行してInterstageシングル・サインオンのリストア/移入を行う際に、エラーメッセージ[0134]が出力されて異常終了する場合は、リストア/移入環境の8.3形式(ショートネーム)のインストールディレクトリがバックアップ/移出資源の8.3形式(ショートネーム)のインストールディレクトリと一致していない可能性があります。

以下の手順で、8.3形式(ショートネーム)のインストールディレクトリを一致させてください。

1. エラーメッセージ[0134]に含まれる8.3形式(ショートネーム)のインストールディレクトリを確認します。
2. リストア/移入環境の8.3形式(ショートネーム)のインストールディレクトリを確認します。



例

本製品のインストールディレクトリが「C:¥Interstage」の場合

1. Interstageのインストールディレクトリに移動します。

```
cd C:¥Interstage
cd ..
```

2. dirコマンドを使用して、8.3形式(ショートネーム)のディレクトリ名を確認します。

```
dir /x
XXXX/XX/XX XX:XX <DIR>     INTERS~1     Interstage
```

上記の例の場合、Interstageのインストールディレクトリに設定されている8.3形式(ショートネーム)のパスは、dirコマンドの実行結果から「C:¥INTERS~1」となります。

3. バックアップ/移出資源の8.3形式(ショートネーム)のインストールディレクトリとリストア/移入環境の8.3形式(ショートネーム)のインストールディレクトリが一致していない場合は、8.3形式(ショートネーム)のインストールディレクトリが一致するようにInterstageを再インストールします。



例

エラーメッセージ[0134]に含まれていたバックアップ/移出資源の8.3形式(ショートネーム)のインストールディレクトリが「C:¥INTERS~2」であった場合

1. Interstageをインストールするトップディレクトリに移動します。

```
cd C:¥
```

2. 8.3形式(ショートネーム)が「INTERS~1」となるようにディレクトリを作成します。

```
mkdir Interstage1
```

3. dirコマンドを使用して、8.3形式(ショートネーム)のディレクトリ名を確認します。

```
dir /x
XXXX/XX/XX XX:XX <DIR>     INTERS~1     Interstage1
```

4. Interstageをインストールします。

5. dirコマンドを使用して、「C:¥Interstage」の8.3形式(ショートネーム)のディレクトリ名を確認します。

```
dir /x
XXXX/XX/XX XX:XX <DIR>     INTERS~2     Interstage
XXXX/XX/XX XX:XX <DIR>     INTERS~1     Interstage1
```

上記の例の場合、先に「Interstage1」というダミーディレクトリを作成することにより、「Interstage」の8.3形式(ショートネーム)のディレクトリ名を「INTERS~2」にしています。

## 7.4 IJServerクラスタ資源のリストア/移入

ここでは、IJServerクラスタ資源のリストア/移入時に発生するトラブルの事例およびその対処について説明します。



参照

IJServerクラスタ資源のリストア/移入方法については、「運用ガイド(基本編)」の「メンテナンス(資源のバックアップ/他サーバへの資源移行/ホスト情報の変更)」を参照してください。

## リストア/移入時にエラーメッセージISJEE\_OM4028を出力して失敗する

ijrestoreコマンドを実行してIJServerクラスタ資源のリストア/移入を行う際に、エラーメッセージISJEE\_OM4028が出力されて異常終了する場合は、リストア/移入環境の8.3形式(ショートネーム)のインストールディレクトリがバックアップ/移出資源の8.3形式(ショートネーム)のインストールディレクトリと一致していない可能性があります。

以下の手順で、8.3形式(ショートネーム)のインストールディレクトリを一致させてください。

1. [IJServerクラスタ資源をバックアップしたディレクトリ]¥JavaEE¥etc¥interstage-config.propertiesファイルのis.homeの値に含まれる8.3形式(ショートネーム)のインストールディレクトリを確認します。



例

### ijrestoreコマンドの-dオプションで指定するディレクトリが「X:¥Backup」の場合

1. X:¥Backup¥JavaEE¥etc¥interstage-config.propertiesをテキストエディタで開きます。
2. is.homeの値を確認します。

```
#Fri Nov 18 14:21:21 JST 2011
~省略~
is.home=C¥:¥¥INTERS~2
```

上記の例の場合、バックアップ/移出時の環境でのInterstageのインストールディレクトリに設定されている8.3形式(ショートネーム)のディレクトリ名は「C:¥INTERS~2」となります。

2. リストア/移入環境の8.3形式(ショートネーム)のインストールディレクトリを確認します。



例

### 本製品のインストールディレクトリが「C:¥Interstage」の場合

1. Interstageのインストールディレクトリに移動します。

```
cd C:¥Interstage
cd ..
```

2. dirコマンドを使用して、8.3形式(ショートネーム)のディレクトリ名を確認します。

```
dir /x
XXXX/XX/XX XX:XX <DIR> INTERS~1 Interstage
```

上記の例の場合、Interstageのインストールディレクトリに設定されている8.3形式(ショートネーム)のディレクトリ名は、dirコマンドの実行結果から「C:¥INTERS~1」となります。

3. バックアップ/移出資源の8.3形式(ショートネーム)のインストールディレクトリとリストア/移入環境の8.3形式(ショートネーム)のインストールディレクトリが一致していない場合は、8.3形式(ショートネーム)のインストールディレクトリが一致するようにInterstageを再インストールします。



例

### [IJServerクラスタ資源をバックアップしたディレクトリ]¥JavaEE¥etc¥interstage-config.propertiesファイルのis.homeの値に含まれている8.3形式(ショートネーム)のインストールディレクトリが「C:¥INTERS~2」であった場合

1. Interstageをインストールするトップディレクトリに移動します。

```
cd C:¥
```

2. 8.3形式(ショートネーム)が「INTERS~1」となるようにディレクトリを作成します。

```
mkdir Interstage1
```

3. dirコマンドを使用して、8.3形式(ショートネーム)のディレクトリ名を確認します。

```
dir /x
XXX/XX/XX XX:XX <DIR> INTERS~1 Interstage1
```

4. Interstageを「C:¥Interstage」にインストールします。
5. dirコマンドを使用して、「C:¥Interstage」の8.3形式(ショートネーム)のディレクトリ名を確認します。

```
dir /x
XXX/XX/XX XX:XX <DIR> INTERS~2 Interstage
XXX/XX/XX XX:XX <DIR> INTERS~1 Interstage1
```

.....

上記の例の場合、先に「Interstage1」というダミーディレクトリを作成することにより、「Interstage」の8.3形式(ショートネーム)のディレクトリ名を「INTERIS~2」にしています。

## 第8章 Interstage HTTP Server運用時の異常

Interstage HTTP Serverの運用でトラブルが発生した場合の対処方法について説明します。

Interstage HTTP Serverでは、運用時のエラーを、Windows(R)システムの場合はイベントログ、Solarisシステム/Linuxシステムの場合はシステムログにメッセージを出力します。「メッセージ集」の「メッセージ番号がihsで始まるメッセージ」を参照し、出力されたメッセージに対する対処を実施してください。

また、アクセスログ、エラーログ、およびトレースログも出力します。各ログの詳細については、「Interstage HTTP Server 運用ガイド」の「ログ」を参照してください。

ここでは、Interstage HTTP Serverの運用時に発生したトラブルの事例、およびその対処について説明します。

### 8.1 Webサーバの起動/停止時の異常

Webサーバの起動/停止時に異常が発生した場合の対処方法について説明します。

- Webサーバの起動/停止時にエラーメッセージが出力された場合 [Windows32/64](#)
- Webサーバの停止時にエラーメッセージが出力された場合 [Linux32/64](#)
- Webサーバの起動に時間がかかる
- 起動タイムアウト時間が有効とならない [Windows32/64](#)
- Apache HTTP Serverの操作中にWebサーバ(Interstage HTTP Server)が停止される [Linux32/64](#)
- Oracle製品がインストールされている場合に、エラーメッセージが出力されてInterstage HTTP Server運用コマンドの実行に失敗する [Linux32/64](#)

#### Webサーバの起動/停止時にエラーメッセージが出力された場合 [Windows32/64](#)

Webサーバの起動/停止時、イベントログに以下のエラーメッセージが出力された場合、デスクトップヒープが使い果たされている可能性があります。

##### エラーメッセージ(1)

```
Apache.exe - アプリケーション エラー :
アプリケーションを正しく初期化できませんでした (0xc0000142)。
[OK] をクリックしてアプリケーションを終了してください。
```

##### エラーメッセージ(2)

```
ihsrlog.exe - アプリケーション エラー :
"0x00000000"の命令が"0x00000000"のメモリを参照しました。
メモリが"read"になることはできませんでした。
プログラムを終了するには [OK] をクリックしてください。
プログラムをデバックするには [キャンセル] をクリックしてください。
```

##### エラーメッセージ(3)

```
障害が発生しているアプリケーション ihsrlog.exe、バージョン XX.X.X.X、
タイム スタンプ 0xXXXXXXXX、障害が発生しているモジュール kernel32.dll、
バージョン X.X.XXXX.XXXX、タイム スタンプ 0xXXXXXXXX、
例外コード 0xc0000142、障害オフセット 0xXXXXXXXX、 プロセス D0xXXX、
アプリケーションの開始時刻 0xXXXXXXXXXXXXXXXXXX。
```

#### エラーメッセージ(4)

```
ihs81101: Failed to start Interstage HTTP Server. Detail: (82-xxx:0:11:0:0:128-0-0) [Webサーバ名]
```

#### エラーメッセージ(5)

```
ihs81101: Failed to start Interstage HTTP Server. Detail: (82-xxx:0:11:0:0:3221225794-0-0) [Webサーバ名]
```

本現象が発生した場合は、デスクトップヒープを拡張する必要があります。デスクトップヒープを拡張する手順については、「[5.5 デスクトップヒープが枯渇した場合](#)」を参照してください。

### Webサーバの停止時にエラーメッセージが出力された場合 Linux32/64

SELinuxが有効な環境において、Interstage管理コンソールでWebサーバの停止時に以下のエラーメッセージが出力された場合、ihsstart/isstart/apachectlコマンドで起動したWebサーバを、Interstage管理コンソールで停止した可能性があります。

#### エラーメッセージ(1)

```
ihs81216: Interstage HTTP Server停止時に異常が発生しました。詳細コード: retcode=10, /bin/sh: line 0: kill: (%s2) - 許可がありません Webサーバ名: %s1
```

**注)** [可変情報]

%s1: Webサーバ名

%s2: プロセスID

#### エラーメッセージ(2)

```
ihs81216: Interstage HTTP Server停止時に異常が発生しました。詳細コード: /bin/sh: line 0: kill: (%s2) - 許可がありません Webサーバ名: %s1
```

**注)** [可変情報]

%s1: Webサーバ名

%s2: プロセスID

本現象が発生した場合は、以下のどちらかの方法で停止してください。

- ihsstopコマンドで、Webサーバ%s1を停止します。
- ismngconsolestopコマンドでInterstage管理コンソールをいったん停止し、ismngconsolestartコマンドでInterstage管理コンソールを起動してから、Interstage管理コンソールでWebサーバ%s1を停止します。

### Webサーバの起動に時間がかかる

Interstage シングル・サインオンにおいてSSOリポジトリにロール定義およびサイト定義を多量に登録してリポジトリサーバを運用している場合、Webサーバの起動に時間がかかることがあります。本現象の詳細および回避方法については、「[16.2.5 リポジトリサーバに関するトラブル](#)」を参照してください。

### 起動タイムアウト時間が有効とならない Windows32/64

Interstage シングル・サインオンにおいてSSOリポジトリにロール定義およびサイト定義を多量に登録してリポジトリサーバを運用している場合、ihsconfigコマンドで設定した起動タイムアウト時間が無効となることがあります。

## Apache HTTP Serverの操作中にWebサーバ(Interstage HTTP Server)が停止される Linux32/64

RHEL5(x86)/RHEL5(Intel64)において、Apache HTTP Server (基本ソフトウェアにバンドルされるhttpd)を停止した場合に、Webサーバ(Interstage HTTP Server)も停止されてしまうことがあります。

本現象は、Apache HTTP Serverのシェル(/etc/rc.d/init.d/httpd)を実行してApache HTTP Serverを停止する場合に、プロセス名“httpd”のすべてのプロセスを終了するため、プロセス名が“httpd”であるWebサーバ(Interstage HTTP Server)のプロセスも終了されることにより、Webサーバ(Interstage HTTP Server)が停止されてしまいます。

本現象が発生した場合は、以下のどちらかの対処を行ってください。

- Apache HTTP Serverをhttpd-2.2.3-43.el5以降のパッケージに移行して運用してください。なお、Apache HTTP Serverのアンインストール時は、アンインストールの処理においてもシェル(/etc/rc.d/init.d/httpd)が実行されるため、同様の現象が発生するので、注意してください。
- Apache HTTP Serverを停止する場合は、Apache HTTP Serverのapachectlコマンドを使用して停止するようにしてください。

## Oracle製品がインストールされている場合に、以下のエラーメッセージが出力されてInterstage HTTP Server運用コマンドの実行に失敗する Linux32/64

■(コマンド名): error while loading shared libraries: /xxx/libexpat.so.0: file too short

本現象は、Oracle製品がインストールされている場合に、Oracle製品の資源にシステムライブラリ(libexpat.so.0)と同名のライブラリが存在し、かつ環境変数LD\_LIBRARY\_PATHにそのライブラリのパスが設定されているために発生します。

本現象が発生した場合は、環境変数LD\_LIBRARY\_PATHの設定値を確認し、以下のパスを先頭に設定してください。

- Linux for x86/Linux for Intel64 (32ビット互換モード)の場合: “/lib”
- Linux for Intel64 (64ビットモード)の場合: “/lib64”

## 8.2 Webサーバの運用時の異常

Webサーバの運用時に異常が発生した場合の対処方法について説明します。

- Webサーバから応答がない/ Webサーバからのレスポンスが遅延する
- ログファイルが出力されなくなった
- Webブラウザからアクセスすると、意図しないページが表示される

### 参照

Webアプリケーション運用時の異常については、以下のマニュアルを参照してください。

- Java EE機能を使用している場合  
「第10章 Java EE使用時の異常」-「10.3 Webアプリケーションの開発・運用時の異常」

**EE SJE**

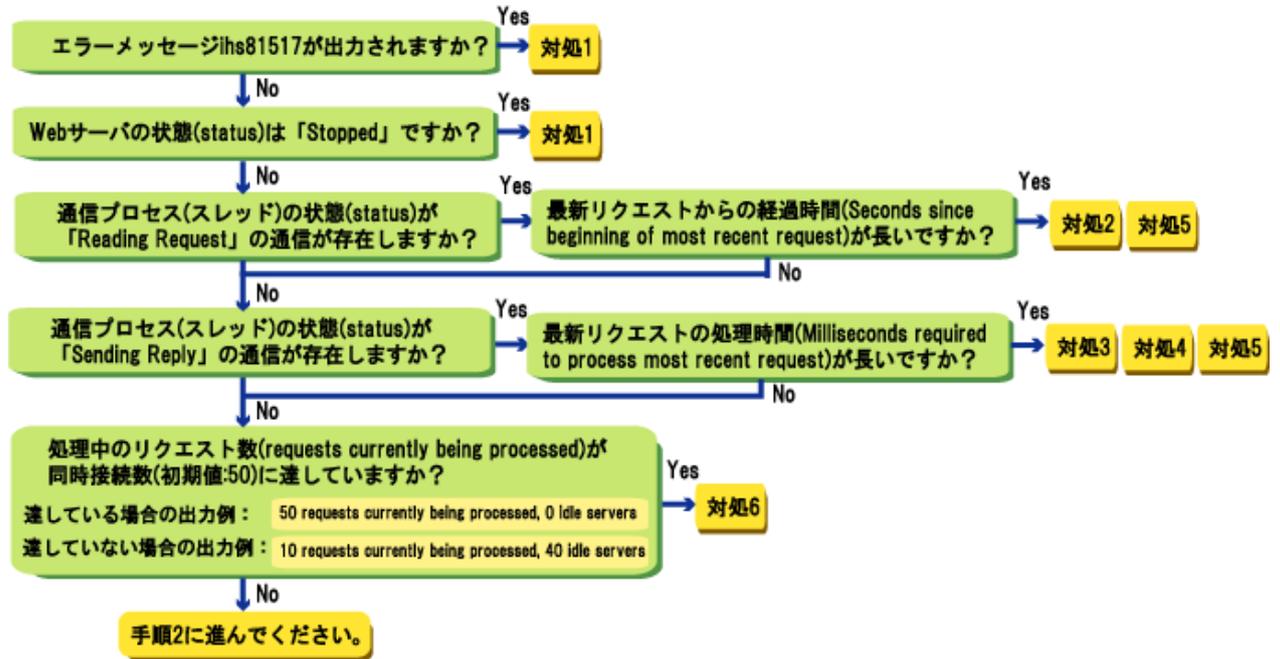
- J2EE機能を使用している場合  
「J2EE ユーザーズガイド(旧版互換)」の「J2EEアプリケーション開発・運用時の異常」-「Webアプリケーションの開発・運用時の異常」

### Webサーバから応答がない/ Webサーバからのレスポンスが遅延する

本現象が発生した場合、以下の手順で問題がないかを確認し、それぞれの対処を行ってください。

## ■手順1(Webサーバの状態確認)

ihsdispコマンドを-aオプションを指定して実行し、運用中のWebサーバの状態を表示します。その表示内容について、以下の図を参考にして問題がないかを確認し、それぞれの**対処**を行ってください。Webサーバの動作状態の表示方法については、「Interstage HTTP Server 運用ガイド」の「運用・保守」-「動作状態表示」を参照してください。



## ■手順2(トレースログの確認)

トレースログを参照し、本現象が発生した時間帯のトレース情報の出力内容から問題がないかを確認します。トレースログの詳細については、「Interstage HTTP Server 運用ガイド」の「ログ」-「トレースログ」を参照してください。

1. トレースログのイベント(conn, recv, call, rtn, send, disc)出力を以下の方法で抽出します。イベントの出力の流れについては、以下の図を参考にしてください。

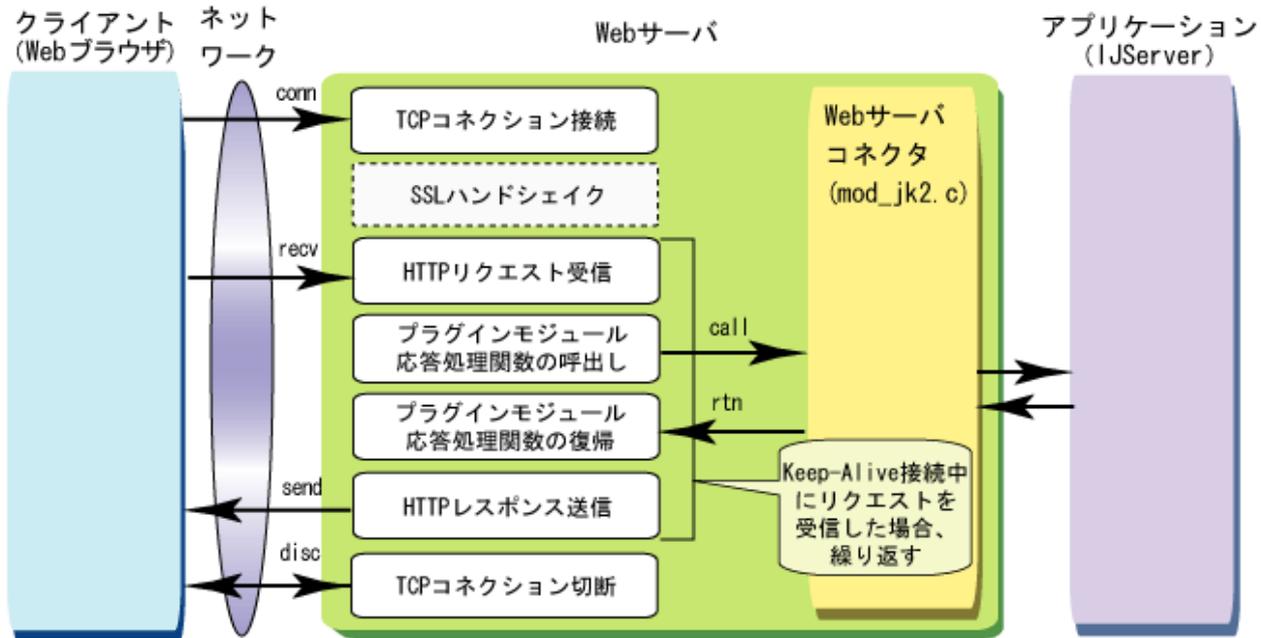
### Windows32/64

トレース情報を、同一通信スレッドのスレッドID単位で抽出します。本現象発生時のクライアントのIPアドレスがわかっている場合は、そのクライアントIPアドレスが出力されているconnイベントのスレッドID単位で抽出してください。

### Solaris32/64 Linux32/64

トレース情報を、同一通信プロセスのプロセスID単位で抽出します。本現象発生時のクライアントのIPアドレスがわか

かっている場合は、そのクライアントIPアドレスが出力されているconnイベントのプロセスID単位で抽出してください。



2. 1.で抽出したトレース情報の出力内容について、以下の図を参考にしてそれぞれの対処を行います。



## ■ 対処1

Webサーバが起動されていません。  
Webサーバを起動してください。

## ■ 対処2

Webサーバにリクエストが届いていない可能性があります。  
クライアントからリクエストデータが送信されているかを確認してください。

## ■ 対処3

POSTリクエストの場合、クライアントからボディデータが届いていない可能性があります。  
クライアントからPOSTリクエストのボディデータが送信されているかを確認してください。

## ■ 対処4

Servletサービスのアプリケーションの処理が遅延している可能性があります。  
Webサーバコネクタおよびアプリケーションに問題が発生していないかを確認してください。  
**注)**「手順2(トレースの確認)」では、トレース情報の出力内容が以下の状態の場合に該当します。

- callイベントのあとに、rtmイベントが出力されていない
- callイベントとrtmイベントの時間差が大きい

## ■ 対処5

リクエスト送信元のクライアントと送信先のWebサーバ間のネットワーク機器(ルータ、負荷分散装置など)で遅延が発生している可能性があります。  
クライアントとWebサーバ間のネットワーク機器(ルータ、負荷分散装置など)が正常に動作しているかを確認してください。

## ■ 対処6

処理中のリクエスト数(requests currently being processed)が同時接続数に達している場合は、必要に応じて、以下の手順でクライアントの同時接続数の設定値を増加してください。

- Interstage管理コンソールで設定する場合  
Webサーバの環境設定画面で設定します。Interstage管理コンソールの定義方法については、Interstage管理コンソールのヘルプを参照してください。
- 環境定義ファイル(httpd.conf)を編集して設定する場合
  - **Windows32/64**  
ThreadsPerChildディレクティブの設定値を変更します。
  - **Solaris32/64** **Linux32/64**  
MaxClientsディレクティブの設定値を変更します。

環境定義ファイル(httpd.conf)の設定方法の詳細については、「Interstage HTTP Server 運用ガイド」の「環境設定」-「環境定義ファイル」-「クライアントの同時接続数の設定」を参照してください。

なお、Webサーバの起動後、リクエスト数がはじめてクライアントの同時接続数に達した際に、以下のメッセージがエラーログに出力されます。エラーログのメッセージの内容と対処については、「メッセージ集」の「Interstage HTTP Server が出力するメッセージ」-「Interstage HTTP Serverのエラーログ」を参照してください。

**Windows32/64**

Server ran out of threads to serve requests. Consider raising the ThreadsPerChild setting

[Solaris32/64](#) [Linux32/64](#)

The number of requests received simultaneously reached the value set in the MaxClients directive.

#### ■ 対処7

SSL通信においてSSLハンドシェイクに失敗した可能性があります。

エラーログが出力されていないかを確認してください。エラーログが出力されている場合は、そのメッセージの異常の原因を取り除いてください。エラーログのメッセージの内容と対処については、「メッセージ集」の「Interstage HTTP Serverが出力するメッセージ」-「Interstage HTTP Serverのエラーログ」を参照してください。

#### ■ 対処8

ログのローテーション機能により、問題が発生した時間帯のトレースログが削除された可能性があります。

トレースログのファイルサイズおよびファイル数の上限値を確認してください。トレースログの設定については、「Interstage HTTP Server 運用ガイド」の「環境定義ファイル」-「トレースログの設定」を参照してください。

#### ■ 対処9

iscollectinfoコマンドを使用して調査情報を採取した後、技術員に連絡してください。資料採取の詳細については、「メッセージ集」の「技術員連絡の前に」を参照してください。

### ログファイルが出力されなくなった

本現象が発生した場合、最新のログファイルを誤って操作した可能性があります。Webサーバ運用中は、最新のログファイルの編集/削除などの操作を行うことはできません。最新のログファイルを誤って操作してしまった場合は、Webサーバを再起動してください。

### Webブラウザからアクセスすると、意図しないページが表示される

本現象が発生した場合、Interstage HTTP Serverのポート番号(省略値:80)と他のWebサーバ(基本ソフトウェアにバンドルされるApache HTTP Serverなど)のポート番号が重複している可能性があります。

Webサーバを共存して運用する場合は、すべてのWebサーバに異なるポート番号を設定する必要があります。

他のWebサーバが同一のポート番号で起動されている場合は、Interstage HTTP Serverのポート番号を変更して、Webサーバを再起動してください。Interstage HTTP Serverのポート番号の設定方法については、「Interstage HTTP Server 運用ガイド」の「環境設定」-「環境定義ファイル」-「ポート番号とIPアドレスの設定」を参照してください。

## 8.3 よくある質問とその対処方法

Interstage HTTP Serverの運用について、よくある質問とその対処方法を以下に示します。

- ・ 「Interstage HTTP Server 運用ガイド」の「ディレクティブ一覧」に記載されていないディレクティブを使用できますか
- ・ Apache HTTP Server Version 2.0.59をベースとしているが、セキュリティホール関連の問題については対応されていますか
- ・ Interstage HTTP Serverを複数起動できますか
- ・ Interstage HTTP Serverと他のWebサーバ(Apache HTTP Serverなど)を共存できますか

- Interstage HTTP ServerとMicrosoft(R) Internet Information Servicesで同時にWebサーバコネクタを利用できますか  

- Interstage HTTP ServerとSun Java System Web Serverで同時にWebサーバコネクタを利用できますか  

- OpenSSLで発行した証明書を使用できますか
- サードパーティのプラグインモジュールをサポートしていますか
- 静的コンテンツに対して、どのようなHTTPメソッドを使用できますか

## 「Interstage HTTP Server 運用ガイド」の「ディレクティブ一覧」に記載されていないディレクティブを使用できますか

Interstage HTTP Serverでは、Apache HTTP サーバ バージョン 2.0で提供されているディレクティブ(一部のディレクティブを除く)を使用できます。

「Interstage HTTP Server 運用ガイド」の「ディレクティブ一覧」に説明されていないApache HTTP Server Version 2.0のディレクティブについては、「Apache HTTP サーバ バージョン 2.0ドキュメント」を参照して使用してください。



### 参照

- Interstage HTTP Serverで使用できない一部のディレクティブについては、「Interstage HTTP Server 運用ガイド」の「ディレクティブ一覧」-「ディレクティブの注意事項」を参照してください。
- Apache HTTP サーバ バージョン 2.0ドキュメントの参照方法については、「Interstage HTTP Server 運用ガイド」の「Apache HTTP サーバ バージョン 2.0ドキュメントの参照方法」を参照してください。

## Apache HTTP Server Version 2.0.59をベースとしているが、セキュリティホール関連の問題については対応されていますか

本製品では、Apache HTTP Serverで検出されたセキュリティホールにおいて、Interstage HTTP Serverに影響がある修正については、Apache HTTP Server Version 2.0.65まで対応しています。また、本製品出荷以降、セキュリティホール関連の問題が発生した場合は、緊急修正にて随時対応していきます。

Interstage HTTP Serverのセキュリティホール関連の対応については、以下のサイトで提供している富士通製品のセキュリティ情報にて確認してください。

<http://software.fujitsu.com/jp/security/index.html>

## Interstage HTTP Serverを複数起動できますか

Interstage HTTP Serverでは、1つのシステム上に複数のWebサーバの構築・運用を行うことができます。



### 参照

Webサーバの構築・運用については、「Interstage HTTP Server 運用ガイド」の「複数Webサーバの運用」を参照してください。

## Interstage HTTP Serverと他のWebサーバ(Apache HTTP Serverなど)を共存できますか

Interstage HTTP Serverでは、他のWebサーバ(Apache HTTP Serverなど)と共存できます。ただし、1つのポート番号を複数のWebサーバで利用できません。必ずWebサーバごとに異なるポート番号を設定してください。

## Interstage HTTP ServerとMicrosoft(R) Internet Information Servicesで同時にWebサーバコネクタを利用できますか

Interstage HTTP Serverでは、Microsoft(R) Internet Information Servicesと同時にWebサーバコネクタを利用できません。

## Interstage HTTP ServerとSun Java System Web Serverで同時にWebサーバコネクタを利用できますか

Interstage HTTP Serverでは、Sun Java System Web Serverと同時にWebサーバコネクタを利用できません。

## OpenSSLで発行した証明書を使用できますか

Interstage HTTP Serverでは、OpenSSLで発行した証明書を使用できません。



証明書については、「セキュリティシステム運用ガイド」の以下の項を参照してください。

- Interstage証明書環境のSSLを使用する場合  
「Interstage証明書環境の構築と利用」-「証明書と秘密鍵について」
- SMEEコマンドで構築する証明書/鍵管理環境のSSLを使用する場合  
「SMEEコマンドによる証明書/鍵管理環境の構築と利用」-「証明書/鍵管理環境で使用できるSSLライブラリについて」-「証明書/鍵管理環境について」

## サードパーティのプラグインモジュールをサポートしていますか

Interstage HTTP Serverでは、mod\_perl/mod\_phpなどを含むサードパーティのプラグインモジュールはサポートしていません。富士通製品で提供しているプラグインモジュールだけ使用できます。

なお、Interstage HTTP Serverのサポート対象外のプラグインモジュールを導入した場合は、Interstage HTTP Server全体の動作についてもサポート対象外となります。

## 静的コンテンツに対して、どのようなHTTPメソッドを使用できますか

Interstage HTTP Serverでは、静的コンテンツに対して、以下のHTTPメソッドを使用できます。POSTメソッドは、使用できません。

- GET
- HEAD
- OPTIONS
- TRACE

## 第9章 J2EEアプリケーション開発・運用時の異常

J2EEアプリケーションの開発や運用中に異常が発生した場合の対処方法については、「J2EE ユーザーズガイド(旧版互換)」を参照してください。

## 第10章 Java EE使用時の異常

Java EEアプリケーションの運用時に発生するトラブルの解決方法について説明します。

### 10.1 コマンド実行時の異常

コマンド実行時の以下のトラブルについて対処方法を説明します。

- ・ コマンド実行時にパスワード入力を要求される
- ・ エラーメッセージの出力
- ・ 例外メッセージの出力
- ・ コマンド復帰の遅延
- ・ 定義項目の更新時、意図しない値が保存される

#### コマンド実行時にパスワード入力を要求される

コマンド実行時のパスワード入力を省略したい場合は、「loginサブコマンド」を参照して対処を行ってください。

#### エラーメッセージの出力

コマンド実行時に異常が発生した場合、コマンドを実行したターミナルへエラーメッセージが出力されます。「メッセージ集」-「Java EEのメッセージラベルのあるメッセージ」を参照して対処してください。

#### 例外メッセージの出力

コマンド実行時に例外メッセージが出力された場合は、以下を参照して対処してください。

##### **java.lang.OutOfMemoryError**

メモリ不足が発生している可能性があります。以下のマニュアルを参照して対処を行ってください。

- ・ チューニングガイド
  - メモリ領域不足事象発生時のメッセージ出力機能の強化
  - java.lang.OutOfMemoryErrorがスローされた場合

java.lang.OutOfMemoryErrorの出力例を以下に示します。

```
Exception in thread "RMI
RenewClean-[192.0.2.1:42964,com.sun.appserv.management.client.AdminRMISSLClientSocketFactory@4f2ec]"
java.lang.OutOfMemoryError: Java heap space
at java.io.ByteArrayOutputStream.<init>(ByteArrayOutputStream.java:59)
The memory was exhausted on Java heap space. : requested 48 bytes
Java heap size / max Java heap size = 59703288 / 67108864
Java perm size / max Java perm size = 12748800 / 67108864
```

##### **java.net.SocketTimeoutException**

高負荷によりInterstage Java EE DASサービス、Interstage Java EE Node Agentサービス、またはIIServerクラスタとの通信が失敗した可能性があります。

システムの負荷状況を確認し、システムリソースに空きができれば再度コマンドを実行してください。

#### コマンド復帰の遅延

処理開始から長時間経過してコマンドが異常復帰する場合、複数の操作が同時に実行されたことでInterstage Java EE DASサービスに接続できない状態となっている可能性があります。

この場合、コマンドの実行はエラーとなりますが、Java EEの定義・運用には影響はありません。同時に他の操作が行われていないことを確認し、再度コマンドを実行してください。

## 定義項目の更新時、意図しない値が保存される

他のユーザーが以下により同時に操作を行ったため、指定値が上書きされた可能性があります。

- Interstage Java EE管理コンソール
- asadminコマンド

同時に操作を行っているユーザーがいないか確認し、必要に応じて操作を再度実行してください。

## 10.1.1 初期化コマンドに関する異常

---

初期化コマンド実行時の以下のトラブルについて対処方法を説明します。

### 実行に失敗する

以下のプロセスが残存している可能性があります。該当するプロセスが残存しているかどうか確認してください。いずれかのプロセスが残存している場合、プロセスを終了した後、再実行してください。

- メッセージブローカー

- 確認方法

#### Windows32/64

タスクマネージャのプロセス一覧からイメージ名「imqbrokersvc.exe」が存在するかどうか確認してください。

#### Solaris32/64

以下のコマンドを実行して、imqbrokerdコマンドのプロセスが存在するかどうか確認してください。

##### [global zoneの場合]

```
ps -efz global | grep imqbrokerd
```

##### [non-global zoneの場合]

```
ps -ef | grep imqbrokerd
```

#### Linux32/64

以下のコマンドを実行して、imqbrokerdコマンドのプロセスが存在するかどうか確認してください。

```
ps -ef | grep imqbrokerd
```

- 停止方法

「10.8 JMSの開発・運用時の異常」の「メッセージブローカを停止できない」を参照してください。

- Java DB(クライアント/サーバー環境)

- 停止方法

Java DBのシステムディレクトリ配下に格納されたderby.logからJava DBのポート番号を特定し、そのポート番号を指定してasadminコマンドのstop-databaseサブコマンドを実行してください。

残存プロセスをすべて停止しても、初期化コマンドの実行に失敗する場合は、システムを再起動した後、再度初期化コマンドを実行してください。

## 10.2 運用中の異常

---

運用中に発生するトラブルについて説明します。

### 10.2.1 運用環境に関する異常

---

Java EEアプリケーションの運用環境に関する以下のトラブルについて対処方法を説明します。

- IIServerクラスタ起動時の異常
- IIServerクラスタ停止時の異常

- サーバーインスタンスの異常終了
- サーバーインスタンスの状態取得コマンドの復帰に時間がかかる場合
- Interstage Java EE Node Agentサービス起動時の異常
- Interstage Java EE Node Agentサービス停止時の異常
- Interstage Java EE Node Agentサービスの異常終了
- Interstage Java EE Node Agentサービスのプロセス回収手順
- Interstage Java EE Node Agentサービスに関するタイムアウト値のチューニング
- Interstage Java EE DASサービス起動時の異常
- Interstage Java EE DASサービス停止時の異常
- Interstage Java EE DASサービスの異常終了
- Interstage Java EE DASサービスが操作を受け付けない場合
- Interstage Java EE DASサービスの状態取得コマンドの復帰に時間がかかる場合
- Interstage Java EE DASサービスのプロセス回収手順
- Interstage Java EE DASサービスに関するタイムアウト値のチューニング
- Interstage Java EE DASサービスでのアプリケーション運用時の異常

## IJServerクラスタ起動時の異常

IJServerクラスタの起動に失敗した場合は、以下のログにエラーメッセージが出力されていないか確認し、「メッセージ集」-「Java EEのメッセージラベルのあるメッセージ」を参照して起動失敗の原因を調査してください。

- イベントログ(Windows)/システムログ(Solaris/Linux)
- サーバーログ
- Java VMログ
- IJServerクラスタの起動処理遅延またはハングアップによりIJServerクラスタ起動待ち時間を超過した場合、Java VMログにJava VMプロセスのフルスレッドダンプが出力されます。フルスレッドダンプを参照し、遅延やハングアップの原因となっているアプリケーションがないか確認してください。
- 通信ポートを獲得できないために起動に失敗している場合は、「システム設計ガイド」-「ポート番号」を参照してポート番号をチューニングしてください。
- Java VMログにJava VMの異常を示すメッセージが出力されている場合は、「トラブルシューティング集」の「Java実行環境運用時の異常」を参照し、起動に失敗した原因を調査してください。
- 各種ログに出力された情報から原因を特定できない場合は、`iscollectinfo`コマンドで資料採取を行った上で技術員に連絡してください。

## IJServerクラスタ停止時の異常

IJServerクラスタ停止時に`core`ファイルが出力される場合、`java.lang.System`クラスや`java.lang.Runtime`クラスの`runFinalizersOnExit(Boolean)`メソッドの引数に`"true"`を指定してファイナライズを有効にしているアプリケーションがないか確認してください。本メソッドは非奨励のため、利用しないでください。

## サーバーインスタンスの異常終了

IJServerクラスタプロセスが異常終了した場合は、以下のログにエラーメッセージが出力されていないか確認し、「メッセージ集」-「Java EEのメッセージラベルのあるメッセージ」を参照して異常終了の原因を調査してください。なお、プロセスの自動再起動機能により業務継続が可能です。

- イベントログ(Windows)/システムログ(Solaris/Linux)

- ・ サーバーログ
- ・ Java VMログ

Java VMログおよびサーバーログにはJDK/JREの異常メッセージが出力される場合があります。調査の際は、以下のマニュアルも参照してください。

- ・ 「チューニングガイド」の「JDK/JREのチューニング」

各種ログに出力された情報から原因を特定できない場合は、`iscollectinfo`コマンドで資料採取を行った上で技術員に連絡してください。

なお、運用中にIIServerクラスタプロセスが異常終了した場合、プロセスの自動再起動機能により業務継続が可能です。詳細は、「Java EE運用ガイド」-「Java VMプロセスの生死監視」を参照してください。

### サーバーインスタンスの状態取得コマンドの復帰に時間がかかる場合

サーバーインスタンスがメモリ不足などにより高負荷状態となっている場合、サーバーインスタンスの状態取得コマンド(`asadmin list-instances`コマンド)の復帰に時間がかかり、かつ、結果として「unknown」が表示される場合があります。

このような場合、しばらく経ってからコマンドを再度実行してください。

なお、高負荷状態のサーバーインスタンスを停止したい場合は、IIServerクラスタ、または、サーバーインスタンスの強制停止を利用してください。

### Interstage Java EE Node Agentサービス起動時の異常

Interstage Java EE Node Agentサービスの起動に失敗した場合は、以下のログにエラーメッセージが出力されていないか確認し、「メッセージ集」-「Java EEのメッセージラベルのあるメッセージ」を参照して起動失敗の原因を調査してください。

- ・ イベントログ(Windows)/システムログ(Solaris/Linux)
- ・ サーバーログ

通信ポートを獲得できないために起動に失敗している場合は、「システム設計ガイド」-「ポート番号」を参照してポート番号をチューニングしてください。

Interstage Java EE Node Agentサービス起動時に、タイムアウトが発生して起動に失敗する場合は、[Interstage Java EE Node Agentサービスに関するタイムアウト値のチューニング](#)にしたがってタイムアウト値のチューニングを行ってください。

各種ログに出力された情報から原因を特定できない場合は、`iscollectinfo`コマンドで資料採取を行った上で技術員に連絡してください。

### Interstage Java EE Node Agentサービス停止時の異常

Interstage Java EE Node Agentサービス停止時に、タイムアウトが発生して停止に失敗する場合、[Interstage Java EE Node Agentサービスのプロセス回収手順](#)にしたがって停止させた後、[Interstage Java EE Node Agentサービスに関するタイムアウト値のチューニング](#)にしたがってタイムアウト値のチューニングを行ってください。

### Interstage Java EE Node Agentサービスの異常終了

Interstage Java EE Node Agentサービスが異常終了した場合、`iscollectinfo`コマンドで資料採取を行った上で技術員に連絡してください。

なお、Interstage Java EE Node Agentサービスが異常終了すると、以下の機能は動作しません。

- ・ IIServerクラスタのJava VMプロセスの生死監視
- ・ IIServerクラスタプロセス異常時の自動再起動
- ・ アプリケーション最大処理時間の監視 (注1)
- ・ Javaヒープ/Permanent世代領域不足時の制御
- ・ Java VMログ(jvm.log)への情報出力
- ・ IIServerクラスタの運用状態表示

**注1)**Javaヒープ/Permanent世代領域不足時の制御ではInterstage Java EE Node Agentサービスが異常終了していると、選択している制御方法により、以下の動作となります。

- 「プロセスを再起動する」の場合  
プロセスの再起動を行うことができないため、プロセスダウンとなります。
- 「アプリケーションにjava.lang.OutOfMemoryErrorを返却する」の場合  
動作に影響はありません。  
アプリケーションにjava.lang.OutOfMemoryErrorが返却されます。

ただし、運用中のアプリケーションの疎通への影響はありません。  
リカバリは、以下の手順で行います。

1. asadminコマンドのlist-instancesサブコマンドで運用中のサーバーインスタンスを確認します。
2. 起動しているサーバーインスタンスに対して、以下のコマンドを実行します。

**Windows32/64**

[Java EE共通ディレクトリ]¥nodeagents¥ijna¥<サーバーインスタンス名>¥bin¥stopserv

**Solaris32/64 Linux32/64**

[Java EE共通ディレクトリ]/nodeagents/ijna/<サーバーインスタンス名>/bin/stopserv

3. 以下の手順でInterstage Java EE Node Agentサービスを起動します。

**Windows32/64**

Interstage Java EE Node Agentサービスを停止して、再起動します。

**Solaris32/64 Linux32/64**

Interstage Java EE Node Agentサービスを起動します。

4. IJServerクラスタを再度起動します。

## 注意

リカバリ手順を行わずにInterstage Java EE Node Agentサービスを起動した場合、リカバリを行っていないサーバーインスタンスに対する操作を正しく行うことができません。リカバリ手順を実施することで正常な状態に復旧することが可能です。リカバリ手順を行わずにInterstage Java EE Node Agentサービスを起動した状態で、リカバリを行う対象のサーバーインスタンスを見分けるためには、Interstage Java EE Node Agentサービスの起動時刻とサーバーインスタンスの起動時刻を比較することで判別可能です。サーバーインスタンスの起動時刻が小さい方をリカバリ対象と判断してください。Interstage Java EE Node Agentサービスの起動時刻はシステムログ・イベントログに出力されるijna10000のメッセージの時刻を、サーバーインスタンスの起動時刻はサーバーログに出力されるSERVER1007の時刻をそれぞれ参照ください。

## Interstage Java EE Node Agentサービスのプロセス回収手順

以下の手順により、Interstage Java EE Node Agentサービスのプロセスを回収してください。

1. 以下のファイルを参照して、Interstage Java EE Node AgentサービスのプロセスIDを確認してください。

**Windows32/64**

[Java EE共通ディレクトリ]¥nodeagents¥ijna¥agent¥config¥.\_\_com\_sun\_appsर्व\_pid

**Solaris32/64 Linux32/64**

[Java EE共通ディレクトリ]/nodeagents/ijna/agent/config/.\_\_com\_sun\_appserv\_pid

(注) ファイルが存在していない場合、またはjavaプロセスが存在しない場合、プロセスは存在していません。

2. 1で確認したIDのプロセスがjavaプロセスとして存在しているか、以下の方法で確認してください。

**Windows32/64**

Windowsのタスクマネージャで確認してください。

**Solaris32/64 Linux32/64**

psコマンドを使用して確認してください。

3. 1で確認したプロセスIDに対して、以下の方法でプロセスを回収してください。

**Windows32/64**

Windowsのタスクマネージャでプロセスを終了させてください。

**Solaris32/64 Linux32/64**

killコマンドを使用してプロセスを削除してください。

上記手順と合わせて、サーバーインスタンスのプロセスが存在している場合は、[Interstage Java EE Node Agentサービスの異常終了](#)を参照してサーバーインスタンスのプロセスの回収も行ってください。

## Interstage Java EE Node Agentサービスに関するタイムアウト値のチューニング

下記のファイルを編集してください。

- [Java EE共通ディレクトリ]¥domains¥interstage¥config¥domain.xml
- [Java EE共通ディレクトリ]¥nodeagents¥ijna¥agent¥config¥domain.xml

下記のname属性とvalue属性を定義した、system-property要素を追加してください。

```
(省略)
<domain application-root=" . . .
  :
(省略)
  :
  <system-property name="タイムアウト名" value="タイムアウト値"/> ★追加
  <property name="administrative.domain.name" value="interstage"/>
</domain>
```

指定可能なタイムアウト名は下記です。

### NA\_STARTING\_TIMEOUT

起動処理で、状態がstartingに変化する時間  
指定可能範囲は180～65535[秒]、デフォルト180です。

### NA\_STOPPED\_TIMEOUT

停止処理で、状態がnot runningに変化する時間  
指定可能範囲は60～65535[秒]、デフォルト60です。

## Interstage Java EE DASサービス起動時の異常

Interstage Java EE DASサービスの起動に失敗した場合は、以下のログにエラーメッセージが出力されていないか確認し、「メッセージ集」-「Java EEのメッセージラベルのあるメッセージ」を参照して起動失敗の原因を調査してください。

- イベントログ(Windows)/システムログ(Solaris/Linux)
- Interstage Java EE DASサービスのサーバーログ

サーバーログにメモリ不足を示すエラーメッセージが出力されている場合は、「チューニングガイド」-「Interstage Java EE DASサービスのヒープ領域サイズとアドレス空間」を参照して対処を行ってください。

ログファイルについての詳細は、「Java EE運用ガイド」-「Java EEアプリケーションのログ」を参照してください。

**Windows32/64**

以下のログファイルに”java.lang.NoClassDefFoundError”が出力されている場合、環境変数PATHにダブルクォート(”)が含まれている可能性があります。

[JServerクラスタのログ出力ディレクトリ]¥start.log

※) [IJServerクラスタのログ出力ディレクトリ]のデフォルトは、[Java EE共通ディレクトリ]¥nodeagents¥ijna¥agent¥logsです。環境変数PATHにダブルクォートが含まれているとInterstage Java EE DASサービスが起動できないため、環境変数PATHを確認してダブルクォートが含まれている場合は、ダブルクォートを削除してください。

## 通信ポートの獲得失敗による起動失敗

通信ポートを獲得できないために起動に失敗している場合は、「システム設計ガイド」-「ポート番号」を参照してポート番号をチューニングしてください。

なお、他のサービス・プロセスとのポート番号重複によりInterstage Java EE DASサービスの起動に失敗している場合、重複したポート番号を利用しているサービス、プロセスを一旦停止した上で、Interstage Java EE DASサービスを起動して設定変更を行ってください。

## メモリ不足による起動失敗

IJServerクラスタ、およびサーバーインスタンスを多く定義した場合、メモリ不足によりInterstage Java EE DASサービスの起動に失敗する場合があります。

メモリ不足によりInterstage Java EE DASサービスが起動しない場合は、下記手順に従いdomain.xmlを編集してください。

### 1) Javaヒープ領域の拡張

以下手順によりInterstage Java EE DASサービスのJavaヒープ領域を拡張します。

domain.xmlの格納先:

**Windows32/64**

[Java EE共通ディレクトリ]¥domains¥interstage¥config¥domain.xml

**Solaris32/64 Linux32/64**

[Java EE共通ディレクトリ]/domains/interstage/config/domain.xml

1. Interstage Java EE Node Agentサービスを停止してください。
2. Interstage Java EE DASサービスを停止してください。通常停止を実施できない場合には、本手順実施後にシステムを再起動してください。
3. domain.xmlを編集し、ヒープ領域サイズを増やしてください。  
テキストエディタなどでファイルを開き、下記 `-Xmx512m` を `-Xmx1024m` に変更してください。

[編集する要素]

```
<domain>
  <configs>
    <config dynamic-reconfiguration-enabled="true" name="server-config">
      <java-config>
        <jvm-options>-Xmx512m</jvm-options>
      </java-config>
    </config>
  </configs>
</domain>
```

4. `asadmin verify-domain-xml`サブコマンドを利用し、定義の内容を検証してください。
5. Interstage Java EE DASサービスを起動してください。通常停止を行う際に失敗していた場合は、システムを再起動してください。
6. Interstage Java EE Node Agentサービスを起動してください。

### 2) IJServerクラスタの削除

クラスタの定義項目"ハートビート"が有効に設定されている場合、定義済みのIJServerクラスタを削除することにより、Interstage Java EE DASサービスのメモリ使用量を削減できる可能性があります。

"物理メモリ使用量"に記載した見積もり式に当てはめて削除する必要のあるIJServerクラスタ数を算出し、以下の手順でIJServerクラスタを削除してください。作業を行う前に、事前にIJServerクラスタの配備資源やdomain.xml、

workers2.propertiesのバックアップをとることを推奨します。バックアップの方法は、「運用ガイド(基本編)」-「資源のバックアップとリストア」を参照してください。

#### domain.xmlの格納先:

##### Windows32/64

[Java EE共通ディレクトリ]\domains\interstage\config\domain.xml

##### Solaris32/64 Linux32/64

[Java EE共通ディレクトリ]/domains/interstage/config/domain.xml

#### workers2.propertiesの格納先:

##### Windows32/64

C:\Interstage\F3FMjs5\conf\jk2\[Webサーバ名]\workers2.properties

##### Solaris32/64 Linux32/64

/opt/FJsvjs5/conf/jk2/[Webサーバ名]/workers2.properties

1. Interstage Java EE Node Agentサービスを停止してください。通常停止を実施できない場合には、プロセスを強制停止してください。
2. Interstage Java EE DASサービスを停止してください。通常停止を実施できない場合には、プロセスを強制停止してください。
3. Webサーバを停止してください。詳細は、「Interstage HTTP Server 運用ガイド」を参照してください。
4. Interstage管理コンソールを停止してください。詳細は、「運用ガイド(基本編)」を参照してください。
5. 以下のとおりdomain.xmlを編集してください。

#### heartbeatの無効化

すべてのIJServerクラスタに対応したcluster要素の"heartbeat-enabled"を"false"に設定してください。

```
<cluster config-ref="{IJServer名}-config"  
heartbeat-address="{ハートビートアドレス}"  
heartbeat-enabled="false"  
heartbeat-port="{ハートビートポート}" name="{IJServerクラスタ名}">  
~  
</cluster>
```

6. asadmin verify-domain-xmlサブコマンドを利用し、定義の内容を検証してください。
7. Webサーバを起動してください。詳細は、「Interstage HTTP Server 運用ガイド」を参照してください。
8. Interstage管理コンソールを起動してください。詳細は、「運用ガイド(基本編)」を参照してください。
9. Interstage Java EE DASサービスを起動してください。
10. asadminコマンド、またはGUIを利用して、削除対象のIJServerクラスタとサーバーインスタンスを削除してください。
11. Interstage Java EE Node Agentサービスを起動してください。

### Java EE共通ディレクトリが参照できないことによる起動失敗

Java EE共通ディレクトリが参照できない場合、Interstage Java EE DASサービスの起動に失敗します。クラスタサービス連携のためにJava EE共通ディレクトリを共用ディスクに配置している場合は、必ずJava EE共通ディレクトリを参照可能な運用ノードから操作を行ってください。

## タイムアウトによる起動失敗

Interstage Java EE DASサービス起動時に、タイムアウトが発生して起動に失敗する場合、[Interstage Java EE DASサービスに関するタイムアウト値のチューニング](#)にしたがってタイムアウト値のチューニングを行ってください。

各種ログに出力された情報から原因を特定できない場合は、`iscollectinfo`コマンドで資料採取を行った上で技術員に連絡してください。

## Interstage Java EE DASサービス停止時の異常

Interstage Java EE DASサービス停止時に、タイムアウトが発生して停止に失敗する場合、[Interstage Java EE DASサービスのプロセス回収手順](#)にしたがって停止させた後、[Interstage Java EE DASサービスに関するタイムアウト値のチューニング](#)にしたがってタイムアウト値のチューニングを行ってください。

## Interstage Java EE DASサービスの異常終了

Interstage Java EE DASサービスが異常終了した場合、システムログ/イベントログにエラーメッセージが通知されます。`iscollectinfo`コマンドで資料採取を行った上で技術員に連絡してください。

なお、Interstage Java EE DASサービスが異常終了すると、Interstage Java EE管理コンソール、および`asadmin`コマンドは利用できません。再度、Interstage Java EE DASサービスを起動することで、上記機能が有効になります。

## Interstage Java EE DASサービスが操作を受け付けられない場合

メモリ不足により、Interstage Java EE DASサービスが操作を受け付けできない状態となっている可能性があります。なお、この場合、Interstage Java EE DASサービスが起動していても、`ijdasstat`コマンドが「unknown」を表示することがあります。Interstage Java EE DASサービスのサーバーログを参照し、メモリ不足を示すエラーメッセージが出力されていないか確認してください。

ログファイルについての詳細は、「Java EE運用ガイド」-「Java EEアプリケーションのログ」を参照してください。

サーバーログにメモリ不足を示すエラーメッセージが出力されている場合は、[Interstage Java EE DASサービスのプロセス回収手順](#)を参照してプロセスを回収した上、以下マニュアルを参照してヒープサイズのチューニングを実施してください。

「チューニングガイド」-「Interstage Java EE DASサービスのヒープ領域サイズとアドレス空間」

## Interstage Java EE DASサービスの状態取得コマンドの復帰に時間がかかる場合

Interstage Java EE DASサービスがメモリ不足などにより高負荷状態となっている場合、Interstage Java EE DASサービスの状態取得コマンド(`ijdasstat`コマンド)の復帰に時間がかかり、かつ、結果として「unknown」が表示される場合があります。このような場合、しばらく経ってからコマンドを再度実行してください。

なお、高負荷状態のInterstage Java EE DASサービスを停止したい場合は、[Interstage Java EE DASサービスのプロセス回収手順](#)を参照してプロセスを回収してください。

## Interstage Java EE DASサービスのプロセス回収手順

以下の手順により、Interstage Java EE DASサービスのプロセスを回収してください。

1. 以下のファイルを参照して、Interstage Java EE DASサービスのプロセスIDを確認してください。

**Windows32/64**

```
[Java EE共通ディレクトリ]¥domains¥interstage¥config¥.__com_sun_appservlet_pid
```

**Solaris32/64** **Linux32/64**

```
[Java EE共通ディレクトリ]/domains/interstage/config/.__com_sun_appservlet_pid
```

(注) ファイルが存在していない場合、またはjavaプロセスが存在しない場合、プロセスは存在していません。

2. 1で確認したIDのプロセスがjavaプロセスとして存在しているか、以下の方法で確認してください。

**Windows32/64**

Windowsのタスクマネージャで確認してください。

**Solaris32/64** **Linux32/64**

psコマンドを使用して確認してください。

3. 1で確認したプロセスIDに対して、以下の方法でプロセスを回収してください。

**Windows32/64**

Windowsのタスクマネージャでプロセスを終了させてください。

**Solaris32/64 Linux32/64**

killコマンドを使用してプロセスを削除してください。

## Interstage Java EE DASサービスに関するタイムアウト値のチューニング

下記のファイルを編集してください。

- [Java EE共通ディレクトリ]¥domains¥interstage¥config¥domain.xml

下記のname属性とvalue属性を定義した、system-property要素を追加してください。

```
(省略)
<domain application-root=" . . .
  :
(省略)
  :
  <system-property name="タイムアウト名" value="タイムアウト値"/> ★追加
  <property name="administrative.domain.name" value="interstage"/>
</domain>
```

指定可能なタイムアウト名は下記です。

### DAS\_STARTING\_TIMEOUT

起動処理で、状態がstartingに変化する時間  
指定可能範囲は180～65535[秒]、デフォルト180です。

### DAS\_STARTED\_TIMEOUT

起動処理で、状態がrunningに変化する時間  
指定可能範囲は1200～65535[秒]、デフォルト1200です。

### DAS\_STOPPED\_TIMEOUT

停止処理で、状態がnot runningに変化する時間  
指定可能範囲は60～65535[秒]、デフォルト60です。

## Interstage Java EE DASサービスでのアプリケーション運用時の異常

Interstage Java EE DASサービスでアプリケーションを運用する際の異常については、「[10.2.3 アプリケーションの運用に関する異常](#)」を参照してください。

## 10.2.2 Java EEアプリケーションで文字化けが発生する場合の対処

文字化けの対処方法、および文字化けに影響するコード変換の設定について説明します。

### 文字化け状況

文字化けが発生した場合、原因を追求するために文字化けが起きた状況を確認します。

- 文字化けする文字と文字化けしない文字の区別をする。
- 文字化けが発生している箇所を特定する。
- 文字化けが発生している箇所の文字を16進で表示する。

## チェックポイント

Java EEアプリケーションの運用で文字化けが発生した場合のチェックポイントを以下に示します。

- **ロケールに必要なパッケージをインストールしていますか。**  
ロケールに必要なパッケージをインストールしていない場合は、インストールしてください。
- **ロケールの設定は正しいですか。**  
ロケールを正しく設定していない場合は、ロケールを正しく設定してください。  
ロケールの設定については、「[ロケールの設定](#)」を参照してください。
- **文字化けしている文字に特殊文字「～」が含まれていませんか。**  
日本語コード(JIS、EUC、シフトJIS)をUnicodeに変換する際の変換規則が異なるために文字化けが発生している可能性があります。コード系は統一してください。統一できない場合は、Java EEアプリケーションでUnicodeから変換する直前にフィルターをかけ対処してください。  
その他、Unicodeに変換する際の変換規則が違う文字については「[文字化け特殊文字](#)」を参照してください。
- **「JISAutoDetect」を使用していませんか。**  
「JISAutoDetect」(自動変換)は、短い日本語文字列などはコード系の認識が正しく行われずコード変換が正しく行われないことがあります。例えば、日本語EUCの文字をShift-JISの半角カタカナだと誤認識した場合、半角カタカナを含む判読不可能な文字列と認識されます。  
必要がない限り「JISAutoDetect」を使用しないでください。使用する場合は上記の問題が含まれることを認識した上で使用してください。
- **「Shift\_JIS」を使用していませんか。**  
JDK/JRE 1.4.1からエンコーディング名「Shift\_JIS」の扱いが変更されました。このため、JDK/JRE 1.4.0以前のバージョンで動作していたJava EEアプリケーションを動作させると文字化けが発生することがあります。  
JDK/JRE 1.3と同様に動作させるには、「windows-31j」を指定してください。
- **コード変換が必要な機能の設定は正しいですか。**  
データを入力/出力する際に正しくコード変換されない場合は、コード変換の設定を見直し、正しく設定してください。  
Webアプリケーションの場合、「使用上の注意」-「Webアプリケーションの注意事項」の「文字コードについて」もあわせて参照してください。

## ロケールの設定

Java VM、javac、native2asciiなどのツールは、ロケールの設定に基づいて動作します。Java EEアプリケーションも同様にロケールの設定に基づいて動作します。

注) ロケール(Locale)は、共有する言語や文化に依存する様々な属性をひとまとめにして定義しています。

ロケールの設定方法について以下に説明します。

### Windows32/64

[コントロールパネル]の[地域と言語のオプション]でロケールを設定します。

注) Windows(R) 2003の場合の説明です。使用するOSにより操作方法は異なる場合があります。

### Solaris32/64 Linux32/64

環境変数LANGにロケールを設定します。

コード系	ロケール
EUC	Ja ja_JP.eucJP
シフトJIS	ja.JP.PCK
UTF-8	ja_JP.UTF-8

## 文字化け特殊文字

以下に、Unicodeに変換する際の変換規則が違う文字の対応表を示します。

JIS X 0208		Unicodeへの変換	
33 ('~')	EUC :a1c1	Solaris上での変換	U+301c (WAVE DASH)
	SJIS:8160	Windows(R)上での変換	U+ff5e (FULLWIDTH TILDE)
01-34 ('//')	EUC :a1c2	Solaris上での変換	U+2016 (DOUBLE VERTICAL LINE)
	SJIS:8161	Windows(R)上での変換	U+2225 (PARALLEL TO)
01-61 ('-')	EUC :a1dd	Solaris上での変換	U+2212 (MINUS SIGN)
	SJIS:817c	Windows(R)上での変換	U+ff0d (FULLWIDTH HYPHEN-MINUS)
01-81 ('¢')	EUC :a1f1	Solaris上での変換	U+00a2 (CENT SIGN)
	SJIS:8191	Windows(R)上での変換	U+ffe0 (FULLWIDTH CENT SIGN)
01-82 ('£')	EUC :a1f2	Solaris上での変換	U+00a3 (POUND SIGN)
	SJIS:8192	Windows(R)上での変換	U+ffe1 (FULLWIDTH POUND SIGN)
02-44 ('¬')	EUC :a2cc	Solaris上での変換	U+00ac (NOT SIGN)
	SJIS:81ca	Windows(R)上での変換	U+ffe2 (FULLWIDTH NOT SIGN)

## 10.2.3 アプリケーションの運用に関する異常

以下のトラブルについて対処方法を説明します。

- IJServerクラスタ起動時、および運用時にアプリケーションでエラーが発生した場合
- アプリケーションの再配備時、または配備解除時にAPPDEP1010が出力された場合
- 運用時にアプリケーションの応答がない場合、遅延が発生した場合
- アプリケーション内のクラスファイルやリソースファイルの参照に失敗する場合
- アプリケーションの再配備時にDEP4112が出力された場合
- アプリケーションの運用時にWEB1502が出力された場合

### IJServerクラスタ起動時、および運用時にアプリケーションでエラーが発生した場合

IJServerクラスタ起動時、および運用時にアプリケーションでエラーが発生した場合、アプリケーションがJava EE規約に準拠していない可能性があります。

Java EE規約に準拠していることを確認する場合には、配備時の検証を有効にして再配備を行ってください。

### アプリケーションの再配備時、または配備解除時にAPPDEP1010が出力された場合

アプリケーションの再配備時、または配備解除時に、APPDEP1010の詳細メッセージに以下が出力された場合、IJServerクラスタの内部情報に不整合が発生している可能性があります。

- Failure removing policy context directory: ディレクトリ
- Failure removing policy file: ファイル

本問題が発生した場合は以下の手順でリカバリを行ってください。

1. アプリケーションを配備
2. アプリケーションを配備解除
3. 再配備の場合:アプリケーションを配備

### 運用時にアプリケーションの応答がない場合、遅延が発生した場合

Enterprise Beanごとのスレッドプール設定を行わず、デフォルトスレッドプールを使用している場合、EJBアプリケーションの呼出しが多くなると、応答がない現象や処理遅延が発生することがあります。「チューニングガイド」-「スレッドプーリング」を参照し、Enterprise Beanごとのスレッドプール設定を行ってください。

## アプリケーション内のクラスファイルやリソースファイルの参照に失敗する場合

- ・アプリケーションがロードされたクラスローダからクラスファイルやリソースファイルが参照できない可能性があります。「Java EE運用ガイド」-「クラスローダ」を参照し、アプリケーションがロードされたクラスローダからクラスファイルやリソースファイルが参照できるか確認してください。参照できない場合は、適切に設定してください。
- ・アプリケーションが大容量の場合、アプリケーション展開処理でメモリ不足が発生した可能性があります。「チューニングガイド」-「配備時のチューニング」を参照し、アプリケーション資産を展開するプロセスをチューニングしてください。

## アプリケーションの再配備時にDEP4112が出力された場合

再配備を行うと、配備済みアプリケーションの配備取消が行われたあと、アプリケーションの配備が行われます。

このため、「Webサーバコネクタの制限事項」の項番1の制限により、アプリケーションの配備に失敗する場合があります。

この場合、しばらく時間をおいてから再操作するか、Webサーバの停止後に再配備操作を行ってください。

制限事項の詳細は、「使用上の注意」の「Webサーバコネクタの制限事項」を参照してください。

## アプリケーションの運用時にWEB1502が出力された場合

下記条件に該当する場合、Webコンテナでリクエストの受付処理を実行するタイミングで、クライアント起因の理由により接続が切断された可能性があります。リクエストの受付時にエラーとなりますが、アプリケーションの処理は実行されておらず運用上の影響はありませんので、WEB1502のメッセージは無視してください。

### <条件>

- ・ Webサーバを経由しない運用(注1)。かつ、
- ・ リクエスト受付中のタイミングでクライアント側の要因で切断(注2)する。

**注1)** Webサーバを経由する環境でも発生する可能性はありますが、クライアントで切断した場合に、WebサーバやWebサーバコネクタで検知してエラーを返す可能性があるため、Webサーバを経由しない運用の場合に顕著化します。

**注2)** ブラウザをクローズしたり、リロードを連打した場合など。

## 10.2.4 Dependency Injection失敗時の異常

Dependency Injectionに失敗すると以下のような例外が発生します。この場合には出力される情報と、例外のスタックトレースを参照して対処します。スタックトレースの[Caused by: ]に出力されているエラーが根本原因となる例外です。

以下の例外情報から異常が発生したオブジェクトを特定し、スタックトレースの[Caused by: ]に出力されている例外情報で原因を特定して対処してください。根本原因の例外情報については「メッセージ集」-「Java EE使用時に出力される例外情報」を参照して対処してください。

- ・ @EJBのDependency Injectionに失敗した場合の例外
- ・ @ResourceのDependency Injectionに失敗した場合の例外
- ・ @WebServiceRefのDependency Injectionに失敗した場合の例外

### @EJBのDependency Injectionに失敗した場合の例外

**com.sun.enterprise.InjectionException: Exception attempting to inject Resolved Ejb-Ref {0}/{1}@jndi: {2} -> {3} into class {0}**

[可変情報]

{0}:クラス名。オブジェクトを注入する対象のクラス名です。

{1}:フィールド名。オブジェクトを注入するフィールド名です。

{2}:JNDI名。注入するオブジェクトのJNDI名です。

{3}:Enterprise Bean名。注入するオブジェクトのEnterprise Bean名です。

[出力例]

```
com.sun.enterprise.InjectionException: Exception attempting to inject Resolved Ejb-Ref
com.mycompany.jndi CallerClass/remote@jndi: CallerEJB -> CallerBean into class com.mycompany.jndi CallerClass
```

## @ResourceのDependency Injectionに失敗した場合の例外

**com.sun.enterprise.InjectionException: Exception attempting to inject Resolved Res-Ref-Env-Property {0}/ {1}@{2}@{3}**

[可変情報]

{0}:クラス名。オブジェクトを注入する対象のクラス名です。  
{1}:フィールド名。オブジェクトを注入するフィールド名です。  
{2}:フィールド型名。オブジェクトを注入するフィールドの型名です。  
{3}:その他の属性情報。

[出力例]

```
com.sun.enterprise.InjectionException: Exception attempting to inject Res-Ref-Env-Property:
com.mycompany.jndi CallerClass/ref@javax.sql.DataSource@ resolved as: jndi: jndiName@res principal: null@mail: null
```

## @WebServiceRefのDependency Injectionに失敗した場合の例外

**com.sun.enterprise.InjectionException?: Exception attempting to inject Env-Prop: {0}/{1}@Field-Injectable Resource. Class name = {0} Field name={1}@javax.jws.WebServiceRef@@@ into class {0}**

[可変情報]

{0}:クラス名。オブジェクトを注入する対象のクラス名です。  
{1}:フィールド名。オブジェクトを注入するフィールド名です。

[出力例]

```
com.sun.enterprise.InjectionException?: Exception attempting to inject Env-Prop: client.JAXWSClient/service@Field-Injectable Resource. Class name = client.JAXWSClient Field name=service@javax.jws.WebServiceRef@@@ into class client.JAXWSClient
```

## 10.2.5 Windows Server(R) 2003でnonpaged poolがリークする

Windows Server(R) 2003でWebアプリケーションを運用している場合、リクエスト処理に伴い、IIS Serverクラスタのプロセスでnonpaged pool領域のリークが発生することがあります。

### 現象

nonpaged pool領域のリークサイズが増大することで、以下の現象となる場合があります。

- ・ サーバログに次の例外が出力され、リクエストが処理できない/操作できないなど、IIS ServerクラスタやInterstage管理コンソールなどの動作が不安定となります。

```
java.net.SocketException: No buffer space available (maximum connections reached?): connect
```

- ・ イベントログに以下のメッセージが出力され、ネットワークを使用する処理やファイル操作が不可能となるなど、OSおよびOS上で動作するアプリケーションの動作が不安定となります。  
イベントID: 2019

```
The server was unable to allocate from the system nonpaged pool because the pool was empty.
プールが空であるため、サーバはシステムの非ページプールから割り当てることができませんでした。
```

### 原因

OSまたはNIC(Network Interface Card)のドライバの問題と考えられます。

### 復旧方法

マシンを再起動してください。

### 対処/予防方法

OSの修正や最新のドライバが提供されている場合、最新の修正の適用やドライバの更新により問題が解決する可能性があります。

問題が解決しない場合は、以下の設定により運用での回避を行ってください。

- ・「ブロック化」を有効にします
- ・上記とあわせて、次のどちらかの設定とします。
  - － 以下の大小関係を満たすようにします。  
WebサーバコネクタのWebコンテナへの最大接続数(注) < Webコンテナの同時処理数  
注) IJServerクラスタが複数のWebサーバと連携している場合、各WebサーバのWebサーバコネクタの、Webコンテナへの最大接続数の設定値の合計と比較します。
  - － WebサーバコネクターWebコンテナ間のKeepAliveを行わない設定とします。

設定については以下を参照してください。

また、「チューニングガイド」-「Webコンテナのチューニング」も参考にしてください。

- ・ Interstage Java EE管理コンソールヘルプ
- ・ 「リファレンスマニュアル(コマンド編)」-「Java EE運用コマンド」の「asadmin」および下記「getサブコマンド」/「setサブコマンド」/「listサブコマンド」、「configs.config.http-serviceの定義項目」

## 10.2.6 大規模ファイル使用時の異常

大規模ファイルを使用した場合に、以下のメッセージが出力され処理に失敗することがあります。

```
ERROR: ISJEE_APPDEP3010: file open failure: file = {0}: detail = {1}
```

上記のメッセージが出力された場合は、ファイルを分割して再度処理を行ってください。

## 10.2.7 SSL通信ができない

SSL通信ができない場合に考えられる原因とその対処を以下に説明します。

- ・ 証明書環境に誤りがあります  
証明書が正しくインストールされているか確認してください。
- ・ 使用するキーストアの設定に誤りがあります  
キーストアが正しく設定されているか確認してください。  
キーストアの設定方法の詳細は、「Java EE運用ガイド」-「SSL」を参照してください。
- ・ サイト証明書のニックネーム名に誤りがあります  
サイト証明書のニックネーム名が正しいか確認してください。
- ・ サポートされていない暗号化方法を使用しています  
Webサーバコネクタ、Webコンテナ間をSSLで通信する場合に使用可能な暗号化方法には制限があります。暗号化方法が正しいか確認してください。  
使用可能な暗号化方法の詳細は、「Java EE運用ガイド」-「SSL」を参照してください。

## 10.2.8 Interstageディレクトリサービスへ接続できない

admin-realmにLDAPを使用時、サーバ障害やネットワークなどの問題でInterstageディレクトリサービスに接続できなくなると、Interstage Java EE DASサービスやInterstage Java EE Node Agentサービスの操作や、Interstage Java EE管理コンソールおよびasadminコマンドでのログイン認証ができなくなります。

その場合、以下によりadmin-realmによる認証方法をfileレルムへ切り替えてください。

- ・ Interstage Java EE DAS サービス、Interstage Java EE Node AgentサービスおよびIJServerクラスタを停止する。
- ・ 「10.2.9 認証に失敗する旨のメッセージが出力される場合」の「Interstage Java EE DASサービスの再起動に失敗する場合」の【fileレルム使用時】の記述に従って対処する。
- ・ asadmin change-service-adminuser サブコマンドにより、fileレルムに登録されている管理ユーザーをサービス管理ユーザーとして設定する。
- ・ Interstage Java EE DASサービスを起動する。

- ・「Java EE運用ガイド」-「レルムの種類の変更方法」を参照して、各サービスおよびIIServerクラスタについて、fileレルムを使用するように設定する。
- ・ Interstage Java EE Node AgentサービスおよびIIServerクラスタを起動する。

## 10.2.9 認証に失敗する旨のメッセージが出力される場合

以下に該当する場合、出力されたメッセージの内容にしたがって対処を行ってください。

- ・ Interstage Java EE管理コンソールのログイン認証に失敗した
- ・ asadminコマンドのログイン認証に失敗、ユーザー名やパスワードに誤りがある、または、権限がない旨のメッセージが出力された
- ・ Interstage Java EE DASサービス/Interstage Java EE Node Agentサービス/IIServerクラスタ/サーバーインスタンスでの異常発生時、ログに認証に失敗、ユーザー名やパスワードに誤りがある、または、権限がない旨のメッセージが出力された

また、以下のマニュアルもあわせて参照してください。

- ・「Java EE運用ガイド」-「admin-realmおよび管理ユーザーの情報更新時の注意」
- ・「使用上の注意」-「Solarisでrootを使用する場合の注意事項」

上記で問題が解決しない場合、以下の可能性があります。

- ・ Interstageディレクトリサービスが起動していない  
この場合はInterstageディレクトリサービスを起動してください。
- ・ 管理ユーザーの情報を更新(ユーザーの追加/削除、パスワードやグループの更新など)を行ったが、認証処理に反映されていない
- ・ admin-realmの設定を変更したが、認証処理に反映されていない

該当する操作を行った場合は、Interstage Java EE DASサービス/Interstage Java EE Node Agentサービスを再起動することで変更が反映され、問題が解決することがあります。

admin-realmの設定を変更後、Interstage Java EE DASサービス/Interstage Java EE Node Agentサービス/IIServerクラスタの再起動に失敗する場合は変更後の設定に誤りがある可能性があります。

以下の対処を行ってください。

- ・ Interstage Java EE Node Agentサービス/IIServerクラスタの再起動に失敗する場合  
asadminコマンドまたはInterstage Java EE管理コンソールでadmin-realmの設定を修正後再起動してください。
- ・ Interstage Java EE DASサービスの再起動に失敗する場合  
domain.xml(※)の「name="server-config"」のあるconfigタグ配下の「name="admin-realm"」のあるauth-realmタグについて、下記太字部分を修正後、再起動してください。  
admin-realmの設定については、「Java EE運用ガイド」-「レルムの種類の変更方法」を参照してください。

【OSレルム使用时】

```
<config dynamic-reconfiguration-enabled="true" name="server-config">
....
<security-service activate-default-principal-to-role-mapping="false" anonymous-role="AttributeDeprecated" audit-
enabled="true" audit-modules="default" default-realm="admin-realm" jacc="default">
<!-- All administrative users use this realm by default. -->
<auth-realm classname="com.fujitsu.interstage.javaee.security.auth.realm.os.OSRealm" name="admin-
realm">
<property name="jaas-context" value="osRealm"/>
<property name="assign-groups" value="asadmin"/>
</auth-realm>
....
```

### 【ldapレールム使用時】

```
<config dynamic-reconfiguration-enabled="true" name="server-config">
....
<security-service activate-default-principal-to-role-mapping="false" anonymous-role="AttributeDeprecated" audit-
enabled="true" audit-modules="default" default-realm="admin-realm" jacc="default">
<!-- All administrative users use this realm by default. -->
<auth-realm classname="com.sun.enterprise.security.auth.realm.ldap.LDAPRealm" name="admin-
realm">
<property name="jaas-context" value="ldapRealm"/>
<property name="base-dn" value="ou=interstage,o=fujitsu,dc=com"/>
<property name="directory" value="ldap://localhost:389"/>
<property name="group-search-filter" value="member=%d"/>
....
</auth-realm>
....
```

プロパティdirectoryおよびbase-dnは環境に応じて値を変更してください。

また、group-search-filter、search-filter、group-base-dn、assign-groups、group-targetについても必要に応じて設定してください。

### 【fileレールム使用時】

```
<config dynamic-reconfiguration-enabled="true" name="server-config">
....
<security-service activate-default-principal-to-role-mapping="false" anonymous-role="AttributeDeprecated" audit-
enabled="true" audit-modules="default" default-realm="admin-realm" jacc="default">
<!-- All administrative users use this realm by default. -->
<auth-realm classname="com.sun.enterprise.security.auth.realm.file.FileRealm" name="admin-realm">
<property name="jaas-context" value="fileRealm"/>
<property name="file" value="{com.sun.aas.instanceRoot}/config/admin-keyfile"/>
<property name="assign-groups" value="asadmin"/>
</auth-realm>
....
```

プロパティassign-groupsは必要であれば設定します。

他の値は上記のとおりを設定します。

(※)編集対象のdomain.xmlの格納ディレクトリ

**Windows32/64**

[Java EE共通ディレクトリ]¥domains¥interstage¥config

**Solaris32/64 Linux32/64**

[Java EE共通ディレクトリ]/domains/interstage/config

## 10.2.10 追加プロパティに関する異常

Interstage Java EE管理コンソール/asadminコマンドを使用してリソースなどを作成する際に、誤った追加プロパティを指定して作成を実行した場合、作成処理が異常復帰しますが、該当の資産が作成されている場合があります。

このような場合、該当の資産は作成処理が正しく完了していないため、正常に動作することができません。該当する資産を削除する操作を実行することで環境を復旧してください。

## Interstage Java EE管理コンソールの場合

### 例1: JMS接続ファクトリの作成で現象が発生した場合

「リソース > JMSリソース > 接続ファクトリ」画面で該当の資産を削除してください。

### 例2: JavaMailセッションの作成で現象が発生した場合

「リソース > JavaMailセッション」画面で該当の資産を削除してください。

## asadminコマンドの場合

### 例3: create-custom-resourceサブコマンドで現象が発生した場合

delete-custom-resourceサブコマンドで該当の資産を削除してください。

### 例4: create-connector-connection-poolサブコマンドで現象が発生した場合

delete-connector-connection-poolサブコマンドで該当の資産を削除してください。

## 10.2.11 jersey利用時の異常

Java EEアプリケーションにオープンソースソフトウェアのjerseyを含めたり、クラスパスに設定して利用する場合、アプリケーションの実行時に予期しない例外が発生する場合があります。

このような場合、以下の手順を行うことで、現象を回避することができます。

1. 以下のサービスが起動している場合、停止します。
  - Interstage Java EE DASサービス
  - Interstage Java EE Node Agentサービス
2. ファイル退避用のディレクトリを作成します。



例

Windows32/64

C:¥Backup

Solaris32/64 Linux32/64

/backup

3. 作成した退避用ディレクトリに、Interstageインストール先の以下に示すファイルを退避します。

退避元ファイルの格納ディレクトリ

Windows32/64

C:¥Interstage¥F3FMisjee¥lib (C:¥Interstageにインストールした場合)

Solaris32/64 Linux32/64

/opt/FJSVisjee/lib

退避するファイル

- jersey-bundle-1.0.3.1.jar
- jsr311-api-1.0.jar

- jettison-1.0.1.jar
  - jackson-asl-0.9.4.jar
4. Interstage Java EE DASサービスを起動します。
  5. Interstage Java EE Node Agentサービスを起動します。

## 注意

上記の手順を行った後にJava EEのパッチを適用した場合は、上記の手順を再度実施する必要があります。

## 10.3 Webアプリケーションの開発・運用時の異常

Webアプリケーションの開発・運用時に発生する以下のトラブルについて対処方法を説明します。

- JSPのコンパイルに失敗した場合
- JSPまたはJSPから静的includeされているファイルを更新しても実行時に反映されない
- エラーページとして指定したページがWebブラウザに表示されない場合
- Webブラウザにステータスコード「404 Not Found」が通知される
- WebサーバコネクタのログにメッセージIJSERVER12027が出力されている場合
- Interstage Java EE DASサービスに配備したWebアプリケーションとの通信に失敗する場合
- Webブラウザにステータスコード「400 Bad Request」が通知される
- Webブラウザにステータスコード「500 Internal Server Error」が通知される
- 故障監視機能を使用しているが、故障が検出されない場合がある
- svmondspstatコマンドで表示されない振り分け先がある

### JSPのコンパイルに失敗した場合

以下の可能性が考えられます。

- **JSPに誤りがある**  
使用できるのはJSP 2.1に準拠したJSPです。  
JSPの文法に誤りがないか確認してください。  
JSP 2.1の文法についてはJSP 2.1の仕様を参照してください。
- **JSPの文字コードを正しく指定していない**  
JSPファイルの文字コードを正しく指定していないためにJSPをコンパイルする際に文字化けが発生している可能性があります。  
JSPの文字コードを正しく指定してください。  
文字コードの指定方法についてはJSP 2.1の仕様を参照してください。
- **JSPから使用しているクラスが存在しない**  
JSPから使用するクラスは同じアプリケーション内に含めるかIJSERVERクラススタに設定する必要があります。  
IJSERVERクラススタの設定については、「Java EE運用ガイド」-「クラスローダ」の「IJSERVERクラススタで使用するクラスの設定について」を参照してください。
- **無名パッケージのクラスを使用している**  
JSPからは無名パッケージのクラスを使用することはできません。  
JSPから使用するクラスにはパッケージ名を指定してください。

- **JSPファイルのサイズが上限値を超えている**

JSPファイルのサイズが大きすぎます。  
以下の対処を行ってください。

- タグ(アクションやスクリプティング要素)の間の不要な記載(スペース、コメント、改行など)がある場合は削除してください。
- JSPファイルを分割し、動的にincludeを行ってください。

- **JREで運用を行っている**

Java Runtime Environment(JRE)環境ではJSPのコンパイルはできません。

JREで運用する場合については、「Java EE運用ガイド」-「Webコンテナの提供機能」の「JSP事前コンパイル」を参照してください。

## **JSPまたはJSPから静的includeされているファイルを更新しても実行時に反映されない**

以下の可能性があります。

- JSPコンパイル中にJSPまたはJSPから静的includeされているファイルの更新を行った。
- JSPまたはJSPから静的includeされているファイルの置き換えを行ったが、ファイルの日付が前回JSPのコンパイルを行った日付よりも過去であった。

以下の対処を行ってください。

- JSPまたはJSPから静的includeされているファイルの日付をマシンの現在時刻に更新してください。

## **エラーページとして指定したページがWebブラウザに表示されない場合**

「使用上の注意」-「Webアプリケーションの注意事項」の「エラーページについて」の「Webブラウザの設定」を参照してください。

## **Webブラウザにステータスコード「404 Not Found」が通知される**

IIServerクラスタ資源をリストア・移入した場合、isprintbackuppscコマンドで表示されたバックアップ対象サービスの資源もリストア・移入しているか確認してください。

## **WebサーバコネクタのログにメッセージIIServer12027が出力されている場合**

メッセージの説明に記載されている対処を行っても解決しない場合、以下の可能性があります。

- クライアントからのリクエストが、Webコンテナの同時処理数を超えている。かつ
- Webアプリケーションの処理に時間がかかっており、レスポンスが遅延している。

以下の対処を行ってください。

- 「Webコンテナのチューニング」の「同時処理数」を増やす。
- 「HTTPサービスの定義項目」の「ブロック化」を有効にする。

## **Interstage Java EE DASサービスに配備したWebアプリケーションとの通信に失敗する場合**

以下の可能性があります。

- 配備したWebアプリケーションに対して大量のリクエストを送信している。

以下の対処をおこなってください。

- Interstage Java EE DASサービスに配備したWebアプリケーションの実行は、開発環境でのみ実施してください。大量のリクエストを送信する場合、運用環境に構築したIIServerクラスタ上で実施してください。

## **Webブラウザにステータスコード「400 Bad Request」が通知される**

Webコンテナのへアクセスする運用形態と「HTTPリスナー」のプロパティ「requestCheck」の設定値が合っているか確認してください。

詳細は、「Java EE運用ガイド」-「HTTPサービスの定義項目」を参照してください。

## Webブラウザにステータスコード「500 Internal Server Error」が通知される

以下を参照してください。

「メッセージ集」の「Webサーバから通知されるステータスコード」-「Server Error (500番台のステータスコード)」-「500」

## 故障監視機能を使用しているが、故障が検出されない場合がある

WebサーバコネクタとWebサーバコネクタ(Interstage HTTP Server 2.2用)で、同一のWebサーバ名を使用している可能性があります。

故障監視を行っているWebサーバ名が、WebサーバコネクタとWebサーバコネクタ(Interstage HTTP Server 2.2用)の両方で使用されている場合、正しく故障監視を行うことができません。

以下の方法で、それぞれのWebサーバ名を確認してください。

- Webサーバコネクタの場合

[Interstage管理コンソール] > [Interstage Application Server] > [システム] > [サービス] > [Webサーバ]

- Webサーバコネクタ(Interstage HTTP Server 2.2用)の場合

wscadmin list-web-serversサブコマンド

list-web-serversサブコマンドについては、「Java EE運用ガイド(Java EE 6編)」-「Java EE 6運用コマンド」-「wscadmin」-「list-web-serversサブコマンド」を参照してください。

故障監視を行っているWebサーバ名が、WebサーバコネクタとWebサーバコネクタ(Interstage HTTP Server 2.2用)の両方で使用されている場合、Webサーバコネクタ(Interstage HTTP Server 2.2用)で使用するWebサーバ名を変更してください。

1. Webサーバ名を変更するWebサーバを停止します。

Webサーバの停止方法については、「Interstage HTTP Server 2.2 運用ガイド」の「運用・保守」-「起動・停止」-「停止」を参照してください。

2. wscadmin list-web-serversサブコマンドの--detailオプションでWebサーバコネクタの設定を出力し、記録します。  
list-web-serversサブコマンドについては、「Java EE運用ガイド(Java EE 6編)」-「Java EE 6運用コマンド」-「wscadmin」-「list-web-serversサブコマンド」を参照してください。

3. wscadmin delete-envサブコマンドでWebサーバコネクタの動作環境を削除します。

delete-envサブコマンドについては、「Java EE運用ガイド(Java EE 6編)」-「Java EE 6運用コマンド」-「wscadmin」-「delete-envサブコマンド」を参照してください。

4. テキストエディタなどでWebサーバ一覧ファイル(servers.conf)を開いて、変更対象のWebサーバ名を新しいWebサーバ名に変更します。

Webサーバ一覧ファイル(servers.conf)の設定方法については、「Interstage HTTP Server 2.2 運用ガイド」の「チューニング」-「環境定義」-「Webサーバ一覧ファイル(servers.conf)」を参照してください。

5. wscadmin create-envサブコマンドで、新しいWebサーバ名を指定し、Webサーバコネクタの動作環境を作成します。  
create-envサブコマンドについては、「Java EE運用ガイド(Java EE 6編)」-「Java EE 6運用コマンド」-「wscadmin」-「create-envサブコマンド」を参照してください。

6. asadmin、またはwscadminコマンドを使用し、2.で記録したWebサーバコネクタの設定と同様になるように、設定を行ってください。

asadmin、wscadminコマンドについては、「Java EE運用ガイド(Java EE 6編)」-「Java EE 6運用コマンド」を参照してください。

7. Webサーバ名を変更したWebサーバを起動します。

Webサーバの起動方法については、「Interstage HTTP Server 2.2 運用ガイド」の「運用・保守」-「起動・停止」-「起動」を参照してください。

## svmondspstatコマンドで表示されない振り分け先がある

WebサーバコネクタとWebサーバコネクタ(Interstage HTTP Server 2.2用)で、同一のWebサーバ名を使用している可能性があります。

「故障監視機能を使用しているが、故障が検出されない場合がある」を参照してください。

## 10.4 EJBアプリケーションの開発・運用時の異常

---

EJBアプリケーションの開発・運用時に発生する以下のトラブルについて対処方法を説明します。

- ・ @Statelessアノテーションを定義しているBeanが、運用時にStateful SessionBeanとして動作する場合

### @Statelessアノテーションを定義しているBeanが、運用時にStateful SessionBeanとして動作する場合

以下の条件に該当している可能性があります。

1. @Statelessアノテーションを定義している、かつ
2. 1のBeanをdeployment descriptor(ejb-jar.xml)に定義している、かつ
3. 2の定義に<session-type>タグが定義されていない

以下の対処を行ってください。

deployment descriptorに<session-type>タグを定義し、「Stateless」を指定してください。

## 10.5 EJBディスパッチ機能の異常

---

### EJBディスパッチ機能によるロードバランス利用時にIIOP通信要求がロードバランスされない

以下のような原因が考えられます。

[原因]

- ・@EJBアノテーションを利用したオブジェクトの注入を行っている。
- ・Enterprise JavaBeansのBeanクラスのコンストラクタ・ライフサイクルモジュールの起動処理・サーブレットの初期化(initメソッド)などの初期化処理においてJNDIのlookupメソッドを実行してオブジェクトリファレンスを取得し、各処理でオブジェクトリファレンスを使いまわしている。
- ・Enterprise JavaBeansの呼び出しで異常が発生してcatch節などの例外処理において処理をリトライする際にオブジェクトリファレンスを使いまわしている。

[対処]

- ・@EJBアノテーションを利用している場合、lookupメソッドを使用するようにアプリケーションを変更してロードバランスが必要なタイミングでlookupメソッドを再実行してください。
- ・オブジェクトリファレンスを使いまわしている場合、アプリケーションを変更してロードバランスが必要なタイミングでHome/Remoteインタフェースを利用する場合にはcreateメソッド、ビジネスインタフェースを利用する場合にはlookupメソッドから再実行してください。

## 10.6 Interstage Java EE管理コンソール操作中の異常

---

### 10.6.1 Interstage Java EE管理コンソール起動時の異常

---

Interstage Java EE管理コンソールの起動時に、以下のエラーメッセージが出力される

- Internet Explorerではこのページは表示できません
- Internal Server Error

Interstage Java EE管理コンソールを使用するために必要なサービスが起動されていない場合、[Interstage Java EE管理コンソール]ノードを選択した際にエラーとなります。

#### Windows32/64

Windows(R)のサービスから、Interstage Java EE DASサービスが起動しているかを確認してください。

起動していない場合は、Interstage Java EE DASサービスを起動してください。

#### Solaris32/64 Linux32/64

Interstage Java EE DASサービスが起動しているかを確認してください。

起動していない場合は、以下のコマンドを実行し、Interstage Java EE DASサービスを起動してください。

```
# /opt/FJSVisjee/bin/ijdasstart
```



#### 注意

本現象は、ウィンドウ全体ではなく、フレーム単位に発生することがあります。

### スタートメニューまたはショートカットからInterstage Java EE管理コンソールを起動すると、Interstage Java EE管理コンソールの起動に失敗する

本現象が発生した場合は、Interstage Java EE管理コンソールが使用していたWebブラウザのプロセスが残存している可能性があります。以下のいずれかの対処を行うか、または以下の回避方法により、Interstage Java EE管理コンソールを起動してください。

#### ■対処方法

- ・ 本現象が発生したシステムを再起動します。
- ・ 以下の手順で、残存しているWebブラウザのプロセスを終了します。
  1. Windows(R)の「タスクマネージャ」を起動します。
  2. [プロセス]タブをクリックします。
  3. イメージ名「iexplore.exe」をクリックします。
  4. [プロセスの終了]をクリックします。



#### 注意

Webブラウザ以外のプロセスを終了させることがないように、十分に注意してください。

#### ■回避方法

以下の手順で、Interstage Java EE管理コンソールを起動します。

##### Internet Explorer 8/9/10の場合

1. Webブラウザを起動します。
2. 1.で起動したWebブラウザの[ファイル]メニューの[新規セッション]をクリックし、新しくWebブラウザを起動します。
3. 2.で起動したWebブラウザで、Interstage Java EE管理コンソールのURLを指定します。

##### Internet Explorer 7の場合

1. Webブラウザを起動します。
2. Interstage Java EE管理コンソールのURLを指定します。

## 10.6.2 ログイン時の異常

### Interstage Java EE管理コンソールにログインできない、またはボタンを押下しても画面が遷移しない

本現象が発生した場合は、Interstage Java EE管理コンソールを動作させるために必要なWebブラウザの設定が無効になっている可能性があります。

以下の手順で、Interstage Java EE管理コンソールが動作しているゾーンのWebブラウザの設定を有効にしてください。

1. Interstage Java EE管理コンソールを表示した状態で、Webブラウザの[ツール]メニューの[インターネット オプション]で、[セキュリティ]タブをクリックします。  
この時点で選択されているゾーンを変更しないでください。
2. [レベルのカスタマイズ]をクリックします。
3. [スクリプト]セクションで、[アクティブスクリプト]を「有効にする」に変更します。
4. Webブラウザを再起動し、Interstage Java EE管理コンソールにログインします。

### Interstage Java EE管理コンソールにログインしたが、[コンソール切替]ボタンが表示されない

本機能を使用する場合は、Interstage管理コンソールがインストールされている必要があります。Interstage管理コンソールがインストールされていない場合は、インストールしてください。

Interstage管理コンソールがインストールされている場合は、Interstage管理コンソールにおいて、「Interstage Java EE管理コンソール」ツリーノードをクリックして起動したInterstage Java EE管理コンソール画面で操作している可能性があります。または、以前にInterstage管理コンソールを操作していたWebブラウザを使って、Interstage Java EE管理コンソール画面を操作している可能性があります。

#### ■対処方法

新しくWebブラウザを起動して、Interstage Java EE管理コンソールにログインしてください。Interstage Java EE管理コンソールの運用方法については、「Java EE運用ガイド」の「Java EEアプリケーションの運用」-「Interstage Java EE管理コンソールを利用した運用」-「Interstage Java EE管理コンソールの運用方法について」を参照してください。

### rootでInterstage Java EE管理コンソールにログインできない Solaris32/64

本現象が発生した場合は、Solarisでrootが役割として設定されている可能性があります。

以下のマニュアルを参照して、対処してください。

- ・「システム設計ガイド」-「動作OSの違いによる注意事項」-「Solaris 11で使用する場合の注意事項」
- ・「使用上の注意」-「Java EEの注意事項」-「Solarisでrootを使用する場合の注意事項」

## 10.6.3 画面の異常

### Interstage Java EE管理コンソールの[クラスタ] > [JServerクラスタ名] > [アプリケーション] > [配備]で、[OK]ボタンを押下後、配備画面が正しく表示されない場合がある

以下の可能性が考えられます。

- ・ **ディスク容量が不足している**  
Interstageをインストールしているサーバのテンポラリディレクトリのあるディスク容量が不足している可能性があります。不要なファイルを削除してディスクの空き容量を増やした後、再度処理を実行してください。  
配備時に必要なディスクの空き容量については、「チューニングガイド」-「運用時に必要なディスク容量」を参照してください。
- ・ **配備ファイルのサイズが2Gバイトを超えている**  
2Gバイトを超える配備ファイルの配備はできません。2Gバイト未満の配備ファイルを指定して配備を行ってください。

## Interstage Java EE管理コンソールの各配備/再配備画面で「The specified file exceeds the maximum allowable size of <最大サイズ> byte」が表示される

アップロードファイルの最大サイズを超えたファイルが指定されました。

アップロードファイルの最大サイズを変更する場合は、「チューニングガイド」-「アップロードファイルの最大サイズのチューニング」を参照してください。

## Interstage Java EE管理コンソールのヘルプ画面で、HTTP Status 500となる

Interstage Java EE管理コンソールが動作するサーバで、Interstage Java EE管理コンソールを表示するために指定したホスト名(またはIPアドレス)をアドレス解決できていない可能性があります。

以下のいずれかの対処を行ってください。

- サーバのhostsファイルを設定します。
- ネットワーク管理者に連絡して、ネットワーク環境を見直します。



例

### hostsファイルに追加する例

- Interstage Java EE管理コンソールへのURLが、https://host1:12001/javaee\_adminの場合

```
xxx.xxx.xxx.xxx      host1
```

- Interstage Java EE管理コンソールへのURLが、https://host1.domain.com:12001/javaee\_adminの場合

```
xxx.xxx.xxx.xxx      host1.domain.com
```

## Interstage Java EE管理コンソールの操作中に、以下のエラーメッセージが出力されて、そのあとの操作ができない

### ■Internet Explorerではこのページは表示できません

本現象が発生した場合は、Interstage Java EE管理コンソールを使用するために必要なサービスに問題が発生している可能性があります。

以下のサービスが起動しているかを確認してください。

#### Windows32/64

Windows(R)のサービスから、Interstage Java EE DASサービスが起動しているかを確認してください。

起動していない場合は、Interstage Java EE DASサービスを起動してください。

#### Solaris32/64 Linux32/64

Interstage Java EE DASサービスが起動しているかを確認してください。

起動していない場合は、以下のコマンドを実行し、Interstage Java EE DASサービスを起動してください。

```
# /opt/FJSVisjee/bin/ijdasstart
```

### ■HTTP Status 404

本現象が発生した場合は、Interstage Java EE管理コンソールを使用するために必要な資源に問題が発生している可能性があります。

以下の方法で資源を復旧してください。

#### Windows32/64

Windows(R)のサービスから、Interstage Java EE DASサービスを再起動してください。

#### Solaris32/64 Linux32/64

以下のコマンドを実行し、Interstage Java EE DASサービスを再起動してください。

```
# /opt/FJSVisjee/bin/ijdasstop
# /opt/FJSVisjee/bin/ijdasstart
```

## 注意

本現象は、ウィンドウ全体ではなく、フレーム単位に発生することがあります。

### Interstage Java EE管理コンソールの操作中に、無応答となる、またはブランクページが表示される

本現象が発生した場合は、以下の原因が考えられます。

- 画面の描画が完了する前に、次の操作を行った可能性があります。Interstage Java EE管理コンソールの操作は、画面の描画が完了してから次の操作を行う必要があります。
- サーバのシステム時間を大幅に戻した可能性があります。

以下の対処を行ってから操作を行ってください。

1. 現在作業中のWebブラウザを閉じます。
2. 新しいWebブラウザを起動します。
3. 再度Interstage Java EE管理コンソールにログインします。

### Interstage Java EE管理コンソールの操作中にセッションタイムアウト時間が経過する前にセッションタイムアウトが発生する

本現象が発生した場合は、サーバのシステム時間を大幅に進めた可能性があります。

再度Interstage Java EE管理コンソールにログインした後、操作を行ってください。セッションタイムアウトした旨のアラートが表示されますので、[OK]ボタンを押下してWebブラウザを閉じてください。

### Interstage Java EE管理コンソールから操作を行うと、画面が遷移しない

本現象が発生した場合は、Interstage Java EE管理コンソールを動作させるために必要なWebブラウザの設定が無効になっている可能性があります。

以下の手順で、Interstage Java EE管理コンソールが動作しているゾーンのWebブラウザの設定を有効にしてください。

1. Interstage Java EE管理コンソールを表示した状態で、Webブラウザの[ツール]メニューの[インターネット オプション]で、[セキュリティ]タブをクリックします。  
この時点で選択されているゾーンを変更しないでください。
2. [レベルのカスタマイズ]をクリックします。
3. [スクリプト]セクションで、[アクティブスクリプト]を「有効にする」に変更します。
4. Webブラウザを再起動します。
5. Interstage Java EE管理コンソールにログインします。

### ズーム機能を使用した場合、画面表示が崩れる

ズーム機能を使用した場合、画面表示が崩れる場合があります。画面表示が崩れる場合は、標準倍率で使用してください。

### Interstage Java EE管理コンソールの画面遷移および操作に異常が発生する

■Internet Explorer 8/9/10において、複数のWebブラウザから操作している場合

1つの端末から複数のInterstage Java EE管理コンソールを操作することはできません。本操作を行った場合、Interstage Java EE管理コンソールの画面の遷移、および操作に異常が発生する可能性があります。

1つの端末から複数のInterstage Java EE管理コンソールを操作する場合は、以下の対処を行ってください。

- Internet Explorerの[ファイル]メニューの[新規セッション]をクリックして、新しくWebブラウザを起動します。
- Internet Explorer 7を使用して、Interstage Java EE管理コンソールを操作します。

#### ■タブブラウズ機能を使用している場合

1つのWebブラウザ上で、タブブラウズ機能を使用して、複数のタブからInterstage Java EE管理コンソールを起動することはできません。本操作を行った場合、Interstage Java EE管理コンソールの画面の遷移、および操作に異常が発生する可能性があります。

Interstage Java EE管理コンソールを使用する場合は、複数のタブを使用しないでください。

また、複数のInterstage Java EE管理コンソールを起動する場合は、セッションが異なる複数のWebブラウザを起動してInterstage Java EE管理コンソールを操作してください。

#### ■タブブラウズ機能を使用していない場合

Internet Explorerにおいて、ショートカットの起動時にウィンドウを再使用する設定が無効になっている可能性があります。以下の手順で、ショートカットの起動時にウィンドウを再使用する設定を有効にしてください。

1. Internet Explorerの[メニュー]–[ツール]–[インターネットオプション]–[詳細設定]で、[ショートカットの起動時にウィンドウを再使用する(タブブラウズが無効である場合)]のチェックボックスが選択されているかを確認します。
2. 1.で、[ショートカットの起動時にウィンドウを再使用する(タブブラウズが無効である場合)]のチェックボックスが選択されていない場合は、チェックボックスを選択します。

#### Interstage Java EE管理コンソールの画面表示中に、画面が崩れて表示されることがある

本現象は、特に問題ありません。使用中のWebブラウザまたはオペレーティングシステムの状況により、画面レンダリングに時間がかかることがあります。この場合、表示中の画面が画面表示完了後の画面と異なったり、一時的に崩れたり、存在しない項目が表示されたりすることがあります。

#### Interstage Java EE管理コンソールのツリーノードの開閉ができなくなる

使用しているWebブラウザのインターネットゾーンにおけるセキュリティ設定で、[アクティブスクリプト]が「無効にする」になっている可能性があります。

以下の手順で回避してから操作を行ってください。

1. Internet Explorerの[ツール]メニューの[インターネット オプション]で[セキュリティ]タブをクリックします。
2. 「セキュリティ設定を表示または変更するゾーンを選択してください。」で「インターネット」を選択します。
3. [レベルのカスタマイズ]をクリックします。
4. [スクリプト]セクションで、[アクティブスクリプト]を「有効にする」に変更します。
5. Webブラウザを再起動します。
6. Interstage Java EE管理コンソールにログインします。

#### 注意

インターネットゾーンのアクティブスクリプトを有効にすると、不正なスクリプトを含むインターネットサイトへアクセスした場合、そのスクリプトが実行されるおそれがあります。

## 10.6.4 その他の異常

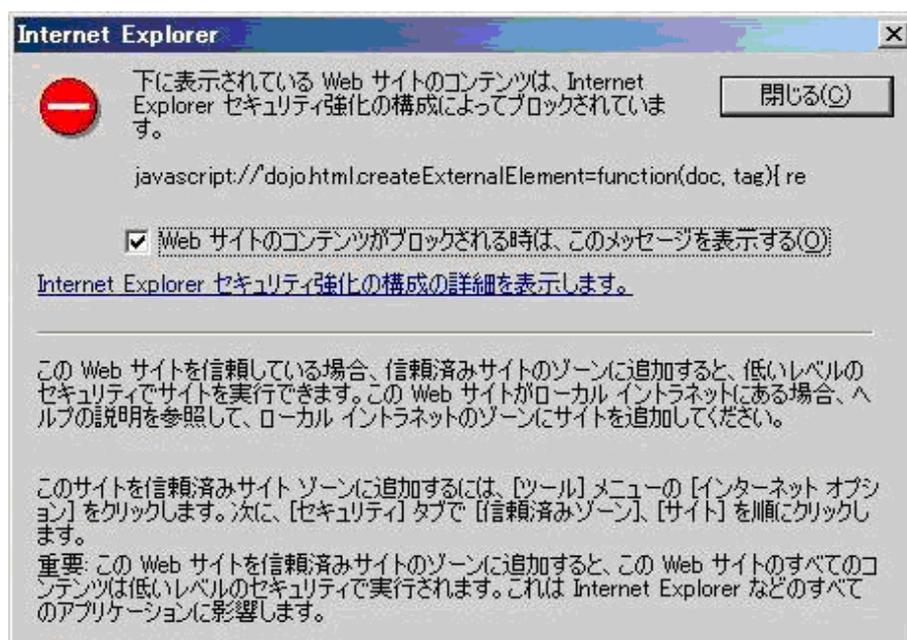
### 定義項目の更新時、意図しない値が保存される

他のユーザーが以下により同時に操作を行ったため、指定値が上書きされた可能性があります。

- Interstage Java EE管理コンソール
- asadminコマンド

同時に操作を行っているユーザーがいないかを確認し、必要に応じて操作を再度実行してください。

### Interstage Java EE管理コンソールを操作中に、Webサイトのコンテンツがブロックされたことを表すダイアログが出力される場合がある



上記ダイアログが出力されても、セキュリティ上の問題はありません。[閉じる]ボタンを押下し、ダイアログを閉じて処理を続行してください。

### 注意

上記ダイアログは、一例です。javascript以降のメッセージは、異なる場合があります。

### 表示しているページのタイトルや[バージョン]ボタンを押して表示される画面で「Unknown Version」と表示される場合がある

本現象が発生した場合は、Interstage Java EE DASサービスのサーバーログを参照し、出力されているメッセージに従って対処してください。

### [コンソール切替]ボタン押下時に、セッションタイムアウトが発生した旨のダイアログが表示される

本現象が発生した場合は、切替え先のInterstage管理コンソールでセッション切れ(セッションタイムアウト/サービスの再起動)が発生した可能性があります。

セッション切れが発生する前に、Interstage管理コンソールに切り替えてください。また、頻繁にセッションタイムアウトが発生する場合は、セッションタイムアウト時間に問題がないかを確認してください。

## ログアウト時に、セキュリティ情報に関する警告ダイアログが表示される

本現象が発生した場合は、GUI統合運用を行っているInterstage Java EE管理コンソールとInterstage管理コンソールの間で、SSL暗号化通信の使用有無の設定が異なっている可能性があります。

Interstage Java EE管理コンソールとInterstage管理コンソールの間で、SSL暗号化通信の使用有無の設定を合わせてください。

## Interstage Java EE管理コンソールで、ファイルのダウンロードができない

本現象が発生した場合は、Interstage Java EE管理コンソールを動作させるために必要なWebブラウザの設定が無効になっている可能性があります。

以下の手順で、Interstage Java EE管理コンソールが動作しているゾーンのWebブラウザの設定を有効にしてください。

### Internet Explorer 9/10の場合

1. Interstage Java EE管理コンソールを表示した状態で、Webブラウザの[ツール]メニューの[インターネット オプション]で、[セキュリティ]タブをクリックします。  
この時点で選択されているゾーンを変更しないでください。
2. [レベルのカスタマイズ]をクリックします。
3. [ダウンロード]セクションで、[ファイルのダウンロード]を「有効にする」に変更します。

### Internet Explorer 7/8の場合

1. Interstage Java EE管理コンソールを表示した状態で、Webブラウザの[ツール]メニューの[インターネット オプション]で、[セキュリティ]タブをクリックします。  
この時点で選択されているゾーンを変更しないでください。
2. [レベルのカスタマイズ]をクリックします。
3. [ダウンロード]セクションで、[ファイルのダウンロード]および[ファイルのダウンロード時に自動的にダイアログを表示]を「有効にする」に変更します。



### 注意

上記の対処後にダウンロードを行った際、ブラウザの上部にセキュリティに関する情報バーが表示された場合は、情報バーに表示されている対処を行い、再度ダウンロードしてください。

## SSL暗号化通信中のファイルのダウンロード時に、ダウンロード先のファイル名と異なるファイル名でダウンロードされる

Internet Explorer 9使用時に本現象が発生した場合は、以下の手順で、Webブラウザの設定を変更して回避してください。

1. Internet Explorerの[ツール]メニューの[インターネット オプション]で、[詳細設定]タブをクリックします。
2. [セキュリティ]セクションで、[暗号化されたページをディスクに保存しない]のチェックボックスを選択しない設定に変更します。

## ファイルまたはログのダウンロード時に、「xxxは移動または削除された可能性があります。」というダイアログが表示され、ダウンロードができない

Internet Explorer 9使用時に、ファイルまたはログのダウンロード時のダイアログで[ファイルを開く]ボタンを押下した場合、本現象が発生する可能性があります。以下の方法で対処してください。

1. ファイルまたはログのダウンロード時のダイアログで[保存]ボタンを押下し、一旦ファイルを保存します。
2. 保存したファイルを参照してください。

## 10.7 Webサービス・Webサービスクライアントの開発・運用時の異常

### Webサービスとの通信に失敗する場合

以下を参照して対処してください。

- 例外やメッセージが出力されている場合は、その内容に従って対処してください。
- Webサービスクライアントの作成で使用しているWSDLに誤りがないか、Webサービス提供者に確認してください。

### Webサービスエンドポイントの配備または起動に失敗した場合

以下を参照して対処してください。

- 例外やメッセージが出力されている場合は、その内容に従って対処してください。
- 配備時に配備モジュールの検証を有効にして、アプリケーションの移植性を検証してください。詳細は「Java EE運用ガイド」-「アプリケーションの検証」を参照してください。

また、アプリケーションが以下の条件に該当し、かつ上記の対処を実施しても問題が解決しない場合、

- Webサービスエンドポイントのクラスが@javax.jws.WebServiceアノテーションのendpointInterfaceパラメタでサービスエンドポイントインターフェースを指定している。

この場合は、以下のプロパティをInterstage Java EE DASサービスのシステムプロパティに設定して再起動し、アプリケーションを配備し直すことで問題が回避できる場合があります。

プロパティ	値
com.fujitsu.interstage.javaee.webservice.deploy.91compatiblemode.portcomponent	true

### 配備したアプリケーションのWebサービス名がInterstage Java EE管理コンソールに表示されない場合

Interstage Java EE管理コンソールのWebサービスのノードに、配備したWebサービスアプリケーションのWebサービス名が表示されない場合は、以下を参照して対処してください。

- 例外やメッセージが出力されている場合は、その内容に従って対処してください。
- Webサービスアプリケーションのパッケージングに誤りがある可能性があります。Webサービスアプリケーションのパッケージングについては、「Java EE運用ガイド」-「Webサービスアプリケーションの開発」-「パッケージング」を参照してください。

## 10.8 JMSの開発・運用時の異常

### メッセージブローカのJavaヒープが不足している場合の対処

Windows32/64

- imqsvcadmコマンドを使用してメッセージブローカをサービスに登録している場合  
imqsvcadm removeコマンドを使用して、メッセージブローカのサービスを削除後、imqsvcadm installコマンドに「-vmargs -Xmx<size>」オプションを指定して、メッセージブローカを再登録します。



例

メッセージブローカのJavaヒープを512MBにする場合

```
imqsvcadm install -vmargs -Xmx512m
```

- imqbrokerdコマンドを使用してメッセージブローカを起動している場合  
imqbrokerdコマンドを「-vmargs -Xmx<size>」オプションをつけて実行し、メッセージブローカを起動します。



例

メッセージブローカのJavaヒープを512MBにする場合

```
imqbrokerd -vmargs -Xmx512m
```

**Solaris32/64** **Linux32/64**

- rcスクリプトを使用してメッセージブローカを起動している場合  
ijmqファイルの「VM\_ARGS」の項目に「-Xmx<size>」オプションを追加します。



例

メッセージブローカのJavaヒープを512MBにする場合

```
VM_ARGS="-Xmx512m"
```

- imqbrokerdコマンドを使用してメッセージブローカを起動している場合  
imqbrokerdコマンドを「-vmargs -Xmx<size>」オプションをつけて実行し、メッセージブローカを起動します。



例

メッセージブローカのJavaヒープを512MBにする場合

```
imqbrokerd -vmargs -Xmx512m
```

## メッセージブローカを停止できない

起動しているすべてのメッセージブローカをimqcmdコマンドで停止してください。  
imqcmdコマンドで停止できないメッセージブローカについては、以下の対処を行なってください。

**Windows32/64**

- タスクマネージャのプロセサー一覧からイメージ名「imqbrokersvc.exe」が存在するかどうかを確認します。
- イメージ名「imqbrokersvc.exe」を選択して右クリックし、「プロセスツリーの終了」を選択します。
- タスクマネージャの警告画面が表示されるので、「はい」をクリックします。

**Solaris32/64**

- 当該メッセージブローカのPIDを特定します。
  - 以下のコマンドを実行して、imqbrokerdコマンドのPIDとPPIDを取得します。

**[global zoneの場合]**

```
ps -efz global | grep imqbrokerd | grep -v "grep imqbrokerd" | grep -v /opt/FJSVisje6/mq/bin/imqbrokerd
```

**[non-global zoneの場合]**

```
ps -ef | grep imqbrokerd | grep -v "grep imqbrokerd" | grep -v /opt/FJSVisje6/mq/bin/imqbrokerd
```

- 上記コマンドで取得したPIDとPPIDからメッセージブローカのPIDを取得します。

```
ps -e -o pid,ppid | grep $pid | grep -v -w $ppid | awk '{print $(1)}'
```

2. 当該メッセージブローカプロセスを強制停止します。

```
kill $メッセージブローカのpid
```

#### Linux32/64

1. 当該メッセージブローカのPIDを特定します。

- a. 以下のコマンドを実行して、imqbrokerdコマンドのPIDとPPIDを取得します。

```
ps -ef | grep imqbrokerd | grep -v "grep imqbrokerd" | grep -v /opt/FJSVisjee6/mq/bin/imqbrokerd
```

- b. 上記コマンドで取得したPIDとPPIDからメッセージブローカのPIDを取得します。

```
ps -e -o pid,ppid | grep $pid | grep -v -w $ppid | awk '{print $(1)}'
```

2. 当該メッセージブローカプロセスを強制停止します。

```
kill $メッセージブローカのpid
```

### アクセス制御ファイル(accesscontrol.properties)またはパスワードファイル(passwd)が存在しない場合の対処

アクセス制御ファイル(accesscontrol.properties)またはパスワードファイル(passwd)が存在しないために、コマンドを使用してメッセージブローカの起動または停止ができない場合は、以下の対処を行ってください。

#### アクセス制御ファイルが存在しない場合

下記デフォルトのアクセス制御ファイル「accesscontrol.properties.default」を「accesscontrol.properties」として、コマンド実行時に出力されるエラー情報に記載されたアクセス制御ファイルのパスへコピーしてください。

#### デフォルトのアクセス制御ファイル格納先

##### Windows32/64

```
C:\¥Interstage¥F3FMisjee¥imq¥etc¥accesscontrol.properties.default
```

##### Solaris32/64 Linux32/64

```
/opt/FJSVisjee/imq/etc/accesscontrol.properties.default
```

#### パスワードファイルが存在しない場合

下記デフォルトのパスワードファイル「passwd.default」を「passwd」として、コマンド実行時に出力されるエラー情報に記載されたパスワードファイルのパスにコピーしてください。

#### デフォルトのパスワードファイル格納先

##### Windows32/64

```
C:\¥Interstage¥F3FMisjee¥imq¥etc¥passwd.default
```

##### Solaris32/64 Linux32/64

```
/opt/FJSVisjee/imq/etc/passwd.default
```

### ポイント

- imqcmdコマンドでメッセージブローカを停止する場合は、デフォルトユーザとデフォルトパスワードを指定して実行してください。

- **Solaris32/64** **Linux32/64**  
コピー先のファイルの権限を「root/root 0644」に変更してください。

## 注意

本操作後は、メッセージブローカのユーザ設定が初期化されます。  
「Java EE運用ガイド」の「JMSのセキュリティ機能」に記述されている設定を再度実行してから、メッセージブローカを起動してください。

## JMS送信先リソースにQueueを使用するMessage-driven Beanを配備した場合、Topicの物理格納先に接続される

以下の可能性があります。

- JMS送信先リソースの参照先にIJServerクラスタが指定されていない可能性があります。  
asadmin list-resource-refsサブコマンドを使用して、JMS送信先リソースの参照先にIJServerクラスタが指定されているかを確認してください。IJServerクラスタが指定されていない場合は、asadmin create-resource-refサブコマンドを使用してIJServerクラスタを参照先に指定してください。
- JMS送信先リソースが利用不可になっている可能性があります。  
asadmin get IJServerクラスタ名.resources.admin-object-resource.JNDI名.enabledコマンドを使用して、JMS送信先リソースが利用不可になっていないかを確認してください。JMS送信先リソースが利用不可になっている場合は、有効にしてください。

上記対応後、Message-driven Beanの再配備を行い、必要に応じてTopicの物理格納先を削除してください。

## Windows(R)のサービスから「Interstage Message Queue Broker」の停止ができない **Windows32/64**

imqcmd shutdown bkrサブコマンドに、停止したいメッセージブローカのポート番号を指定して、メッセージブローカを停止してください。

なお、サービスプロセスが多く存在することにより、デスクトップヒープが枯渇している可能性があります。デスクトップヒープの枯渇については、「[5.5 デスクトップヒープが枯渇した場合](#)」を参照してください。

## 10.9 スレッドダンプが出力された場合の対処

スレッドダンプは、アプリケーション最大処理時間の超過時や、IJServerクラスタ起動時・強制停止時のタイムアウト発生時に出力されます。詳細は「Java EE運用ガイド」の以下を参照してください。

- 「Java EEアプリケーションの提供機能」-「IJServerクラスタ」-「アプリケーション最大処理時間の監視とレスポンス監視」
- 「Java EEアプリケーションの開発」-「アプリケーションのデバッグ」-「スレッドダンプを利用したデバッグ」

### ■出力例

以下にスタックトレース部の出力例を示します。

```
"Java Thread dump tool" prio=10 tid=0x000000006654000 nid=0x18ec waiting on condition [0x0000000000000000]
 java.lang.Thread.State: RUNNABLE

"Thread-127" daemon prio=6 tid=0x000000006653800 nid=0x1130 in Object.wait() [0x00000000088ff000]
 java.lang.Thread.State: TIMED_WAITING (on object monitor)
   at java.lang.Object.wait(Native Method)
   - waiting on <0x00000000fae25e18> (a com.sun.appserv.management.util.jmx.NotificationEmitterSupport
$SenderThread)
   at com.sun.appserv.management.util.jmx.NotificationEmitterSupport
```

```

$SenderThread.run(NotificationEmitterSupport.java:414)
  - locked <0x00000000fae25e18> (a com.sun.appserv.management.util.jmx.NotificationEmitterSupport
$SenderThread)
  . . . 中略 . . .

"Finalizer" daemon prio=8 tid=0x0000000014fe000 nid=0xfe4 in Object.wait() [0x00000000648f000]
java.lang.Thread.State: WAITING (on object monitor)
  at java.lang.Object.wait(Native Method)
  - waiting on <0x00000000f000bf00> (a java.lang.ref.ReferenceQueue$Lock)
  at java.lang.ref.ReferenceQueue.remove(ReferenceQueue.java:118)
  - locked <0x00000000f000bf00> (a java.lang.ref.ReferenceQueue$Lock)
  at java.lang.ref.ReferenceQueue.remove(ReferenceQueue.java:134)
  at java.lang.ref.Finalizer$FinalizerThread.run(Finalizer.java:159)

"Reference Handler" daemon prio=10 tid=0x0000000014f5000 nid=0xfe8 in Object.wait() [0x00000000638f000]
java.lang.Thread.State: WAITING (on object monitor)
  at java.lang.Object.wait(Native Method)
  - waiting on <0x00000000f000bf30> (a java.lang.ref.Reference$Lock)
  at java.lang.Object.wait(Object.java:485)
  at java.lang.ref.Reference$ReferenceHandler.run(Reference.java:116)
  - locked <0x00000000f000bf30> (a java.lang.ref.Reference$Lock)

"VM Thread" prio=10 tid=0x0000000014e7000 nid=0xe04 runnable

"GC task thread#0 (ParallelGC)" prio=6 tid=0x000000001439000 nid=0x17c8 runnable

"GC task thread#1 (ParallelGC)" prio=6 tid=0x00000000143a800 nid=0x1458 runnable

"VM Periodic Task Thread" prio=10 tid=0x0000000006505800 nid=0xe74 waiting on condition

"RAS Control Thread" prio=10 tid=0x0000000014e4800 nid=0xfdc runnable

```

## ■スタックトレースの見方

スタックトレースは、javaのjava.lang.ThrowableクラスのprintStackTraceメソッドが出力する形式と同様です。メソッドの呼び出し元から呼び出し先に向かって下から上にスタックに積まれ、現在走行中のメソッドが最上位に表示されます。

### 出力例

```

"httpWorkerThread-28090-2" daemon prio=6 tid=0x000000000984e800 nid=0x108c waiting on condition
[0x00000000d58e000]
  java.lang.Thread.State: TIMED_WAITING (sleeping)
  at java.lang.Thread.sleep(Native Method)
  at examples.TestServlet.doGet(TestServlet.java:15)
  at javax.servlet.http.HttpServlet.service(HttpServlet.java:734)
  at javax.servlet.http.HttpServlet.service(HttpServlet.java:847)
  . . . 以下省略 . . .

```

上記出力例では、TestServletのdoGetメソッドの15行目でsleepしていることがわかります(太字の部分)。

### 解析方法

スタックトレース中に自分で作成したクラスがあれば、意図せずに待ち状態になっていないか確認してください。行番号が表示されている場合は、ソースコードを参照することで実行中の処理を確認することができます。

スタックトレースの解析時には、「チューニングガイド」-「JDK/JREのチューニング」-「チューニング/デバッグ技法」-「スレッドダンプ」もあわせて参照してください。

以下は調査対象となる代表的なスレッドのスタックトレースの出力例です。

#### 【サーブレットのリクエスト処理中のスレッド】

リクエスト処理中のスレッドではサーブレット/JSP/フィルター等が処理されます。

#### サーブレット処理中のスレッドの出力例

サーブレットの場合はdoGetやdoPost等、オーバーライドして作成したサーブレットのメソッドより上位のスタックが調査対象となります。

```
"httpWorkerThread-28090-9" daemon prio=6 tid=0x000000009f9c800 nid=0xef8 runnable
[0x00000000ebbe000]
  java.lang.Thread.State: RUNNABLE
    at examples.TestServlet.doGet(TestServlet.java:36)
    at javax.servlet.http.HttpServlet.service(HttpServlet.java:734)
    at javax.servlet.http.HttpServlet.service(HttpServlet.java:847)
    at org.apache.catalina.core.ApplicationFilterChain.servletService(ApplicationFilterChain.java:427)
    at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:
333)
    at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:214)
    . . . 以下省略 . . .
```

上記出力例ではTestServletのdoGetメソッドより上位のスタックが調査対象となります。

ここではTestServletのdoGetメソッドが処理中であることがわかります(太字の部分)。

### JSP処理中のスレッドの出力例

JSP の場合は以下のようにorg.apache.jspから始まるパッケージ名のクラスに変換されています。通常HttpJspBaseのserviceメソッドより上位のスタックが調査対象となります。このスレッドからは、スクリプトレット部分から呼び出されるjavaクラスやカスタムタグクラス等が呼び出されます。

```
"httpWorkerThread-28090-10" daemon prio=6 tid=0x00000000be66000 nid=0x1790 waiting on condition
[0x00000000edce000]
  java.lang.Thread.State: TIMED_WAITING (sleeping)
    at java.lang.Thread.sleep(Native Method)
    at examples.TestTag.doEndTag(TestTag.java:41)
    at org.apache.jsp.test_jsp._jspx_meth_hello_sayHello_0(test_jsp.java from :102)
    at org.apache.jsp.test_jsp._jspService(test_jsp.java from :77)
    at org.apache.jasper.runtime.HttpJspBase.service(HttpJspBase.java:109)
    at javax.servlet.http.HttpServlet.service(HttpServlet.java:847)
    at org.apache.jasper.servlet.JspServletWrapper.service(JspServletWrapper.java:426)
    at org.apache.jasper.servlet.JspServlet.serviceJspFile(JspServlet.java:486)
    at org.apache.jasper.servlet.JspServlet.service(JspServlet.java:380)
    at javax.servlet.http.HttpServlet.service(HttpServlet.java:847)
    . . . 以下省略 . . .
```

上記出力例では、test.jspから呼び出されているカスタムタグTestTagのdoEndTagメソッドの41行目でsleepしていることがわかります(太字の部分)。

### フィルター処理中のスレッドの出力例

フィルターの場合はオーバーライドして作成したdoFilterメソッドより上位のスタックが調査対象となります。

```
"httpWorkerThread-28090-0" daemon prio=6 tid=0x000000000984d000 nid=0x11bc waiting on condition
[0x00000000d38e000]
  java.lang.Thread.State: TIMED_WAITING (sleeping)
    at java.lang.Thread.sleep(Native Method)
    at examples.TestFilter2.doFilter(TestFilter2.java:27) (2)
    at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:
246)
    at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:214)
    at examples.TestFilter1.doFilter(TestFilter1.java:23) (1)
    at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:
246)
    at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:214)
    . . . 以下省略 . . .
```

上記出力例ではフィルターが2つ定義されているため(1)と(2)の2箇所に作成したフィルターのスタックが出力されており、TestFilter1、TestFilter2の順番で呼び出され TestFilter2のdoFilterメソッドの27行目でsleepしていることがわかります(太字の部分)。

#### 【サーブレットのセッションタイムアウト時のスレッド】

サーブレットのセッションタイムアウトを監視するスレッドでは、HttpSessionListenerのsessionDestroyed等のセッションが破棄された場合の処理が動作します。

```
"ContainerBackgroundProcessor [StandardEngine [com. sun. appserv]. StandardHost [server]. StandardContext [/examples]]" daemon prio=6 tid=0x00000000d7e4800 nid=0x1844 runnable [0x00000000d10f000]
  java. lang. Thread. State: RUNNABLE
    at examples. TestListener. sessionDestroyed (TestListener. java:16)
    at org. apache. catalina. session. StandardSession. expire (StandardSession. java:865)
    - locked <0x00000000f1d7c890> (a org. apache. catalina. session. StandardSession)
    at org. apache. catalina. session. StandardSession. expire (StandardSession. java:823)
    at org. apache. catalina. session. StandardSession. isValid (StandardSession. java:739)
    at org. apache. catalina. session. StandardManager. processExpires (StandardManager. java:1014)
    at org. apache. catalina. core. StandardContext. backgroundProcess (StandardContext. java:5731)
    at org. apache. catalina. core. ContainerBase
$ContainerBackgroundProcessor. processChildren (ContainerBase. java:1835)
  at org. apache. catalina. core. ContainerBase$ContainerBackgroundProcessor. run (ContainerBase. java:1824)
  at java. lang. Thread. run (Thread. java:662)
```

上記出力例ではTestListenerのsessionDestroyedメソッドの16行目が処理中であることがわかります(太字の部分)。なお、セッションのタイムアウト時に呼び出される処理は、スレッドダンプ自動採取の対象外です。

#### 【IJServerクラスタ起動時のスレッド】

IJServerクラスタ起動時には、load-on-startupを指定したサーブレットのinit等が実行されます。

```
"pool-1-thread-7" prio=6 tid=0x000000008fb1800 nid=0x1020 runnable [0x00000000a8af000]
  java. lang. Thread. State: RUNNABLE
    at examples. TestServlet. init (TestServlet. java:39)
    at javax. servlet. GenericServlet. init (GenericServlet. java:270)
    at org. apache. catalina. core. StandardWrapper. loadServlet (StandardWrapper. java:1100)
    - locked <0x00000000fb3cd040> (a org. apache. catalina. core. StandardWrapper)
    at org. apache. catalina. core. StandardWrapper. load (StandardWrapper. java:1023)
    - locked <0x00000000fb3cd040> (a org. apache. catalina. core. StandardWrapper)
    at org. apache. catalina. core. StandardContext. loadOnStartup (StandardContext. java:4999)
    at org. apache. catalina. core. StandardContext. start (StandardContext. java:5406)
    - locked <0x00000000f0a9f4c8> (a com. sun. enterprise. web. WebModule)
    at com. sun. enterprise. web. WebModule. start (WebModule. java:345)
    - locked <0x00000000f0a9f4c8> (a com. sun. enterprise. web. WebModule)
    at com. sun. enterprise. web. LifecycleStarter. doRun (LifecycleStarter. java:58)
    at com. sun. appserv. management. util. misc. RunnableBase. runSync (RunnableBase. java:304)
    at com. sun. appserv. management. util. misc. RunnableBase. run (RunnableBase. java:341)
    at java. util. concurrent. Executors$RunnableAdapter. call (Executors. java:439)
    at java. util. concurrent. FutureTask$Sync. innerRun (FutureTask. java:303)
    at java. util. concurrent. FutureTask. run (FutureTask. java:138)
    at java. util. concurrent. ThreadPoolExecutor$Worker. runTask (ThreadPoolExecutor. java:886)
    at java. util. concurrent. ThreadPoolExecutor$Worker. run (ThreadPoolExecutor. java:908)
    at java. lang. Thread. run (Thread. java:662)
```

上記出力例ではTestServletのinitメソッドの39行目が処理中であることがわかります(太字の部分)。

#### 【IJServerクラスタ停止時のスレッド】

IJServerクラスタの停止時には、サーブレットのdestroy等が実行されます。

```
"RMI TCP Connection (23)-127. 0. 0. 1" daemon prio=6 tid=0x00000000988f800 nid=0x1030 runnable [0x00000000bb0d000]
  java. lang. Thread. State: RUNNABLE
    at examples. TestServlet. destroy (TestServlet. java:43)
```

```
at org.apache.catalina.core.StandardWrapper.unload(StandardWrapper.java:1461)
- locked <0x00000000f1dc8120> (a org.apache.catalina.core.StandardWrapper)
at org.apache.catalina.core.StandardWrapper.stop(StandardWrapper.java:1824)
at org.apache.catalina.core.StandardContext.stop(StandardContext.java:5531)
- locked <0x00000000f1dc8418> (a com.sun.enterprise.web.WebModule)
. . . 中略 . . .
at java.util.concurrent.ThreadPoolExecutor$Worker.runTask(ThreadPoolExecutor.java:886)
at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:908)
at java.lang.Thread.run(Thread.java:662)
```

上記出力例ではTestServletのdestroyメソッドの43行目が処理中であることがわかります(太字の部分)。

## 10.10 Java EEセキュリティの異常

### 認証または認可に失敗する

認証または認可に失敗する場合、以下の可能性が考えられます。

- **レルムの設定に誤りがある**  
「Java EE運用ガイド」-「デフォルトレルムの設定」を確認してください。  
Webアプリケーションでレルムの設定を行っている場合、「Java EE運用ガイド」-「Web application deployment descriptor (web.xml)」にある、<realm-name>タグの設定を見直してください。  
「Java EE運用ガイド」-「fileレルムの設定」を確認してください。  
「Java EE運用ガイド」-「ldapレルムの設定」を確認してください。
- **Interstage deployment descriptorの設定に誤りがある**  
「Java EE運用ガイド」-「Interstage deployment descriptor」の設定を見直してください。  
「Java EE運用ガイド」-「SSL」を参照し、SSLの環境設定を見直してください。
- **キーファイルが存在しない**  
IIServerクラスタの再起動を行ってください。  
キーファイルで指定したパス上のディレクトリにアクセス権があることを確認してください。
- **Interstageディレクトリサービスに接続できない**  
Interstageディレクトリサービスを起動してください。  
Interstageディレクトリサービスを運用するマシンとIIServerクラスタを運用するマシン間を繋ぐネットワークに異常がないことを確認してください。
- **クライアントからの接続方法に誤りがある**  
「Java EE運用ガイド」-「クライアントとWebサーバ間の通信データの保証に関する要求」を使用している場合、WebアプリケーションにあるリソースへのアクセスはSSLで通信してください。

## 10.11 ログの異常

ログ関連の以下のトラブルについて対処方法を説明します。

### 長すぎるログ出力ディレクトリを設定した

ログ出力ディレクトリに、ファイル名(ローテーション後のファイル名)を含めてOSの最大値を超えるパス名を指定した場合、通常の削除処理ではログ出力ディレクトリを削除できません。長すぎるログ出力ディレクトリを削除するには、Microsoft サポートオンラインの「文書番号: 320081 NTFS ファイル システム上のファイルまたはフォルダを削除できない」を参照してください。

ログ出力ディレクトリに、ファイル名(ローテーション後のファイル名)を含めてOSの最大値を超えるパス名を指定すると、一括情報採取に失敗します。ログ出力ディレクトリのパスの長さは、ファイル名(ローテーション後のファイル名)を含めてOSの最大値の範囲内になるようにしてください。

## 10.12 アンインストール時の異常



### アンインストール中に“ijwscinit”から始まるエラーメッセージが出力された

この場合、IIServerクラスタの定義情報がWebサーバコネクタの定義ファイルに残存した状態になり、IIServer(J2EE)の運用で異常が発生する場合があります。

残存した定義情報を削除するために、以下の手順でWebサーバコネクタの定義ファイルを復旧してください。

1. IIServerクラスタが連携していたWebサーバを一旦削除して再作成します。
2. Webサーバコネクタを再作成、またはIIServer(J2EE)が連携するWebサーバを選択しなおします。

# 第11章 CORBAサービス使用時の異常

CORBAサービスの運用でトラブルが発生した場合の対処方法について説明します。

## 11.1 CORBAサービス運用中のシステム例外発生

### CORBAサービス運用中にCOMM\_FAILUREが発生する

CORBAサービス運用中にシステム例外(COMM\_FAILURE)が発生した場合、以下の原因が考えられます。

#### configファイルの設定による問題

configファイル(CORBAサービスの動作環境ファイル)の以下のパラメタ値で不足が発生した場合、COMM\_FAILUREが発生します。

なお、サーバ機能では、max\_IIOp\_resp\_conの値および実際の消費量をodprtcuparamコマンドにより確認することができます。また、oddumprespコマンドを使用すると、max\_IIOp\_resp\_conの使用状況(接続の状態)について確認できます。

- **max\_IIOp\_resp\_con**

[発生マシン]

サーバ

[発生条件]

サーバでクライアントから受け付ける接続数を超えて、リクエストを受け付けた場合に発生します。動作ログを採取し、ログファイルに以下の情報があれば、これが原因と考えられます。

Error gen (OM\_con\_lib.c,xxxx): responder: max connection limit exceeded

[発生頻度]

通常、クライアントアプリが終了するまで継続して発生します。

- **period\_receive\_timeout、period\_server\_timeout(タイマ監視パラメタ)**

[発生マシン]

クライアント/サーバ

[発生条件]

以下の監視時間が設定値を超えた場合に発生します。

period\_receive\_timeout

リクエスト発行から応答受信までの時間

period\_server\_timeout

起動タイプがpersistent以外のサーバで、fork()/exec()からCORBA\_ORB\_init()までの時間

[発生頻度]

瞬間的に発生します。

#### initial\_hosts/inithostファイルの設定による問題

initial\_hosts(inithost)ファイル(サービスの所在問い合わせ先定義)に以下の問題が発生した場合、COMM\_FAILUREが発生します。

- **ホスト名、ポート番号の誤り**

[発生マシン]

クライアント/サーバ

[発生条件]

誤ったホスト名やポート番号が設定された場合に発生します。

**[発生頻度]**

再設定まで継続して発生します。

**アプリケーション/運用上の問題**

運用中に以下のような問題が発生した場合、COMM\_FAILUREが発生します。

・ **サービス未起動**

**[発生マシン]**

クライアント/サーバ

**[発生条件]**

CORBAサービスが起動されていない、またはinitial\_hosts/inihostファイルやオブジェクトリファレンスに定義されたサービスが動作していない場合に発生します。

**[発生頻度]**

サービス起動まで継続して発生します。

・ **インタフェースリポジトリとの通信不可**

**[発生マシン]**

クライアント/サーバ

**[発生条件]**

TypeCodeインタフェースの以下のメソッド(TypeCode情報取得)を発行時、インタフェースリポジトリと通信できなかった場合に発生します。

CORBA\_TypeCode\_name()

CORBA\_TypeCode\_member\_count()

**[発生頻度]**

メソッド発行時

・ **ネットワークエラー**

**[発生マシン]**

クライアント/サーバ

**[発生条件]**

LANの負荷が高く、TCP/IPへの送信/受信でエラーになった場合に発生します。

**[発生頻度]**

CORBAサービス起動まで継続して発生します。

・ **サーバマシンのシステムダウン**

**[発生マシン]**

クライアント

**[発生条件]**

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を異なるマシン上で運用している際、TCP/IPの仕様のよりサーバマシンのシステムダウンをクライアント側で検出できなかった場合に発生します。これは、クライアントにおいてconfigファイルのperiod\_receive\_timeoutパラメタで設定した待機時間を超過した際に通知されます。

**[発生頻度]**

ネットワーク構成に依存するため、特定できません。

**[対処]**

サーバマシン、ルータなどのネットワーク装置の状態を確認してください。

## CORBAサービス運用中にNO\_RESOURCESが発生する

CORBAサービス運用中にシステム例外(NO\_RESOURCES)が発生した場合、以下の原因が考えられます。

### configファイルの設定による問題

configファイル(CORBAサービスの動作環境ファイル)の以下のパラメタ値で不足が発生した場合、NO\_RESOURCESが発生します。

なお、サーバ機能ではmax\_IIOp\_resp\_requestsの値および実際の消費量をodprtcparamコマンドにより確認することができます。また、oddumprespコマンドを使用すると、max\_IIOp\_resp\_requestsの使用状況(コネクションの状態)について確認できます。

- **max\_IIOp\_resp\_requests**

**[発生マシン]**

サーバ

**[発生条件]**

サーバ側で同時に処理できるリクエスト数を超えた場合に発生します。

**[発生頻度]**

瞬間的に発生します。

## CORBAサービス運用中にUNKNOWNが発生する

CORBAサービス運用中にCORBA\_ORB\_init関数でシステム例外(UNKNOWN)が発生した場合、以下の原因が考えられます。

### configファイルの設定による問題

configファイル(CORBAサービスの動作環境ファイル)の以下のパラメタ値で不足が発生した場合、CORBA\_ORB\_init関数でUNKNOWN例外が発生します。

なお、max\_processesの値および実際の消費量をodprtcparamコマンドにより確認することができます。

- **max\_processes**

**[発生条件]**

同時に起動できるCORBAアプリケーションのプロセス数を超えた場合に発生します。

システムログ(Windows(R)ではイベントログ)にod10936メッセージが出力されていれば、これが原因と考えられます。

**[発生頻度]**

システム構成に依存するため、特定できません。

### odprocwatchプロセス停止による問題

CORBAサービスが起動中の状態で、odprocwatchプロセス(Windows(R)版ではイメージ名はodprocwatch.exe)が起動していない場合、CORBA\_ORB\_init関数でUNKNOWN例外が発生します。

なお、クライアントパッケージではCORBAサービスが起動しないため該当しません。

#### Windows32/64

タスクマネージャでプロセス一覧を確認し、イメージ名がOdstart.exeのプロセスが存在し、odprocwatch.exeのプロセスが存在しない場合は、これが原因と考えられます。

#### Solaris32/64 Linux32/64

psコマンドでシステム上のプロセス一覧を確認し、OD\_startプロセスが存在し、odprocwatchプロセスが存在しない場合は、これが原因と考えられます。

## [対処]

一括情報採取ツールを使用して調査情報を採取した後、Interstageを再起動してください。

## 11.2 クライアントへのレスポンスが悪い

---

### サーバアプリケーションの処理が遅延している場合

クライアントからの要求を処理しているサーバアプリケーションの処理が遅延している可能性があります。サーバアプリケーションの処理が遅延していないか調査してください。

ワークユニットで動作している場合、性能監視ツールを使用することが可能です。性能監視ツールにより、遅延場所の調査を行ってください。

### メッセージにod11110またはod11111が出力されている場合

クライアントからの要求数に比べ、キューが使用するバッファ数が足りない状態であることが考えられます。各メッセージの[ユーザの対処]を参照のうえ、バッファ数のチューニングを行ってください。

### リクエストの受け付けに時間がかかっている場合

キュー制御に使用されるバッファよりもリクエスト電文の方が大きい場合に、一つ前のリクエスト電文をサーバアプリケーションが受信するのを待っている可能性があります。“OLTPサーバ運用ガイド”の“バッファ制御”を参照のうえ、バッファのチューニングを行ってください。

## 11.3 アプリケーションが停止しない

---

### CORBAワークユニットが停止しない

以下に記載されている方法でCORBAワークユニットの停止処理を行ったにもかかわらず、CORBAワークユニットが停止処理中の状態のまま停止が完了しない場合があります。

- isstopwuコマンドで通常停止または同期停止を実行
- Interstage管理コンソールを使用して通常停止または同期停止を実行

### アプリケーションの実装と登録情報の不一致による問題

サーバアプリケーションの実装と登録情報が一致していないときに、この現象が発生する可能性があります。この現象が発生した場合、強制停止を実行することによりCORBAワークユニットを停止することが可能です。

サーバアプリケーションの動作モードに“COMPATIBLE”を指定した場合は特別な処理は不要ですが、動作モードに“SYNC\_END”を指定した場合は活性化メソッド(注)から復帰した後に(exit関数の発行など)明示的にプロセスを終了させる必要があります。ユーザ論理のスレッドがすべて終了してもCORBA機能用の管理スレッドが存在するため、プロセスは終了しません。そのため、“SYNC\_END”の場合に活性化メソッド後のプロセス終了処理が行われていないと、CORBAワークユニットの停止処理を行ったにもかかわらず、CORBAワークユニットが停止しない(CORBAワークユニットのプロセスが残存する)という現象が発生します。

たとえば、V7以前のサンプルでは活性化メソッド後にスレッドの終了処理(Solaris版: thr\_exit関数の発行、Windows(R)版: ExitThread関数の発行、Linux版: pthread\_exit関数の発行)を行っていますが、これは“COMPATIBLE”を期待した処理です。動作モードを“SYNC\_END”に変更する場合は、活性化メソッド後にexit関数を発行するなどの修正をする必要があります。

動作モードが“SYNC\_END”で活性化メソッド後にプロセスを終了させていない場合は、以下のどちらかの対処を行ってください。

- 動作モードを“COMPATIBLE”に変更する。
- 活性化メソッド復帰後にプロセスを終了するよう処理を修正する。

注) 活性化メソッドの一覧は以下のとおりとなります。

C : CORBA\_BOA\_impl\_is\_ready(), CORBA\_BOA\_obj\_is\_ready()  
C++ : CORBA::BOA::impl\_is\_ready(), CORBA::BOA::obj\_is\_ready(), PortableServer::POAManager::activate()  
Java : org.omg.PortableServer.POAManager.activate()  
COBOL : ORBA-BOA-IMPL-IS-READY, CORBA-BOA-OBJ-IS-READY  
OOCOBOL : CORBA-BOA-IMPL\_IS\_READY, CORBA-BOA-OBJ\_IS\_READY

### COBOLアプリケーションの標準出力切り替えを実施していないことによる問題

ワークユニットではコンソールが無いため、COBOLアプリケーションではバッチ処理での実行と同じ扱いになります。そのためCOBOLアプリケーションでDISPLAY文を使用している場合は、データ出力先をファイルに変更するか環境変数“@WinCloseMSG”にOFFを設定するなどの対処を行わないとワークユニット停止時にアプリケーションがハングアップする可能性があります。COBOLのバッチ処理についての詳細は、ご利用されているCOBOLのマニュアルを参照してください。

### CORBAサーバアプリケーションが停止しない

odcntlqueコマンドを使用してCORBAサーバアプリケーションの停止を行ったにもかかわらずCORBAサーバアプリケーションのプロセスが終了しない場合、サーバアプリケーションの実装と登録情報が一致していない可能性があります。

サーバアプリケーションの動作モードに“COMPATIBLE”を指定した場合は特別な処理は不要ですが、動作モードに“SYNC\_END”を指定した場合は活性化メソッドから復帰した後に(exit関数の発行など)明示的にプロセスを終了させる必要があります。ユーザ論理のスレッドがすべて終了してもCORBA機能用の管理スレッドが存在するため、プロセスは終了しません。そのため、“SYNC\_END”の場合に活性化メソッド後のプロセス終了処理が行われていないと、CORBAサーバアプリケーションの停止処理を行ったにもかかわらず、CORBAサーバアプリケーションのプロセスが残存するという現象が発生します。

動作モードが“SYNC\_END”で活性化メソッド後にプロセスを終了させていない場合は、以下のどちらかの対処を行ってください。

- ・ 動作モードを“COMPATIBLE”に変更する。
- ・ 活性化メソッド復帰後にプロセスを終了するよう処理を修正する。

## 11.4 アプリケーションがエラーとなる

### OOCOBOLアプリケーションがエラーとなる

使用している製品とは異なるバージョンで作成したOOCOBOLアプリケーションを実行している可能性があります。OOCOBOLアプリケーションは、異なるバージョン間でのバイナリ互換がありません。移行する場合には、再ビルドを行ってください。

### CORBAアプリケーションがエラーとなる

 Windows 32/64

Interstageのインストールフォルダ、およびその配下のフォルダとファイルが暗号化されていないことを確認してください。インストールフォルダ、およびその配下のフォルダとファイルを暗号化した場合、CORBAアプリケーションの実行時にエラーが発生する可能性があります。

### IDLコンパイラが生成したJava言語マッピングのスタブ・スケルトンでjava.lang.NullPointerExceptionが発生する

クライアントアプリケーションのin、inoutパラメタ、およびサーバアプリケーションのout、inoutパラメタ、復帰値において、以下のデータ型に“nullオブジェクト”を設定していないかを確認してください。“nullオブジェクト”を指定している場合は、“nullオブジェクト”を設定しないように修正してください。

- ・ 文字列型
- ・ any型

- ・ シーケンス型
- ・ 構造体
- ・ 共用体
- ・ 配列

**“org.omg.CORBA.COMM\_FAILURE: vmcid: SUN minor code: 201 completed: No”が発生する  
または、  
“java.lang.ClassCastException”が発生する**

ORB(Object Request Broker)が正しく指定されているかを確認してください。

ORBは、以下の方法で必ず指定する必要があります。

- ・ Javaアプリケーション実行時、javaコマンドの-Dオプションにプロパティ情報を指定します。
- ・ プロパティ情報を指定したファイル「orb.properties」を作成して、所定のディレクトリ配下に格納します。

ORB(Object Request Broker)の指定方法の詳細については、“アプリケーション作成ガイド(CORBAサービス編)”の“アプリケーションの開発(Java言語)” – “CORBAアプリケーションの実行” – “ORB(Object Request Broker)の指定”を参照してください。

**“org.omg.CORBA.INITIALIZE: can't instantiate default ORB implementation  
com.fujitsu.ObjectDirector.CORBA.ORB vmcid: 0x0 minor code: 0 completed: No”が発生する**

以下の対応を行ってください。

- ・ 環境変数CLASSPATHに必要なファイルが設定されていない可能性があります。環境変数CLASSPATHが正しく設定されているかを確認してください。環境変数CLASSPATHの設定については、“アプリケーション作成ガイド(CORBAサービス編)”の“アプリケーションの開発(Java言語)” – “サーバの環境設定”を参照してください。
- ・ プレインストール型Javaライブラリを使用してアプレット運用を行っている場合、アプレット運用に必要な権限が不足しています。アプレットおよびJavaライブラリの環境設定を確認してください。アプレットおよびJavaライブラリの環境設定については、“アプリケーション作成ガイド(CORBAサービス編)”の“アプリケーションの開発(Java言語)” – “アプレットのデジタル署名”を参照してください。

## 11.5 CORBAワークユニットの標準出力/標準エラー出力がファイルに出力されない

Windows32/64

### CORBAワークユニットの標準出力(標準エラー出力)がstdoutファイル(stderrファイル)に出力されない

C/C++アプリケーションの場合、Interstage V6以前で作成したアプリケーションを再ビルドせずに使用している可能性があります。Interstage V7ではCORBAサービスが提供しているライブラリのコンパイルオプションが変更されたため、Interstage V6以前で作成したアプリケーションを移行する場合、アプリケーションのコンパイルオプションを変更して再ビルドする必要があります。詳細については“移行ガイド”の“CORBAサービスの資源”、“Windowsアプリケーションのコンパイルオプション変更について”を参照してください。

## 11.6 インスタンス保持機能が正常に動作しない

### CORBAアプリケーション情報定義ファイルに“iswitch=ON”を指定しているにもかかわらず、インスタンスデータが保持されない

クライアント側無通信監視タイムアウトにより、接続が切断された可能性があります。config(CORBAサービスの動作環境ファイル)の“period\_client\_idle\_con\_timeout”の値を見直してください。

## 11.7 アプリケーションコンパイル時の異常

---

Solaris32/64

### Solarisでアプリケーションのコンパイルに失敗する

Oracle Solaris Studio 12.2以降を使用してアプリケーションを作成する場合に以下のコンパイルエラーとなる場合があります。

```
Error: ..... requires an lvalue.  
エラー: ..... を初期化するには左辺値が必要です.
```

CCコマンドに"-features=rvaluerel"を指定してコンパイルしてください。

## 第12章 コンポーネントランザクションサービス使用時の異常

コンポーネントランザクションサービスの使用に際して、トラブルが発生した場合の対象方法について説明します。

### 12.1 アプリケーション処理要求時の異常発生時の対処

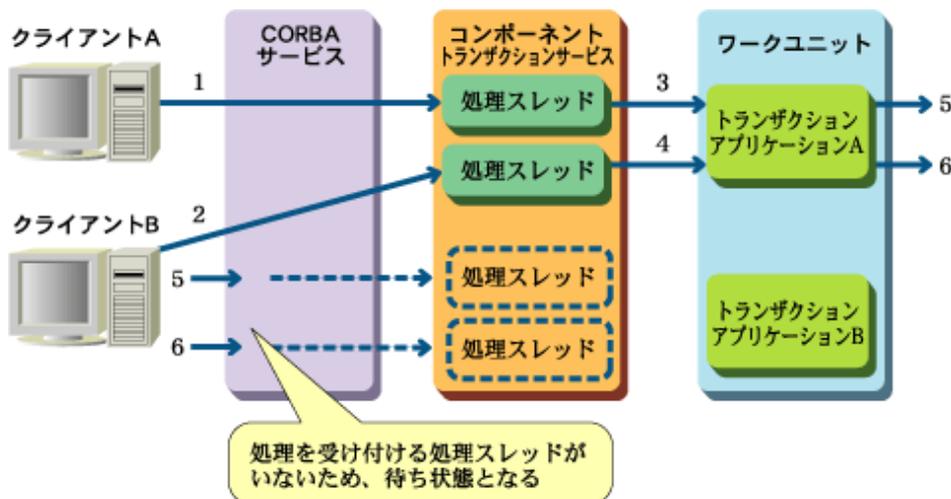
ここでは、アプリケーション実行時に発生する異常終了の事例とその対処方法について説明します。

#### 多階層システムのアプリケーション間連携時に処理要求が無応答となる

3階層システムなどでコンポーネントランザクションサービスを構成すると、以下の図に示すとおり接続している端末数に対して多くの接続資源を消費します。

各クライアントアプリケーションはランザクションアプリケーションAに処理要求を行い、さらにランザクションアプリケーションAはランザクションアプリケーションBへ処理要求を行います。

図中の1から4の処理で、コンポーネントランザクションサービスの処理スレッドが使い切られているため、アプリケーション間の処理要求時(5および6)に必要な処理スレッドが不足し、CORBAサービスで処理要求が待ち状態となり、クライアントAおよびクライアントBの処理要求のいずれも復帰することができません。したがって、処理スレッドが解放されず、デッドロックが発生して、クライアントでは処理要求が無応答となります。



このため、各処理要求に対する受付順番によりデッドロック状態となり、処理要求が無応答となることがあります。この場合、システムログに以下のメッセージが出力されます。

```
TD: 警告: td12038: 最大接続クライアント数に達しました(ORB- PID)
ORB: ワークユニット種別を表示します。
PID: 最大接続クライアント数を表示します。
```

処理要求が無応答となり、上記メッセージが出力されている場合は、メッセージの[ユーザの対処]に記載しているように、システム規模を見直す必要があります。



## 参照

システム規模の変更については、「運用ガイド(基本編)」を参照してください。

### サーバから不完全な文字列型データが復帰する

以下の条件の場合に、クライアントアプリケーションに復帰したデータが途切れるなど、不完全な状態になることがあります。

- ・ サーバアプリケーションにCOBOLを使用している。かつ、
- ・ IDL定義に文字列型で宣言している。かつ、
- ・ サーバアプリケーションで、文字列以外のデータ(バイナリデータや数値データなど)を設定している。

この場合、サーバアプリケーションで設定したデータに0x00のコードが含まれている可能性があります。

サーバアプリケーションの処理を見直してください。

なお、本現象については、「アプリケーション作成ガイド(コンポーネントトランザクション編)」の「トランザクションアプリケーション作成上の注意点」を参照してください。

### 富士通標準コード変換(iconv\_open関数)が異常復帰する Solaris32

富士通標準コード変換のiconv\_open関数が復帰コード:22(指定された変換元コード系と変換先コード系の組み合わせはサポートされていません)で復帰した場合、富士通標準コード変換が動作せずに、libc.soに含まれる関数が動作している可能性があります。

この場合、tdlinknormapmコマンドでAPMを再作成して、富士通標準コード変換のライブラリをAPMの先頭に結合してください。

### アプリケーションで出力した標準出力／標準エラー出力が、カレントフォルダ配下のstdoutファイルまたはstderrファイルに出力されない Windows32/64

アプリケーションより標準出力／標準エラー出力に対して出力されたデータは、カレントフォルダ配下のstdoutファイル／stderrファイルに出力されますが、Microsoft(R) Visual C++ .NET／Microsoft(R) Visual C++ 2005を使用してビルドしたアプリケーションでは、標準出力／標準エラー出力に対して出力されたデータが、カレントフォルダ配下のstdoutファイル／stderrファイルに出力されません。

正しく出力するためには、アプリケーションのプログラムの先頭に以下のコードを追加してください。

```
freopen("stdout", "w", stdout);
freopen("stderr", "w", stderr);
```



## 注意

- ・ Microsoft(R) Visual C++ 2005を使用してビルドした場合、以下の警告が出力されることがありますが、動作上の問題はありません。

```
warning C4996: 'freopen' が古い形式として宣言されました。
```

- ・ 前出口プログラムを使用する場合は、前出口プログラムの先頭に追加してください。前出口プログラムに追加した場合、本処理および後出口プログラムへの対処は必要ありません。
- ・ 前出口プログラムを使用しない場合は、本処理の先頭に追加し、かつ、初回呼び出し時のみ実行されるよう対処してください。
- ・ ワークユニット出口プログラム／プロセス回収出口プログラムを使用する場合は、各出口プログラムの先頭に追加してください。

## 12.2 アプリケーション作成時の異常発生時の対処

---

アプリケーション作成時に異常が発生した場合は、「第11章 CORBAサービス使用時の異常」-「11.7 アプリケーションコンパイル時の異常」を参照して対処してください。

## 12.3 ワークユニット使用時の異常発生時の対処

---

ワークユニットを使用している場合に異常が発生した場合の対処について、以下の内容で説明します。

- ・ ワークユニット起動失敗
- ・ ワークユニット起動コマンド無応答
- ・ ワークユニット停止コマンド無応答
- ・ ワークユニットが異常終了
- ・ サーバアプリケーション無応答
- ・ コンソールからログインできない **Solaris32/64**
- ・ コマンドの実行が失敗する **Windows32/64**

### 12.3.1 ワークユニット起動失敗

---

ワークユニットの起動に失敗した場合、起動に失敗した原因がメッセージに出力されています。使用者はメッセージを参照し、対処を行ってください。

以下にワークユニット起動に失敗する原因の可能性について説明します。

#### ワークユニット定義が登録されていない

ワークユニット定義が登録されていない場合、以下のどちらかのメッセージが出力されます。ワークユニット定義を登録してください。

```
tdstartwu: エラー: td21015: コマンドの実行に失敗しました 理由コード(3)
isstartwu: エラー: is30807: コマンドの実行に失敗しました 理由コード(3)
```

#### カレントディレクトリが存在しない

ワークユニット定義で指定されたディレクトリが存在しない場合、メッセージEXTP4317が出力されます。カレントディレクトリを作成するか、ワークユニット定義を正しく修正してください。

#### カレントフォルダを使用中である **Windows32/64**

ワークユニット起動時は、以前使用していたフォルダを削除したあと、カレントフォルダを作成します。したがって、エクスプローラなどによりカレントフォルダ配下が使用中である場合、または使用中のプロセスが存在する場合は、メッセージEXTP4316が出力されます。カレントフォルダが使用されている原因を調査してください。

#### ワークユニット出口プログラム(前処理)が異常終了した

ワークユニット起動時に呼び出されるワークユニット出口プログラム(前処理)が異常終了した場合、メッセージEXTP4659が出力されます。異常終了した原因を調査してください。これは使用者が作成したプログラムであるため、使用者が調査する必要があります。

## ワークユニット出口プログラム(前処理)が異常復帰した

ワークユニット起動時に呼び出されるワークユニット出口プログラム(前処理)が異常終了した場合、メッセージEXTP4661が出力されます。異常復帰した原因を調査してください。これは使用者が作成したプログラムであるため、使用者が調査する必要があります。

## 実行ファイルのオープンで異常が発生した

以下の場合、メッセージEXTP4640が出力され、ワークユニットの起動に失敗します。

- ・ワークユニット定義のアプリケーションライブラリパスに設定されたフォルダに、実行ファイルが存在しません。ワークユニット定義のアプリケーションライブラリパスおよび実行ファイル名が正しく設定されているかを確認してください。
- ・アプリケーションプログラムの実行ファイルにアプリケーションが使用するライブラリがリンクされていません。必要なライブラリがすべてリンクされているかを確認してください。
- ・ワークユニット定義のアプリケーション使用パス(Windows版)、またはアプリケーション使用ライブラリパス(Solaris版、Linux版)に、アプリケーションプログラムが動作時に使用するライブラリのパスが設定されていません。すべてのライブラリへのパスが設定されているかを確認してください。
- ・メモリ不足が発生しています。メモリに十分な空き領域が存在するかを確認してください。

## デスクトップヒープ領域が不足している Windows32/64

正常に起動していたワークユニットが、プロセス多重度を増加してワークユニットを起動すると、ポップアップダイアログが出力されて起動に失敗する場合、デスクトップヒープ領域が不足している可能性があります。デスクトップヒープ領域を拡張して、再起動してください。

## スワップ領域が不足している Solaris32/64 Linux32/64

正常に起動していたワークユニットが、プロセス多重度を増加すると、ワークユニットの起動に失敗する場合、スワップ領域が不足している可能性があります。この場合、メモリ不足を示すInterstageのメッセージ、およびスワップ領域不足を示すOSのメッセージが出力されます。

スワップ領域が不足している場合は、スワップ領域を追加してください。また、以下のコマンドを実行し、システムにスワップ領域が正しく追加されているかを確認してください。

```
/usr/sbin/swap -l
```

## インプリメンテーションリポジトリに登録されたサーバアプリケーションの動作モードに「COMPATIBLE」が指定されている

CORBAワークユニットの起動に失敗した場合、インプリメンテーションリポジトリに登録されたサーバアプリケーションの活性化後の動作モードがデフォルトの「COMPATIBLE」が指定されている可能性があります。以下に該当する場合、サーバアプリケーションの活性化後の動作モードは「SYNC\_END」を指定してください。

- ・サーバアプリケーションをプロセスモードで作成した場合
- ・Javaのサーバアプリケーションの場合
- ・COBOLのサーバアプリケーションの場合
- ・OOCOBOLのサーバアプリケーションの場合

## 環境変数またはプロセス起動パラメタが不足している

CORBAワークユニットの起動に失敗した場合、アプリケーションの起動に必要な環境変数や起動パラメタが不足している可能性があります。

この場合、ワークユニット定義の「Start Log」(プロセス起動ログ採取)ステートメントに「YES」を指定し、プロセス起動時の

ログ出力機能を有効にして、ワークユニットを起動してください。

これにより、カレントディレクトリ配下に、プロセス起動時に設定されている環境変数と起動パラメタがログファイルに出力されます。ログファイルを参照し、環境変数と起動パラメタに過不足がないかを確認してください。過不足がある場合は、ワークユニット定義を修正し、再度実行してください。

## 注意

実行ファイル名および実行ファイルのパスに誤りがある場合、プロセス起動時のログは出力されません。

### クラスパスにCORBAサービスが提供するJavaライブラリ設定されていない Solaris32/64 Linux32/64

クラスパスに使用するJDK/JREのバージョンに応じたCORBAサービスが提供するライブラリを設定する必要があります。詳細については、「アプリケーション作成ガイド (CORBAサービス編)」の「CORBAアプリケーションの実行」を参照してください。

### 実行環境とバージョンの異なるJDKでコンパイル(javac)したアプリケーションをCORBAワークユニットに配備している

CORBAワークユニットに配備したCORBA-Javaアプリケーションが、アプリケーションの格納ディレクトリに指定したjavaコマンドと異なるバージョンのjavacコマンドでコンパイルされています。アプリケーションの格納ディレクトリに指定したjavaコマンドと同一バージョンのjavacコマンドでアプリケーションをコンパイルし、アプリケーションを置き換えたあと、再度CORBAワークユニットを起動してください。

#### 確認方法

##### Windows32/64

CORBA-Javaアプリケーションにおいて、ファイルに標準エラー出力を割り当てている場合は、ファイル内にjava.lang.IncompatibleClassChangeErrorが出力されていないかを確認してください。

##### Solaris32/64 Linux32/64

カレントディレクトリ配下の標準エラー出力ファイル(stderr)を参照して、java.lang.IncompatibleClassChangeErrorが出力されていないかを確認してください。

## 12.3.2 ワークユニット起動コマンド無応答

以下にワークユニット起動コマンド無応答となる原因の可能性について説明します。

### ワークユニット出口プログラム(前処理)がハングアップ

ワークユニット起動時に呼び出されるワークユニット出口プログラム(前処理)が復帰していない場合、コマンドが無応答となります。

### モジュール不整合によるハングアップ Solaris32/64 Linux32/64

以下の条件の場合、ワークユニット起動時に参照するリンクライブラリの参照先を誤り、コマンドが無応答となります。

- C++言語で作成したアプリケーションを使用している。
- C言語/COBOLアプリケーション(COBOLはSolarisのみ使用可能)で作成したプロセスモードのアプリケーションで、サーバアプリケーション間連携を使用した3階層モデルを使用している。
- ワークユニット定義の「LIBRARY for Application」に、以下が指定されていない。

##### Solaris32/64

```
/opt/FSUNod/lib/nt
```

Linux32/64

```
/opt/FJSVod/lib/nt
```

この場合、ワークユニット定義で指定したカレントディレクトリのstderrに以下のメッセージが出力されて、ワークユニット起動コマンドがハングアップします。

```
libc internal error: _rmutex_unlock: rmutex not hold.
```

本現象が発生した場合は、ワークユニット起動コマンドプロセスとサーバアプリケーションプロセスをkillして、ハングアップ状態を解除してください。

ワークユニット定義の「LIBRARY for Application」に以下を指定し、再度ワークユニット起動コマンドを実行してください。

Solaris32/64

```
/opt/FSUNod/lib/nt
```

Linux32/64

```
/opt/FJSVod/lib/nt
```

### 12.3.3 ワークユニット停止コマンド無応答

以下にワークユニット停止コマンド無応答となる原因の可能性について説明します。

#### ワークユニット出口プログラム(後処理)がハングアップ

ワークユニット停止時に呼び出されるワークユニット出口プログラム(後処理)が復帰していない場合、コマンドが無応答となります。

#### アプリケーションエラーが発生し、ポップアップダイアログ出力 Windows32/64

トランザクションアプリケーション処理中にワークユニットを強制停止した場合、アプリケーションエラーのポップアップダイアログが出力されることがあります。これは、アプリケーションにリンクされているライブラリにおいて例外が発生しています。また、アプリケーション自身の矛盾が発生し、アプリケーションエラーのポップアップダイアログが出力された場合でも同様です。

本現象が発生した場合は、ポップアップダイアログを終了することによりワークユニットが強制停止されます。

#### ワークユニット運用中にアプリケーションの入れ替えを行った Solaris32/64 Linux32/64

ワークユニットを停止しないで、アプリケーションを入れ替えると、ハングアップすることがあります。アプリケーションを入れ替える場合は、必ずワークユニットを停止してから入れ替えてください。

### 12.3.4 ワークユニットが異常終了

ワークユニットが異常終了した場合、メッセージEXTP4400が出力され、ワークユニットが異常終了した原因がEXTP4400の前に出力されます。メッセージを参照して対処してください。

以下にワークユニットが異常終了となる原因の可能性について説明します。

## 連続異常終了回数に達した

アプリケーションの連続異常終了の回数が、ワークユニット定義の「Control Optionセクション」の「Application Retry Count」に設定された連続異常終了回数に達した場合、ワークユニットが異常終了します。この場合、メッセージEXTP4400の前に以下のメッセージが出力されます。

- EXTP4656
- EXTP4657

## 自動再起動が失敗した

アプリケーションが異常終了した、またはアプリケーションのタイムアウトによりアプリケーションが強制終了し、アプリケーションプロセスの自動再起動処理が失敗した場合、ワークユニットが異常終了します。この場合、メッセージEXTP4400の前に出力されたメッセージが自動再起動処理の失敗した原因を示しています。出力されたメッセージに対する対処を行ってください。

## ワークユニット停止コマンドを使用せずにアプリケーションプロセスを停止した

ワークユニット停止コマンドを使用しないで、アプリケーションプロセスのプロセス自体を強制的に停止すると、ワークユニットが異常終了します。アプリケーションプロセスを停止する場合は、必ずワークユニット停止コマンドを使用してください。

## ワークユニット配下で動作するアプリケーションプロセスの異常終了の原因を特定できない Solaris32/64

### Linux32/64

ワークユニット配下で動作するアプリケーションプロセスがメッセージEXTP4703,EXTP4371,EXTP4656,EXTP4354を出力し、異常終了したが原因を特定できない場合、以下のログファイルを参照することにより、アプリケーションプロセスがexitにより終了したか、シグナルにより終了したかを特定できます。

### デフォルトシステムのログファイル

#### Solaris32/64

```
/var/opt/FJSVisas/system/default/FSUNextp/td001/tmp/is/proc_term_status.log
```

#### Linux32/64

```
/var/opt/FJSVisas/system/default/FJSVextp/td001/tmp/is/proc_term_status.log
```

### 拡張システムのログファイル

#### Solaris32

```
/var/opt/FJSVisas/system/[システム名]/FSUNextp/[システム名]/tmp/is/proc_term_status.log
```

### ログファイルの出力形式

```
-----  
SYSTEM NAME   :default  
OUTPUT TIME   :10/11/2004 14:22:51  
-----  
              TIME           PID EXIT SIGNAL CORE  
10/11/2004 14:22:51 4009     0     -     NO
```

10/11/2004 14:36:54	4018	-	9	NO
10/11/2004 14:54:28	4023	-	11	YES

#### SYSTEM NAME

Interstageのシステム名。

#### OUTPUT TIME

Interstage起動時または本ログファイルを作成した日時。

#### TIME

プロセスが終了した日時。

#### PID

終了したプロセスのプロセスID。

#### EXIT

プロセスが正常終了した場合、終了したプロセスの返り値を表示します。返り値とは、`exit()`や`_exit()`の引数として設定された値、および`main`プログラムの`return`文で引数として与えられた値(0~255)。負の値および256以上の値は、正しく表示されません。

#### SIGNAL

プロセスがシグナルにより終了した場合、プロセスを終了させたシグナル番号を表示します。

#### CORE

プロセスのcore出力の有無。YESの場合、ワークユニットのカレントディレクトリにcoreが出力されます。

上記のログファイルの場合、プロセスID「4009」のプロセスは正常終了しています。プロセスID「4018」および「4023」のプロセスは、シグナル番号よりそれぞれSIGKILL、SIGSEGVで終了していると特定できます。

### 注意

- ・ 本ログファイルは、プロセス終了時に必ず出力されるため、ワークユニットが正常終了した場合にも出力されます。
- ・ 本ログファイルは、ファイルサイズが1Mバイトを超えると新しいファイルに切り替わります。使用していたログファイルは「`proc_term_status.old`」として保存され、次にログファイルのサイズが1Mバイトを超えた際に削除されます。

## 12.3.5 サーバアプリケーション無応答

以下にサーバアプリケーションが無応答となる原因の可能性について説明します。

### サーバアプリケーションがハングアップ

サーバアプリケーションプロセスのCPU使用率が高くなっている場合、サーバアプリケーション内でループしています。ループ原因を調査してください。これは使用者が作成したプログラムであるため、使用者が調査する必要があります。

### サーバアプリケーションが監視時間超過で異常終了した

サーバアプリケーション内でループまたはサーバアプリケーションの処理に時間がかかっています。ループまたは処理に時間がかかる原因を調査してください。これは使用者が作成したプログラムであるため、使用者が調査する必要があります。

## ワークユニット運用中にアプリケーションの入れ替えを行った Solaris32/64 Linux32/64

ワークユニットを停止しないで、アプリケーションを入れ替えると、ハングアップすることがあります。アプリケーションを入れ替える場合、必ずワークユニットを停止してから入れ替えてください。

## 12.3.6 コンソールからログインできない Solaris32/64

以下にコンソールからログインできなくなる原因の可能性について説明します。

### RCプロシージャからInterstage起動およびワークユニット起動を行うとコンソールログインができなくなる場合

RCプロシージャにInterstage起動およびワークユニット起動を行うスクリプトを登録して、サーバ・ブート時に起動スクリプトからInterstage起動およびワークユニット起動を行うと、コンソール端末からのログイン時に以下のメッセージが出力され、ログインできなくなることがあります。

```
console login: root
Warning - ttymon cannot allocate controlling tty on "/dev/console", there may be another session active on this port.
```

この場合、起動スクリプトにコンソールへのメッセージ出力を抑制する環境変数(IS\_ISV\_CONSOLE\_OUTPUT=OFF)を設定するか、以下の処理を追加してください。

なお、コンソールへのメッセージ出力を抑制することで、コンソールへのメッセージ出力が遅延するトラブルなども防止できるため、コンソールへのメッセージ出力を抑制する方法での対処を推奨します。コンソールへのメッセージ出力を抑制する方法は、「RCプロシージャからInterstageおよびワークユニットを起動するとコンソールログインができなくなった場合」を参照してください。

1. Interstageおよびワークユニットの起動前処理

```
#!/bin/sh
/usr/lib/saf/ttymon -g -h -T sun -d /dev/console -l console -m ldterm,ttcompat &
```

2. Interstageおよびワークユニットの起動後処理

```
#!/bin/sh
pid=`/usr/bin/ps -e | grep -w ttymon | grep -w console | /usr/bin/sed -e 's/^ *//' -e 's/ .*//'`
kill $pid
```

### 例

#### Interstageおよびワークユニットの起動スクリプト

```
#!/bin/sh
/usr/lib/saf/ttymon -g -h -T sun -d /dev/console -l console -m ldterm,ttcompat &

isstart
isstartwu TDSAMPLE1

#!/bin/sh
pid=`/usr/bin/ps -e | grep -w ttymon | grep -w console | /usr/bin/sed -e 's/^ *//' -e 's/ .*//'`
kill $pid
```

## 12.3.7 コマンドの実行が失敗する Windows32/64

以下にコマンドの実行が失敗する原因の可能性について説明します。

## InterstageがNTFSでフォーマットされたドライブでEveryoneグループにフルコントロールのアクセス許可が設定されていないフォルダにインストールされている

例えば、以下のようなフォルダが該当します。

- NTFSでフォーマットされたシステムドライブの“Program Files”フォルダ
- Microsoft(R) Windows(R) XPのシステムドライブのフォルダ
- Microsoft Windows Server 2003のシステムドライブのフォルダ

上記に該当する場合で、一般ユーザ(Usersグループのみに所属するユーザ)またはWindows Server(R) 2008/Windows Server(R) 2012においてビルトインアカウントの「Administrator」以外のユーザによるコマンド操作を行う場合は、Interstageインストールフォルダのプロパティで[セキュリティ]タブを表示し、該当ユーザに対するアクセス許可をフルコントロールに設定してください。

なお、Microsoft(R) Windows(R) XPでフォルダのプロパティに[セキュリティ]タブが表示されない場合は、次の手順を実行することで表示されます。

1. [スタート]—[コントロールパネル]から「フォルダオプション」を起動します。
2. [表示]タブをクリックし、「簡易ファイルの共有を使用にする(推奨)」のチェックを解除して「OK」をクリックします。

## 12.3.8 Interstageに対する操作が無応答となる Solaris32/64 Linux32/64

以下にInterstageに対する操作が無応答となる原因の可能性について説明します。

### Interstageのプロセスおよびワークユニット配下で動作するプロセスに対して、シグナル9番(SIGKILL)を送信した。

Interstageのプロセスおよびワークユニット配下で動作するプロセスに対して、シグナル9番(SIGKILL)を送信した場合、Interstageに対する操作が無応答となることがあります。この場合、マシンを再起動して環境を復旧してください。

## 12.4 性能監視ツール使用時の異常発生時の対処

ここでは、性能監視ツールの以下の機能を使用した場合のトラブル事例およびその対処方法について説明します。

- 性能ログファイルへのログ出力機能
- ネットワーク管理マネージャによる性能情報のリアルタイム監視機能(MIBによる監視) Windows32/64 Solaris32/64

### 12.4.1 性能情報が採取できない

性能監視ツールにより性能情報の採取操作を行っているにもかかわらず、性能情報が採取されない場合は、以下の原因が考えられます。

#### 参照

性能監視ツールの使用方法の詳細については、「運用ガイド(基本編)」の「性能監視」を参照してください。

### 性能監視ツールの環境が作成されていない

性能監視環境が作成されていない可能性があります。  
ismakeenvコマンドを実行し、性能監視環境を作成してください。

## ispstartコマンドで、オブジェクトを指定していない

性能監視の対象とするオブジェクトが指定されていない可能性があります。  
ispstartコマンドを実行し、性能監視の対象とするオブジェクトを指定してください。  
ispstatusコマンドにより、性能監視対象のオブジェクト名を確認できます。

## ispstartコマンドでトランザクションアプリケーションまたはラッパーのオブジェクトを指定していない Windows32/64

Solaris32/64

性能監視ツールで監視できるオブジェクトは、トランザクションアプリケーションまたはラッパーのオブジェクトだけです。  
ispstartコマンドでは、トランザクションアプリケーションまたはラッパーのオブジェクトを指定してください。  
ispstatusコマンドにより、性能監視対象のオブジェクト名を確認できます。

## 監視対象としているオブジェクトが閉塞状態である

性能監視ツールで監視しているオブジェクトが閉塞されている可能性があります。  
isinfobjコマンドを使用し、オブジェクトの状態を確認してください。オブジェクトの状態が閉塞中である場合は、サーバアプリケーションで処理が行われないため、性能情報は採取されません。

## ワークユニットが起動していない

性能監視ツールで監視しているオブジェクトが含まれるワークユニットが起動していない可能性があります。  
islistwuコマンドを使用し、ワークユニットの運用状態を確認してください。ワークユニットの運用状態が起動済み状態でない場合は、サーバアプリケーションで処理が行われないため、性能情報は採取されません。

## サーバアプリケーションが異常終了している

性能監視ツールで監視しているオブジェクトのサーバアプリケーションが異常終了している可能性があります。  
サーバアプリケーションが異常終了している場合は、異常事象の発生を通知する異常メッセージが出力されますので、異常メッセージが出力されていないかを確認してください。

## クライアントアプリケーションへの復帰値が正しく設定されていない

トランザクションアプリケーションのサーバアプリケーションが正常に復帰していない可能性があります。この場合、性能情報は採取されません。  
トランザクションアプリケーションでは、クライアントアプリケーションへの復帰値として0～10000の値が復帰した場合に、正常終了と判断します。

## 性能監視環境の共用メモリが不足している

性能監視環境の共用メモリが不足している可能性があります。  
性能監視環境の共用メモリが不足していないかを確認してください。

## ispmakeenvコマンドの実行前にワークユニットが起動されている

ispmakeenvコマンドの発行前に、ワークユニットが起動されている可能性があります。  
ispmakeenvコマンド発行後に、ワークユニットを起動してください。

## 性能ログファイルを削除または移動した

性能ログファイルの削除／移動を行った可能性があります。  
性能ログファイルを削除した場合は、採取した性能情報はなくなります。  
性能ログファイルを移動した場合は、移動先の性能ログファイルに対してispreportコマンドを実行してください。

## 12.4.2 性能情報のリアルタイム監視ができない Windows32/64 Solaris32/64

性能情報のリアルタイム監視ができない場合は、「12.4.1 性能情報が採取できない」に記載した原因か、以下の原因が考えられます。

### 参照

性能監視ツールの使用方法の詳細については、「運用ガイド(基本編)」の「性能監視」を参照してください。

### SNMPサービスが起動されていない

SNMPサービスが起動されていない可能性があります。  
この場合、SNMPサービスを起動してください。  
Solaris 10以降の場合、Solstice Enterprise Agentsソフトウェア(SEA)を使用しているかを確認してください。

### ispsetagtコマンドが実行されていない Windows32/64

ispsetagtコマンドを実行し、SNMPサービスに対して性能監視ツールが登録されていない可能性があります。  
この場合、ispsetagtコマンドを実行し、SNMPサービスに対して性能監視ツールを登録してください。

### 性能監視ツールのリアルタイム監視準備操作が行われていない Solaris32/64

性能監視ツールのリアルタイム監視のための準備操作を行ってください。

### 性能監視ツールで異常が発生している Solaris32/64

性能監視ツールで異常が発生している可能性があります。システムログを参照し、異常メッセージが出力されていないかを確認してください。異常メッセージが出力されている場合は、そのメッセージに対する対処を行ってください。

EE

## 12.5 Interstage運用APIの異常発生時の対処 Windows32 Solaris32

Linux32

ここでは、Interstage運用APIを使用した場合のトラブル事例およびその対処について説明します。

### 注意

- Windows32 Solaris32 Linux32  
Interstage運用APIは、カスタムインストールで以下のどちらかの機能をインストールした場合に使用できます。
  - マルチ言語サービス
  - J2EE互換
- Windows64 Solaris64 Linux64  
Interstage運用APIは、使用できません。

### 参照

Interstage運用APIの使用方法については、「OLTPサーバ運用ガイド」の「Interstage運用API」を参照してください。

## Interstage運用APIを使用したプログラムのコンパイル・リンクに失敗する

Interstage運用APIは、C言語インタフェースだけサポートしています。  
Interstage運用APIを使用するプログラムが、C言語以外の言語で、記述されている可能性があります。  
プログラムを、C言語以外の言語で記述している場合は、C言語で記述してください。

## Interstage運用APIが正しく動作しない

Interstage運用APIが、マルチスレッドで動作するプロセス上で、複数のスレッドから同時に発行されている可能性があります。  
Interstage運用APIは、複数スレッドからの同時発行に対応していません。同時に発行されないような制御を行うように、プログラムを修正してください。

## 12.6 EXTPのアンインストールが失敗する Windows32/64

Interstageのアンインストール時に、メッセージEXTP0698が出力され失敗する場合があります。  
この場合、いったんシステムを再起動し、tdunsetupを行った後、再度Interstageをアンインストールしてください。



## 12.7 セッション情報管理機能を使用した場合の異常への対処

### リスナの異常

セッション情報管理機能を使用し、事象通知リスナを作成してセッション情報域のタイムアウトを監視している環境で、セッション情報域のタイムアウトが発生してもセッション情報管理の事象通知リスナにタイムアウト通知が行われない場合、事象通知リスナの実装時に必要なライブラリをリンクしていないか、またはインプリメンテーションリポジトリへの登録に誤りがある可能性があります。  
この場合、システムログファイルにメッセージtd14003が以下のエラー情報を付加して出力されます。

#### エラー情報1

```
ISTD_SMO_LISTENER2_timeout  
ISTD_SMO_LISTENER_timeout
```

#### エラー情報2

```
IDL:CORBA/StExcep/BAD_OPERATION:1.0
```

上記の条件に一致する場合、事象通知リスナの作成時に正しいライブラリをリンクしているかを確認してください。  
リンクライブラリについては、「アプリケーション作成ガイド (コンポーネントトランザクションサービス編)」の「セッション情報管理機能を使用したトランザクションアプリケーションの作成」を参照してください。  
ライブラリのリンクに誤りがあった場合は、正しいライブラリをリンクして事象通知リスナを再作成してください。

データベース連携サービスで異常が発生した場合の対処方法について、以下の内容を説明します。

- ・ アプリケーション運用中の異常
- ・ OTSシステムが動作するマシンのシステムダウン
- ・ リソース管理プログラムが動作するマシンのシステムダウン
- ・ OTSシステムの異常
- ・ リソース管理プログラムの異常
- ・ OTSシステムとリソース管理プログラムの通信異常
- ・ OTSシステムとリソース管理プログラムの起動時または停止時の異常
- ・ Oracleを使用した場合の異常
- ・ OTS動作環境の異常
- ・ Interstage管理コンソール使用時の異常

## 13.1 アプリケーション運用中の異常

 Windows32/64 Solaris32 Linux32/64

アプリケーションの運用中に異常が発生した場合の対処方法について、以下の内容を説明します。

- ・ クライアントアプリケーションの異常
- ・ サーバアプリケーションの異常
- ・ サーバアプリケーションの処理結果異常
- ・ アプリケーション連携中の通信回線異常
- ・ メソッド呼出し時の異常
- ・ クライアントアプリケーションでUserTransactionのlookupが失敗する

### クライアントアプリケーションの異常

クライアントアプリケーションが異常終了した場合やループ状態になった場合は、原因を取り除いたあと、クライアントアプリケーションを再起動してください。

なお、グローバルトランザクション連携時に、クライアントアプリケーションが処理していたトランザクションは、トランザクションのタイムアウト時間の経過後、rollbackが指示されます。

ただし、トランザクションのタイムアウトに「0」（タイムアウト:無効）を設定し、以下の条件に該当する場合は、システム管理者がotstranlistコマンドで未解決(処理中)のトランザクションを終了してください。

- ・ 完了処理(commit/rollback)を発行しないで終了した場合
- ・ クライアントアプリケーションが異常終了した場合
- ・ クライアントアプリケーションがループ状態になった場合
- ・ クライアントアプリケーションが動作するマシンの回線が切断された場合
- ・ クライアントアプリケーションが動作するマシンがシステムダウンした場合

## 注意

トランザクションのタイムアウトは、クライアントアプリケーションのCurrentインタフェースのbegin発行前に、Currentインタフェースのset\_timeoutを発行して指定します。set\_timeoutを発行しない場合は、データベース連携サービスの環境定義(config)のTRAN\_TIME\_OUTの設定値が有効となります。

### サーバアプリケーションの異常

サーバアプリケーションが異常終了した場合やループ状態になった場合は、原因を取り除いたあと、サーバアプリケーションを再起動してください。

なお、グローバルトランザクション連携時に、サーバアプリケーションが処理していたトランザクションは、トランザクションのタイムアウト時間の経過後、rollbackが指示されます。

ただし、トランザクションのタイムアウトに「0」（タイムアウト:無効）を設定し、以下の条件に該当する場合は、システム管理者がotstranlistコマンドで未解決のトランザクションを終了してください。

#### サーバアプリケーションが異常終了した場合

クライアントアプリケーションの復帰値に「10004」が通知され、システムログメッセージファイルにエラーメッセージEXTP4656が出力された場合は、トランザクションアプリケーションの異常によりアプリケーションプロセスが異常終了しています。これは、アプリケーションの不具合が原因です。異常終了した原因を調査してください。

#### サーバアプリケーションがループ状態になった場合

サーバアプリケーションがループ状態になった場合は、クライアントアプリケーションにタイムアウトなどが検出されたエラーが通知されます。エラーが通知されたクライアントアプリケーションは、rollbackを発行してトランザクションを終了してください。

### サーバアプリケーションの処理結果異常

サーバアプリケーションの処理において期待した処理結果が得られない場合の対処方法を以下に示します。

#### Oracleの環境設定ミスによる処理結果異常

環境変数ORACLE\_SIDの設定に誤りがある場合、正しいデータベースにアクセスできずに、処理結果異常となります。正しい値を設定してください。

#### Oracleの環境構築ミスによる処理結果異常

Oracleの環境が正しく構築されていない場合、Interstageとの接続にエラーが発生し、処理結果異常となります。正しい値を設定してください。「13.7 Oracle使用時の異常」を参考に対処してください。

#### トランザクション完了命令の発行ミスによる処理結果異常

データベース連携サービスを使用する場合は、必ずCurrentインタフェース/JTAを使用してトランザクションを制御する必要があります。したがって、アプリケーションにおいてSQL文で単独にCOMMIT命令を呼び出した場合などは、処理結果異常となります。トランザクションを制御する場合は、Currentインタフェース/JTAを使用してください。

### アプリケーション連携中の通信回線異常

アプリケーション連携中に通信回線異常が発生した場合の対処方法を以下に示します。

#### 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間の通信異常

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間で通信異常が発生した場合、クライアントアプリケーションからオペレーションを呼び出した復帰値にエラーが通知されます。

以下の対処を行ってください。

- クライアントアプリケーションでbeginを発行している場合は、rollbackを発行してトランザクションを終了させてください。通信異常が通知されない場合、トランザクションは、トランザクションのタイムアウト時間の経過後にrollbackされます。通信回線の回復後、クライアントアプリケーションを再起動してください。
- クライアントアプリケーションでbeginを発行していない場合は、通信回線の回復後、クライアントアプリケーションを再起動してください。

#### クライアントアプリケーションとOTSシステム間の通信異常

クライアントアプリケーションが発行した完了処理 (commit / rollback) において通信異常で終了した場合、トランザクションのタイムアウト時間の経過後、トランザクションがrollbackされます。通信回線の回復後、クライアントアプリケーションを再起動してください。

### メソッド呼出し時の異常

アプリケーションからメソッド呼出し時に異常が発生した場合の対処方法を以下に示します。

#### beginメソッド発行時に、NO\_IMPLEMENT例外が発生する場合

OTSシステムが起動されていない可能性があります。isstatコマンド / otsaliveコマンドでOTSシステムが起動されているかを確認してください。停止している場合は、isstartコマンド / otsstartコマンドでOTSを再起動したあと、再度beginメソッドを発行してください。

#### beginメソッド発行時に、NO\_RESOURCES例外が発生する場合

OTSシステムに対して、Interstage動作環境定義の「OTS maximum Transaction」の設定値 / セットアップ情報ファイルの「TRANMAX」の設定値を超えた要求が発行されています。

### クライアントアプリケーションでUserTransactionのlookupが失敗する

クライアントアプリケーションにおいてUserTransactionをlookupした際に以下の例外が発生する場合は、クライアントアプリケーション起動時にプロパティを指定してください。

```
java.lang.IllegalStateException: has not set GlobalTransactionMode
```

プロパティに指定する文字列を以下に示します。大文字 / 小文字は、区別されます。

```
-Dcom.fujitsu.ObjectDirector.CORBA.GlobalTransactionMode=True
```

また、使用しているJDK/JREがInterstage Application Serverに同梱されているJDK/JREであるかを確認してください。



#### 参照

他のデータベース連携サービスの例外については、「メッセージ集」の「データベース連携サービスから通知される例外情報」を参照してください。

## 13.2 アプリケーション作成時の異常 Windows32/64 Solaris32 Linux32/64

アプリケーション作成時にエラーが発生した場合は、「第11章 CORBAサービス使用時の異常」-「11.7 アプリケーションコンパイル時の異常」を参照して対処してください。

## 13.3 マシンのシステムダウン Windows32/64 Solaris32 Linux32/64

マシンのシステムダウンの対処方法について、以下の内容を説明します。

- OTSシステムが動作するマシンのシステムダウン

- ・ リソース管理プログラムが動作するマシンのシステムダウン

### OTSシステムが動作するマシンのシステムダウン

OTSシステムが動作するマシンでシステムダウン／電源断が発生した場合は、システムを再起動したあと、以下の対処を行ってください。本対処により、未解決のトランザクションが保証されます。

1. データベースシステムを使用している場合は、データベースシステムを再起動します。再起動時、データベースシステムのリカバリ処理が行われます。
2. isstartコマンドでInterstageを再起動します。
3. リソース管理プログラムを使用している場合は、リソース管理プログラムを再起動します。2フェーズコミットの処理中にシステムダウンが発生した場合は、再起動時にリソース管理プログラムのリカバリ処理が行われます。
4. サーバアプリケーションを使用している場合は、サーバアプリケーションを再起動します。
5. 他システム上でサーバアプリケーションを動作させていた場合は、他システム上のサーバアプリケーションを再起動します。サーバアプリケーション終了時、サーバアプリケーションがデータベースを排他していた場合は、解放されます。
6. 他システム上でリソース管理プログラムを使用していた場合は、リソース管理プログラムの再起動を促すメッセージが通知される可能性があります。メッセージが通知された場合は、リソース管理プログラムを再起動します。

### リソース管理プログラムが動作するマシンのシステムダウン

リソース管理プログラムが動作するマシンでシステムダウン／電源断が発生した場合は、システムを再起動したあと、以下の対処を行ってください。

1. データベースシステムを再起動します。再起動時、データベースシステムのリカバリ処理が行われます。
2. isstartコマンドでInterstageを再起動します。
3. リソース管理プログラムを再起動します。2フェーズコミットの処理中にシステムダウンした場合は、再起動時、リソース管理プログラムのリカバリ処理が行われます。
4. サーバアプリケーションを再起動します。

## 13.4 OTSシステムの異常 Windows32/64 Solaris32 Linux32/64

OTSシステムの異常とその対処方法について、以下の内容を説明します。

- ・ OTSシステムの異常終了／強制終了
- ・ OTSのシステムログファイルでI/Oエラーが発生した場合

### OTSシステムの異常終了／強制終了

OTSシステムが異常終了した場合、またはOTSシステムを強制終了させた場合、以下の対処を行ってください。

1. すべてのリソース管理プログラムおよびサーバアプリケーションを停止します。
2. isstartコマンドでInterstageを再起動します。
3. すべてのリソース管理プログラムおよびサーバアプリケーションを再起動します。

### OTSのシステムログファイルでI/Oエラーが発生した場合

OTSシステムが使用するシステムログファイルでディスク破壊などによるI/Oエラーが発生した場合は、以下の対処を行ってください。

1. すべてのリソース管理プログラムおよびサーバアプリケーションを終了します。
2. I/Oエラーの原因を取り除いたあと、isstartコマンドでInterstageを再起動します。
3. すべてのリソース管理プログラムおよびサーバアプリケーションを再起動します。

## 13.5 リソース管理プログラムの異常 Windows32/64 Solaris32 Linux32/64

リソース管理プログラムの異常とその対処方法について、以下の内容を説明します。

- リソース管理プログラムで異常が発生した場合
- データベースで異常が発生した場合
- インダウト状態が発生した場合

### リソース管理プログラムで異常が発生した場合

リソース管理プログラムが異常終了した場合、またはリソース管理システムを強制終了させた場合は、リソース管理プログラムが動作しているマシンのコンソールにメッセージが通知されます。リソース管理プログラムを再起動してください。なお、完了処理(commit/rollback)が失敗した場合に、メッセージが通知されることがあります。この場合、データベースがロックされた状態になっているため、サーバアプリケーションを再起動してください。

### データベースで異常が発生した場合

データベースが異常終了した場合、またはデータベースを強制終了した場合は、リソース管理プログラムが動作しているマシンのコンソールにメッセージが通知されます。

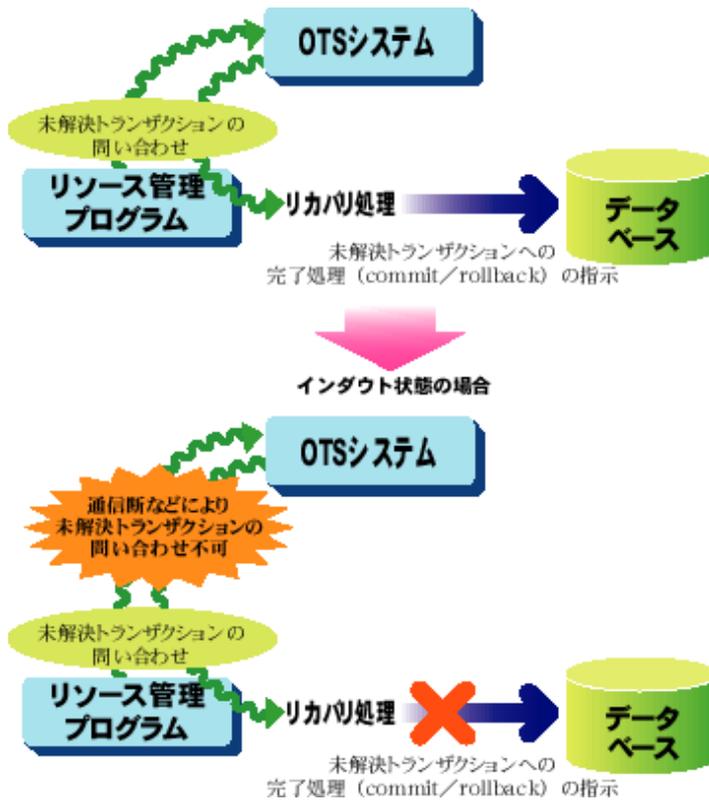
リソース管理プログラムの再起動時にデータベースのリカバリ処理が行われるため、リソース管理プログラムを再起動してください。

1. 異常が発生したデータベースに関連するリソース管理プログラムを終了します。
2. サーバアプリケーションを終了します。
3. データベース異常の原因を取り除いたあと、データベースを再起動します。
4. リソース管理プログラムを再起動します。
5. サーバアプリケーションを再起動します。

### インダウト状態が発生した場合

インダウト状態とは、通信断などにより、本来リソース管理プログラムの再起動時に行われるデータベースのリカバリ処理が行われない状態のことをいいます。

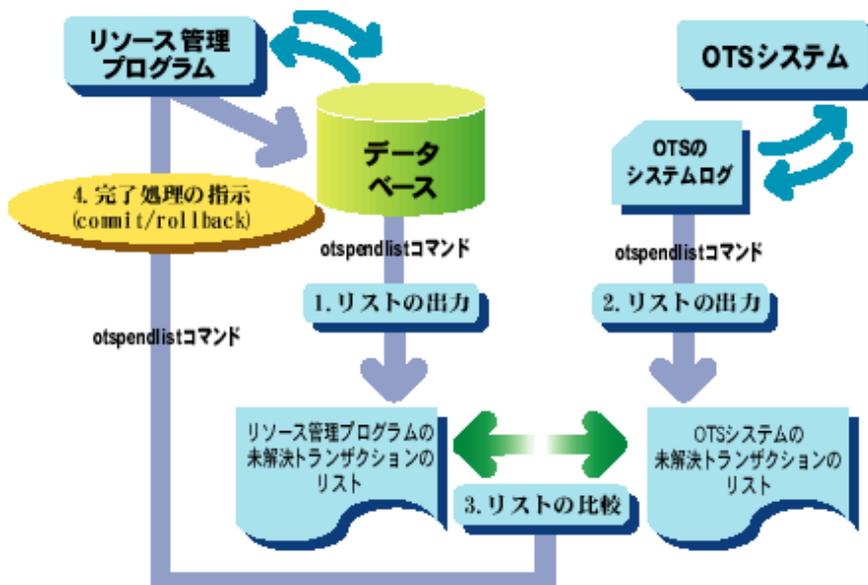
インダウト状態の図を以下に示します。



インダウト状態が発生した場合は、メッセージが通知されます。以下の対処を行ってください。

### 注意

otspendlistコマンドでインダウト状態を解決しますが、otspendlistコマンドはシステム管理者がトランザクションの終了を任意に選択できるため、データベースに不整合が発生する可能性があります。必ず、OTSシステムとリソース管理プログラムの未解決のトランザクションリストを比較して、未解決トランザクションの処理を指示してください。



1. otspendlistコマンドでリソース管理プログラムの未解決トランザクションのリストを出力します。

```
otspendlist {-s | -r リソース定義ファイル名}
```

2. otspendlistコマンドでOTSシステムの未解決トランザクションのリストを出力します。
3. システム管理者は、1.および2.で出力したリストを比較し、未解決トランザクションの完了処理(commit/rollback)を決定します。
4. otspendlistコマンドで未解決トランザクションに完了処理(commit/rollback)を指示します。

## 13.6 OTSシステムとリソース管理プログラムの異常 Windows32/64 Solaris32

Linux32/64

OTSシステムとリソース管理プログラムの異常とその対処方法について、以下の内容を説明します。

- ・ トランザクション完了処理(commit/rollback)時に通信異常が発生した場合
- ・ データベース連携サービス停止時に資源不足が発生した場合
- ・ 「otsstartsc -j」が失敗する

### トランザクション完了処理(commit/rollback)時に通信異常が発生した場合

OTSシステムとリソース管理プログラム間の通信エラーにより、クライアントアプリケーションが発行したトランザクション完了処理(commit/rollback)で異常終了した場合、メッセージが通知されます。データベースがロックされているため、サーバアプリケーションを再起動してください。

また、本現象が発生した場合は、データベースの一貫性が失われている可能性があります。リソース管理プログラムの再起動を促すメッセージが通知されるため、通信回線の回復後、リソース管理プログラムを再起動してリカバリ処理を行ってください。

### データベース連携サービス停止時に資源不足が発生した場合 Windows32/64

データベース連携サービス停止時に資源不足が発生した場合は、データベース連携サービスの環境を正しく削除できません。データベース連携サービスを再起動する場合は、以下の対処を行ってください。

1. コントロールパネルのサービスでObjectTransactionServiceを停止します。
2. 以下のファイルで資源不足したパラメタを見直します。

```
C:\%Interstage%\ots\etc\ots.ini
```

3. コントロールパネルのサービスでObjectTransactionServiceを起動します。

### 「otsstartsc -j」が失敗する

「otsstartsc -j」実行時に以下のメッセージが出力されてリソース管理プログラムの起動に失敗する場合は、環境変数OD\_IMPLIDが設定されています。環境変数OD\_IMPLIDを削除した状態で再度実行してください。

```
OTS:エラー:ots9740: ObjectDirectorが利用できる状態ではありません。:EXCEPTION=org.omg.CORBA.OBJ_ADAPTER  
CODE=0x00000000 0x00000002
```

## 13.7 Oracle使用時の異常 Windows32/64 Solaris32 Linux32/64

Oracle使用時の異常とその対処方法について、以下の内容を説明します。

- ・ インダウトランザクションの手動ロールバック機能を使用した場合

- JTSとEJBを使用するデータベースを作成する場合
- rollback処理時に、XAER\_NOTAが発生した場合
- Oracleアクセス時に、メッセージが出力される場合
- Oracleと連携時に、OTS用リソース管理プログラム／サーバアプリケーションの起動が遅延する場合

### インダウトランザクションの手動ロールバック機能を使用した場合

Oracleのインダウトランザクションの手動ロールバック機能を使用して、ROLLBACK FORCEにより強制的にロールバックさせたあと、再度リソース管理プログラムを起動すると、リカバリ時に以下のエラーメッセージが表示され、未解決ランザクションが残ります。

```
OTS:エラー:ots9666: システムメモリまたはシステムリソースが不足しました
```

また、otspendlistコマンドでcommitした場合は、以下のエラーメッセージが表示されます。

```
OTS:エラー:ots9348: commit処理でエラーが発生しました
```

本現象が発生した場合、otspendlistコマンドでrollbackさせることによりインダウト状態を解除してください。

EE

### JTSとEJBを使用するデータベースを作成する場合

EJBがグローバルランザクションを使用する場合は、データベース作成時にOracle JVMが有効となるように設定してください。データベースの設定方法については、Oracleのマニュアルを参照してください。

### rollback処理時に、XAER\_NOTAが発生した場合

2つのリソースに対するrollback処理時に、一方のリソースは成功し、もう一方のリソースではXAER\_NOTAが発生した場合は、Oracleが正しくインストールされていない、またはデータベースが正しく作成されていない可能性があります。データベース再作成時にOracle JVMが有効となるように設定してください。

### Oracleアクセス時に、メッセージが出力される場合

Oracleアクセス時に以下のメッセージが出力される場合は、OracleのSIDが複数存在するなど、ORACLE\_SIDの設定に誤りがある可能性があります。ORACLE\_SIDの設定を確認し、正しいORACLE\_SIDを設定してください。

```
java.sql.Exception: ORA-12560: TNS: プロトコル・アダプタ・エラー
```

### Oracleと連携時に、OTS用リソース管理プログラム／サーバアプリケーションの起動が遅延する場合

Windows32/64

Oracle連携時にOTS用リソース管理プログラム／サーバアプリケーションの起動が遅延する場合は、OracleのWindows(R)認証機能の処理が遅延している可能性があります。sqlnet.oraの以下の設定をコメントアウトするなど、Windows(R)認証機能が無効となるように設定してください。

```
SQLNET.AUTHENTICATION_SERVICES = (NTS)
```

なお、Windows(R)認証機能を使用する場合は、「NetBIOS over TCP/IP」を無効にすることで起動性能が改善されることがあります。

## 13.8 OTS動作環境の異常 Windows32/64 Solaris32 Linux32/64

OTSの動作環境に異常が発生した場合の環境の初期化方法について、以下に説明します。

## 注意

本操作は、手動でOTSの動作環境の初期化を行います。十分に注意して操作してください。

1. サイトからサーバグループに参加しているサーバを離脱します。マルチサーバ環境において、対象サーバがサイトに参加している場合は、サイトから削除してください。
2. 以下のOTS動作環境ファイルが残っている場合は、OTS動作環境ファイルを削除します。

### Windows32/64

```
C:\Interstage\ots\etc\setup.inf
C:\Interstage\ots\etc\sysconfig
C:\Interstage\ots\etc\sysconfig.ser
C:\Interstage\ots\etc\rmppconfig.ser
```

### Solaris32

```
/opt/FSUNots/etc/otssetup.inf
/opt/FSUNots/etc/sysconfig
/opt/FSUNots/etc/sysconfig.ser
/opt/FSUNots/etc/rmppconfig.ser
```

### Linux32/64

```
/opt/FJSVots/etc/otssetup.inf
/opt/FJSVots/etc/sysconfig
/opt/FJSVots/etc/sysconfig.ser
/opt/FJSVots/etc/rmppconfig.ser
```

3. Interstage管理コンソール/セットアップ情報ファイル/Interstage動作環境定義に指定したトランザクションログファイルが残っている場合は、トランザクションログファイルを削除します。
4. データベース連携サービスが登録した以下のネーミングサービス登録情報を削除します。

— CosTransactions::RecoveryCoordinator

— JTSRMP:xxxxxx (注)

注) xxxxxxは、環境により異なります。odlistnsコマンドで登録情報を確認してください。

```
# odlistns
Name (Type)      IntfID
JTSRMP:220A7D9D (o) IDL:com/fujitsu/interstage/ots/jts/FjtsResourceFactory:1.0
CosTransactions::RecoveryCoordinator (o) IDL:CosTransactions/RecoveryCoordinator:1.0
# OD_or_adm -d -n CosTransactions::RecoveryCoordinator
# OD_or_adm -d -n JTSRMP:220A7D9D
```

5. データベース連携サービスが登録したインプリメンテーションリポジトリ情報を削除します。

```
# OD_impl_inst -d -r IDL:CosTransactions/RecoveryCoordinator:1.0
# OD_impl_inst -d -r IDL:CosTransactions/TransactionFactory:1.0
# OD_impl_inst -d -r OTS_JTSRMP
```

6. データベース連携サービスが登録したイニシャルサービス情報を削除します。

```
# OD_or_adm -d -i TransactionService
```

## 13.9 Interstage管理コンソール使用時の異常

Windows32/64 Solaris32

### Linux32/64

### **Interstage管理コンソールの[トランザクションサービスの詳細設定]で変更した定義値が反映されない**

[トランザクションサービスの詳細設定]で定義値を変更した場合、イベントログ／システムログに出力される情報メッセージots9139により、トランザクションサービス起動後に設定した定義値が有効になっているかを確認してください。定義値が有効になっていない場合は、Interstageを再起動してください。

## 第14章 イベントサービス運用時の異常

イベントサービスの運用でトラブルが発生した場合の対処方法について説明します。

### 14.1 コマンド実行時の異常

イベントサービスでコマンド実行時に異常が発生した場合の対処方法について説明します。

#### ユニットの作成に失敗する場合

特定のユニットが、ユニットの作成(esmkunitコマンドの実行)に失敗する場合は、以下の手順で確認してください。

##### ■ 対処1

ユニット定義ファイル(defファイル)のtrandir/sysdir/userdirに指定されているディレクトリが、他のユニット定義ファイルの指定内容と重複していないかを確認します。

該当する場合は、trandir/sysdir/userdirに指定したディレクトリを修正後に、再度esmkunitコマンドを実行してください。

##### ■ 対処2

ユニット定義ファイル(defファイル)のtrandir/sysdir/userdirに指定されているディレクトリ配下に、以下のファイル(以前作成された古いファイル)がないかを確認します。なお、パッケージの再インストールを行う場合などに発生することがあります。

- esmsgfile0
- essysmsgfile
- essysqfile
- estranfile
- esuserqfile

該当する場合は、trandir/sysdir/userdirに指定したディレクトリ配下の上記のファイルを削除後に、再度esmkunitコマンドを実行してください。

##### ■ 対処3

対処1および対処2に該当しない場合は、作成に失敗するユニットを、esrmunitコマンド(このとき-unitオプションでユニットを指定)で削除後、再度esmkunitコマンドでユニットを作成してください。

この場合、esrmunitコマンド実行時に「ユニットがセットアップされていません」という内容のエラーメッセージが出力されることがありますが、問題ありません。その後、esmkunitコマンドを実行してください。

### 14.2 アプリケーション運用中の異常

イベントサービスでアプリケーション運用中に異常が発生した場合の対処方法について説明します。

コンシューマアプリケーション運用中に異常が発生した場合、発生したタイミングにより対処を行う必要があります。また、対処方法は、コンシューマアプリケーションが接続情報を保存している場合としていない場合で異なるため、それぞれについて説明します。

#### アプリケーション運用中に接続が切断される場合

アプリケーション運用中に、ユーザ例外「Disconnected」が復帰された場合、イベントチャネルの作成時に「接続情報を自動回収する処理」を有効と設定(esmkchnlコマンド実行時に-autodisconオプションを指定)したため、CORBAサービスのクライアント側無通信監視時間超過により接続が切断された可能性があります。

アプリケーションを継続して運用する場合は、イベントチャンネルへの接続から行ってください。

## コンシューマアプリケーションの異常終了(接続情報を保存している場合)

### ■ イベントデータの待合せ中に異常終了した場合

イベントチャンネルは、イベントデータ待合せ中のコンシューマアプリケーションが異常終了しても、イベントチャンネル内ではproxyのイベントデータの待合せ状態が解除されず、サブライヤからイベントデータが送信されるまで(またはイベントデータの待合せ時間まで)、イベントデータを待ち合わせます。このため、コンシューマアプリケーションの再起動後に、同一proxyで受信処理を行った場合は、イベントチャンネル内でpullメソッド/pull\_waitメソッドが2重に発行されることになり、以下の例外が復帰される可能性があります。

システム例外 : BAD_PARAM マイナーコード : 0x464a09e5
--

本現象が発生した場合、pull\_cancelメソッドを発行し、イベントデータの待合せ状態を解除した後に、pullメソッドおよびpull\_waitメソッドを発行してください。

### ■ トランザクション運用でイベントデータ受信後、commit/rollback処理前に異常終了した場合

トランザクション運用中のアプリケーションが異常終了しても、イベントチャンネルはトランザクションのタイムアウト時間までトランザクション処理中となります。

このため、受信済みのイベントデータを再受信する場合は、必ずrollbackメソッドを発行後にbeginメソッドを発行して再度トランザクション処理を行ってください。

なお、トランザクション処理中にdisconnectメソッドを発行した場合、イベントチャンネルは自動的にcommit処理を実行しません。

## コンシューマアプリケーションの異常終了(接続情報を保存していない場合)

### ■ トランザクション運用中、イベントデータ受信後、commit/rollback処理前に異常終了した場合

Point-To-Pointモデルの運用など、コンシューマアプリケーションが接続情報を保存せずに、コンシューマの起動ごとにproxyを作成してローカルトランザクションを実施している場合、ローカルトランザクションのタイムアウト時間に大きい値を設定していると、前回受信したデータ(トランザクションが完了していない状態のデータ)を早く受信できません。

以下のどちらかの対処を行ってください。本対処によりトランザクションタイムアウトを早く検出し、前回受信したデータ(トランザクションが完了していない状態のデータ)を早く受信できます。

- ・ イベントチャンネル起動前に、esstcfnfchnlコマンド(-ltrntimeオプション指定)で、ローカルトランザクションのタイムアウト時間を300秒程度(初期設定値:300秒)に変更します。
- ・ イベントサービス起動前に、esstcfnfコマンド(-ltrntimeオプション指定)で、ローカルトランザクションのタイムアウト時間を300秒程度(初期設定値:300秒)に変更します。

なお、イベントチャンネルの不揮発チャンネル運用中の場合は、トランザクションタイムアウトの検出を待たずに、イベントチャンネルの再起動を行うことにより、ローカルトランザクションは解除され、前回受信したイベントデータを復元できます。

## 動的イベントチャンネルとの通信時に、システム例外「COMM\_FAILURE」およびマイナーコード「0x464a01ec/0x464a09ec」が復帰される場合

動的イベントチャンネルとの通信時に、Windows(R)の場合はイベントログ上、Solaris/Linuxの場合はシステムログ上に、エラーメッセージod10939が出力された場合は、以下のインタフェースに指定したホスト名またはポート番号に誤りがある可能性があります。

- ・ イベントサービス使用時
  - C言語の場合: EventFactory\_create()

- C++言語の場合: EventFactory::create()
- Java言語の場合: com.fujitsu.ObjectDirector.EventService.EventFactory.create()
- COBOLの場合: EVENTFACTORY-CREATE
- ノーティフィケーションサービス使用時
  - C言語の場合: CosNotifyChannelAdmin\_EventChannelFactory\_create\_channel()
  - C++言語の場合: CosNotifyChannelAdmin::EventChannelFactory::create\_channel()
  - Java言語の場合: org.omg.CosNotifyChannelAdmin.EventChannelFactory.create\_channel()
  - COBOLの場合: COSNOTIFYCHANNELADMIN-EVENTCHANNELFACTORY-CREATE-CHANNEL

アプリケーションプログラムにおいて、上記のインタフェースに指定したホスト名およびポート番号に誤りがないかを確認してください。

また、当該イベントファクトリを停止する場合は、esstopfctryコマンドを以下のオプションで実行し、すべての動的イベントチャンネルを停止してください。

```
esstopfctry -o off
```

### 動的イベントチャンネルの作成が遅延して、エラーメッセージod10942が出力される場合

イベントファクトリは、1プロセスかつ1スレッドのシングルスレッドのサービスであるため、同時に複数の要求を処理できません。アプリケーションがイベントファクトリに同時に複数の処理要求を発行した場合、2件目以降の処理要求はCORBAサービスにキューイングされ、イベントファクトリの処理が可能になるまで待ち状態となります。このため、アプリケーションがイベントファクトリに大量の処理要求を発行した場合、キューイング中のリクエストにタイムアウトが通知されて、エラーメッセージod10942が出力される可能性があります。

イベントファクトリに大量の処理要求を発行する場合は、エラーメッセージod10942の出力を抑止するため、エラーメッセージod10942のユーザの対処を行ってください。エラーメッセージod10942のユーザの対処については、「メッセージ集」の「メッセージ番号がodで始まるメッセージ」-「od10900番台のメッセージ」-「od10942」を参照してください。

### その他

イベントサービスのアプリケーションは、CORBAサービスのアプリケーションです。アプリケーション運用中に異常が発生した場合、「第11章 CORBAサービス使用時の異常」-「11.4 アプリケーションがエラーとなる」も参考にして対処してください。なお、Java言語のアプリケーション作成時の環境設定ORBおよび環境変数CLASSPATHの設定については、「アプリケーション作成ガイド(イベントサービス編)」の「アプリケーションの開発 (Java言語)」-「アプリケーションの作成方法」-「クライアントアプリケーションの作成方法」も参照してください。

## 14.3 アプリケーション作成時の異常

アプリケーション作成時にエラーが発生した場合は、「第11章 CORBAサービス使用時の異常」-「11.7 アプリケーションコンパイル時の異常」を参照して対処してください。



## 14.4 クラスタサービス運用中の異常

クラスタサービス運用中に異常が発生した場合の対処方法について説明します。

ホットスタンバイ運用中の環境において、クラスタサービス停止時に、待機側で以下のメッセージが表示される場合 [Solaris32/64](#) [Linux32/64](#)

■ FJSVcluster:ERROR: clmonproc: 6657: The specified resource is not being monitored.(rid:50)

### ■メッセージの意味

指定されたリソースは、監視されていません。「(rid:50)」は、ES\_INTERSTAGEで出力されたことを示します。

### ■対処

状態遷移プロシジャのサンプル「ES\_INTERSTAGE」を使用した場合、本メッセージが表示されますが、動作上は問題ありません。特に対処を行う必要はありません。

## 第15章 クラスタサービス機能使用時の異常

クラスタサービス機能を使用している場合に異常が発生した場合の対処について、以下の内容で説明します。

- ・ クラスタサービスの起動が失敗する
- ・ クラスタサービスの切り替えが失敗する

### 15.1 クラスタサービスの起動が失敗する

#### Interstage動作環境定義の設定ミス

クラスタサービス機能を使用する場合、Interstage動作環境定義にCORBAサービスが使用するホスト名(Corba Host Name)を指定している場合、待機系でのInterstageの事前起動に失敗します。クラスタサービス機能を使用する場合CORBAサービスが使用するホスト名(Corba Host Name)を使用しないでください。

#### 状態遷移プロシジャの起動優先順序の設定ミス

クラスタサービスで設定する状態遷移プロシジャの起動優先順序が正しく設定されていない場合、クラスタサービスの起動が失敗する場合があります。状態遷移プロシジャの起動優先順序が正しく設定されているか確認してください。

状態遷移プロシジャの起動優先順序については、“高信頼性システム運用ガイド”の“クラスタサービスの設定”を参照してください。

#### Webサーバの起動に失敗する Windows32/64

クラスタサービス機能使用時に、イベントログのシステムログに以下のメッセージを出力してWebサーバの起動に失敗する場合があります。

FJapache (注) は、サービス固有エラー 1 (0x1) で終了しました。

または、

汎用サービス 'FJapache' (注) はエラー '1' で失敗しました。

**注)**「Interstage HTTP Server(Webサーバ名)」と出力される場合があります。

本現象が発生した場合は、以下の対処を行ってください。

- ・ Windows Server(R) 2008の場合、以下のコマンドを実行して、セットアップパラメタを削除してください。

Cluster Resource “リソース名” /priv StartupParameters=""

- ・ Windows Server(R) 2012の場合、フェールオーバークラスターマネージャーを使用して、登録したリソースの[プロパティ]-[全般]タブを開き、「スタートアップパラメータ」を削除してください。

### 15.2 クラスタサービスの切り替えが失敗する

#### 状態遷移プロシジャの起動優先順序の設定ミス

クラスタサービスで設定する状態遷移プロシジャの起動優先順序が正しく設定されていない場合、クラスタサービスの切り替えが失敗する場合があります。状態遷移プロシジャの起動優先順序が正しく設定されているか確認してください。

状態遷移プロシジャの起動優先順序については、“高信頼性システム運用ガイド”の“クラスタサービスの設定”を参照してください。

## 第16章 Interstage シングル・サインオン運用時の異常

Interstage シングル・サインオンのトラブル発生時の対処方法について説明します。

### 16.1 トラブル発生時の対処

以下にトラブル発生から対処までの流れを説明します。

#### 16.1.1 トラブル発生から対処までの流れ

##### ■利用者

利用者は、Webブラウザに表示されるメッセージより対処を行います。

ただし、“500 Internal Server Error”など、利用者側で対処できない問題が発生した場合は、業務サーバ管理者に問い合わせてください。

その際に、以下の情報が必要です。

- ・ 問題が発生した日時
- ・ 認証に使用したユーザID、または証明書
- ・ Webブラウザに表示されたメッセージ、およびステータスコード

##### ■業務サーバ管理者

業務サーバ管理者の役割を以下に示します。

- ・ 業務サーバの調査
- ・ SSOリポジトリ/リポジトリサーバ/認証サーバに関する調査、および設定変更の依頼

利用者より問い合わせがあった場合は、以下の順序で原因を調査してください。

1. 利用者から通知されたWebブラウザに表示されたメッセージ、およびステータスコードを参照し、要因を取り除いてください。  
Webブラウザに表示されたメッセージ、およびステータスコードについては、“メッセージ集”の“Webサーバから通知されるステータスコード”を参照してください。  
“Webサーバから通知されるステータスコード”の[業務サーバ管理者の対処]に、SSOリポジトリ/リポジトリサーバ/認証サーバに関する対処が記述されていた場合は、利用者から通知された情報をSSO管理者に伝えて調査、または設定変更を依頼してください。
2. 利用者から通知された“問題が発生した日時”と“認証に使用したユーザID、または証明書”をもとに、業務サーバのアクセスログを参照し、要因を取り除いてください。  
業務サーバのアクセスログについては、“メッセージ集”の“シングル・サインオンが出力するログメッセージ”-“シングル・サインオンのアクセスログ”を参照してください。
3. 業務サーバのアクセスログより原因が特定できない場合は、業務サーバのシステムのログを参照し、要因を取り除いてください。  
業務サーバのシステムのログについては、“メッセージ集”の“メッセージ番号がssoで始まるメッセージ”を参照してください。

業務サーバのシステムのログより原因が特定できない場合は、利用者から通知された情報をSSO管理者に伝えて、調査を依頼してください。

##### ■SSO管理者

SSO管理者の役割を以下に示します。

- ・ SSOリポジトリの設定変更

- 認証サーバの調査、および設定変更
- リポジトリサーバの調査、および設定変更
- 認証サーバ間連携サービスの調査、および設定変更

業務サーバ管理者より調査、または設定変更の依頼があった場合は、以下の順序で原因の調査、または設定変更を行ってください。

1. 業務サーバ管理者より設定変更の依頼があった場合は、Webブラウザに表示されたステータスコードをもとに、“メッセージ集”の“Webサーバから通知されるステータスコード”の[SSO管理者の対処]を参照し、設定を変更してください。
2. 業務サーバ管理者より調査の依頼があった場合は、業務サーバ管理者より通知された“問題が発生した日時”と“認証に使用したユーザID、または証明書”をもとに認証サーバのアクセスログを参照し、要因を取り除いてください。認証サーバのアクセスログについては、“メッセージ集”の“シングル・サインオンが出力するログメッセージ” – “シングル・サインオンのアクセスログ”を参照してください。
3. 認証サーバのアクセスログより原因が特定できない場合は、認証サーバのシステムのログを参照し、要因を取り除いてください。  
認証サーバのシステムのログについては、“メッセージ集”の“メッセージ番号がssoで始まるメッセージ”を参照してください。  
また、認証サーバ間連携を行っている場合は、認証サーバ間連携ログを参照し、認証サーバ間連携サービスへのアクセス状況を確認してください。  
認証サーバ間連携ログについては、“メッセージ集”の“シングル・サインオンが出力するログメッセージ” – “シングル・サインオンの認証サーバ間連携ログ”を参照してください。
4. 認証サーバのシステムのログより原因が特定できない場合は、業務サーバ管理者より通知された“問題が発生した日時”と“認証に使用したユーザID、または証明書”をもとにリポジトリサーバのアクセスログを参照し、要因を取り除いてください。  
リポジトリサーバのアクセスログについては、“メッセージ集”の“シングル・サインオンが出力するログメッセージ” – “シングル・サインオンのアクセスログ”を参照してください。  
また、セッションの管理を行っている場合は、セッション管理ログを参照し、セッション状態を確認してください。  
セッション管理ログについては、“メッセージ集”の“シングル・サインオンが出力するログメッセージ” – “シングル・サインオンのセッション管理ログ”を参照してください。
5. リポジトリサーバのアクセスログより原因が特定できない場合は、リポジトリサーバのシステムのログを参照し、要因を取り除いてください。  
リポジトリサーバのシステムのログについては、“メッセージ集”の“メッセージ番号がssoで始まるメッセージ”を参照してください。

## 16.1.2 ログの出力先

シングル・サインオンのログの出力先は次のとおりです。

### システムのログの出力先

**Windows32/64**

イベントビューア(アプリケーションのログ)

**Solaris32/64**

/var/adm/messages

**Linux32/64**

/var/log/messages

### 業務サーバのアクセスログの出力先

Interstage管理コンソールを使用して、[システム] > [セキュリティ] > [シングル・サインオン] > [業務システム] > [業務システム名] > [環境設定] タブの[アクセスログ] > [ファイル名] に設定されたファイルに出力されます。

アクセスログファイルの格納先

Windows32/64

C:\Interstage\F3FMso\ssoatzag\log

Solaris32/64 Linux32/64

/var/opt/FJSV\ssoaz/log

#### 認証サーバのアクセスログの出力先

Interstage管理コンソールを使用して、[システム] > [セキュリティ] > [シングル・サインオン] > [認証基盤] > [認証サーバ] > [環境設定] タブの [アクセスログ] > [ファイル名] に設定されたファイルに出力されます。

アクセスログファイルの格納先

Windows32/64

C:\Interstage\F3FMso\ssoatcag\log

Solaris32/64 Linux32/64

/var/opt/FJSV\ssoac/log

#### リポジトリサーバのアクセスログの出力先

Interstage管理コンソールを使用して、[システム] > [セキュリティ] > [シングル・サインオン] > [認証基盤] > [リポジトリサーバ] > [環境設定] タブの [アクセスログ] > [ファイル名] に設定されたファイルに出力されます。

アクセスログファイルの格納先

Windows32/64

C:\Interstage\F3FMso\ssoatcsv\log

Solaris32/64 Linux32/64

/var/opt/FJSV\ssoav/log

#### セッション管理ログの出力先

Interstage管理コンソールを使用して、[システム] > [セキュリティ] > [シングル・サインオン] > [認証基盤] > [リポジトリサーバ] > [環境設定] タブの [セッション管理ログ] > [ファイル名] に設定されたファイルに出力されます。

セッション管理ログファイルの格納先

Windows32/64

C:\Interstage\F3FMso\ssoatcsv\log

Solaris32/64 Linux32/64

/var/opt/FJSV\ssoav/log

#### 認証サーバ間連携ログの出力先

Interstage管理コンソールを使用して、[システム] > [セキュリティ] > [シングル・サインオン] > [認証基盤] > [認証サーバ] > [環境設定] タブの [認証サーバ間連携ログ] > [ファイル名] に設定されたファイルに出力されます。

認証サーバ間連携ログファイルの格納先

Windows32/64

C:\Interstage\F3FMso\ssofsv\log

Solaris32/64 Linux32/64

/var/opt/FJSV\ssofs/log

## 16.2 トラブル事例

---

一般的に発生するトラブルについて、以下の項目に分類して説明します。

## 16.2.1 認証に関するトラブル

---

### 業務システムの保護リソースにアクセスしたが、認証が行われずにコンテンツが表示される

以下の点を確認してください。

- 業務サーバの環境設定の[Webサーバの設定]の[使用しているWebサーバ]のポート番号の値が、業務サーバが動作するWebサーバのポート番号と一致しているか確認してください。(注1)
- 業務サーバを組み込んだWebサーバのアクセスログを確認し、リクエストを受信しているか確認してください。リクエストを受信していない場合、他のWebサーバにアクセスしている可能性があります。
- Webサーバに業務サーバを組み込んだ後にWebサーバを再起動していない可能性があります。Webサーバを再起動していない場合は、再起動してください。
- 業務サーバの組み込み手順に誤りがないか確認してください。  
業務サーバの組み込み手順については、“シングル・サインオン運用ガイド”の“環境構築(業務サーバ管理者編)”の“Webサーバへの組み込み”を参照してください。

### 業務システムの保護リソースにアクセスしたが、予期せぬコンテンツが表示される

業務サーバの環境設定の[業務システムの情報]の[公開URL]、または[認証基盤の情報]の[URL]が正しく設定されているか確認してください。(注2)

### 業務システムの保護リソースにアクセスしたが、応答に1分以上かかる

業務システムの保護リソースのURLに含まれる業務サーバのFQDNを、認証サーバで名前解決できるか確認してください。

### 業務システムの保護リソースにアクセスしたが、応答が返ってこない

以下の点を確認してください。

- 業務サーバおよび認証サーバが起動しているか確認してください。
- 業務サーバの環境設定の[業務システムの情報]の[公開URL]、または[認証基盤の情報]の[認証基盤のURL]が正しく設定されているか確認してください。(注2)
- リポジトリサーバ、または認証サーバが構築されているマシンにおいて、通信を妨げるWebサーバ、バーチャルホストなどが存在していないか確認してください。
- Interstage Portalworksと連携する業務システムの場合には、以下の点について確認してください。
  - 業務サーバの構築時に使用した業務システム構築ファイルが、Interstage Portalworksと連携する業務システム用に作成されたものである。(注3)
  - 業務サーバの環境設定の[認証情報]の[クライアントのIPアドレスのチェック]が“チェックしない”に設定されている。(注1)
- 認証サーバの前にSSLアクセラレータ、ロードバランサ、またはInterstage Security Directorを配置している場合には、それらの設定が正しく行われていることを確認してください。(注4)

### ユーザ情報の有効期間が切れる前に再認証が発生する

リポジトリサーバ、認証サーバ、および業務サーバのシステム時刻が同一となっているか確認してください。(注5)

### ユーザ情報の有効期間が切れても再認証が発生しない

以下の点を確認してください。

- 利用者に対して再度認証が要求されない場合はWebブラウザのキャッシュ情報を表示している可能性があります。
- リポジトリサーバ、認証サーバ、および業務サーバのシステム時刻が同一となっているか確認してください。(注5)

## サインオフ後に保護リソースが表示される

クライアントのWebブラウザにFirefoxを使用し、フレームを使用したコンテンツにアクセスしている場合、Webブラウザのキャッシュ情報を表示している可能性があります。フレームページの保護リソースの設定が正しく行われているか確認してください。

フレームページの保護リソースの設定については、“シングル・サインオン運用ガイド”の“環境構築(業務サーバ管理者編)”―“業務システムの設計”―“コンテンツ作成時の注意事項”―“フレームページ使用時の注意”を参照してください。

## 業務システムの保護リソース、または認証基盤のURLにアクセスしても認証画面が表示されない Webブラウザに「他の画面で認証されるのを待っています。」が表示された後に、[継続]ボタンを押下しても正しく動作しない

Webブラウザに表示するメッセージファイルのカスタマイズが正しく行われていない可能性があります。以下の同時認証確認画面のメッセージファイル(注6)を確認し、正しくカスタマイズしなおしてください。(注7)

- 200check\_duplication\_auth\_en.template
- 200check\_duplication\_auth\_ja.template

## ICカードを利用した証明書認証をパスワード認証に切り替えられない

ブラウザの設定が自動的に証明書を送信する設定となっている場合、認証方式が「証明書認証またはパスワード認証」の利用者は、パスワード認証が行えません。

ブラウザの設定を確認してください。

## 保護パスにアクセスして認証を行ったが、保護パスが表示されない 保護パスにアクセスして認証を行ったが、認証画面が表示されたままとなる 保護パスにアクセスして認証を行ったが、認証画面が空白のコンテンツとして表示されたままとなる

Windows(R) Internet Explorer(R) 7.0以降のインターネットオプションのセキュリティ設定において、インターネットゾーンや、ローカルイントラネットゾーンなどの各ゾーンとして登録したInterstage シングル・サインオンの業務サーバとInterstage シングル・サインオンの認証サーバに対する“保護モード”の設定が異なっている可能性があります。業務サーバと認証サーバに対する“保護モード”の設定を合わせてください。

## COOKIEが有効なWebブラウザで「Please enable COOKIE on your browser.」というメッセージが表示される

利用者が、Webブラウザを使用して、非SSL通信で認証サーバにアクセスしました。

認証サーバへの通信はSSLで行なうようにしてください。SSLアクセラレータを設置している場合は、SSLアクセラレータ経由で認証サーバにアクセスするようにしてください。

## 【リポジトリサーバ(更新系)の負荷分散を行っている場合のトラブル】

### 保護パスにアクセスして認証を行ったが、保護パスが表示されずに、保護パスのURLと認証基盤のURLが繰返し表示され続ける

### 保護パスにアクセスし、正しいユーザID/パスワードで認証を行うが、認証に失敗する 強制サインオン問い合わせページにて“はい”ボタンをクリックしたが、認証画面が表示される

ロードバランサの設定が正しく行われているか確認してください。

ロードバランサの設定については、“シングル・サインオン運用ガイド”の“環境構築(SSO管理者編)”―“リポジトリサーバの構築”―“負荷分散のためのリポジトリサーバ(更新系)の追加”を参照してください。

**注1)**[Webサーバの設定]の[使用しているWebサーバ]、および、[認証情報]の[クライアントのIPアドレスのチェック]については、Interstage管理コンソールを使用して、[システム]>[セキュリティ]>[シングル・サインオン]>[業務システム]>[業務システム名]>[環境設定]タブの[詳細設定[表示]]で確認/変更してください。

**注2)**[業務システムの情報]の[公開URL]および[認証基盤の情報]の[URL]については、Interstage管理コンソールを使用して、[システム]>[セキュリティ]>[シングル・サインオン]>[業務システム]>[業務システム名]>[環境設定]タブの[詳細設定[表示]]を確認してください。

値を変更する場合は、業務サーバを再度構築してください。

業務サーバの再構築については、Interstage管理コンソールを使用して、[システム]>[セキュリティ]>[シングル・サイン

オン] > [業務システム]画面の[一覧]タブにより業務サーバを削除してから、[業務サーバの追加]タブにより業務サーバを追加してください。

**注3)** Interstage Portalworksと連携する業務システム用の業務システム構築ファイルを作成する場合には、リポジトリサーバ(更新系)のInterstage管理コンソールの[セキュリティ] > [シングル・サインオン] > [認証基盤] > [業務システム構築ファイル]の[Interstage Portalworksとの連携]で“連携する”を選択し、認証が有効となる範囲をドメイン名で指定してください。

**注4)** SSLアクセラレータの設定については、“シングル・サインオン運用ガイド”の“概要”-“高性能・高信頼性システム”-“SSLアクセラレータとの連携”を参照してください。ロードバランサの設定については、“概要”-“高性能・高信頼性システム”-“負荷分散”を参照してください。Interstage Security Directorの設定については、“概要”-“高性能・高信頼性システム”-“Interstage Security Directorとの連携”を参照してください。

**注5)** 各サーバのシステム時刻を確認する場合には、タイムゾーンの設定に注意してください。

**注6)** 同時認証確認画面のメッセージファイルについては、“シングル・サインオン運用ガイド”の“シングル・サインオンのカスタマイズ”-“Webブラウザに表示するメッセージのカスタマイズ”-“カスタマイズできるメッセージ”を参照してください。

**注7)** 同時認証確認画面用タグの仕様については、“シングル・サインオン運用ガイド”の“シングル・サインオンのカスタマイズ”-“Webブラウザに表示するメッセージのカスタマイズ”-“同時認証確認画面用タグの仕様”を参照してください。

## 16.2.2 POSTリクエストに対する認証に関するトラブル

### ブラウザに未認証画面が表示されるが、認証画面が表示されない

以下の点を確認してください。

- ブラウザの設定でJavaScriptが無効となっていないか確認してください。無効となっている場合は、JavaScriptを有効にした上で再度、認証を実施してください。
- 以下の未認証画面用メッセージファイル(注1)の「var target = "<!--SSO\_POST\_REQUEST\_AUTH\_TARGET-->";」部分が削除されている可能性があります。削除されている場合は、未認証画面用メッセージファイルに「var target = "<!--SSO\_POST\_REQUEST\_AUTH\_TARGET-->";」を正しく記述してください。(注2)
  - 200postauth\_ja.template、または403postauth\_ja.template
  - 200postauth\_en.template、または403postauth\_en.template

### 認証画面に何も表示されない

以下の未認証画面用メッセージファイル(注1)の「var target = "<!--SSO\_POST\_REQUEST\_AUTH\_TARGET-->";」が変更されている可能性があります。変更されている場合は、未認証画面用メッセージファイルに「var target = "<!--SSO\_POST\_REQUEST\_AUTH\_TARGET-->";」を正しく記述してください。(注2)

- 200postauth\_ja.template、または403postauth\_ja.template
- 200postauth\_en.template、または403postauth\_en.template

### 認証画面に「Waiting」が表示される

表示する画像、または他のページへのハイパーリンクなど、以下の未認証画面用メッセージファイルに他のコンテンツが指定されている場合、運用中の業務サーバの保護パス配下に格納されたファイルを指定している可能性があります。保護パス配下に格納されたファイルを指定しないようメッセージファイルを見直してください。

- 200postauth\_ja.template、または403postauth\_ja.template
- 200postauth\_en.template、または403postauth\_en.template

### 認証画面は自動的に閉じるが、元のウィンドウが未認証画面から画面遷移しない

### 認証後、認証画面が自動的に閉じず、認証画面にコンテンツが表示される、または「認証に成功しました。」が表示される

以下の点を確認してください。

- Webコンテンツに記述するPOSTリクエストの送信先(<FORM>タグのaction属性)に記述されているホスト名、およびポート番号が、業務システムの公開URLのホスト名、およびポート番号と一致していない可能性があります。一致していない場合は、ホスト名、およびポート番号が一致するように記述を見直してください。
- POSTリクエストの送信先が、スキームが“http”でポート番号が“80”の場合、またはスキームが“https”でポート番号が“443”の場合、Webコンテンツに記述するPOSTリクエストの送信先(<FORM>タグのaction属性)にポート番号が記述されている可能性があります。記述されている場合は、以下のようにポート番号を削除し、正しく指定してください。  
(誤) : https://business\_server.fujitsu.co.jp:443/post  
(正) : https://business\_server.fujitsu.co.jp/post
- 以下の未認証画面用メッセージファイル(注1)の「<!--SSO\_POST\_REQUEST\_AUTH\_NEED-->」部分が削除、または変更されている可能性があります。削除、または変更されている場合は、未認証画面用メッセージファイルに「<!--SSO\_POST\_REQUEST\_AUTH\_NEED-->」を正しく記述してください。(注2)
  - 200postauth\_ja.template、または403postauth\_ja.template
  - 200postauth\_en.template、または403postauth\_en.template

## POSTデータが正しく送信されない

認証後の画面の日本語が、文字化けして表示される場合など、POSTデータが正しく送信されない場合は、以下の未認証画面用メッセージファイル(注1)の<meta>タグのcharset属性に指定した文字コードと、Webコンテンツで使用する文字コードが一致していない可能性があります。一致していない場合は、文字コードが一致するよう設定を見直してください。

- 200postauth\_ja.template、または403postauth\_ja.template
- 200postauth\_en.template、または403postauth\_en.template

## 未認証画面に認証画面を表示するリンクが表示されない

利用者が、POSTリクエストに対する認証操作を途中で中断した可能性があります。認証操作は完了していますので、継続して業務システムを利用してください。

業務システムの利用を中止する場合には、セッションの管理を行っている業務システムではサインオフ操作を実施し、セッションの管理を行っていない業務システムではWebブラウザを閉じてください。

## 認証後の画面に、「認証に成功しました。」が表示される

以下の点を確認してください。

- POSTリクエストに対する認証操作を途中で中断し、再度認証操作を行った可能性があります。認証は完了していますので、継続して業務システムを利用してください。
- 以下のメッセージファイルのカスタマイズに誤りがある可能性があります。メッセージファイルの<head>タグ中に記載されている<script>タグ内のJavaScript、および<body>タグに設定されているJavaScriptのイベントハンドラ「onload="close\_sso\_window()」部分が変更、または削除されていないか確認してください。変更、または削除されている場合は、元の正しい記述に戻してください。
  - 200closeerr\_en.template、または403closeerr\_en.template
  - 200closeerr\_ja.template、または403closeerr\_ja.template
- 業務サーバへのリクエストURLに“fj-is-sso”で始まるクエリストリングを使用している可能性があります。使用している場合は、“fj-is-sso”で始まるクエリストリングを使用しないよう、設定を見直してください。
- Windows(R) Internet Explorer(R) 7.0以降のインターネットオプションのセキュリティ設定において、インターネットゾーンや、ローカルイントラネットゾーンなどの各ゾーンとして登録したInterstage シングル・サインオンの業務サーバとInterstage シングル・サインオンの認証サーバに対する“保護モード”の設定が異なっている可能性があります。業務サーバと認証サーバに対する“保護モード”の設定を合わせてください。

注1)未認証画面用メッセージファイルについては、“シングル・サインオン運用ガイド”の“シングル・サインオンのカスタマイズ” – “Webブラウザに表示するメッセージのカスタマイズ” – “カスタマイズできるメッセージ”を参照してください。

**注2)**未認証画面用メッセージファイルのカスタマイズについては、“シングル・サインオン運用ガイド”の“シングル・サインオンのカスタマイズ” – “Webブラウザに表示するメッセージのカスタマイズ” – “未認証画面用タグの仕様”を参照してください。

## 16.2.3 統合Windows認証に関するトラブル

---

### 業務システムの保護リソースへのアクセス時、または認証基盤のURLへのアクセス時に、認証画面が表示されず、Webブラウザに何も表示されない

統合Windows認証時に表示されるメッセージファイル(**注1**)に必要なFORMタグが設定されていない可能性があります。以下のメッセージファイルを確認し、FORMタグを正しく記述してください。(注2)

- 200\_confirm\_winauth\_en.html
- 200\_confirm\_winauth\_ja.html
- 200\_confirm\_timeout\_en.html
- 200\_confirm\_timeout\_ja.html
- 200\_confirm\_expired\_en.html
- 200\_confirm\_expired\_ja.html

### Webブラウザに表示するメッセージが文字化けする

統合Windows認証時に表示されるメッセージファイル(**注1**)に必要な<meta>タグの“charset”属性の設定に誤りがある可能性があります。メッセージファイルを確認し、<meta>タグの“charset”属性を正しく記述してください。(注2)

### 統合Windows認証を実施しているのに、基本認証ダイアログが表示される

Webブラウザの設定を見直してください。Webブラウザの設定については、“シングル・サインオン運用ガイド”の“Active Directoryと連携するための設定”を参照してください。

### 統合Windows認証によるサインオン取り消し時に、「統合Windows認証に失敗しました。」が表示される

認証操作によって表示されたすべてのWebブラウザ画面を閉じて、再度認証操作を実施してください。

### Active Directoryの設定を変更した後、統合Windows認証を行うと「統合Windows認証に失敗しました。」が表示される

ドメインからいったんログオフし、再度ログオンした後、統合Windows認証を行ってください。

### 連携しているActive Directoryのドメインにログオンしても、統合Windows認証に失敗する

統合Windows認証に失敗するユーザIDでWindowsログオンした際、Kerberos認証のキー配布センターから、チケット許可チケットが正しく発行されているか確認してください。

### Webブラウザに「ページを表示できません」が表示され、システムのログにsso07000番台のメッセージが出力される

Webブラウザに表示する以下のメッセージファイルのサイズが512バイト未満です。ファイルのサイズを512バイト以上にカスタマイズしなおしてください。(注1)

- 500\_internal\_error\_en.html
- 500\_internal\_error\_ja.html

**注1)**統合Windows認証時に表示されるメッセージファイルについては、“シングル・サインオン運用ガイド”の“シングル・サインオンのカスタマイズ” – “Webブラウザに表示するメッセージのカスタマイズ” – “カスタマイズできるメッセージ”を参照してください。

注2)統合Windows認証時に表示されるメッセージのカスタマイズについては、“シングル・サインオン運用ガイド”の“シングル・サインオンのカスタマイズ”―“Webブラウザに表示するメッセージのカスタマイズ”―“統合Windows認証時に表示されるメッセージ用のタグの仕様”を参照してください。

## 16.2.4 業務サーバの認可に関するトラブル

### SSOリポジトリに保護リソースやパス定義の追加/変更/削除を行ったが、正しく認可されない

保護リソースやパス定義の追加/変更/削除を行った場合は、必ずアクセス制御情報の更新を行ってください。(注1)  
なお、ロール定義も合わせて変更した場合は、必ずロール情報を更新した後、アクセス制御情報の更新を行ってください。(注1)(注2)

### 拡張ユーザ情報がWebアプリケーションに通知されない

以下の点を確認してください。

- ・ シングル・サインオンシステムを構築するサーバのバージョンが古い可能性があります。シングル・サインオンシステムを構築する各サーバのバージョンが、拡張ユーザ情報をWebアプリケーションに通知するシステムとして可能な組み合わせかどうか確認してください。(注3)
- ・ 保護パスに、拡張ユーザ情報が設定されていない可能性があります。保護パスに、通知する拡張ユーザ情報が設定されているか確認してください。(注4)
- ・ アクセス制御情報が更新されていない可能性があります。アクセス制御情報の更新を行ってください。(注1)
- ・ リポジトリサーバ(更新系)を複数台配置して負荷分散するシステム構成で運用している場合、負荷分散しているすべてのリポジトリサーバ(更新系)の環境設定において、[拡張ユーザ情報]の設定が一致していない可能性があります。すべてのリポジトリサーバ(更新系)の環境設定を合わせ、リポジトリサーバ(更新系)を起動しなおしてください。(注5)
- ・ 認証サーバ間連携を行っている場合、相手シングル・サインオンシステムから拡張ユーザ情報が通知されていない可能性があります。相手シングル・サインオンシステム管理者に、拡張ユーザ情報の通知の有無を確認してください。
- ・ 認証サーバ間連携を行い、ユーザ情報の変換を行っている場合、拡張ユーザ情報の変換ルールが定義されていない可能性があります。拡張ユーザ情報の変換ルールを確認してください。
- ・ リポジトリサーバ(更新系)とリポジトリサーバ(参照系)に分散するシステム構成で運用している場合、リポジトリサーバ(参照系)を構築した後、リポジトリサーバ(更新系)の環境設定で[拡張ユーザ情報]を変更した可能性があります。[拡張ユーザ情報]を変更した場合は、リポジトリサーバ(参照系)を作成しなおしてください。(注6)

### アクセス制御情報の更新に失敗する

業務サーバのInterstage管理コンソールを使用して、アクセス制御情報の更新を行った場合、sso04139のエラーメッセージが表示される場合があります。

リポジトリサーバのシステムログにエラーが出力されているか確認し、エラーが出力されている場合は、リポジトリサーバのシステムログの対処に従ってください。

### アクセス制御情報ファイルの更新を行ったが、正しく認可されない

- ・ アクセス制御情報更新コマンド(ssorfinfaz)の実行時に、引数のhost:portに指定した値、および-acに指定した値に誤りがないか確認してください。

**Solaris32/64** **Linux32/64**

- ・ 以下のいずれかの操作を行った場合、1台のマシンに構築されているすべての業務サーバを再起動してください。
  - ― Interstage管理コンソールを使用し、業務サーバの環境設定を変更した。
  - ― 業務サーバの環境定義ファイルを更新した。
  - ― 業務サーバの追加を行った。
  - ― 業務サーバの削除を行った。

— ssoimpazコマンドを実行した。

## 業務システムの保護リソースにアクセスしたが、応答が極端に遅い

アクセス制御の対象とする必要のないパスが、保護パスに設定されている可能性があります。アクセス制御の対象とする必要のないパスが、保護パスの対象外になるように設定してください。(注7)

## 認証前にアクセスした保護リソースと異なる保護リソースが表示される

認証失敗時、または認証キャンセル時に、Webブラウザを閉じずに継続して、保護リソースにアクセスしたため、以前にアクセスした保護リソースの情報が残っている可能性があります。

Webブラウザを終了し再度起動して、保護リソースにアクセスし直してください。

**注1)**アクセス制御情報の更新は、業務サーバのInterstage管理コンソールを使用して、[システム]>[セキュリティ]>[シングル・サインオン]>[業務システム]>[業務システム名]>[アクセス制御情報の更新]タブの[更新]ボタンをクリックしてください。

**注2)**ロール情報の更新は、リポジトリサーバ(更新系)のInterstage管理コンソールを使用して、[システム]>[セキュリティ]>[シングル・サインオン]>[認証基盤]>[リポジトリサーバ]>[ロール情報の更新]タブの[更新]ボタンをクリックしてください。

**注3)**拡張ユーザ情報をWebアプリケーションに通知するシステムとして可能な組み合わせについては、“使用上の注意”の“注意事項”-“Interstage シングル・サインオンの注意事項”-“バージョン・エディション混在でシングル・サインオンシステムを構築する場合の注意事項”を参照してください。

**注4)**通知する拡張ユーザ情報については、リポジトリサーバ(更新系)のInterstage管理コンソールを使用して、[システム]>[セキュリティ]>[シングル・サインオン]>[認証基盤]>[リポジトリサーバ]>[保護リソース]>[サイト定義]>[保護パス]>[パス定義]をクリックし、[パス定義]の[拡張ユーザ情報の通知]を確認してください。

**注5)**リポジトリサーバの停止については“シングル・サインオン運用ガイド”の“運用・保守”-“シングル・サインオンの停止”-“リポジトリサーバの停止”を参照してください。リポジトリサーバの起動については“シングル・サインオン運用ガイド”の“運用・保守”-“シングル・サインオンの起動”-“リポジトリサーバの起動”を参照してください。

**注6)**リポジトリサーバ(参照系)の作成方法については、“シングル・サインオン運用ガイド”の“環境構築(SSO管理者編)”-“リポジトリサーバの構築”-“リポジトリサーバ(参照系)の追加”を参照してください。

**注7)**保護パスの設定方法については、“シングル・サインオン運用ガイド”の“環境構築(SSO管理者編)”-“業務システムの登録”-“保護リソースなどの登録”-“保護パスの登録”を参照してください。

## 16.2.5 リポジトリサーバに関するトラブル

### リポジトリサーバの起動に時間がかかる

SSOリポジトリ内に登録しているロール定義のエントリ数、およびサイト定義のエントリ数によって、Interstage HTTP Serverの起動に時間がかかる場合があります。

特にロール定義にロールセットを多用し、ロールセットに含めるロールとしてロールセット名を指定している場合(ロールセットの入れ子)、以下のように、ロールセットの登録数分、入れ子の階層ごとにInterstage HTTP Serverの起動時間が倍増します。

ロールセットの入れ子が少なくなるようにロール定義を見直すことで、リポジトリサーバの起動時間を短くすることができます。



例

No.	ロール定義の登録パターン	Interstage HTTP Serverの起動時間
1	1000個のロールのみ登録されている場合	1秒 + α (注)
2	No.1に加え、10個のロールを含むロールセットが100個登録されている場合	101秒 + α (注)

No.	ロール定義の登録パターン	Interstage HTTP Server の起動時間
3	No.2に加え、No.2の10個のロールセットを含むロールセットが10個登録されている場合	1101秒 + $\alpha$ (注)

(注)ロール定義に関係なく、Interstage HTTP Serverの起動に要する時間(数秒程度)です。

## 16.2.6 ユーザ情報を登録するディレクトリサービスにActive Directoryを使用する場合のトラブル

リポジトリサーバの起動時にins01027のメッセージが出力され、起動に失敗する [Solaris32/64](#) [Linux32/64](#)

ディレクトリサービスのURLに誤りがないか確認してください。(注1)

### 【シングル・サインオンの拡張スキーマを使用しない場合】

#### 保護パスにアクセスして統合Windows認証を行うが、認証に失敗する

以下の点を確認してください。

- ディレクトリサービスとの接続情報(接続用DN、接続用DNのパスワード)が、すべてのユーザ情報に対して参照権限があるか確認してください。(注2)(注3)
- ユーザ情報の登録先エン트리配下に、対象のユーザ情報が存在するか確認してください。(注3)(注4)
- ユーザ情報の登録先エン트리配下の対象のユーザ情報に、必要な属性が存在するか確認してください。(注3)(注5)
- 連携を行うことのできない、リフェラル機能を使用したActive Directoryを使用している可能性があります。使用しているActive Directoryが、リフェラル機能を使用していないか確認してください。

#### 保護パスにアクセスして統合Windows認証を行うが、保護パスが表示されない

以下の点を確認してください。

- リポジトリサーバの環境設定の[Active Directoryの設定]で指定した[ロールに使用する属性名]の値に誤りがないか確認してください。(注6)
- リポジトリサーバの環境設定の[Active Directoryの設定]で指定した[ロールに使用する属性名]の値が、ユーザ情報に登録されているか確認してください。
  - ユーザ情報に登録されていない場合は、[ロール定義に使用する属性名]の値を登録してください。
  - ユーザ情報に登録されている場合は、アクセスした保護パスに設定したロール/ロールセットのいずれかに設定したロールに使用する属性値に、ユーザ情報に登録されている値が存在するか確認してください。  
なお、アクセスした保護パスに設定したロール/ロールセットが多階層で登録されている場合は、ロールセットの内容に設定したすべてのロール/ロールセットについても確認してください。(注7)
- 上記に問題がない場合は、ロール定義の関連付けを見直してください。(注8)

### 【シングル・サインオンの拡張スキーマを使用する場合】

#### 保護パスにアクセスして統合Windows認証、またはパスワード認証を行うが、認証に失敗する 利用者のロック解除を行うが、利用者を特定できず、ロック解除に失敗する

以下の点を確認してください。

- ディレクトリサービスとの接続情報(接続用DN、接続用DNのパスワード)が、すべてのユーザ情報に対して参照、および更新権限があるか確認してください。(注2)(注3)
- ユーザ情報の登録先エン트리配下に、対象のユーザ情報が存在するか確認してください。(注3)(注4)
- ユーザ情報の登録先エン트리配下の対象のユーザ情報に、必要な属性が存在するか確認してください。(注3)(注5)

- ユーザ情報に、“inetOrgPerson”オブジェクトクラスが含まれているか確認してください。(注3)
- ユーザ情報の“ssoUserStatus”属性に値(“good”、または“locked”)が設定されているか確認してください。(注3)
- シングル・サインオンの拡張スキーマの設定が正しく行われているか、以下の点を確認してください。
  - Active Directoryスキーマに“inetOrgPerson”オブジェクトクラスが存在するか確認してください。(注9)
  - Active Directoryスキーマに“ssoUser”オブジェクトクラスが存在するか確認してください。(注9)
  - Active Directoryスキーマに、シングル・サインオンのすべての属性が存在するか確認してください。(注9)(注10)
  - Active Directoryスキーマの“ssoUser”オブジェクトクラスのオプション属性に、シングル・サインオンのすべての属性が設定されているか確認してください。(注9)(注10)
  - Active Directoryスキーマの“inetOrgPerson”オブジェクトクラスの補助型クラスに、“ssoUser”オブジェクトクラスが設定されているか確認してください。(注9)
- 連携を行うことのできない、リフェラル機能を使用したActive Directoryを使用している可能性があります。使用しているActive Directoryが、リフェラル機能を使用していないか確認してください。

### 保護パスにアクセスして統合Windows認証、またはパスワード認証を行うが、保護パスが表示されない

ユーザ情報に、保護パスにアクセス可能なロール名または、ロールセット名(ssoRoleName)が設定されているか確認してください。(注3)

**注1)**Interstage管理コンソールの[システム] > [セキュリティ] > [シングル・サインオン] > [認証基盤] > [リポジトリサーバ] > [環境設定]タブの[リポジトリサーバ詳細設定[表示]]をクリックし、[リポジトリ] > [Active Directoryの設定]の[ディレクトリサービスのURL]で確認してください。

**注2)**Interstage管理コンソールの[システム] > [セキュリティ] > [シングル・サインオン] > [認証基盤] > [リポジトリサーバ] > [環境設定]タブの[リポジトリサーバ詳細設定[表示]]をクリックし、[リポジトリ] > [Active Directoryの設定]の[接続用DN]および、[接続用DNのパスワード]で確認してください。

**注3)**Microsoftが提供するADSI Editツールなどを使用して確認してください。

**注4)**Interstage管理コンソールの[システム] > [セキュリティ] > [シングル・サインオン] > [認証基盤] > [リポジトリサーバ] > [環境設定]タブの[リポジトリサーバ詳細設定[表示]]をクリックし、[リポジトリ] > [ユーザ情報の登録先エントリ]で確認してください。

**注5)**ユーザ情報に必要な属性については、“シングル・サインオン運用ガイド”の“Active Directoryと連携するための設定”-“ユーザ情報を登録するディレクトリサービスにActive Directoryを使用する”-“Active Directoryのユーザ情報の設定”を参照してください。

**注6)**Interstage管理コンソールの[システム] > [セキュリティ] > [シングル・サインオン] > [認証基盤] > [リポジトリサーバ] > [環境設定]タブの[リポジトリサーバ詳細設定[表示]]をクリックし、[リポジトリ] > [Active Directoryの設定]の[ロールに使用する属性名]で確認してください。

**注7)**アクセスした保護パスに設定したロール/ロールセットの確認方法については、Interstage管理コンソールの[システム] > [セキュリティ] > [シングル・サインオン] > [認証基盤] > [リポジトリサーバ] > [保護リソース] > [サイト定義] > [保護パス]をクリックし、パス定義の一覧からアクセスした保護パスをクリックして、ロール/ロールセット、ロールセットの内容、およびロールに使用する属性値を確認してください。

**注8)**ロール定義の関連付けについては、“シングル・サインオン運用ガイド”の“Active Directoryと連携するための設定”-“ユーザ情報を登録するディレクトリサービスにActive Directoryを使用する”-“SSOリポジトリの設定(ロール定義の関連付け)”を参照してください。

**注9)**Active Directoryの確認方法については、Active Directoryのマニュアルを参照してください。

**注10)**シングル・サインオンの属性は、以下に示す9つです。

- ssoAuthType
- ssoCredentialTTL
- ssoFailureCount

- ssoLockTimeStamp
- ssoNotAfter
- ssoNotBefore
- ssoRoleName
- ssoSessionInfo
- ssoUserStatus

## 16.2.7 Microsoft(R) Internet Information Servicesに関するトラブル

### Microsoft(R) Internet Information Services 使用時に業務サーバのプログラムの読み込みに失敗する

以下のイベントログが出力されているか確認してください。出力されている場合は、システムを再起動してください。

HTTP フィルタ DLL C:\¥Interstage¥F3FM\\$sso¥ssoatzag¥lib¥F3FM\\$ssoatziis.dll の読み込みに失敗しました。このデータはエラーです。

### ISAPIフィルタに業務サーバ以外のプログラムを設定すると正しく認可されない

ISAPIフィルタに業務サーバ以外のプログラムを設定している場合は、Microsoft(R) Internet Information Servicesのプロパティシートの[ISAPIフィルタ]タブにて、業務サーバのプログラムが一番上に表示されるように順番を変更してください。

ISAPIフィルタの設定については、“シングル・サインオン運用ガイド”の“環境構築(業務サーバ管理者編)” – “Webサーバへの組み込み”を参照してください。

### 運用中、システムのログにsso00202のメッセージが表示される。

アプリケーション プールの設定に誤りがないか確認してください。アプリケーション プールの設定については、以下を参照してください。

- Microsoft(R) Internet Information Services 7.0,7.5を使用している場合  
“シングル・サインオン運用ガイド”の“環境構築(業務サーバ管理者編)” – “Webサーバへの組み込み” – “Microsoft(R) Internet Information Services 7.0,7.5 への組み込み”
- Microsoft(R) Internet Information Services 8.0を使用している場合  
“シングル・サインオン運用ガイド”の“環境構築(業務サーバ管理者編)” – “Webサーバへの組み込み” – “Microsoft(R) Internet Information Services 8.0 への組み込み”

## 16.2.8 Webブラウザに関するトラブル

### 複数のウィンドウを使用して業務システムにアクセスした時に、「ページを表示できません」と表示される

インターネット オプションの「詳細設定」 – 「HTTPエラーメッセージを簡易表示する」がチェックされているか確認してください。チェックされている場合は「HTTPエラーメッセージを簡易表示する」のチェックをはずし、ブラウザの「更新」ボタンをクリックしてください。

ICカードを使用して認証を行っている場合は、ICカードが正しく挿入されていることを確認し、ブラウザの「更新」ボタンをクリックしてください。

## 16.2.9 認証サーバ間連携サービスに関するトラブル

### Webブラウザに表示するメッセージが文字化けする

認証サーバ間連携時に表示されるメッセージファイル(注1)に必要な<meta>タグの“charset”属性の設定に誤りがある可能性があります。メッセージファイルを確認し、<meta>タグの“charset”属性を正しく記述してください。(注2)

## 認証を行うシステムの設定画面のURL、または認証を行うシステムの解除画面のURLにアクセスしても、Webブラウザに何も表示されない

認証を行うシステムの設定画面、または認証を行うシステムの解除画面のURLにアクセスした際に表示されるメッセージファイル(注1)に必要なFORMタグが設定されていない可能性があります。以下のメッセージファイルを確認し、FORMタグを正しく記述してください。(注2)

- 200system\_setting\_en.html
- 200system\_setting\_ja.html
- 200system\_clear\_local\_system\_set\_en.html
- 200system\_clear\_local\_system\_set\_ja.html

## 業務システムの保護リソースへのアクセス時、または認証基盤のURLへのアクセス時に、認証画面が表示されず、Webブラウザに何も表示されない

以下の点を確認してください。

- 認証サーバ間連携サービスが配備されているか確認してください。(注3)
- 相手シングル・サインオンシステムの認証基盤のURLに誤りがないか確認してください。(注4)
- 利用者から通知されたURLが、認証を行うシステムの設定画面、または認証を行うシステムの解除画面のURLであるか確認してください。(注5)
- 利用者が、認証を行うシステムの設定画面から操作を行っていた場合、以下のメッセージファイル(注1)のカスタマイズの内容に誤りがないか確認してください。(注2)
  - 200system\_setting\_en.html
  - 200system\_setting\_ja.html
- 利用者が、認証を行うシステムの解除画面から操作を行っていた場合、以下のメッセージファイル(注1)のカスタマイズの内容に誤りがないか確認してください。(注2)
  - 200system\_clear\_local\_system\_set\_en.html
  - 200system\_clear\_local\_system\_set\_ja.html
- 認証サーバ間連携の環境定義ファイルを直接編集している場合、定義項目に正しい値が設定されているか確認してください。
- 認証サーバ間連携サービスのシステムのログにシングル・サインオンのメッセージが出力されているか確認してください。メッセージが出力されている場合は、出力されたメッセージに対する[ユーザの対処]に従って対処してください。
- 利用者からのアクセスが行われていた場合には、認証サーバ間連携ログに出力されている、リクエストを送信する相手シングル・サインオンシステムの管理者に問い合わせてください。
- ネットワーク環境を見直してください。

## 相手シングル・サインオンシステムで認証した後、Webブラウザに何も表示されない

認証サーバ間連携サービスの移行が正しく行われていない可能性があります。移行手順に誤りがないか確認してください。移行手順については、“移行ガイド”の“Interstage運用操作、環境の移行” – “Interstage シングル・サインオンの移行”を参照してください。

## 認証を行うシステムの設定を行ったにもかかわらず、認証を行うシステムの認証画面とは異なる認証画面が表示される

サインオンした状態で、認証を行うシステムの設定を行いました。ブラウザを閉じてから、再度操作を行ってください。

## 認証を行うシステムの設定を行ったにもかかわらず、認証時に「認証を行うシステムが設定されていません。」が表示される

以下の点を確認してください。

- ・ 自シングル・サインオンシステムの環境設定に定義されている相手シングル・サインオンシステムの情報に誤りが  
ないか確認してください。(注6)
- ・ 認証サーバ間連携の環境定義ファイルを直接編集している場合、定義項目に正しい値が設定されているか確認  
してください。
- ・ 認証サーバ間連携サービスを再起動してください。(注7)
- ・ 相手シングル・サインオンシステムの管理者に、相手シングル・サインオンシステムの環境設定に定義されてい  
る自シングル・サインオンシステムの設定に誤りがなく確認するよう依頼してください。
- ・ WebブラウザにWindows(R) Internet Explorer(R) 7.0以降を使用している場合、以下のいずれかの警告メッセ  
ージが表示されていないか確認してください。警告メッセージが表示されている場合は、メッセージに表示された詳細情報、  
またはInternet Explorerのヘルプを確認し、問題を解決してください。
  - Webブラウザ画面に以下の警告メッセージが表示された場合  
“この Web サイトのセキュリティ証明書には問題があります。”
  - 情報バーに以下の警告メッセージが表示された場合  
“セキュリティ保護のため、セキュリティ証明書にエラーのあるコンテンツの表示が Internet Explorer によりブロ  
ックされました。オプションを表示するには、ここをクリックしてください。”
  - 通知バーに以下の警告メッセージが表示された場合  
“セキュリティ証明書にエラーのあるコンテンツの表示が Internet Explorer によりブロックされました。”
- ・ WebブラウザにMozilla(R) Firefoxを使用している場合、以下のいずれかの警告メッセージが表示されてい  
ないか確認してください。警告メッセージが表示されている場合は、メッセージに表示された詳細情報を確認し、問題を解決  
してください。
  - 安全な接続ができませんでした
  - 接続の安全性を確認できません

### 認証を行うシステムの解除を行ったにもかかわらず、認証画面が表示される

サインオンした状態で、認証を行うシステムの解除を行いました。ブラウザを閉じてください。

### 認証に成功した後、現在アクセスしているシングル・サインオンシステムとは異なるシングル・サインオンシ ステムの保護リソースにアクセスすると、認証に成功した認証画面とは異なる認証画面が表示される

認証を行うシステムが正しく設定されていない可能性があります。以下の手順に従い、認証を行うシステムを解除し、認  
証を行うシステムを設定しなおしてください。(注5)

1. 認証に成功した認証画面とは異なる認証画面を表示したシングル・サインオンシステムで、認証を行うシステム  
の解除を行ってください。
2. 認証に成功したシングル・サインオンシステムで、認証を行うシステムの解除を行ってください。
3. 認証を行うシステムの設定を行ってください。

### Webブラウザに「ページを表示できません」が表示され、システムのログにsso08000番台のメッセージが出 力される

Webブラウザに表示する以下のメッセージファイルのサイズが512バイト未満です。ファイルのサイズを512バイト以上  
にカスタマイズしなおしてください。(注1)

- ・ 500internal\_error\_en.html
- ・ 500internal\_error\_ja.html

### 認証を行うシステムの設定画面のURLにアクセスしても、認証を行うシステムの設定画面が表示されない 認証を行うシステムの解除画面のURLにアクセスしても、認証を行うシステムの解除画面が表示されない 保護リソースにアクセスしても、認証画面が表示されない

Webブラウザに表示するメッセージファイルのカスタマイズが正しく行われていない可能性があります。以下の同時処理  
確認画面のメッセージファイル(注1)を確認し、正しくカスタマイズしなおしてください。(注2)

- 200check\_duplication\_process\_en.html
- 200check\_duplication\_process\_ja.html

**注1)** 認証サーバ間連携時に表示されるメッセージファイルについては、“シングル・サインオン運用ガイド”の“認証サーバ間連携”―“カスタマイズ”―“Webブラウザに表示するメッセージのカスタマイズ”―“カスタマイズできるメッセージ”を参照してください。

**注2)** 認証サーバ間連携時に表示されるメッセージのカスタマイズについては、“シングル・サインオン運用ガイド”の“認証サーバ間連携”―“カスタマイズ”―“Webブラウザに表示するメッセージのカスタマイズ”―“メッセージのカスタマイズ方法”を参照してください。

**注3)** 認証サーバ間連携サービスの配備については、“シングル・サインオン運用ガイド”の“認証サーバ間連携”―“導入”―“導入手順”―“認証サーバ間連携サービスの配備”を参照してください。

**注4)** 相手シングル・サインオンシステムの認証基盤のURLについては、Interstage管理コンソールの[システム] > [セキュリティ] > [シングル・サインオン] > [認証基盤] > [認証サーバ] > [環境設定]タブで[認証サーバ間連携サービス詳細設定[表示]]をクリックし、[相手シングル・サインオンシステムの設定]の[認証基盤のURL]を参照してください。

**注5)** 認証を行うシステムの設定、および解除については、“シングル・サインオン運用ガイド”の“認証サーバ間連携”―“運用・保守”―“認証を行うシステムの設定”を参照してください。

**注6)** 相手シングル・サインオンシステムの情報については、Interstage管理コンソールの[システム] > [セキュリティ] > [シングル・サインオン] > [認証基盤] > [認証サーバ] > [環境設定]タブで[認証サーバ間連携サービス詳細設定[表示]]をクリックし、[相手シングル・サインオンシステムの情報]の[相手シングル・サインオンシステムの設定]を参照してください。

**注7)** 認証サーバ間連携サービスの停止、および起動については、“シングル・サインオン運用ガイド”の“認証サーバ間連携”―“運用・保守”―“認証サーバ間連携サービスの起動・停止・削除”を参照してください。

## 16.2.10 Interstage管理コンソールに関するトラブル

### 認証基盤の構築手順を確認したい

Interstage管理コンソールを使用して認証基盤の構築を行う手順を確認する場合は、[システム] > [セキュリティ] > [シングル・サインオン] > [認証基盤] > [認証基盤の構築]タブの[認証基盤:認証基盤の構築]画面で[ヘルプ]ボタンをクリックして表示されるヘルプを参照してください。

### 業務サーバの追加操作で警告メッセージ(sso04604、またはsso04608)が表示される

認証基盤の準備が整っていないなどの理由により、アクセス制御情報の更新に失敗しています。メッセージ中の詳細に表示されたエラーメッセージの対処を行ってください。

なお、対処を行った後に、必ず[システム] > [セキュリティ] > [シングル・サインオン] > [業務システム] > [業務システム名] > [アクセス制御情報の更新]タブの[更新]ボタンをクリックし、アクセス制御情報の更新が成功することを確認してください。

### SSOリポジトリに作成したサイト定義、パス定義が[サイト定義一覧]、[パス定義一覧]、[業務システム構築ファイルのダウンロード]の画面に表示されない

Interstage ディレクトリサービスのエントリ管理ツールやldapmodifyコマンドなどを使用してSSOリポジトリ内のサイト定義やパス定義を不正な状態に変更した可能性があります。

“シングル・サインオン運用ガイド”の“SSOリポジトリに登録するエントリの属性”を参照し、Interstage ディレクトリサービスのエントリ管理ツールやldapmodifyコマンドなどを使用してサイト定義、パス定義を正しく修正してください。SSOリポジトリのエントリ操作については、“ディレクトリサービス運用ガイド”の“エントリの管理”を参照してください。

### リポジトリサーバ起動時、Interstage管理コンソールに“ihs81364: タイムアウトが発生しました。”のメッセージが表示される Solaris32/64 Linux32/64

SSOリポジトリ内に登録しているロール定義のエントリ数、およびサイト定義のエントリ数によりInterstage HTTP Serverの起動に時間がかかる場合があります。その場合、リポジトリサーバを構築したマシンのInterstage管理コンソールに“ihs81364: タイムアウトが発生しました。”のメッセージが表示されます。システムのログを確認し、異常が発生していない場合には、しばらく時間を空けて[システム] > [サービス] > [Webサーバ] > [Webサーバ名] > [Webサーバ名:状態]画面でInterstage HTTP Serverが起動されていることを確認してください。

## Active Directoryの接続に失敗する、または正常に接続できない

マシンのネットワーク設定の変更などにより、Active Directoryの接続に失敗しています。Interstage管理コンソールを再起動した後、再度同じ操作を実施してください。

Interstage管理コンソールの再起動については、“運用ガイド(基本編)”の“Interstage管理コンソールによるInterstage運用”を参照してください。

## サイト定義、またはパス定義の表示が極端に遅い

SSOリポジトリ内に登録してあるユーザ情報、ロール定義、保護リソースなどのエントリ数が多いため、サイト定義、またはパス定義の検索に多くの時間がかかっています。

サイト定義、またはパス定義の情報を参照するのみの場合は、Interstage ディレクトリサービスのエントリ管理ツールやldapsearchコマンドなどを使用してください。サイト定義、またはパス定義は、“保護リソースの登録先エントリ”配下に登録されています。(注)

注)“保護リソースの登録先エントリ”については、[システム]>[セキュリティ]>[シングル・サインオン]>[認証基盤]>[リポジトリサーバ]>[環境設定]タブの[リポジトリサーバ詳細設定[表示]]をクリックし、[リポジトリ]の[保護リソースの登録先エントリ]を参照してください。

## 16.2.11 Interstage ディレクトリサービスのレプリケーション機能使用時のトラブル

### ロール定義を変更／追加し、Interstage管理コンソールでロール情報を更新したが、変更／追加内容が有効にならない

Interstage ディレクトリサービスのレプリケーション機能を使用し、SSOリポジトリのデータの同期を取っている場合、ネットワークの負荷などにより同期が取れるまでに数秒かかることがあります。

同期が取れる前にロール情報を更新すると、ロール定義の内容が正しく反映されません。

この場合、更新系リポジトリサーバのSSOリポジトリの変更内容が、参照系リポジトリサーバのSSOリポジトリに反映されていることを確認した後に、ロール情報を更新してください。

ロール定義の変更、追加については、“シングル・サインオン運用ガイド”の“運用・保守”-“認可に関する操作”-“ロール定義の変更、追加”を参照してください。

SSOリポジトリの登録内容の確認については、エントリ管理ツールなどを使用してください。詳細については、“ディレクトリサービス運用ガイド”の“エントリの管理”を参照してください。

レプリケーション機能については、“ディレクトリサービス運用ガイド”の“概要”を参照してください。

## 16.2.12 旧バージョンからの移行に関するトラブル

### 旧バージョンからの移行後、Interstage管理コンソールの操作、またはリポジトリサーバ、認証サーバ、および業務サーバの起動に失敗する

以下の点に該当する場合、移行手順に誤りがある可能性があります。移行手順に誤りがないか確認してください(注1)

- Interstage管理コンソールの操作中に、「環境が壊れています」という内容のエラーメッセージ(sso04000、sso04304、sso04504、sso04704など)が表示されます。
- リポジトリサーバ、認証サーバ、および業務サーバの起動に失敗します。
- リポジトリサーバ、認証サーバ、および業務サーバへの接続に失敗します。
- 移行時に編集した環境定義ファイルの定義項目が、エラーメッセージ、またはシステムのログに表示されます。

### 旧バージョンからの移行後、アクセスログが出力されない

旧バージョンから移行する際に、環境定義ファイルのアクセスログの定義を正しく設定していない可能性があります。移行手順を参照(注1)し、アクセスログの設定を確認してください。

なお、移行後の環境にて、リポジトリサーバ(参照系)が使用されない運用を行っている場合、リポジトリサーバ(参照系)のアクセスログは出力されません。

リポジトリサーバ(参照系)が使用されない運用については、“移行ガイド”の“Interstage運用操作、環境の移行” – “Interstage シングル・サインオンの移行”の“■旧バージョン・レベルからのサーバ環境の移行”を参照してください。

### **旧バージョンからの移行後、認証に失敗し、システムのログにsso00002が出力される、またはWebサーバ(Interstage HTTP Server)のエラーログに“SSL: Authority error”が表示される。(注2)**

旧バージョンから移行する際に、Webサーバ(Interstage HTTP Server)の移行が正しく行われていない可能性があります。Webサーバ(Interstage HTTP Server)の移行手順に誤りがないか確認してください。(注3)

**注1)**各サーバの移行手順については、“移行ガイド”の“Interstage シングル・サインオンの移行”を参照してください。

**注2)**Webサーバ(Interstage HTTP Server)のエラーログについては、“メッセージ集”の“Interstage HTTP Serverが出力するメッセージ” – “Interstage HTTP Serverのエラーログ”を参照してください。

**注3)**Webサーバ(Interstage HTTP Server)の移行については、“移行ガイド”の“Interstage HTTP Serverの移行”を参照してください。

## 第17章 MessageQueueDirector運用時の異常 Windows32/64

Solaris32 Linux32/64

本章ではMessageQueueDirector(以下MQD)運用時に発生するトラブルの事例および対処について説明します。MQDの機能詳細などについては、“MessageQueueDirector説明書”を参照してください。

イベントチャネル連携サービスを利用する場合は、イベントサービスを利用しますので、トラブル発生時には、イベントサービスのトラブルシューティングも参照してください。

### 17.1 イベントチャネル連携サービス運用時の異常

イベントチャネル連携サービスの運用中に発生するトラブルの事例およびその対処について説明します。

#### イベントチャネル連携サービスを起動後、mqdmsgwprtコマンドで通信状態を確認時、ConnectではなくSuspend (MQD6352)が表示される。

上記現象が発生した場合、以下の可能性があります。

- ・ 相手サーバ側のイベントチャネル、またはメッセージキューが作成されていない。
  - イベントチャネルを作成していない場合には、イベントチャネルを作成後に起動してください。メッセージキューを作成していない場合には、メッセージキューを作成してください。
- ・ 相手サーバ側のイベントチャネルが起動していない。
  - イベントチャネルを起動してください。
- ・ 送信側のサービス定義で設定した相手サーバ側のイベントチャネル名、もしくはメッセージキュー名と、実際に作成した相手サーバ側のイベントチャネル名、もしくはメッセージキュー名が一致していない。
  - 送信側サービス定義を見直して、サービス定義が間違っている場合には、環境の変更をしてください。相手サーバ側のイベントチャネル名、もしくはメッセージキュー名が間違っている場合には、それぞれ再作成してください。
- ・ 相手サーバ側の受信側サービス定義で設定した集信数を超える数の通信を要求した。
  - 以下のいずれかの可能性があります。
    - イベントチャネル連携サービスの受信側サービス定義作成時に、recvnumキーワードの見積もりを誤った。
      - 受信側サービス定義のrecvnumキーワード値の見積もりを実施してください。
    - イベントチャネル連携サービスのシステム構成上、想定されていない送信側サーバからの通信要求が受信側サーバに届いた。
      - 誤った送信サービス定義で運用している送信側サーバを特定し、環境の変更を行ってください。
    - 運用中のチャネルコネクションの初期化をしないままイベントチャネル連携サービスの環境変更をしたために、相手サーバ側に、変更前のチャネルコネクション情報が残ってしまった。
      - 相手サーバ側のチャネルコネクション情報を削除してください。

#### イベントチャネル連携サービスを起動後、mqdmsgwprtコマンドで通信状態を確認時、ConnectではなくSuspend (MQD6470)が表示される。

以下の可能性があります。それぞれの項目を見直して再実行してください。

- ・ 相手サーバが起動していない。
  - 相手サーバを起動してください。

- URLリストファイルを使用して、イベントチャンネル連携サービスの通信を行っている場合に、hostsファイルの記述が正しくされていない、またはURLリストファイルの記述に誤りがある。  
→ hostsファイル、またはURLリストファイルの見直しを行ってください。
- キュー間コネクションのみを運用している場合に、Interstageが起動していない可能性があります。  
→ Interstageを起動してください。
- 何らかの事象により、通信のセッションが切れてしまった。  
→ チャンネルコネクションを初期化後、再開してください。
- ネーミングサービス方式を使用して、イベントチャンネル連携サービスの通信を行っている場合、URLリストファイルで指定したネーミングサービスに受信側のイベントチャンネル連携サービスが登録されていない。  
→ URLリストファイルに指定したネーミングサービスにイベントチャンネル連携サービスの受信側の位置情報が正しく登録されているか、odlistnsコマンドで確認してください。  
相手サーバのイベントチャンネル連携サービスの受信側の位置情報が登録されていない場合、または、正しく登録されていない場合、イベントチャンネル連携サービスのCORBAサービスへの登録を正しく行ってください。  
詳細は、“MessageQueueDirector説明書”の“イベントチャンネル連携サービス”の“環境作成”の“イベントチャンネル連携サービスのCORBAサービスへの登録”を参照してください。  
odlistnsコマンドの詳細については、“リファレンスマニュアル(コマンド編)”を参照してください。
- IORファイル方式を使用して、イベントチャンネル連携サービスの通信を行っている場合、指定したIORファイルが相手サーバのイベントチャンネル連携サービスのIORファイルではない。  
→ 相手サーバのイベントチャンネル連携サービスのCORBAサービスへの登録時に作成したIORか確認してください。
- イベントチャンネル連携サービスの受信側の位置情報(ポート番号)が正しく登録されていない。  
→ イベントチャンネル連携サービスの受信側の位置情報(ポート番号)が正しく登録されているか、odlistnsコマンドで確認してください。  
相手サーバのイベントチャンネル連携サービスの受信側の位置情報(ポート番号)が正しく登録されていない場合、イベントチャンネル連携サービスのCORBAサービスへの登録を正しく行ってください。  
詳細は、“MessageQueueDirector説明書”の“イベントチャンネル連携サービス”の“環境作成”の“イベントチャンネル連携サービスのCORBAサービスへの登録”を参照してください。  
odlistnsコマンドの詳細については、“リファレンスマニュアル(コマンド編)”を参照してください。

### イベントチャンネル連携サービスを起動後、mqdmsgwprtコマンドで通信状態を確認時、ConnectではなくSuspend (MQD6459)が表示される。

以下の可能性があります。それぞれの項目を見直して再実行してください。

- イベントチャンネルを作成する時、-tranオプション、-otsオプションのどちらのオプションも付けずに作成した。  
→ イベントチャンネルがイベントチャンネル連携サービスで使用するための正しい設定で作成されているか、esstetcnfchnlコマンドで以下の項目のどちらかが“yes”に設定されているか確認してください。
  - local transaction mode
  - ots use
 イベントチャンネルがイベントチャンネル連携サービスで使用するための正しい設定で作成されていない場合、正しい設定で再作成してください。  
esstetcnfchnlコマンドの詳細については、“リファレンスマニュアル(コマンド編)”を参照してください。
- イベントチャンネルを作成する時、-notifyオプションを付けずに作成した。  
→ イベントチャンネルがイベントチャンネル連携サービスで使用するための正しい設定で作成されているか、esstetcnfchnlコマンドで以下の項目が“yes”に設定されているか確認してください。
  - Notification channel mode

イベントチャンネルがイベントチャンネル連携サービスで使用するための正しい設定で作成されていない場合、正しい設定で再作成してください。

esstcnfchnlコマンドの詳細については、“リファレンスマニュアル(コマンド編)”を参照してください。

イベントチャンネル連携サービスで使用するイベントチャンネルの正しい設定についての詳細は、“MessageQueueDirector説明書”の“イベントチャンネル連携サービス”の“環境作成”の“環境作成の説明”の“送信キューと受信キューの作成”を参照してください。

### 相手サーバ側のイベントチャンネル連携サービスを再起動後、コネクションの再開ができない。

以下の可能性があります。その場合には、送信サーバにおいてmqdmsgwcommコマンドを-aオプションで実行し、チャンネルコネクションの状態がSuspendとなったチャンネルコネクションの通信を再開してください。

- ・ 通信中に、相手サーバ側のイベントチャンネル連携サービスを再起動した。
- ・ 通信中に、クラスタシステムによるクラスタ遷移をした。
- ・ 相手サーバを起動せずに送信サーバを先に起動したために、チャンネルコネクション状態がRetryとなり、1分おきに通信再開の試みが自動的に行われたが、その回数が送信側サービス定義のMNGRセクションのerrorretryキーワードの値を超えてしまった。

### 受信サーバ側で、イベントデータが正しく受信されない。

以下の可能性があります。

- ・ チャンネル間通信を使用し、かつワイド文字列型のデータの通信を行っている場合に、双方の環境変数に“OD\_CODE\_SET”を設定していなかった。  
→ 環境変数“OD\_CODE\_SET”を設定してください。
- ・ 送信側サービス定義上のチャンネルコネクション定義のイベントデータの形式がstructuredに設定されているのに、any型のイベントデータを通信しようとしていた。または、イベントデータの形式がanyに設定されているのに、structured型のイベントデータを通信しようとした。  
→ イベントデータの形式をあわせてください。

### イベントチャンネル連携サービスを起動後、CORBAサービスの例外情報として“IDL:CORBA/StExcept/DATA\_CONVERSION:1.0”が通知される。

以下の可能性があります。

- ・ 環境変数“OD\_CODE\_SET”の値とマシンのコード系の値が違う。

#### Windows32/64 Solaris32/64

→ワイド文字列型のデータの通信を行う場合には、環境変数“OD\_CODE\_SET”、マシンのコード系、およびイベントチャンネル作成時の日本語コード系の値として、EUCまたはSJISを設定してください。

なお、環境変数“OD\_CODE\_SET”は、MQDを起動するシェルに設定してください。

また、esmkchnlコマンドを用いてイベントチャンネルを作成する場合には、オプション-lにEUCまたはSJISを設定してください。コマンドの詳細については、“リファレンスマニュアル(コマンド編)”を参照してください。Interstage管理コンソールを用いてイベントチャンネルを作成する場合には、“詳細設定”の“日本語コード系”にEUCまたはShift JISを設定してください。Interstage管理コンソールの詳細については、“Interstage管理コンソール ヘルプ”を参照してください。

#### Linux32/64

→ワイド文字列型のデータの通信を行う場合には、環境変数“OD\_CODE\_SET”、マシンのコード系、およびイベントチャンネル作成時の日本語コード系の値として、EUCを設定してください。

なお、環境変数“OD\_CODE\_SET”は、MQDを起動するシェルに設定してください。

また、esmkchnlコマンドを用いてイベントチャンネルを作成する場合には、オプション-lにEUCを設定してください。コマンドの詳細については、“リファレンスマニュアル(コマンド編)”を参照してください。Interstage管理コンソールを用い

てイベントチャネルを作成する場合には、“詳細設定”の“日本語コード系”にEUCを設定してください。Interstage管理コンソールの詳細については、“Interstage管理コンソール ヘルプ”を参照してください。

### イベントチャネル連携サービスを起動後、mqdmsgwprtコマンドで通信状態を確認時、Connectではなく Suspend (MQD6352)が表示され、かつ、出力されたメッセージMQD6352の可変情報(msgid)がMQD6483となっている。

上記現象が発生した場合、以下の可能性があります。

- 送信側サーバでチャネルコネクションの初期化が行われていない状態で送信側サーバまたは受信側サーバのイベントチャネル連携サービスのサービス定義が変更されました。以下の方法で復旧してください。
  - チャネルコネクションを初期化する。  
# mqdmsgwcomm -s MQDシステム名 -r \*  
チャネルコネクションの初期化に失敗した場合、強制モードで再度実行する。  
# mqdmsgwcomm -s MQDシステム名 -c \*
  - 送信側サーバ、受信側サーバ共にイベントチャネル連携サービスを停止する。  
# mqdstpsvc -s MQDシステム名 NSGW)
  - 送信側サーバ、受信側サーバ共にイベントチャネル連携サービスのサービス環境を削除する  
# mqdmsgwunsetup -s MQDシステム名  
環境削除が失敗した場合、強制モードで再度実行する。  
# mqdmsgwunsetup -s MQDシステム名 -c
  - 送信側サーバ、受信側サーバ共にイベントチャネル連携サービスのサービス環境を作成する  
# mqdmsgwsetup -s MQDシステム名
  - 送信側サーバ、受信側サーバ共にイベントチャネル連携サービスのサービス定義を登録する  
# mqdmsgwdef -s MQDシステム名 -a -f サービス定義ファイル名
  - 送信側サーバ、受信側サーバ共にイベントチャネル連携サービスを起動する  
# mqdstrsvc -s MQDシステム名 -n NSGW

## 17.2 アプリケーション実行時の異常

MQDのアプリケーション使用した場合に発生する、トラブルの事例およびその対処について説明します。

### アプリケーションが異常終了する。

以下の可能性があります。それぞれアプリケーションを確認/修正して再実行してください。

- アプリケーションがマルチスレッド用に作成されていない。その場合アプリケーションの動作は保証できず、突然異常終了する場合があります。
- マルチスレッドアンセーフな関数を使用している。OSによってマルチスレッドアンセーフ関数の種類は異なります。マルチスレッドアンセーフな関数を使用すると、アプリケーションの動作に影響を及ぼす場合があります。
- Java からJNI経由でMQDのAPIを利用している。MQDはJavaから利用する形態の動作保証をしていません。
- アプリケーションによってメモリ破壊が行われている。メモリ破壊がないことを調査してください。
- アプリケーションがメモリの開放漏れを起こしている。開放漏れがメモリ空間を圧迫し、最終的にアプリケーションが異常終了する。アプリケーションのメモリークについて調査してください。

#### Solaris32

- シグナルの利用方法が誤っている。シグナルの使用方法が誤っているとアプリケーションの動作に影響を及ぼす場合があります。

### アプリケーションがエラーコード 26 (MQD\_ERR\_OVERTRAN)、36 (MQD\_ERR\_PROCESSOVER) で異常終了する。

1MQDシステムで同時に実行できるMQDアプリケーション数の限界に達したまたは、1MQDシステムで同時に実行できるトランザクションの最大に達しました。以下を確認して再実行してください。

- ・ 起動しているMQDアプリケーションの数が多すぎないか。
- ・ 実行しているトランザクションの数が多すぎないか。

また、以下のようにMQD-APIを使用すると、本エラーが発生する可能性があります。その場合にはアプリケーションを修正してください。

- ・ トランザクションの取り消しを、短い期間に連続して実行する。
- ・ メッセージ操作を伴わないトランザクションの開始、終了、取り消しを、短い期間に連続して実行する。
- ・ メッセージの存在しないメッセージキューに対する非同期モードでのメッセージの受信、参照、カーソル移動、削除を、短い期間に連続して実行する。
- ・ メッセージ参照のみを、短い期間に連続して実行する。

### アプリケーションがエラーコード 33 (MQD\_ERR\_NOMEMORY) で異常終了する。

以下の可能性があります。それぞれについて確認／修正して再実行してください。

- ・ メッセージバッファの最大長が不足している。MQD環境作成時の [MQDConfiguration] セクションの MessageBufferSize を見直してください。
- ・ 1トランザクションで送受信するメッセージ長の合計の最大値を超えた。MQD環境作成時の [Transaction] セクションの Max\_Size を見直してください。
- ・ ヒープ、実メモリ、スワップ領域が足りない。システム全体のメモリ量を見直してください。

### アプリケーションがエラーコード 37 (MQD\_ERR\_ACCESSDENY) で異常終了する。

MQDの各種資源に対するアクセス権限がありません。以下を確認し、再実行してください。

- ・ アプリケーションの実行者がスーパーユーザ(管理者権限をもっている)、もしくは、MQD運用操作グループに属しているか。
- ・ MQDの各種ファイル資源のアクセス権限が書き換えられていないか。

### アプリケーションがエラーコード 52 (MQD\_ERR\_UNITSUMOVER) で異常終了する。

トランザクションサイズの最大に達しました。以下を確認し、再実行してください。

- ・ 1ユニットまたは1トランザクションで扱える最大のサイズを超えました。
- ・ MQD環境作成時のTransactionセクションの Max\_Size を超えるサイズのメッセージを取り扱いました。

### アプリケーションがエラーコード 113 (MQD\_ERR\_QNAME) で異常終了する。

メッセージキュー名が不当です。以下を確認し、再実行してください。

- ・ メッセージ送信の際、メッセージヘッダの返信キュー名を使用しない場合にはNULL文字を設定してください。

### アプリケーションがエラーコード xx で異常終了する。

マニュアルを参照して異常事象 xx に対する対処を実施したあと、再実行してください。

COBOL言語インタフェースの説明と復帰情報の説明のところではエラーコードを10進の数値で説明しています。C言語インタフェース用includeファイルではマクロ名との関連づけを16進数で記述していますので、注意してください。

## 17.3 コマンド実行時の異常

---

MQDの各種コマンドを実行した場合に発生する、トラブルの事例およびその対処について説明します。

### コマンド実行時に負荷が高いという意味のメッセージが出力される。

通常は実行しているコマンドプロセスやアプリケーションプロセスの数が多すぎる場合に出力されますが、MQDのアプリケーションで、以下にあげるようなMQD-APIの使い方をしている場合、アプリケーションプロセスの数が少なくても、出力される場合があります。その場合にも、当該アプリケーションを終了させてから、再度実行してください。また、このようなAPIの使い方は誤った使い方ですので、アプリケーションは修正してください。

- ・ トランザクションの取り消しを、短い期間に連続して実行する。
- ・ メッセージ操作を伴わないトランザクションの開始、終了、取り消しを、短い期間に連続して実行する。
- ・ メッセージの存在しないメッセージキューに対する非同期モードでのメッセージの受信、参照、カーソル移動、削除を、短い期間に連続して実行する。
- ・ メッセージ参照のみを、短い期間に連続して実行する。

## 第18章 Interstage ディレクトリサービス使用時の異常

Interstage ディレクトリサービスの使用でトラブルが発生した場合の対処方法について説明します。

### 18.1 リポジトリを起動できない

リポジトリが起動できない場合の対処方法を説明します。

- リポジトリ作成時には存在していなかった同一のポート番号が、他で使用されている可能性があります。起動しているポートを停止するか、またはリポジトリを再作成してポート番号を変更してください。
- リポジトリのデータベースにRDBを使用して、RDBのサービスの起動を待ち合わせるように設定(irepadminコマンドの-rdbwaitオプションに「yes」を設定)している場合は、RDBのサービスが起動していない可能性があります。RDBの各サービスが起動しているかを確認してください。

Oracleデータベースを使用している場合に、データベース、リスナーともに起動しているにもかかわらず、リポジトリが起動しない場合は、リスナー定義に使用するデータベースの情報が登録されていない可能性があります。リスナー制御ユーティリティのServicesコマンドで状態を確認し、使用するデータベースの情報が登録されていない場合は、データベースを再起動してください。使用するデータベースの情報が登録されている場合は、通信の設定が正しく行われているか再度確認してください。

- Solaris32/64** **Linux32/64**  
omsアカウント権限で動作するプロセスの最大ファイルディスクリプタ数のハードリミットが、65536より大きい可能性があります。「チューニングガイド」の「システムのチューニング」-「Interstage ディレクトリサービスのシステム資源の設定」を参照し、設定値を見直してください。
- Linux32/64**  
omsアカウントのログインシェルでscriptコマンドを実行している場合、OS起動時にInterstage data storeサービスが起動されません。omsアカウントのログインシェルでscriptコマンドを実行していないか確認してください。実行している場合は、scriptコマンドを実行しないように対処してください。

### 18.2 LDAPコマンドが正常に動作しない

LDAPコマンド(ldapdelete、ldapmodify、ldapsearch)が正常に動作しない場合の対処方法を説明します。

#### LDAPコマンド(ldapdelete、ldapmodify、ldapsearch)のオプションが正常に動作しない

##### **Windows32/64**

環境変数Pathの設定により、同じマシンにインストールされている他製品に付属のLDAPコマンドが実行される場合があります。

- 他製品に付属するLDAPコマンドと、Interstage ディレクトリサービスのLDAPコマンドの両方を使用する必要がある場合  
Interstage ディレクトリサービスのLDAPコマンドを絶対パス[C:¥Interstage¥bin¥LDAPコマンド名]で指定してください。
- 他製品に付属するLDAPコマンドを使用する必要がない場合  
環境変数Pathに、Interstage ディレクトリサービスのLDAPコマンドがインストールされているパス[C:¥Interstage¥bin]を、他製品のLDAPコマンドのパスより前に設定してください。

##### **Solaris32/64** **Linux32/64**

環境変数PATHの設定により、OS付属のLDAPコマンドや、同じマシンにインストールされている他製品に付属のLDAPコマンドが実行される場合があります。

- OS付属のLDAPコマンド、または他製品に付属するLDAPコマンドと、Interstage ディレクトリサービスのLDAPコマンドの両方を使用する必要がある場合

Interstage ディレクトリサービスのLDAPコマンドを絶対パス[/opt/FJSVirepc/bin/LDAPコマンド名]で指定してください。

- OS付属のLDAPコマンド、または他製品に付属するLDAPコマンドを使用する必要がない場合

環境変数PATHに、Interstage ディレクトリサービスのLDAPコマンドがインストールされているパス [/opt/FJSVirepc/bin]を、[/usr/bin]と[/bin]、および他製品のLDAPコマンドのパスより前に設定してください。

## 18.3 リポジトリの応答が遅い

---

リポジトリの応答が遅い場合の対処方法を説明します。

- [リポジトリの起動・停止の応答が遅い](#)
- [検索の応答が遅い](#)
- [更新系処理の応答が遅い](#)
- [レプリケーション処理が遅れている、またはスレーブ側リポジトリにデータが反映されない](#)

### リポジトリの起動・停止の応答が遅い

リポジトリ内部の整合性を調整しています。調整が終了するまでお待ちください。調整が終了しない場合は、技術員に連絡してください。

### 検索の応答が遅い

- 検索条件を絞ってください。または、検索結果で通知してほしい属性を指定し、通知される属性数を絞ってください。検索性能のチューニング方法については、「ディレクトリサービス運用ガイド」の「検索のチューニング」を参照してください。
- サーバが混雑している可能性があります。レプリケーション運用の場合、別のサーバで検索してください。
- ネットワークが混雑している可能性があります。ネットワークの状態を調べてください。
- リポジトリのデータベースにSymfoware Server Lite Edition以外のSymfoware/RDBを使用している場合は、データベースの最適化処理を行ってください。データベースの最適化処理の詳細については、「ディレクトリサービス運用ガイド」の「データベースの最適化」を参照してください。
- リポジトリのデータベースにSymfoware Server Lite Editionを使用している場合は、Symfoware Server Lite Editionの使用想定規模の範囲内でデータベースを使用しているかを確認してください。Symfoware Server Lite Editionの使用想定規模の範囲を超えてデータベースを使用している場合は、上位Editionへの移行をご検討ください。Symfoware Server Lite Editionの使用想定規模の範囲については、「使用上の注意」の「注意事項」-「Interstage ディレクトリサービスの注意事項」-「リポジトリのデータベースにRDBを利用する場合の注意事項」-「使用想定規模に関する注意事項」を参照してください。
- リポジトリのデータベースにOracleデータベースを使用している場合は、データベースの統計情報を採取してください。統計情報の詳細については、Oracleデータベースのマニュアルを参照してください。

### 更新系処理の応答が遅い

- サーバが混雑している可能性があります。しばらく時間がたってから、もう一度実行してください。
- ネットワークが混雑している可能性があります。しばらく時間がたってから、もう一度実行してください。

- リポジトリのデータベースにSymfoware Server Lite Editionを使用している場合は、以下を行ってください。
  - エントリ更新時に数秒の遅延が発生する場合があります。Symfoware Server Lite Editionは検索中心の使用を想定しているため、頻繁にデータ更新を行う場合は、上位Editionへの移行を検討してください。
  - ディスク領域の自動拡張を行うため、リポジトリの応答が数十秒程度遅くなることがあります。この場合、Symfoware Server Lite Editionを運用しているマシンのイベントログにメッセージ(qdg02868i、qdg02865i)が出力されます。本遅延が連続して発生する場合は、iscollectinfoコマンドを使用して調査情報を採取した後、技術員に連絡してください。

### レプリケーション処理が遅れている、またはスレーブ側リポジトリにデータが反映されない

標準データベースを使用している場合は、マスタ側のシステムログを参照し、irep15000番台のメッセージが存在しないかを確認してください。

存在する場合は、「メッセージ集」の「irep15000番台のメッセージ」を参照して、該当するメッセージ番号の[ユーザの対処]に記述された方法で対応してください。

存在しない場合は、レプリケーション処理が遅れている可能性があります。しばらく待ってから、再度スレーブ側のリポジトリを参照し、データが反映されているか確認してください。

それでもデータが反映されていない場合は、マスタ側とスレーブ側の双方でiscollectinfoコマンドを使用して調査情報を採取した後、技術員に連絡してください。

## 18.4 リポジトリの応答がない

---

リポジトリの応答がない場合の対処方法を説明します。

リポジトリのデータベースにRDBを使用している場合に、リポジトリまたはInterstage ディレクトリサービス運用コマンドが応答しない場合は、RDBからの応答がない可能性があります。

Symfoware Serverを使用している場合は、Symfoware Serverのマニュアル「トラブルシューティング集」の「無応答状態の対処について」を参照して対処してください。

Oracleデータベースを使用している場合は、Oracleデータベースのマニュアルを参照してください。

### Solaris32/64 Linux32/64

リポジトリのデータベースとして標準データベースを使用している場合には、標準データベースであるInterstage data storeが起動していない可能性がありますので、以下の対処をしてください。

- omsアカウント権限で動作するプロセスの最大ファイルディスクリプタ数のハードリミットが、65536より大きい可能性があります。「チューニングガイド」の「システムのチューニング」-「Interstageディレクトリサービスのシステム資源の設定」を参照して、設定値を見直してください。

## 18.5 トップエントリを操作できない

---

トップエントリ(公開ディレクトリ)を削除したり、名前を変更したりすることはできません。

## 18.6 JNDIアプリケーションで例外が発生する

---

JNDIアプリケーションで例外が発生する場合の対処方法を説明します。

Oracle Corporationのホームページから、以下の文書を参照して対処してください。

- The JNDI Tutorial

- Java 2 プラットフォームAPI仕様(javadoc)

## CommunicationException が発生する

CommunicationException が発生した場合は、次の対処をしてください。

JNDIを使用しているアプリケーションが動作しているマシン上でpingコマンドを使用し、アクセスしたいリポジトリが存在するマシンとネットワーク接続されているかを確認してください。

ネットワーク接続されている場合は、リポジトリが起動しているかを確認してください。

リポジトリが起動していない場合は、Interstage管理コンソールからリポジトリを起動後、再度実行してください。

リポジトリが起動している場合は、アクセスログを参照してタイムアウトが発生していないかを確認してください。アクセスログに「TIMEOUT」が記録されていた場合は、タイムアウトが発生しています。アクセスログについては、「ディレクトリサービス運用ガイド」の「リポジトリの運用監視」を参照してください。

タイムアウトの原因は、ネットワークの輻輳(通信量が増加し、正常に通信できない状態)、リポジトリが起動しているマシンの高負荷、アプリケーションが動作しているマシンの高負荷などが考えられます。

Interstage ディレクトリサービスが過負荷状態の場合は、レプリケーションを使った負荷分散を実施してください。また、アプリケーション側の負荷が高い場合は、アプリケーションの実行環境を見直して、適切な状態で使用してください。

上記以外の場合は、iscollectinfoコマンドを使用して調査情報を採取した後、技術員に連絡してください。

## 18.7 登録データが文字化けしている

登録データが文字化けしている場合の対処方法を説明します。

- Oracleデータベース使用時に、日本語を含むエントリを登録すると文字化けする
- コマンドを使用して移入・移出をした場合に、文字化けした情報が登録、または出力される
- 移出したLDIFファイルをテキストエディタで開くと文字化けしている

### Oracleデータベース使用時に、日本語を含むエントリを登録すると文字化けする

Oracleデータベースの構築時のキャラクタセットが日本語をサポートしないキャラクタセット(ASCIIなど)になっています。日本語を含むエントリを作成する場合は、日本語をサポートしているキャラクタセットを指定してデータベースを構築してください。

データベースのコード系は、以下のSQL文で確認できます。

```
SELECT VALUE FROM NLS_DATABASE_PARAMETERS WHERE PARAMETER = 'NLS_CHARACTERSET' ;
```

### コマンドを使用して移入・移出をした場合に、文字化けした情報が登録、または出力される

コマンド実行時のコード系の指定に誤りがあります。ldapmodifyコマンド、およびldapsearchコマンドの場合は、-Gオプション、irepaddroleコマンド、およびirepmodifyentコマンドの場合は、-Cオプションをそれぞれ使用して、コード系を指定してください。

エントリ管理ツールでは、移入するLDIFファイルのコード系(日本語環境の場合は、Shift-JIS、EUC-JP、またはUTF8、英語環境の場合は、ISO-8859-1、またはUTF-8)を指定できます。移入時コード系が不明な場合は、各コード系の指定を変えながら、移入処理をしてください。

### 移出したLDIFファイルをテキストエディタで開くと文字化けしている

コマンド、またはエントリ管理ツールを使用して移出し、移出したLDIFファイルをテキストエディタで開くと文字化けしていることがあります。

テキストエディタに文字コード自動認識機能がある場合、正しく認識されていない可能性があります。テキストエディタで、移出時に指定したコード系を指定して、再度ファイルを開いてください。

## 18.8 アクセスログを読めない

---

アクセスログを読めない場合の対処方法を説明します。

- ・ [アクセスログが見えない](#)
- ・ [アクセスログが文字化けする](#)

### アクセスログが見えない

アクセスログの[出力の指定]に「出力しない」が設定されています。「出力する」を設定してください。

### アクセスログが文字化けする

アクセスログは、「UTF-8形式」で出力します。

アクセスログにマルチバイト文字が含まれている場合は、「UTF-8形式」に対応したエディタなどで参照してください。「UTF-8形式」が扱えない場合は、アクセスログのコード系を「UTF-8形式」からシステムのコード系に合わせて変換して参照してください。

## 18.9 データファイルを読めない

---

コマンド、またはエントリ管理ツールで、LDIFファイル、ルールファイル、またはCSVファイルを読み込むことができない場合の対処方法を説明します。

ファイルのコード系がUTF-8のLDIFファイル、ルールファイル、またはCSVファイルを、Windows(R)のメモ帳で作成／編集した場合、コマンド、またはエントリ管理ツールでファイルを読み込むことができません。

メモ帳以外のコード系が「UTF-8」に対応したテキストエディタを使用してファイルを作成／編集してください。

## 18.10 クラスタ運用時の異常

---

クラスタ運用時にトラブルが発生した場合の対処方法を説明します。

**オフライン中のノードに対してオフライン実行した場合に、「FJSVcluster: エラー: clmonproc: XXXX: 指定されたリソースは監視されていません。(rid:YYYY)」というコンソールメッセージが表示される** Solaris32/64

Linux32/64

XXXXはclmonprocのPID、YYYYはIREP\_INTERSTAGEのRIDを表します。

サンプルの状態遷移プロシジャ(IREP\_INTERSTAGE)を利用した場合、上記のメッセージが表示されますが、動作には影響しないため、無視してください。

## 18.11 アプリケーションの移行時の異常

---

C言語アプリケーションの移行時にトラブルが発生した場合の対処方法を説明します。

- ・ [移行したアプリケーションが異常終了する](#)

- ・ 移行したアプリケーションのコンパイルに失敗する

### 移行したアプリケーションが異常終了する

以下のマニュアルを参照して、機能差異に該当する関数を使用していないかを確認してください。

- ・ 「移行ガイド」の「旧機能から新機能への移行方法」－「InfoDirectoryからInterstage ディレクトリサービスへの移行」－「C言語アプリケーションの移行手順」
- ・ 「移行ガイド」の「Interstage運用操作、環境の移行」－「Interstage ディレクトリサービスの移行」

### 移行したアプリケーションのコンパイルに失敗する

以下のマニュアルを参照して、機能差異に該当する関数を使用していないかを確認してください。

- ・ 「移行ガイド」の「旧機能から新機能への移行方法」－「InfoDirectoryからInterstage ディレクトリサービスへの移行」－「C言語アプリケーションの移行手順」
- ・ 「移行ガイド」の「Interstage運用操作、環境の移行」－「Interstage ディレクトリサービスの移行」

## 18.12 テーブル作成コマンド実行時にエラーが発生する

リポジトリのデータベースとしてSymfoware/RDBを使用する場合の環境構築で、テーブル作成コマンド (irepcttbl/irepgendb) 実行時にエラーが発生した場合の対処方法を説明します。

テーブル作成コマンドの実行環境を、Interstage ディレクトリサービスをインストールしたマシンとは異なるWindows(R)のマシンにコピーして、テーブル作成コマンドを実行した場合に、「ダイナミックリンクライブラリMSVCR80.dllが指定されたパス～<パスの文字列>～に見つかりませんでした。」というDLLが見つからないことを示すエラーが発生する [Windows32/64](#)

テーブル作成コマンドの実行環境を、Interstage ディレクトリサービスをインストールしたマシンとは異なるWindows(R)のマシンにコピーして使用する場合は、以下のパッケージがインストールされている必要があります。インストールされていない場合は、Microsoftのホームページからダウンロードしてインストールしてください。

- ・ Microsoft Visual C++ 2005 再頒布可能パッケージ (x86)

## 18.13 テーブル作成コマンド実行時にデータベースのエラーが発生する

リポジトリのデータベースとしてRDBを使用する場合の環境構築で、テーブル作成コマンド実行時にデータベースのエラーが発生した場合の対処方法を説明します。

### ■リポジトリのデータベースにSymfoware Serverを使用している場合

Symfoware Serverを使用時にデータベースのエラーが発生した場合は、Symfoware Serverのマニュアルを参照して対処してください。

以下に、主なエラーと対処方法を説明します。

- ・ rdb: ERROR: qdg12226e:DSI定義文の実行で重症エラーを検出しました 詳細メッセージ='JYP4801E 指定されたデータベーススペースの空き領域が不足しています.' エラーが発生した文の先頭位置=%s1 (システム名=%s2)
- ・ rdb: ERROR: qdg12226e:DSI定義文の実行で重症エラーを検出しました 詳細メッセージ='JYP4204E 指定されたデータベーススペース“DSDBSPACE”が、定義されていません.' エラーが発生した文の先頭位置=%s1 (システム名=%s2)

- **rdp: ERROR: qdg12226e:表定義文の実行で重症エラーを検出しました 詳細メッセージ=JYP4784E データベース%s1は定義されていません. 'エラーが発生した文の先頭位置=%s2 (システム名=%s3)**
- **rdp: ERROR: qdg12226e:DSI定義文の実行で重症エラーを検出しました 詳細メッセージ=JYP5071E アーカイブログファイルの不足が発生しました. 'エラーが発生した文の先頭位置=%s1 (システム名=%s2)**

**rdp: ERROR: qdg12226e:DSI定義文の実行で重症エラーを検出しました 詳細メッセージ=JYP4801E 指定されたデータベーススペースの空き領域が不足しています. 'エラーが発生した文の先頭位置=%s1 (システム名=%s2)**

[意味と対処]

%s1は、エラーが発生した文の先頭位置です。

%s2は、RDBシステム名です。

データベーススペースの空き領域が不足しています。

Interstage ディレクトリサービスのデータベース作成に必要なディスク容量を確保してください。必要なディスクスペースの見積もりは、“ディレクトリサービス運用ガイド”の“データベースの構築”-“Symfoware/RDBを使用する場合”-“Symfoware/RDBで使用する資源の見積もり”を参照してください。

**rdp: ERROR: qdg12226e:DSI定義文の実行で重症エラーを検出しました 詳細メッセージ=JYP4204E 指定されたデータベーススペース“DSDBSPACE”が、定義されていません. 'エラーが発生した文の先頭位置=%s1 (システム名=%s2)**

[意味と対処]

%s1は、エラーが発生した文の先頭位置です。

%s2は、RDBシステム名です。

データベーススペース“DSDBSPACE”が定義されていない、またはデータベーススキーマが定義されていません。

データベーススペース“DSDBSPACE”、またはスキーマが正しく定義されているか確認してください。

**rdp: ERROR: qdg12226e:表定義文の実行で重症エラーを検出しました 詳細メッセージ=JYP4784E データベース%s1は定義されていません. 'エラーが発生した文の先頭位置=%s2 (システム名=%s3)**

[意味と対処]

%s1は、データベース名です。

%s2は、エラーが発生した文の先頭位置です。

%s3は、RDBシステム名です。

データベース名が定義されていません。

データベース名が正しく定義されているか確認してください。

**rdp: ERROR: qdg12226e:DSI定義文の実行で重症エラーを検出しました 詳細メッセージ=JYP5071E アーカイブログファイルの不足が発生しました. 'エラーが発生した文の先頭位置=%s1 (システム名=%s2)**

[意味と対処]

%s1は、エラーが発生した文の先頭位置です。

%s2は、RDBシステム名です。

Symfoware Serverのアーカイブログファイルが不足しています。

Symfoware Serverのマニュアル“メッセージ集”の、“JYP5071E”を参照して、対処してください。その後、スキーマ配下の資源をすべて削除してください。スキーマ配下の資源を削除するには、コマンド、または、WebDBtoolsを使用して、以下のSQLを実行してください。

```
DROP SCHEMA SRADMIN CASCADE
```

下線部分はデータベース作成時に指定したスキーマ名です。

## ■リポジトリのデータベースにOracleデータベースを使用している場合

Oracleデータベースを使用時にデータベースのエラーが発生した場合は、Oracleデータベースのマニュアルを参照してください。

## 18.14 テーブル作成コマンドの応答がない

リポジトリのデータベースとしてRDBを使用する場合の環境構築で、テーブル作成コマンド実行時に応答がない場合の対処方法を説明します。

## ■リポジトリのデータベースにSymfoware Serverを使用している場合

Symfoware Serverがインストールされているマシンのシステムログ、およびSymfoware/RDBのメッセージログファイルを確認してください。

Symfoware Serverのマニュアル「メッセージ集」、または「トラブルシューティング集」を参照して対処してください。

### Windows32/64

テーブル作成コマンドは、コマンドを実行するマシンのコンソール、またはコンソールセッションに接続したリモートデスクトップからログインして実行する必要があります。

コマンドを実行するマシンのコンソール、またはコンソールセッションに接続したリモートデスクトップからログインして実行していることを確認してください。

上記対処を実施後、スキーマ配下の資源をすべて削除してください。スキーマ配下の資源を削除するには、コマンド、またはWebDBtoolsを使用して、以下のSQLを実行してください。

```
DROP SCHEMA DSADMIN CASCADE
```

下線部分はデータベース作成時に指定したスキーマ名です。

## ■リポジトリのデータベースにOracleデータベースを使用している場合

Oracleデータベースがインストールされているマシンのシステムログ、およびOracleデータベースのメッセージログファイルを確認してください。

Oracleデータベースのマニュアルを参照して対処してください。

### Windows32/64

テーブル作成コマンドは、コマンドを実行するマシンのコンソール、またはコンソールセッションに接続したリモートデスクトップからログインして実行する必要があります。

コマンドを実行するマシンのコンソール、またはコンソールセッションに接続したリモートデスクトップからログインして実行していることを確認してください。

上記対処を実施後、Oracle Enterprise Manager を使用して、cascadeオプション付きでスキーマを削除してください。または、SQL\*Plusで以下のSQLを実行して、スキーマを削除してください。

```
DROP USER DSADMIN CASCADE;
```

下線部分はデータベース作成時に指定したアカウントIDです。

## 18.15 エントリ管理ツールのショートカットキーが使用できない

エントリ管理ツールをキーボードから操作する際に使用できるショートカットキーの一部(コピーキー(Ctrl+C)、貼り付けキー(Ctrl+V)など)が動作しない場合の対処方法を説明します。

使用できないショートカットキーでの操作については、メニュー、またはポップアップメニューから該当する機能を選択して操作をしてください。

## 18.16 エントリ管理ツールの文字入力フィールドで、文字を入力できない Linux32/64

RHEL5(x86)/RHEL5(Intel64)で、エントリ管理ツールの接続情報設定画面、またはエントリ追加画面などの文字入力フィールドに、文字を入力できない場合の対処方法を説明します。

- ・ 同一画面内に、複数の入力フィールドがある場合は、他の入力フィールドにいったんカーソルを移動します。その後、もとの入力フィールドにカーソルを戻してください。
- ・ エントリ管理ツールのヘルプをいったん開いてから、もとの画面にフォーカスを戻してください。
- ・ クライアントパッケージのエントリ管理ツールを使用してください。

## 18.17 アクセス制御が正しく機能しない

リポジトリに対してアクセス制御リストの設定を行っても、アクセス制御が正しく機能しない場合の対処方法を説明します。

以下の点について、アクセス制御リストが正しく定義されているかを確認してください。アクセス制御リストの定義内容の詳細については、「セキュリティシステム運用ガイド」の「Interstage ディレクトリサービスのアクセス制御の設定」を参照してください。

- ・ userPassword属性に対して、匿名ユーザ(anonymous)に認証(auth)を許可しているか。  
本指定がないと、管理者用DN以外のユーザからアクセスできません。
- ・ accessディレクティブを複数指定する場合は、アクセス対象の中で、より限定的な対象が定義ファイルの先頭行から順に記述されているか。
- ・ 同じアクセス対象に対するアクセス制御を、複数のaccessディレクティブで指定していないか。  
同じアクセス対象に対するアクセス制御は、1つのaccessディレクティブで記述する必要があります。複数のaccessディレクティブで記述すると、定義ファイルの先頭行からみて、最初に出現したaccessディレクティブのアクセス制御のみが適用されます。
- ・ 日本語などのマルチバイト文字を使用している場合、UTF-8で記述しているか。

## 18.18 スキーマ拡張に関する異常

リポジトリに対してスキーマ拡張、およびスキーマ拡張で定義したエントリの更新時に発生するエラーについて対処方法を説明します。

### スキーマ拡張時にirep10626エラーが発生する

リポジトリに対するスキーマ拡張で登録できる「属性型の名前(属性名)」の長さを以下に示します。

リポジトリのデータベース	「属性型の名前」の最大長
標準データベース	56 バイト
Symfoware/RDB	属性名の最大長を拡張しない場合:32 バイト
Oracle Database	属性名の最大長を拡張した場合:64 バイト

スキーマ拡張で定義した属性名が最大長を超えていないかを確認してください。

リポジトリのデータベースにSymfoware/RDB、またはOracle Databaseを使用している場合、33バイト以上の属性名を使用するためには、データベースのテーブル作成時に、属性名の最大長を拡張しておく必要があります。

### エントリの更新時にLDAP\_UNDEFINED\_TYPEエラーが発生する

指定した属性名がリポジトリのスキーマ定義に存在しません。以下を確認してください。

- 指定した属性名に誤りがないかを確認してください。
- スキーマ拡張が設定されているかを確認してください。

スキーマ拡張の詳細については、「ディレクトリサービス運用ガイド」の「リポジトリの作成」-「スキーマ拡張」を参照してください。

## 18.19 Active Directoryとの通信に関するトラブル

---

Active Directoryに接続した場合に接続エラーとなる場合の対処方法について説明します。

- `ldapsearch`コマンドで、「ref: ldap://xxx」が復帰する
- LDAP-SDK(JNDI, C)による検索で、LDAP\_OPERATIONS\_ERROR、またはLDAP\_REFERRALエラーが復帰する

### ldapsearchコマンドで、「ref: ldap://xxx」が復帰する

「ref: ldap://xxx」が返る場合は、Active Directoryからリフェラルが通知されています。

ldapsearchコマンドの検索条件である「-bオプション(サーチベース)」を見直す、またはActive Directoryの構成を変更して、リフェラルを返さないようにしてください。

### LDAP-SDK(JNDI, C)による検索で、LDAP\_OPERATIONS\_ERROR、またはLDAP\_REFERRALエラーが復帰する

Active Directoryからリフェラルが通知されている可能性があります。以下のいずれかの対処を行ってください。

LDAP-SDK(JNDI, C)に指定するサーチベース・パラメータを見直す、またはActive Directoryの構成を変更して、リフェラルを返さないようにしてください。

## 第19章 Java実行環境運用時の異常

Javaプログラムを実行したときに異常が発生した場合の対処方法は、“チューニングガイド”の“JDK/JREのチューニング”を参照してください。

### 19.1 FJVMのトラブルシューティング機能

FJVMのトラブルシューティング機能の詳細は、“チューニングガイド”の“JDK/JREのチューニング”の“FJVM”を参照してください。

## 第20章 Interstage data store使用時の異常

Interstage data storeの使用でトラブルが発生した場合の対処方法について説明します。

### 20.1 データストアのイベントログにエラーメッセージが出力される

データストアのイベントログに、エラーメッセージが出力された場合の対処方法を説明します。

データストアのイベントログは、以下の場所に作成されます。

#### Windows32/64

```
%IS_HOME%\Enabler\server\etc\errlogOMS\[データストア名].log
```

#### Solaris32/64

```
/opt/FJSVena/server/etc/errlogOMS/[データストア名]
```

#### Linux32/64

```
/opt/FJSVena/Enabler/server/etc/errlogOMS/[データストア名]
```

以下に、エラーと対処方法を説明します。

[日本語メッセージ]

**ERROR 9291 (データストアが開始していません: データストアが矛盾しています.データストアを回復するには,omsstart を -r オプションをつけて実行してください.詳細はデータストアのイベントログを参照してください.) in omsd.exe (omsd.c,438): stop with adm\_errno=9291**

[英語メッセージ]

**ERROR 9291 (Datastore not started: Datastore is inconsistent. To recover the datastore, call omsstart with the -r option. For details, refer to the datastore event log.) in omsd (omsd.c,438): stop with adm\_errno=9291**

[意味と対処]

本エラーが出力された後のメッセージを確認してください。出力されているメッセージにより、意味と対処が異なります。以下に確認すべきメッセージについて説明します。

- 以下のエラーメッセージが出力されている場合

**ERROR in omsd.exe (omsutil.c:416), tid=0, errno=9: Size of file %s1 [index:1] is %s2; it must be a multiple of 4096!**  
リポジトリのデータベースのファイルが壊れている可能性があります。リポジトリを起動できません。各運用形態に合わせて以下のマニュアルを参照し、該当リポジトリを再作成してください。

- スタンドアロン形態での運用中に発生した場合  
“ディレクトリサービス運用ガイド”の“リポジトリの修復”-“スタンドアロン形態の場合”-“環境が壊れている場合”
- レプリケーション形態での運用中にマスタサーバのリポジトリで発生した場合  
“ディレクトリサービス運用ガイド”の“レプリケーション形態での運用監視とリポジトリの修復”-“マスタサーバのリポジトリの修復”
- レプリケーション形態での運用中にスレーブサーバのリポジトリで発生した場合  
“ディレクトリサービス運用ガイド”の“レプリケーション形態での運用監視とリポジトリの修復”-“スレーブサーバのリポジトリの修復”-“環境が壊れている場合”

- 「Recovery process finished!」メッセージが出力されている場合  
データストア内の矛盾を検出したため、回復処理を実行後、データストアを起動しました。  
問題はありませんので、対処は必要ありません。
- 上記メッセージが出力されていない場合  
データストア内で矛盾を検出したため、データストアの起動に失敗しました。  
iscollectinfoコマンドを使用して調査情報を採取した後、技術員に連絡してください。

# 付録A Javaツール機能

## A.1 メソッドトレース機能

Windows32/64 Solaris32/64 Linux32/64

### ■メソッドトレース機能とは

メソッドトレース機能とは、Javaアプリケーションのメソッドの呼出しをトレースする機能です。メソッドトレース機能が出力するトレース情報により、次のことがわかります。

- アプリケーションの処理の流れ
- Java Servlet APIのメソッドに渡される引数、およびメソッドから返される戻り値の情報
- EJBアプリケーションのメソッドに渡される引数、およびメソッドから返される戻り値の情報

たとえば、次のような現象が発生する場合、処理がどこまで実行されているか、どの処理で停止や異常が起きているかを診断するうえでの有用な情報を得ることができます。

- アプリケーションの応答がなくなった場合
- 出力や処理結果の異常
- Exceptionの発生

メソッドトレース機能からの情報をヒントに、アプリケーションのログや各ログ取得機能、デバッグ、APIドキュメントを参照して原因を特定する必要があります。

### ■動作環境

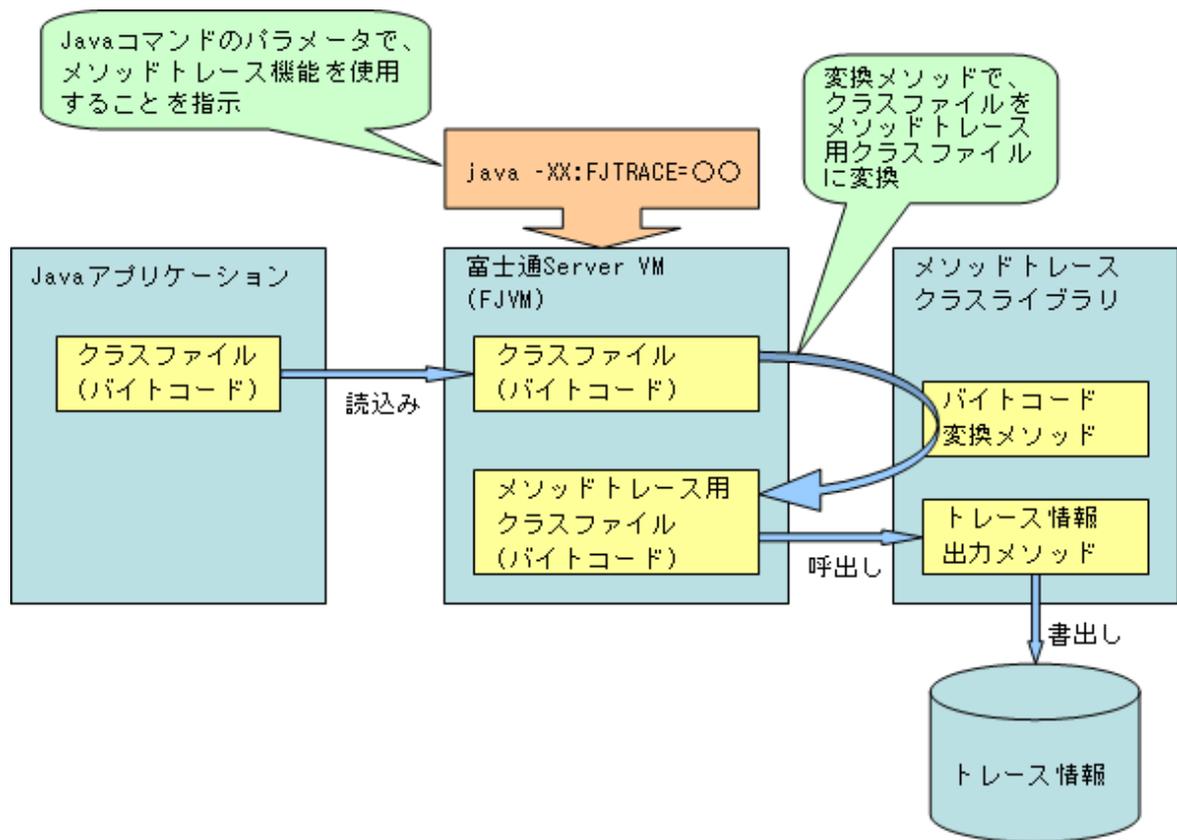
メソッドトレース機能の動作には、製品添付のJDK/JREが必要です。

### A.1.1 動作原理

メソッドトレース機能は、製品添付のJDK/JREおよびJava VMから呼び出されるメソッドトレースクラスライブラリで実現されます。

メソッドトレース機能は、以下のような流れで実現します。

1. Java VMにメソッドトレース機能を有効にする起動オプションを指定して、アプリケーションを起動します。  
メソッドトレース機能を有効にする起動オプションの詳細は、“[A.1.5 使用方法](#)”を参照してください。
2. アプリケーションのクラスファイル(バイトコード)が、クラスローダ経由でJava VMに読み込まれます。
3. Java VMからメソッドトレースクラスライブラリのバイトコード変換メソッドが呼び出されます。  
なお、バイトコード変換メソッドの呼び出しは、Java VM標準のJVMTI機能によって実現しています。
4. バイトコード変換メソッドによって、アプリケーションのバイトコードに、メソッドトレースクラスライブラリのトレース情報出力メソッドを呼び出すコードが埋め込まれます。
5. 変換されたバイトコードがJava VMに渡され、アプリケーションが開始されます。
6. 開始されたアプリケーションのバイトコード内に、トレース情報出力メソッドの呼出しコードがあると、メソッドトレースクラスライブラリによって、トレース情報が出力されます。



## A.1.2 使用手順

次の手順を行うことにより、メソッドトレース機能を使用することができます。

1. メソッドトレース機能の動作を設定するための“制御ファイル”を作成します。  
制御ファイルの作成方法は、“[A.1.4 制御ファイルの作成方法](#)”を参照してください。
2. 環境変数PATHまたはCLASSPATHにメソッドトレース機能の動作に必要なパスを追加します。  
メソッドトレースの格納先は“[A.1.3 格納先](#)”、環境変数の設定方法は“[A.1.5 使用方法](#)”を参照してください。
3. Java VMにメソッドトレース機能を有効にする起動オプションを指定して、アプリケーションを起動します。  
起動オプションの指定方法は、“[A.1.5 使用方法](#)”を参照してください。

## A.1.3 格納先

### ■メソッドトレース機能の格納先

メソッドトレース機能は次の場所にインストールされます。

#### Windows32/64

以下、製品を“C:\¥Interstage”にインストールした場合を想定しています。

- C:\¥Interstage¥jdk6¥tools¥fjtrace (JDK 6)
- C:\¥Interstage¥jre6¥tools¥fjtrace (JRE 6)
- C:\¥Interstage¥jdk7¥tools¥fjtrace (JDK 7)
- C:\¥Interstage¥jre7¥tools¥fjtrace (JRE 7)

#### Solaris32/64 Linux32/64

- /opt/FJSVawjbc/jdk6/tools/fjtrace (JDK 6)

- /opt/FJSVawjbc/jre6/tools/fjtrace (JRE 6)
- /opt/FJSVawjbc/jdk7/tools/fjtrace (JDK 7)
- /opt/FJSVawjbc/jre7/tools/fjtrace (JRE 7)

## ■メソッドトレース機能の動作に必要なモジュール

メソッドトレース機能のインストール先には、次のモジュールが格納されています。

### Windows32/64

- mt.jar
- bcel-5.1.jar
- jakarta-regexp-1.2.jar
- fjtrace.dll
- TraceAgent.dll

### Solaris32/64 Linux32/64

- mt.jar
- bcel-5.1.jar
- jakarta-regexp-1.2.jar
- libfjtrace.so
- libtraceagent.so

## A.1.4 制御ファイルの作成方法

ここでは、制御ファイルの作成方法を説明します。

制御ファイルは、メソッドトレース機能の動作を定義するためのファイルです。

制御ファイル名は、メソッドトレース機能を使用するJava VMの起動時に指定します。

トレース対象のメソッド名、トレース情報の出力形式、トレース情報の出力先、ファイルローテーションの動作などを定義できます。

### ■記述形式

制御ファイルは、以下の形式で記述します。

定義名=定義値
---------

- 1行1項目で指定します。
- コメントを記述する場合は、“#”以降に記述します。
- 定義名を重複して指定した場合は、最後に正しく指定した項目が有効になります。ただし、重複指定可能な“`traceMethod`”および“`not_traceMethod`”を除きます。
- 1行の記述が“定義名=定義値”の形式になっていない場合、または、以下の“制御ファイルの定義項目”に記載されていない定義名を指定した場合、その行は無効となります。
- 定義値の指定に誤りがある場合の解釈は、定義項目によって異なります。詳細は、それぞれの定義項目を参照してください。
- 定義名および定義値に使用する文字列は、大文字と小文字は区別され、異なる文字として扱われます。

なお、メソッドトレース機能の格納先に、制御ファイルのサンプル(`fjtrace.conf`)がありますので、参考にしてください。

### ■定義項目の一覧

制御ファイルに定義できる項目を、次の表に示します。

定義名	説明
<code>statementStyle</code>	トレース対象のメソッドの指定に正規表現を用いるかを指定します。 制御ファイルの記述の誤りについて通知するメッセージの出力にも影響を与えます。
<code>traceMethod</code>	トレース対象のメソッドを指定します。
<code>not_traceMethod</code>	非トレース対象のメソッドを指定します。
<code>classPath</code>	動的に決定されるクラスパスを指定します。
<code>valueLevel</code>	メソッドの引数および戻り値情報の出力形式を指定します。
<code>timeStamp</code>	タイムスタンプ出力の有無を指定します。
<code>traceFile</code>	トレース情報を出力するファイル名を指定します。
<code>traceDirectory</code>	トレース情報出力ファイルを格納するディレクトリを指定します。
<code>traceFileCount</code>	トレース情報出力ファイルのローテーション数を指定します。
<code>traceFileSize</code>	トレース情報出力ファイルの最大サイズを指定します。

## ■ 定義項目の詳細

制御ファイルに定義する項目の詳細を説明します。

### ◆ `statementStyle`

#### 記述形式

```
statementStyle=REGEX | BASIC
```

#### 説明

トレース対象のメソッドの指定に正規表現を使用するかを指定します。  
以下のいずれかの定義値を指定します。

REGEX: 正規表現を使用する

BASIC: 正規表現を使用しない(省略値)

“REGEX”、“BASIC”以外の定義値を指定した場合、その行は無効となります。また、“statementStyle”の記述がない場合、“BASIC”が指定されたものと解釈します。

#### 注意

本指定に誤りがある場合は、制御ファイルの記述に誤りがあったことを通知するメッセージを標準出力(または、使用方法記載のログファイル)に出力します。

(“statementStyle”の記述が無い場合は、メッセージの出力は行われません。)

### ◆ `traceMethod`

### ◆ `not_traceMethod`

#### 記述形式

```
traceMethod=クラス名 メソッド名 [シグネチャ]
```

```
not_traceMethod=クラス名 メソッド名 [シグネチャ]
```

[ ]は省略可能です。

#### 説明

“traceMethod”は、トレース対象のメソッドを指定します。

“not\_traceMethod”は、非トレース対象のメソッドを指定します。

“traceMethod”、“not\_traceMethod”ともに複数指定できます。

“traceMethod”が1つも指定されていない場合は、トレースを行わずにアプリケーションを実行します。

“not\_traceMethod”、“statementStyle=REGEX”が指定されている場合にのみ有効です。  
 (“statementStyle=BASIC”の場合、無効となります。)

クラス名、メソッド名、シグネチャはそれぞれ半角空白で区切ります。

“traceMethod”、“not\_traceMethod”とも、“statementStyle”の指定に影響を受けます。

定義値の記述内容を下表に示します。

【statementStyle=REGEX指定の場合】

<p>クラス名</p>	<p>クラス名を指定します。パッケージ階層の区切りは/です。</p> <p>例) javax/servlet/http/HttpServletRequest</p> <p>クラス名には、インターフェース名やスーパークラス名を指定することもできます。                  インターフェース名を指定した場合、トレース対象のプログラム内で指定のインターフェースを実装しているクラス(インターフェースメソッド)をトレースします。                  スーパークラス名を指定した場合、トレース対象のプログラム内で指定のクラスをスーパークラスに持つクラス(オーバーライドしたメソッド)をトレースします。</p> <p>正規表現を用いることで、“すべてのクラス”の指定ができます。</p> <ul style="list-style-type: none"> <li>• .*                      すべてを意味します(“.”は任意の1文字、“*”は繰り返しを示します)</li> <li>• [a-z]                      範囲を示す表記です。                      この場合、aからz(小文字のアルファベット)を意味します。                      “[a-z]*”とすることで、小文字のアルファベットを含む文字列を意味します。</li> <li>• ^クラス名\$                      完全一致を意味します(^は前方一致、\$は後方一致を示します)</li> </ul> <p>正規表現の詳細は、apacheの提供する、jakarta-regexp V1.2のorg.apache.regexp.REのAPI仕様を参照してください(※注)。</p> <p>(※注) 正規表現 “[:graph:]”、“[:alpha:]”、“[:alnum:]”、“%w”には、既存のバグが存在するため正しく動作しません。                  これらの正規表現の代わりに、それぞれ、“[:print:]”、“[a-zA-Z]”、“[a-zA-Z0-9]”、“[a-zA-Z0-9_]”を使用してください。</p> <p>正規表現の構文が誤っていた場合、その行を無効とし、Warningメッセージ(FJTRACE:400)を標準出力(または、使用方法記載のログファイル)に出力します。</p> <p>注).*を用いた場合、実行性能が極度に劣化します。パッケージ名を指定するなど、部分的な総称となるように指定してください。</p>
<p>メソッド名</p>	<p>メソッド名を指定します。</p> <p>クラス名の指定に応じて、次の意味になります。</p> <p>クラス名:実装したメソッド名                  インターフェース名:インターフェースメソッド名                  スーパークラス名:サブクラスがオーバーライドしたメソッド名</p> <p>正規表現の指定ができます。</p> <ul style="list-style-type: none"> <li>• .*                      すべてを意味します(“.”は任意の1文字、“*”は繰り返しを示します)</li> <li>• [a-z]*                      アルファベットの小文字を含む文字列を意味します</li> <li>• ^メソッド名\$                      完全一致を意味します(^は前方一致、\$は後方一致を示します)</li> </ul>

	<p>正規表現の詳細は、apacheの提供する、jakarta-regexp V1.2のorg.apache.regexp.REのAPI仕様を参照してください(※注)。</p> <p>(※注) 正規表現 "[:graph:]", "[:alpha:]", "[:alnum:]", "¥w"には、既存のバグが存在するため正しく動作しません。</p> <p>これらの正規表現の代わりに、それぞれ、"[:print:]", "[a-zA-Z]", "[a-zA-Z0-9]", "[a-zA-Z0-9_]"を使用してください。</p> <p>正規表現の構文が誤っていた場合、その行を無効とし、Warningメッセージ(FJTRACE:400)を標準出力(または、使用方法記載のログファイル)に出力します。</p>
シグネチャ(省略可)	<p>シグネチャを指定します。省略可能です。</p> <p>指定するとクラス名、メソッド名で限定されたメソッドのうち、さらにシグネチャの一致する特定のメソッドを対象とします。たとえば、クラス内にメソッド名に一致するメソッドが複数存在する(オーバーロードメソッド)場合に、特定のメソッドにトレースを限定できます。</p> <p>シグネチャは、引数を示すデータ型を"()"で囲み、戻り値を示すデータ型をその後に続けて指定します。データ型の表記方法は、“<a href="#">データ型の表記方法</a>”を参考にしてください。</p> <p>例1) 引数を持たず、戻り値がvoid型のメソッドの場合 引数がないため"()"内は記述しません。戻り値void型は")"の後ろに"V"を記述して示します。 "()V"</p> <p>例2) String型の配列とboolean型の2つ引数を持ち、戻り値がint型のメソッドの場合 "([Ljava/lang/String;Z)I"</p>

【statementStyle=BASIC指定の場合】

クラス名	<p>クラス名を指定します。パッケージ階層の区切りは'.'です。</p> <p>例) org.apache.catalina.connector.RequestFacade</p> <p>注) *による“すべてのクラス”の指定はできません。</p>
メソッド名	<p>メソッド名を指定します。指定には、次の2つの方法があります。</p> <ul style="list-style-type: none"> <li>メソッド名:クラス名に指定したクラス内で完全一致するメソッドすべてを対象とします。</li> <li>* :クラス名で指定したクラス内すべてのメソッドを対象とします。</li> </ul>
シグネチャ(省略可)	<p>シグネチャを指定します。省略可能です。</p> <p>指定するとクラス名、メソッド名で限定されたメソッドのうち、さらにシグネチャの一致する特定のメソッドを対象とします。たとえば、クラス内にメソッド名に一致するメソッドが複数存在する(オーバーロードメソッド)場合に、特定のメソッドにトレースを限定できます。</p> <p>シグネチャは、引数を示すデータ型を"()"で囲み、戻り値を示すデータ型をその後に続けて指定します。データ型の表記方法は、“<a href="#">データ型の表記方法</a>”を参考にしてください。</p> <p>例1) 引数を持たず、戻り値がvoid型のメソッドの場合 引数がないため"()"内は記述しません。戻り値void型は")"の後ろに"V"を記述して示します。 "()V"</p> <p>例2) String型の配列とboolean型の2つ引数を持ち、戻り値がint型のメソッドの場合 "([Ljava/lang/String;Z)I"</p>

データ型の表記方法

型	シグネチャの文字
boolean	Z
byte	B
char	C
short	S
int	I

型	シグネチャの文字
long	J
float	F
double	D
void	V(戻り値の場合のみ)
クラス(注)	Lクラス名;
配列	[

注) クラスは、'L'で始まり、';'で終わります。

クラス名は、パッケージ名を含む完全なクラス名を使用します。この時の区切り文字は'/'です。

例) Stringクラスの場合

"Ljava/lang/String;"

### 参考

メソッドのシグネチャを確認するには、JDKに含まれるjavapコマンドに-sオプションを指定します。

```
javap -s クラス名
```

### 注意

以下のパッケージ配下のクラスは、メソッドトレースの指定はできません。指定しても、無効となります。

- com.fujitsu.jvm.mt
- org.apache.bcel
- org.apache.regexp

また、“[トレース不可のクラス](#)”に示すクラスについては、トレース情報を出力できません。

mainメソッドを含むクラスに実装されたメソッドがトレース対象となった場合、クラスをロードするタイミングでメソッドトレースの使用するオブジェクトと競合を起こし、まれにデッドロックが発生することがあります。その場合、以下の指定を追加することで回避できることがあります。

statementStyle=REGEX指定の場合、以下の指定を追加してください。

- traceMethod=java/net/URLClassLoader.\*.\*

statementStyle=BASIC指定の場合、以下の指定を追加してください。

- traceMethod=java.net.URLClassLoader\$1 \*

## ◆classPath

### 記述形式

```
classPath=動的設定されるクラスパス
```

### 説明

メソッドトレース対象のアプリケーションが動的に決定するクラスパス(環境変数CLASSPATHに記述せず、アプリケーション内で設定するクラスパス)を持つ場合に指定します。

クラスパスの設定は、メソッドトレース機能が動作するOS環境の決まりに従います(環境変数CLASSPATHと同様の記述を行います)。

“classPath”の指定がない場合、環境変数CLASSPATHに指定されたパスが使われます。

“classPath”の指定がある場合、指定したパスの後ろに環境変数CLASSPATHのパスをつなげたものが使われます。

“classPath”は、“statementStyle=REGEX”が指定されている場合にのみ有効です。

(“statementStyle=BASIC”の場合、無効となります。)

## 注意

複数のクラスパスを記述する必要がある場合、区切り文字“;”でつないで1行で制御ファイルに記載します。改行はしないでください。

```
classPath=C:\Interstage\lib\isjmxrt.jar;C:\Interstage\lib\isjmxcommon.jar;[…略]
```

- ・ 上記は、Windows(R)上で、製品をC:\Interstageにインストールした場合です。
- ・ SolarisおよびLinuxの場合は、区切り文字はコロン(:)となります。

## ◆valueLevel

### 記述形式

```
valueLevel=full | addHex | none
```

### 説明

メソッドの引数および戻り値の出力形式を指定します。  
以下のいずれかの定義値を指定します。

full	:	引数および戻り値の情報を出力する(省略値)
addHex	:	“full”の機能に加え、文字列の16進数表示を出力する
none	:	引数および戻り値の情報を出力しない

“full”、“addHex”、“none”以外の定義値を指定した場合、その行は無効となります。また、“valueLevel”の記述がない場合、“full”が指定されたものと解釈します。

引数、戻り値の出力形式の詳細は、“[A.1.6 トレース情報出力の形式](#)”を参照してください。

## ◆timeStamp

### 記述形式

```
timeStamp=true | false
```

### 説明

タイムスタンプ出力の有効/無効を指定します。  
以下のいずれかの定義値を指定します。

true	:	タイムスタンプを出力する
false	:	タイムスタンプを出力しない(省略値)

“true”、“false”以外の定義値を指定した場合、その行を無効とします。また、“timeStamp”の記述がない場合、“false”が指定されたものと解釈します。

## 注意

“timeStamp=true”を指定した場合、メソッドトレース機能は日付と時刻の表示のため、以下のクラスを使用しています。

- ・ java.util.Calendar
- ・ java.text.SimpleDateFormat

これらのクラスおよび関連するクラス(※注)を直接または間接的に、複数のスレッド間で同時に使用するアプリケーションでは、デッドロックを引き起こす場合があります。

その場合、本指定をfalseに変更するか、または、トレースを行わないクラスとして“[not\\_traceMethod](#)”で指定してください。

※注:たとえば、以下のクラスです。

- ・ java.util.GregorianCalendar
- ・ java.util.SimpleTimeZone

- java.util.TimeZone

## ◆traceFile

### 記述形式

```
traceFile=トレース情報出力ファイル名
```

### 説明

トレース情報を出力するファイルのフルパスを指定します。  
パス中のディレクトリは存在している必要があります。  
パス中のディレクトリが存在しない場合、この行は無効となります。  
また、同名のファイルがすでに存在した場合、上書きされます。  
メソッドトレース機能を一般ユーザで起動した場合には、その一般ユーザに対する書き込み権が付与されているファイルを指定します。

### 注意

- “traceDirectory”と同時に指定された場合、本指定が優先されます。
- 本指定が省略された場合、“traceDirectory”が指定されたものとして処理します。
- 本指定と“traceDirectory”が省略された場合は、標準出力(または、使用方法記載のログファイル)にトレース情報が出力されます。
- 本指定と同時に“traceFileCount”、“traceFileSize”を指定した場合、“traceFileCount”、“traceFileSize”の指定は無効となります。
- ファイル出力中、出力先のディスク容量が不足した場合、それ以降のトレースは出力されません。

## ◆traceDirectory

### 記述形式

```
traceDirectory=トレース情報出力ファイルを格納するディレクトリ
```

### 説明

トレース情報出力ファイルを格納するディレクトリのフルパスを指定します。  
ディレクトリは存在している必要があります。  
存在しないディレクトリを指定した場合は、この行は無効となります。  
また、メソッドトレース機能を一般ユーザで起動した場合には、その一般ユーザに対する書き込み権が付与されているディレクトリを指定します。

指定のディレクトリには、次のルールでトレース情報出力ファイルを作成します。

```
mt<プロセスID>_n.log
```

mt : トレース情報出力ファイルであることを示す固有の文字列です。  
<プロセス ID> : Javaアプリケーションが動作しているプロセスIDです。  
\_n : ファイル番号を示します。nは、0から始まり、最大で“traceFileCount”に指定した値-1までの番号が振られます。  
.log : ファイルの拡張子です。

### 注意

- “traceFile”と同時に指定された場合、本指定は無効となります。
- “traceFile”と本指定が省略された場合は、標準出力(または、使用方法記載のログファイル)にトレース情報が出力されます。

- 作成されるメソッドトレース情報出力ファイルは、“`traceFileCount`”に指定した個数作成され、さらに出力がある場合、ファイル番号0のファイルから上書きします。
- 同名のメソッドトレース情報出力ファイルが存在した場合、上書きします。
- ファイル出力中、出力先のディスク容量が不足した場合、それ以降のトレースは出力されません。

## ◆`traceFileCount`

### 記述形式

```
traceFileCount=トレース情報出力ファイルのローテーション数
```

### 説明

トレース情報出力ファイルのローテーション数を指定します。1～10(個)が指定可能です。省略値は、2(個)です。“`traceFileCount`”の記述がない場合、省略値と解釈します。0以下の定義値を指定した場合は1と解釈します。11以上の定義値を指定した場合は10と解釈します。数字以外を指定した場合は、省略値と解釈します。

### 注意

- 本指定は“`traceDirectory`”が指定された場合に有効です。“`traceFile`”が指定された場合は、無効となります。

## ◆`traceFileSize`

### 記述形式

```
traceFileSize=トレース情報出力ファイルの最大サイズ
```

### 説明

トレース情報出力ファイルの1個あたりの最大サイズ(MB)を指定します。1～2147483647(MB)が指定可能です。省略値は、1(MB)です。“`traceFileSize`”の記述がないか、以下を指定した場合は、省略値と解釈します。

- 1より小さな値
- 2147483647より大きな値
- 整数値以外の値
- 数字以外を指定

### 注意

- 本指定は“`traceDirectory`”が指定された場合に有効です。“`traceFile`”が指定された場合は、無効となります。
- 指定された定義値を境にファイルローテーションされます。
- 1行で示されるenter情報またはexit情報が2つのファイルに分割されるのを防ぐため、ファイルのサイズが定義値より大きくなる場合があります。

## ■制御ファイルの記述例

制御ファイルの記述例を、次に示します。

```
-----
#
# Interstage Component FJTRACE sample configuration file.
#
statementStyle=REGEX
valueLevel=full
timeStamp=false
traceDirectory=D:%tmp
```

```

traceFileCount=2
traceFileSize=1

traceMethod=^com/fujitsu/interstage/XXXX$ ^methodA$
traceMethod=^com/fujitsu/interstage/XXXX$ ^methodB$ (Ljava/lang/String;)V
traceMethod=^com/fujitsu/interstage/XXXX/YYYY$ .*

```

上記の記述例では、以下のようになります。

```

トレース情報出力形式      :   メソッドの引数および戻り値の情報を出力する
タイムスタンプ出力        :   タイムスタンプを出力しない
ファイル出力先ディレクトリ :   D:¥tmp 配下
ファイルローテーション数   :   2(個)
ファイルの最大サイズ       :   1ファイルあたり1(MB)
トレース対象メソッド       :   "com.fujitsu.interstage.XXXX"クラスの
                                "methodA"メソッド
                                "methodB"メソッド(ただし、メソッドのシグネチャが"(Ljava/lang/String;)V"のメソッド
                                のみ)
                                "com.fujitsu.interstage.XXXX.YYYY"クラスのすべてのメソッド

```

statementStyle=BASIC(指定を省略)とした場合、以下は上記例と同じ意味を示しています。

```

-----
#
# Interstage Component FJTRACE sample configuration file.
#

valueLevel=full
timeStamp=false
traceDirectory=D:¥tmp
traceFileCount=2
traceFileSize=1

traceMethod=com.fujitsu.interstage.XXXX methodA
traceMethod=com.fujitsu.interstage.XXXX methodB (Ljava/lang/String;)V
traceMethod=com.fujitsu.interstage.XXXX.YYYY *
-----

```



### ◆Servletサービスの記述例

statementStyle=REGEXとし、総称指定を使用してServletサービスのトレースを採取する場合、ClassNotFoundException というエラーメッセージが発生する場合があります。

このメッセージを止めたい場合は、トレースするクラスを詳細に指定し直すか、あるいは使用中のIIServerに合わせ(○印が指定)、以下のjarファイルのパスをclassPath指定に追加してください。

※以下の説明は、Windows(R)上で、製品をC:¥Interstageにインストールした場合です。

クラス名	V9.0以降のIIServer
C:¥Interstage¥lib¥isjmxrt.jar	○
C:¥Interstage¥lib¥isjmxcommon.jar	○
C:¥Interstage¥lib¥log4j-1.2.8.jar	○
C:¥Interstage¥lib¥commons-logging-api.jar	○
C:¥Interstage¥lib¥commons-digester.jar	○
C:¥Interstage¥lib¥commons-collections.jar	○
C:¥Interstage¥lib¥commons-modeler.jar	○

C:\¥Interstage¥lib¥commons-beanutils.jar	○
C:\¥Interstage¥lib¥javax77.jar	○
C:\¥Interstage¥lib¥j2ee-sv-server.jar	○
C:\¥Interstage¥lib¥servlet-server.jar	○
C:\¥Interstage¥lib¥servlet-common.jar	○
C:\¥Interstage¥lib¥ejb-server.jar	○
C:\¥Interstage¥J2EE¥lib¥isj2ee.jar	○
C:\¥Interstage¥J2EE¥lib¥isdeploy60.jar	○
C:\¥Interstage¥J2EE¥lib¥ijserverconfig.jar	○
C:\¥Interstage¥jmx¥addons¥jmxremote¥lib¥jmxremote.jar	○
C:\¥Interstage¥F3FMjs5¥common¥jslib配下のjarファイル (注1)	○
C:\¥Interstage¥F3FMjs5¥common¥lib配下のjarファイル (注1)	○
C:\¥Interstage¥F3FMjs5¥server¥jslib配下のjarファイル (注1)	○
C:\¥Interstage¥F3FMjs5¥server¥lib配下のjarファイル (注1)	○
コンテナログにIJServer31009のメッセージで出力されるjarファイル	○

(注1) 必ずこの順番で記載してください。

Webアプリケーションで使用しているユーザが作成したクラスで、ClassNotFoundExceptionが発生する場合があります。これは、Webアプリケーションから使用されるjar群、class群は、war形式に沿ってWEB-INF/libやWEB-INF/classesに配置され動的に決定される場合があるためです。

この場合、配備されるパスをclassPathに追加します。

SampleServer.warというアプリケーションの場合の例を、以下に示します。

C:\¥Interstage¥J2EE¥var¥deployment¥ijserver¥SampleServer¥apps¥SampleServer.war¥WEB-INF¥classes

C:\¥Interstage¥J2EE¥var¥deployment¥ijserver¥SampleServer¥apps¥SampleServer.war¥WEB-INF¥lib¥app-library.jar

※上記は、Windows(R)上で、製品をC:\¥Interstageにインストールした場合です。

また、他にもアプリケーションで使用しているjarファイルがある場合 (IJServerディレクトリのShared/libディレクトリ配下など)、これらもあわせてclassPathに追加してください。

Servletサービスのトレースを採取する場合の設定ファイル例 (statementStyle=BASIC) を以下に示します。

[V9.0以降のIJServerを使用している場合]

```
#
# Interstage Component FJTRACE sample configuration file.
#

# javax.servlet.ServletConfigの実装クラス
traceMethod=org.apache.catalina.core.StandardWrapperFacade *

# javax.servlet.ServletContextの実装クラス
traceMethod=org.apache.catalina.core.ApplicationContextFacade *

# javax.servlet.http.HttpServletRequestの実装クラス
traceMethod=org.apache.catalina.connector.RequestFacade *

# javax.servlet.http.HttpServletRequest(ディスパッチ後)
traceMethod=org.apache.catalina.core.ApplicationHttpRequest *

# javax.servlet.http.HttpServletResponseの実装クラス
traceMethod=org.apache.catalina.connector.ResponseFacade *
```

```

# javax.servlet.http.HttpServletResponse (include後)
traceMethod=org.apache.catalina.core.ApplicationHttpServletResponse *

# javax.servlet.http.HttpSessionの実装クラス
traceMethod=org.apache.catalina.session.StandardSessionFacade *

# javax.servlet.RequestDispatcherの実装クラス
traceMethod=org.apache.catalina.core.ApplicationDispatcher *

# javax.servlet.ServletInputStreamの実装クラス
traceMethod=org.apache.catalina.connector.CoyoteInputStream *

# javax.servlet.ServletOutputStreamの実装クラス
traceMethod=org.apache.catalina.connector.CoyoteOutputStream *

# HttpServletResponseから取り出すjava.io.PrintWriter
traceMethod=org.apache.catalina.connector.CoyoteWriter *

# JSP outオブジェクトの実装クラス
traceMethod=org.apache.jasper.runtime.JspWriterImpl *

#サーブレットの実装クラス
traceMethod=<サーブレットクラス名> *

# javax.servlet.jsp.HttpJspPageの実装クラス
traceMethod=org.apache.jsp.<JSP file name>_jsp *

valueLevel=addHex
timeStamp=true

```

サーブレットクラスのメソッドトレースを採取する場合、サーブレットクラス名はweb.xmlファイル等から確認してください。  
JSPで作成したクラスのメソッドトレースを採取する場合、次のように指定します。

```
traceMethod=org.apache.jsp.<JSP file name>_jsp *
```

パッケージ“org.apache.jsp.”にクラス名<JSP file name>\_jsp(JSP file nameにJSPソースのファイル名を指定し、その後ろに“\_jsp”をつけます)と指定します。

Webアプリケーションのルートディレクトリ配下のサブディレクトリにJSPが格納されている場合は、サブディレクトリ名がorg.apache.jsp.に続くパッケージ名に含まれます。

メソッド名、シグネチャに関しては、通常と同じ指定が可能です。

### 注意

状況に応じて必要のないクラスはコメントアウトしてください。

上記各実装クラスやJSPで作成したクラスの名前は、使用状況やInterstageのバージョン・レベル等により変化する場合があります。



## ◆EJBサービスの記述例

statementStyle=BASICを指定した場合、EJBアプリケーションのデバッグ時にメソッドトレース機能を使用する場合は、作成したEJBアプリケーションのEnterprise Beanクラスを指定してください。

CMP2.0 Entity Beanの場合、Enterprise Beanクラスに“\_PM”を加えた文字列を指定してください。



例

CMP2.0 Entity BeanのEnterprise BeanクラスがflightSample.CustomerBeanの場合

```
flightSample.CustomerBean_PM
```

## ■制御ファイルの更新を反映する方法

制御ファイルは、アプリケーションの起動時に一度だけ読み込まれます。

制御ファイルの内容を更新し、その内容をメソッドトレースに反映させるためには、javaコマンドの再起動が必要になります。

IJServerクラスタの場合では、IJServerクラスタの再起動が必要になります。

なお、IJServerクラスタの操作は、Interstage Java EE管理コンソール、またはasadminコマンドにより行います。詳細は、“Java EE運用ガイド”を参照してください。

EE SJE

IJServerの場合では、ワークユニットの再起動が必要になります。

なお、ワークユニットの操作方法は、“Interstage管理コンソール ヘルプ”の[Interstage管理コンソール] > [Interstage Application Server] > [システム] > [ワークユニット] > [ワークユニットごとの操作] > [IJServer] > [操作]を参照してください。

## A.1.5 使用方法

ここでは、メソッドトレース機能を有効にする際に必要な環境変数の設定方法、および起動オプションの指定方法を説明します。

なお、ここでは次のように記述します。

- ・ 特に断りが無い場合、動作環境としてWindows(R)を想定しています。
- ・ 制御ファイル名を、D:¥tmp¥trace.confとしています。
- ・ Javaアプリケーションを、TestAppとしています。
- ・ JDKが“C:¥Interstage”配下にインストールされたものとしています。
- ・ “カレントディレクトリ”とは、“プログラムが動作するディレクトリ”のことです。
- ・ javaコマンドのコマンドラインオプション、およびJava VMの起動オプションを“起動オプション”と略記します。
- ・ Java EE機能やJ2EE機能のアプリケーション実行環境では、標準出力をログファイルに出力します。ログファイルの詳細は、以下を参照してください。

- IJServerクラスタ(Java EE機能)

“Java EE運用ガイド”

- IJServer(J2EE機能)

“Interstage管理コンソール ヘルプ”等

メソッドトレース機能を有効にするには、起動オプションに次に指定します。

Windows32/64

```
-agentlib:traceagent=制御ファイル名
または
-agentpath:C:¥Interstage¥jdk6¥tools¥fjtrace¥TraceAgent.dll=制御ファイル名
または
-agentpath:C:¥Interstage¥jdk7¥tools¥fjtrace¥TraceAgent.dll=制御ファイル名
(製品を“C:¥Interstage”にインストールした場合を想定しています。)
```

Solaris32/64 Linux32/64

```
-agentlib:traceagent=制御ファイル名
または
-agentpath:/opt/FJSVawjbc/jdk6/tools/fjtrace/libtraceagent.so=制御ファイル名
または
-agentpath:/opt/FJSVawjbc/jdk7/tools/fjtrace/libtraceagent.so=制御ファイル名
```

起動オプションで制御ファイル名を指定する際、次の事項に注意してください。

- ・ 制御ファイル名はフルパスで指定します。
- ・ “=制御ファイル名”を省略した場合(“-agentlib:traceagent”のみ、“-agentpath:[TraceAgent.dll (libtraceagent.so)へのフルパス]”のみ、または“-XX:FJTRACE”のみ)、または、制御ファイル名に指定したファイルが存在しない場合、カレントディレクトリにある“fjtrace.conf”というファイルが読み込まれます。
- ・ 制御ファイル名に指定したファイルが存在しない場合、かつ、カレントディレクトリに“fjtrace.conf”というファイルも存在しない場合は、次のメッセージを標準出力(または、使用方法記載のログファイル)に出力して、アプリケーションを実行します。この場合は、メソッドトレースは行いません。

```
FJTRACE:201: fjtrace.conf (指定されたファイルが見つかりません。)
```

## ■ javaコマンドでの指定方法

ここでは、javaコマンドでのオプションの指定方法を説明します。

### ◆ “-agentlib”オプションを使う場合

#### 【環境変数】

あらかじめ、次の環境変数にメソッドトレース機能の格納先のフルパスを追加してください。

Windows(R)の場合: PATH

Solaris/Linuxの場合: LD\_LIBRARY\_PATH

環境変数の指定が正しく行われていない場合、標準出力(または、使用方法記載のログファイル)にメッセージが出力され、Javaアプリケーションが実行されません。詳細は、下の注意事項“TraceAgent.dll(libtraceagent.so)が見つからない場合”を参照してください。

#### 【指定例】

```
java -agentlib:traceagent=D:¥tmp¥trace.conf TestApp
```

### ◆ “-agentpath”オプションを使う場合

#### 【環境変数】

あらかじめ、次の環境変数にメソッドトレース機能の格納先のフルパスを追加してください。

Windows(R)の場合: PATHまたはCLASSPATH

Solaris/Linuxの場合: LD\_LIBRARY\_PATHまたはCLASSPATH

両方の環境変数に追加した場合は、Windows(R)の場合“PATH”、Solaris/Linuxの場合“LD\_LIBRARY\_PATH”の指定が採用されます。

両方の環境変数の指定が正しく行われていない場合、標準出力(または、使用方法記載のログファイル)に次のメッセージが出力され、Javaアプリケーションが実行されません。

```
FATAL ERROR in native method: FJTRACE:701: FJTRACE stopped due to an internal error in the agent library.
```

#### 【指定例】

```
java -agentpath:C:¥Interstage¥jdk7¥tools¥fjtrace¥TraceAgent.dll=D:¥tmp¥trace.conf TestApp
```

## ■ IJServerクラスタでの指定方法

起動オプションは、Java VMオプションとして指定します。Java VMオプションは、Interstage Java EE管理コンソールおよびasadminコマンドを使用して指定できます。詳細については、以下のマニュアルを参照してください。

- ・ Interstage Java EE管理コンソールヘルプ
- ・ “リファレンスマニュアル(コマンド編)”の”Java EE運用コマンド”-”asadmin”-”定義項目参照/更新”
- ・ “リファレンスマニュアル(コマンド編)”の”Java EE運用コマンド”-”asadmin”-”create-jvm-optionsサブコマンド”

環境変数の設定は、asadminコマンドを使用して指定します。詳細については、以下のマニュアルを参照してください。

- Interstage Java EE管理コンソールヘルプ
- “リファレンスマニュアル(コマンド編)”の”Java EE運用コマンド”-”asadmin”-”定義項目参照/更新”

EE SJE

## ■ IJServerでの指定方法

IJServerにおける起動オプションおよび環境変数の設定は、Interstage管理コンソールで行います。

EE SJE

### ◆ V9.0以降のIJServerを使用している場合

1. 次の箇所にて、起動オプション“-agentlib”または“-agentpath”の設定を行います。
  - “環境設定”の[ワークユニット設定] > [Java VMオプション]
2. 次の箇所にて、メソッドトレース機能の格納先のフルパスを設定します。
  - **Windows32/64**  
“環境設定”の[ワークユニット設定] > [パス]
  - **Solaris32/64** **Linux32/64**  
“環境設定”の[ワークユニット設定] > [ライブラリパス]

起動オプションおよび環境変数の設定内容については、“[■ javaコマンドでの指定方法](#)”を参照してください。

Interstage管理コンソールの環境設定の詳細は、“Interstage管理コンソール ヘルプ”の[Interstage管理コンソール] > [Interstage Application Server] > [システム] > [ワークユニット] > [ワークユニットごとの操作] > [IJServer] > [環境設定]を参照してください。

## ■ 注意事項



### TraceAgent.dll (libtraceagent.so)が見つからない場合

注意

次のようなケースで“TraceAgent.dll(libtraceagent.so)”が見つからない場合、標準出力(または、使用方法記載のログファイル)にメッセージが出力され、プログラムが実行されません。

- 環境変数に“TraceAgent.dll(libtraceagent.so)”の格納先へのフルパスが正しく指定されていない
- エージェント名“traceagent”という文字列で指定されていない

その場合に出力されるメッセージは、次のとおりです。(太字はユーザ入力部分)

```
>java -agentlib:traceagent TestApp
Error occurred during initialization of VM
Could not find agent library on the library path or in the local directory: traceagent
```

または、

```
>java -agentpath: C:¥Interstage¥jdk7¥tools¥fjtrace¥fjtrace¥TraceAgent TestApp
Error occurred during initialization of VM
Could not find agent library in absolute path: C:¥Interstage¥jdk7¥tools¥fjtrace¥fjtrace¥TraceAgent
```

なお、エージェント名の指定を行わなかった場合、次のメッセージが標準出力(または、使用方法記載のログファイル)に出力され、プログラムが実行されません。(太字はユーザ入力部分)

```
>java -agentlib: TestApp
Error occurred during initialization of VM
Could not find agent library on the library path or in the local directory:
```

または、

```
>java -agentpath: TestApp
Error occurred during initialization of VM
Could not find agent library in absolute path:
```



メソッドトレース機能へのクラスパスの設定が正しく行われなかった場合

注意

メソッドトレース機能へのクラスパスの設定 (“CLASSPATH”環境変数、または“-Xbootclasspath/p”オプションの指定)が正しく行われていない状態で、メソッドトレース機能を有効にするオプションを指定した場合、次のメッセージが標準出力 (または、使用方法記載のログファイル)に出力され、プログラムが実行されません。SolarisおよびLinuxの場合、プログラムの応答がなくなる現象 (ハングアップ) が起こる場合があります。

また、クラスパスを設定するとき、jarファイルの区切り文字を間違えて使用した場合も同様です。

```
FJPROF: Cannot found com/fujitsu/jvm/prof/FJProf2 class.
Failed to load Main Class: mainを持つクラス名
```

または、

```
Exception in thread "main" java.lang.NoClassDefFoundError: com/fujitsu/jvm/mt/B2B
```



Linux32/64

注意

メソッドトレース機能を指定することで、FJVMの長時間コンパイル処理の検出機能によりJavaプロセスが強制終了する場合があります。その場合は回避策として、-XX:CompileTimeoutオプションを指定してください。

-XX:CompileTimeoutオプションの指定方法は、“チューニングガイド”の“JDK/JREのチューニング”の“長時間コンパイル処理の検出機能”を参照してください。

## A.1.6 トレース情報出力の形式

ここでは、トレース情報出力の形式を説明します。  
トレース情報として出力される情報には、以下の2種類があります。

- enter情報  
トレース対象に指定されたメソッドが呼び出されたことを示す情報です。  
以下に出力形式を示します。

```
[日時,]in,スレッド名,クラス名,メソッド名[,PARAM:引数の情報]
```

- exit情報  
トレース対象に指定されたメソッドから復帰したことを示す情報です。  
以下に出力形式を示します。

```
[日時,]out,スレッド名,クラス名,メソッド名[,RET:戻り値の情報]
```

Java VMは、enter情報およびexit情報を見つけるたびにトレース情報を出力します。  
正常にトレース情報が出力された場合、enter情報とexit情報は、対となって出力されます。  
enter情報およびexit情報は、1行で示されます。



注意

- 1行とは、復帰改行文字までをいいます。
- 復帰改行文字は、メソッドトレースの動作環境に依存します。
- 文字コードは、現在動作中のJavaに定義された文字コードセットに依存します。

## ■各項目の説明

項目名	意味	形式
日時	トレース情報出力時の日時です。 制御ファイルで出力の有無を指定できます。	DD/MM/YYYY hh:mm:ss:msec  DD : 日 MM : 月 YYYY : 年(西暦) hh : 時 mm : 分 ss : 秒 msec : ミリ秒  YYYY、msec以外は、必ず2桁で出力されます。 msecは、必ず3桁で出力されます。  Calendarクラスの初期化中の処理をトレースする場合、時刻の取得ができないため、表示は、“??/?/?/?/? ??:??:??:???”と表示されます。
in	enter情報であることを示します。	ASCIIコード“in”の文字列です。
out	exit情報であることを示します。	ASCIIコード“out”の文字列です。
スレッド名	メソッドを呼び出したスレッドにつけられたスレッドの名前です。	
クラス名	呼び出されたメソッドが配置されたクラス名です。	パッケージ名から出力され、パッケージ、クラスのセパレータは“.”で示されます。
メソッド名	呼び出されたメソッド名です。	
PARAM:	引数の表示の開始を示します。 制御ファイルで出力の有無を指定できます。	ASCIIコード“PARAM:”の文字列です。 このキーワードの後ろにメソッドに渡された引数の情報が出力されます。 引数の情報は、メソッドが引数に持つ順番に、与えられている個数分出力されます。 引数が複数個ある場合、引数の情報ごと“:”で区切られます。 引数がない場合、引数の情報は出力されません。“PARAM:”のみが出力されます。 データ型による詳細な出力形式は、“ <a href="#">■引数や戻り値の出力形式</a> ”を参照してください。  例1)文字列形式の引数が1つ PARAM:(String):ABCD  例2)文字列形式の引数の次に論理型の引数の2つを持つ場合 PARAM:(String):ABCD:(boolean):true  例3)引数がない場合 PARAM:
RET:	戻り値の表示の開始を示します。 制御ファイルで出力の有無を指定できます。	ASCIIコード“RET:”の文字列です。 このキーワードの後ろにメソッドから返された戻り値の情報が出力されます。 戻り値がない場合、“void”という文字列が出力されます。 データ型による詳細な出力形式は、“ <a href="#">■引数や戻り値の出力形式</a> ”を参照してください。

## ■引数や戻り値の出力形式

引数や戻り値は、そのデータ型により、次の形式で出力されます。

データ型	表示形式	例
戻り値なし	“void”という文字列	void
論理型 boolean	(型):ブール値	(boolean):true
数値型 short,int,long, float,double	(型):10進表記の数字列	(int):100
byte型	(型):16進表記の数字列	(byte):61
char型	(型):文字	(char):A
String型	(型):文字列	(String):あいうえお
オブジェクト	(クラス名):toStringからの復帰文字列	(classA):data1=100,dataA=アイウエオ
論理型の配列(1次元)	(型):配列数:[ブール値][ブール値]...	(boolean):3:[true][false][true]
数値型の配列(1次元)	(型):配列数:[数字列][数字列]...	(short):2:[1][2]
byte型の配列(1次元)	(型):配列数:[16進表記の数字列][16進表記の数字列]...	(byte)2:[61][62]
char型の配列(1次元)	(型):配列数:[文字][文字]...	(char)2:[あ][A]
String型の配列(1次元)	(型):配列数:[文字列][文字列]...	(String)2:[あいう][ABC]
オブジェクトの配列(1次元)	(クラス名):配列数:[toStringからの復帰文字列][toStringからの復帰文字列]...	(classA)2:[data1=100,dataA=アイウエオ][data1=200, dataA=ABCDEF]
論理型の配列(2次元以上)	(型):配列数, 配列数...:[ブール値][ブール値]...[[ブール値][ブール値]...]...	(boolean):2,3:[true][false][true]][[true][true][true]]
数値型の配列(2次元以上)	(型):配列数,配列数...:[数字列][数字列]...[[数字列][数字列]...]...	(short):2,2:[1][2]] [[3][4]]
byte型の配列(2次元以上)	(型):配列数,配列数...:[16進表記の数字列][16進表記の数字列]...[[16進表記の数字列][16進表記の数字列]...]...	(char)2,2:[61][62]] [[63][64]]
char型の配列(2次元以上)	(型):配列数,配列数...:[文字][文字]...[[文字][文字]...]...	(char)2,2:[あ][A]] [[え][D]]
String型の配列(2次元以上)	(型):配列数,配列数...:[文字列][文字列]...[[文字列][文字列]...]...	(String)2,2:[あいう][ABC]] [[えおか][DEF]]
オブジェクトの配列(2次元以上)	(クラス名):配列数,配列数...:[toStringからの復帰文字列][toStringからの復帰文字列]...[[toStringからの復帰文字列][toStringからの復帰文字列]...]...	(classA)2,2:[data1=100, dataA=アイウエオ][data1=200,dataA=ABCDEF]] [[data1=300,dataA=かきくけこ][data1=400,dataA=GHIJKL]]



引数や戻り値が、各要素数が一定でない多次元配列の場合、引数や戻り値の情報には、不足している要素をnullの表示(())として出力されます。また、配列数も最大の要素数に合わせて出力されます。

引数や戻り値の例

```
String[][] str = {
    {"Flintstones", "Fred", "Wilma"},
    {"Rubbles", "Barney", "Betty", "Bam Bam"},
    {"Jetsons", "George", "Jane", "Elroy", "Judy"},
    {"Scooby Doo Gang", "Shaggy"}
};
```

出力例(引数の場合です。戻り値の場合は、“PARAM”の代わりに“RET:”が出力されます。)

```
(略) PARAM:(String):4,5:[Flintstones] [Fred] [Wilma] [] [] [[Rubbles] [Barney] [Betty] [Bam Bam] []] [[Jetsons]
[George] [Jane] [Elroy] [Judy]] [[Scooby Doo Gang] [Shaggy] [] [] []]
```

## ■ null値の表示

引数や戻り値がオブジェクトであり、その値がnullの場合、次のように出力されます。

- 配列以外の場合・・・(オブジェクト名):[]
- 配列の場合・・・・(オブジェクト名): []

## ■ 文字関連の16進数表示

文字関連のデータ表示に16進表示を付加するには、制御ファイルに“valueLevel=addHex”と指定します。  
文字関連のデータとは、以下のデータ型を持つ引数や戻り値のことです。

- byte型
- char型
- String型

16進表示は、“[引数や戻り値の出力形式](#)”に示した表示形式の後ろに“Hex=”で区切り、数字列で示します。

データ型	表示形式	例
byte型	(byte):数字列 Hex=xxxx	(byte):61 Hex=0036 0031
char型	(char):文字 Hex=xxxx	(char):a Hex=0061
String型	(String):文字列 Hex=xxxx xxxx[ ...]	(String):abcd Hex=0061 0062 0063 0064



次の場合、制御ファイルに“valueLevel=addHex”の指定がある場合でも、16進表示は抑止されます。

- char型の場合、null文字
- String型の場合、null文字列

## A.1.7 トレース情報の分析

ここでは、Servletのサンプルを利用して、トレース情報の分析方法を説明します。

### ■ サンプルNumGameの場合

サンプルNumGameを例として、トレース情報の見方について説明します。  
ここでは、NumGameを初回呼出し後、Servletコンテナを停止した場合のトレース情報を例に説明します。

#### 制御ファイル

制御ファイルに、以下の定義を追加します。  
設定方法の詳細については、“[A.1.4 制御ファイルの作成方法](#)”を参照してください。

```
-----  
traceMethod=NumGame *  
traceMethod=NumGameParameter *  
traceMethod=pkgNumGameBean. NumGameBean *  
valueLevel=none  
timeStamp=true  
-----
```

#### トレース情報

採取されたトレースは以下ようになります。

注) 左端の数字は、説明のための行番号です。実際には出力されません。

---

```
1: 12/09/2003 11:06:10.187, in, Thread-20, NumGame, init
2: 12/09/2003 11:06:10.218, out, Thread-20, NumGame, init
3: 12/09/2003 11:06:10.343, in, Thread-20, NumGame, doGet
4: 12/09/2003 11:06:10.531, in, Thread-20, pkgNumGameBean.NumGameBean, <init>
5: 12/09/2003 11:06:10.531, out, Thread-20, pkgNumGameBean.NumGameBean.<init>
  . . . (略) . . .
6: 12/09/2003 11:06:10.671, in, Thread-20, NumGameParameter, getParameter
7: 12/09/2003 11:06:10.765, out, Thread-20, NumGameParameter, getParameter
  . . . (略) . . .
8: 12/09/2003 11:06:13.546, out, Thread-20, NumGame, doGet
9: 12/09/2003 11:06:24.156, in, Thread-20, NumGame, destroy
10: 12/09/2003 11:06:24.156, out, Thread-20, NumGame, destroy
```

---

### トレース情報の見方

init,doGet,destroyメソッドの呼出しと復帰 1,2,3,8,9,10行目

“[時刻], in, ...”はメソッドの呼出しを、“[時刻], out, ...”は復帰を表しています。

上の例では正常に処理が行われているため、init,service,destroyともに“[時刻], in, ...”と“[時刻], out, ...”が対になって出力されています。

初期化(initメソッド) 1,2行目

NumGameサーブレットの初回呼出し時にinitメソッドが呼び出されます。

NumGameのinitメソッドでは処理を行っていないため、init中のトレース情報は出力されていません。

サービス(doGetメソッド) 3~8行目

各メソッドとも、呼出し後正常に復帰しているため、“[時刻], in, ...”と“[時刻], out, ...”が対になって出力されています。

doGetメソッドの中には、Servletの主処理が記述されています。

トレース情報の意味は、“[A.1.6 トレース情報出力の形式](#)”を参照してください。

終了処理(destroyメソッド) 9,10行目

Servletコンテナの停止時に呼び出されるdestroyメソッドのトレース情報です。

NumGameのdestroyメソッドでは処理を行っていないため、destroy中のトレース情報は出力されていません。

## ■異常のあるサーブレットの例

異常のあるサーブレットを例として、トレース情報の見方について説明します。

### 応答がなくなるNumGameのソース

---

```
1: import javax.servlet.http.*;
2: import javax.servlet.*;
3: import java.io.*;
4: public class NumGame extends HttpServlet
5: {
6:     public void doGet(HttpServletRequest req, HttpServletResponse res)
7:     throws ServletException, IOException {
8:         HttpSession session = req.getSession(true);
9:         try {
10:            // アプリケーションオブジェクトの取得
11:            NumGameBean _application =
12:                (NumGameBean) session.getAttribute("gamebean");
13:            if (_application == null) { // 新規またはSessionTimeOut
14:                _application = new NumGameBean();
15:                session.setAttribute("gamebean", _application);
16:                _application.init();
17:            }
18:            // 使用するパラメータークラスの決定
19:            NumGameParameter inform = new NumGameParameter();
20:            // パラメータークラスに入力されたデータを設定
```

```
21:    inform.setRequest(req);
22:    int num;
23:    try {
24:        num = Integer.parseInt(inform.getParameter("Number"));
25:    } catch (NumberFormatException ne) {
26:        num = 0;
27:    }
```

以下省略

---

## 制御ファイル

---

```
traceMethod=NumGame *
traceMethod=NumGameParameter *
traceMethod=pkgNumGameBean.NumGameBean *
valueLevel=none
timeStamp=true
```

---

## トレース情報

---

```
19/10/2003 10:59:53:922, in, Thread-20, NumGame, doGet
19/10/2003 10:59:54:032, in, Thread-20, pkgNumGameBean.NumGameBean, <init>
19/10/2003 10:59:55:434, out, Thread-20, pkgNumGameBean.NumGameBean, <init>
19/10/2003 10:59:58:408, in, Thread-20, NumGameParameter, <init>
19/10/2003 10:59:58:418, out, Thread-20, NumGameParameter, <init>
19/10/2003 10:59:59:630, in, Thread-20, NumGameParameter, setRequest
19/10/2003 10:59:59:640, out, Thread-20, NumGameParameter, setRequest
19/10/2003 10:59:59:670, in, Thread-20, NumGameParameter, getParameter
```

---

## 分析結果

トレース情報より、以下がわかります。

- “19/10/2003 10:59:59:670, in, Thread-20, NumGameParameter, getParameter”が出力されていることから、NumGameParameter#getParameterメソッドは呼び出されている。
- 他方、対となるNumGameParameter#getParameterに対する“..., out, ...”が出力されていないことから、NumGameParameter#getParameterメソッドが復帰していない。

以上より、NumGameParameter#getParameterメソッドの中で処理が止まっていたことがわかります。

EJBやJavaアプリケーションの場合も同様です。ユーザアプリケーションのクラスを設定ファイルに設定し、トレース情報を採取します。“in, ...”に対して“out, ...”がなく、ログが出力されていないような場合、応答がないということがわかります。

---

## A.1.8 トレース不可のクラス

---

制御ファイルの“traceMethod”に指定してもトレース情報に出力できないクラスがあります。ここではトレース情報に出力されないクラスを表で列挙します。

### クラス名

トレース情報に出力されないクラスの名前を示します。

### 対象範囲

トレース情報に出力されない対象のOS名およびJDK/JREのバージョンを示します。何の記述がない場合は、OSおよびJDK/JREのバージョンに関係なくトレース情報が出力されないことを示します。対象範囲における表記は、次のとおりです。

#### Windows

Windows(R)上で動作する場合

Solaris

Solaris上で動作する場合

Linux

Linux上で動作する場合

JDK6

JDK/JRE 6を使用する場合

JDK7

JDK/JRE 7を使用する場合

## ■トレースを出力できないクラス

表A.1 表トレース情報に出力できないクラス

クラス名	対象範囲					
	Windows		Solaris		Linux	
	JDK6	JDK7	JDK6	JDK7	JDK6	JDK7
java.io.BufferedInputStream						
java.io.BufferedOutputStream						
java.io.BufferedReader						
java.io.BufferedWriter						
java.io.Closeable						
java.io.ExpiringCache						
java.io.ExpiringCache\$1						
java.io.ExpiringCache\$Entry						
java.io.File						
java.io.FileDescriptor						
java.io.FileDescriptor\$1		○	○	○	○	○
java.io.FileInputStream						
java.io.FileOutputStream						
java.io.FileReader						
java.io.FileSystem						
java.io.FilterInputStream						
java.io.FilterOutputStream						
java.io.Flushable						
java.io.InputStream						
java.io.InputStreamReader						
java.io.IOException	○		○		○	
java.io.ObjectStreamClass	○		○		○	
java.io.ObjectStreamField						
java.io.OutputStream						
java.io.OutputStreamWriter						
java.io.PrintStream						
java.io.Reader						

java.io.Serializable						
java.io.UnixFileSystem			○	○	○	○
java.io.Writer						
java.lang.AbstractStringBuilder						
java.lang.Appendable						
java.lang.ArithmeticException						
java.lang.ArrayStoreException						
java.lang.AssertionStatusDirectives	○		○		○	
java.lang.AutoCloseable		○		○		○
java.lang.Boolean						
java.lang.BootstrapMethodError		○		○		○
java.lang.Byte						
java.lang.Character						
java.lang.CharacterData		○		○		○
java.lang.CharacterDataLatin1						
java.lang.CharSequence						
java.lang.Class						
java.lang.Class\$1						
java.lang.Class\$3						
java.lang.ClassCastException						
java.lang.ClassFormatError	○		○		○	
java.lang.ClassLoader						
java.lang.ClassLoader\$3						
java.lang.ClassLoader\$NativeLibrary						
java.lang.ClassLoader\$ParallelLoaders		○		○		○
java.lang.ClassNotFoundException						
java.lang.Cloneable						
java.lang.Comparable						
java.lang.Compiler						
java.lang.Compiler\$1						
java.lang.Double						
java.lang.Error						
java.lang.Exception						
java.lang.Float						
java.lang.IllegalArgumentException	○		○		○	
java.lang.IllegalMonitorStateException						
java.lang.IncompatibleClassChangeError						
java.lang.Integer						
java.lang.invoke.AdapterMethodHandle		○		○		○
java.lang.invoke.BoundMethodHandle		○		○		○

java.lang.invoke.CallSite		<input type="radio"/>		<input type="radio"/>		<input type="radio"/>
java.lang.invoke.ConstantCallSite		<input type="radio"/>		<input type="radio"/>		<input type="radio"/>
java.lang.invoke.CountingMethodHandle		<input type="radio"/>		<input type="radio"/>		<input type="radio"/>
java.lang.invoke.DirectMethodHandle		<input type="radio"/>		<input type="radio"/>		<input type="radio"/>
java.lang.invoke.MemberName		<input type="radio"/>		<input type="radio"/>		<input type="radio"/>
java.lang.invoke.MethodHandle		<input type="radio"/>		<input type="radio"/>		<input type="radio"/>
java.lang.invoke.MethodHandleNatives		<input type="radio"/>		<input type="radio"/>		<input type="radio"/>
java.lang.invoke.MethodType		<input type="radio"/>		<input type="radio"/>		<input type="radio"/>
java.lang.invoke.MethodTypeForm		<input type="radio"/>		<input type="radio"/>		<input type="radio"/>
java.lang.invoke.MutableCallSite		<input type="radio"/>		<input type="radio"/>		<input type="radio"/>
java.lang.invoke.VolatileCallSite		<input type="radio"/>		<input type="radio"/>		<input type="radio"/>
java.lang.invoke.WrongMethodTypeException		<input type="radio"/>		<input type="radio"/>		<input type="radio"/>
java.lang.Iterable						
java.lang.LinkageError						
java.lang.Long						
java.lang.Math						
java.lang.NoClassDefFoundError						
java.lang.NoSuchMethodError						
java.lang.NullPointerException						
java.lang.Number						
java.lang.Object						
java.lang.OutOfMemoryError						
java.lang.Package	<input type="radio"/>		<input type="radio"/>		<input type="radio"/>	
java.lang.Readable						
java.lang.ref.Finalizer						
java.lang.ref.Finalizer\$FinalizerThread						
java.lang.ref.FinalReference						
java.lang.ref.PhantomReference						
java.lang.ref.Reference						
java.lang.ref.Reference\$Lock						
java.lang.ref.Reference\$ReferenceHandler						
java.lang.ref.ReferenceQueue						
java.lang.ref.ReferenceQueue\$Lock						
java.lang.ref.ReferenceQueue\$Null						
java.lang.ref.SoftReference						
java.lang.ref.WeakReference						
java.lang.reflect.AccessibleObject						
java.lang.reflect.AnnotatedElement						
java.lang.reflect.Array						
java.lang.reflect.Constructor						

java.lang.reflect.Field						
java.lang.reflect.GenericDeclaration						
java.lang.reflect.Member						
java.lang.reflect.Method						
java.lang.reflect.Modifier						
java.lang.reflect.ReflectAccess						
java.lang.reflect.ReflectPermission						
java.lang.reflect.Type						
java.lang.ReflectiveOperationException		○		○		○
java.lang.Runnable						
java.lang.Runtime						
java.lang.RuntimeException						
java.lang.RuntimePermission						
java.lang.Short						
java.lang.StackOverflowError						
java.lang.StackTraceElement						
java.lang.String						
java.lang.String\$CaseInsensitiveComparator						
java.lang.StringBuffer						
java.lang.StringBuilder						
java.lang.StringCoding	○	○			○	○
java.lang.StringCoding\$StringDecoder	○	○			○	○
java.lang.StringCoding\$StringEncoder	○	○			○	○
java.lang.StringValue	○		○		○	
java.lang.System						
java.lang.System\$2						
java.lang.SystemClassLoaderAction						
java.lang.Terminator						
java.lang.Terminator\$1						
java.lang.Thread						
java.lang.Thread\$UncaughtExceptionHandler						
java.lang.ThreadDeath						
java.lang.ThreadGroup						
java.lang.ThreadLocal						
java.lang.ThreadLocal\$ThreadLocalMap	○	○			○	○
java.lang.ThreadLocal\$ThreadLocalMap\$Entry	○	○			○	○
java.lang.Throwable						
java.lang.VirtualMachineError						
java.lang.Void	○		○		○	
java.net.Parts						

java.net.URL						
java.net.URLClassLoader						
java.net.URLClassLoader\$7						
java.net.URLStreamHandler						
java.net.URLStreamHandlerFactory						
java.nio.Bits						
java.nio.Bits\$1						
java.nio.Buffer						
java.nio.ByteBuffer						
java.nio.ByteOrder						
java.nio.CharBuffer						
java.nio.charset.Charset						
java.nio.charset.Charset\$3	○	○				
java.nio.charset.CharsetDecoder						
java.nio.charset.CharsetEncoder						
java.nio.charset.CoderResult						
java.nio.charset.CoderResult\$1						
java.nio.charset.CoderResult\$2						
java.nio.charset.CoderResult\$Cache						
java.nio.charset.CodingErrorAction						
java.nio.charset.spi.CharsetProvider						
java.nio.HeapByteBuffer						
java.nio.HeapCharBuffer						
java.security.AccessControlContext						
java.security.AccessControlContext\$1	○		○		○	
java.security.AccessController						
java.security.BasicPermission						
java.security.cert.Certificate						
java.security.CodeSource						
java.security.Guard						
java.security.Permission						
java.security.PermissionCollection	○		○		○	
java.security.Principal		○		○		○
java.security.PrivilegedAction						
java.security.PrivilegedExceptionAction						
java.security.ProtectionDomain						
java.security.ProtectionDomain\$1		○		○		○
java.security.ProtectionDomain\$3		○		○		○
java.security.ProtectionDomain\$Key		○		○		○
java.security.SecureClassLoader						

java.util.AbstractCollection						
java.util.AbstractList						
java.util.AbstractMap						
java.util.AbstractSet						
java.util.ArrayList						
java.util.Arrays						
java.util.BitSet						
java.util.Collection						
java.util.Collections						
java.util.Collections\$EmptyList						
java.util.Collections\$EmptyMap						
java.util.Collections\$EmptySet						
java.util.Collections\$ReverseComparator	○		○		○	
java.util.Collections\$SetFromMap		○		○		○
java.util.Collections\$SynchronizedCollection		○		○		○
java.util.Collections\$SynchronizedMap	○		○		○	
java.util.Collections\$SynchronizedSet		○		○		○
java.util.Collections\$UnmodifiableCollection		○		○		○
java.util.Collections\$UnmodifiableList		○		○		○
java.util.Collections\$UnmodifiableRandomAccessList		○		○		○
java.util.Comparator						
java.util.concurrent.atomic.AtomicInteger						
java.util.concurrent.atomic.AtomicReferenceFieldUpdater						
java.util.concurrent.atomic.AtomicReferenceFieldUpdater\$AtomicReferenceFieldUpdaterImpl						
java.util.concurrent.ConcurrentHashMap						
java.util.concurrent.ConcurrentHashMap\$HashEntry						
java.util.concurrent.ConcurrentHashMap\$Segment						
java.util.concurrent.ConcurrentMap						
java.util.concurrent.locks.AbstractOwnableSynchronizer						
java.util.concurrent.locks.AbstractQueuedSynchronizer						
java.util.concurrent.locks.AbstractQueuedSynchronizer\$Node						
java.util.concurrent.locks.Lock						
java.util.concurrent.locks.ReentrantLock						
java.util.concurrent.locks.ReentrantLock\$NonfairSync						
java.util.concurrent.locks.ReentrantLock\$Sync						
java.util.Dictionary						

java.util.Enumeration						
java.util.HashMap						
java.util.HashMap\$Entry						
java.util.HashMap\$EntryIterator		○		○		○
java.util.HashMap\$EntrySet		○		○		○
java.util.HashMap\$HashIterator		○		○		○
java.util.HashSet						
java.util.Hashtable						
java.util.Hashtable\$EmptyEnumerator	○		○		○	
java.util.Hashtable\$EmptyIterator	○		○		○	
java.util.Hashtable\$Entry						
java.util.Hashtable\$EntrySet		○		○		○
java.util.Hashtable\$Enumerator		○		○		○
java.util.Iterator						
java.util.LinkedHashMap						
java.util.LinkedHashMap\$Entry						
java.util.List						
java.util.Locale						
java.util.Locale\$Cache		○		○		○
java.util.Locale\$LocaleKey		○		○		○
java.util.Map						
java.util.Map\$Entry						
java.util.NavigableMap	○	○				
java.util.Properties						
java.util.RandomAccess						
java.util.Set						
java.util.SortedMap	○	○				
java.util.Stack						
java.util.StringTokenizer						
java.util.TreeMap	○	○				
java.util.TreeMap\$Entry	○	○				
java.util.Vector						
java.util.WeakHashMap		○		○		○
java.util.WeakHashMap\$Entry		○		○		○
java.util.WeakHashMap\$KeySet		○		○		○
sun.io.Win32ErrorMode	○	○				
sun.jkernel.DownloadManager	○		○		○	
sun.jkernel.DownloadManager\$1	○		○		○	
sun.jkernel.DownloadManager\$2	○		○		○	
sun.misc.ASCIICaseInsensitiveComparator	○	○				

sun.misc.AtomicLong	<input type="radio"/>		<input type="radio"/>		<input type="radio"/>	
sun.misc.AtomicLongCSImpl	<input type="radio"/>		<input type="radio"/>		<input type="radio"/>	
sun.misc.JavaIOFileDescriptorAccess		<input type="radio"/>				
sun.misc.JavaLangAccess						
sun.misc.JavaNetAccess						
sun.misc.JavaNioAccess						
sun.misc.JavaSecurityAccess						
sun.misc.JavaSecurityProtectionDomainAccess		<input type="radio"/>		<input type="radio"/>		<input type="radio"/>
sun.misc.Launcher						
sun.misc.Launcher\$AppClassLoader						
sun.misc.Launcher\$AppClassLoader\$1						
sun.misc.Launcher\$ExtClassLoader						
sun.misc.Launcher\$ExtClassLoader\$1						
sun.misc.Launcher\$Factory						
sun.misc.MetaIndex						
sun.misc.NativeSignalHandler						
sun.misc.OSEnvironment						
sun.misc.PostVMInitHook	<input type="radio"/>		<input type="radio"/>		<input type="radio"/>	
sun.misc.SharedSecrets						
sun.misc.Signal						
sun.misc.SignalHandler						
sun.misc.SoftCache	<input type="radio"/>		<input type="radio"/>		<input type="radio"/>	
sun.misc.Unsafe						
sun.misc.URLClassPath						
sun.misc.Version						
sun.misc.VM						
sun.net.www.ParseUtil						
sun.net.www.protocol.file.Handler						
sun.net.www.protocol.jar.Handler						
sun.nio.cs.AbstractCharsetProvider	<input type="radio"/>	<input type="radio"/>				
sun.nio.cs.ArrayDecoder		<input type="radio"/>		<input type="radio"/>		<input type="radio"/>
sun.nio.cs.ArrayEncoder		<input type="radio"/>		<input type="radio"/>		<input type="radio"/>
sun.nio.cs.ext.DelegatableDecoder	<input type="radio"/>	<input type="radio"/>				
sun.nio.cs.ext.DoubleByte		<input type="radio"/>				
sun.nio.cs.ext.DoubleByte\$Decoder		<input type="radio"/>				
sun.nio.cs.ext.DoubleByte\$Encoder		<input type="radio"/>				
sun.nio.cs.ext.DoubleByteDecoder	<input type="radio"/>					
sun.nio.cs.ext.DoubleByteEncoder	<input type="radio"/>					
sun.nio.cs.ext.ExtendedCharsets	<input type="radio"/>	<input type="radio"/>				
sun.nio.cs.ext.JIS_X_0201\$Decoder	<input type="radio"/>					

sun.nio.cs.ext.JIS_X_0201\$Encoder	<input type="radio"/>					
sun.nio.cs.ext.MS932	<input type="radio"/>	<input type="radio"/>				
sun.nio.cs.ext.MS932\$Decoder	<input type="radio"/>					
sun.nio.cs.ext.MS932\$Encoder	<input type="radio"/>					
sun.nio.cs.ext.MS932DB\$Decoder	<input type="radio"/>					
sun.nio.cs.ext.MS932DB\$Encoder	<input type="radio"/>					
sun.nio.cs.FastCharsetProvider						
sun.nio.cs.HistoricallyNamedCharset						
sun.nio.cs.ISO_8859_1					<input type="radio"/>	
sun.nio.cs.ISO_8859_1\$Decoder					<input type="radio"/>	
sun.nio.cs.ISO_8859_1\$Encoder					<input type="radio"/>	
sun.nio.cs.SingleByteDecoder	<input type="radio"/>					
sun.nio.cs.SingleByteEncoder	<input type="radio"/>					
sun.nio.cs.StandardCharsets						
sun.nio.cs.StandardCharsets\$Aliases						
sun.nio.cs.StandardCharsets\$Cache						
sun.nio.cs.StandardCharsets\$Classes						
sun.nio.cs.StreamDecoder						
sun.nio.cs.StreamEncoder						
sun.nio.cs.Surrogate	<input type="radio"/>		<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
sun.nio.cs.Surrogate\$Parser	<input type="radio"/>		<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
sun.nio.cs.US_ASCII			<input type="radio"/>	<input type="radio"/>		<input type="radio"/>
sun.nio.cs.US_ASCII\$Decoder			<input type="radio"/>	<input type="radio"/>		<input type="radio"/>
sun.nio.cs.US_ASCII\$Encoder			<input type="radio"/>	<input type="radio"/>		<input type="radio"/>
sun.reflect.ConstantPool						
sun.reflect.ConstructorAccessor						
sun.reflect.ConstructorAccessorImpl						
sun.reflect.DelegatingClassLoader						
sun.reflect.DelegatingConstructorAccessorImpl						
sun.reflect.FieldAccessor						
sun.reflect.FieldAccessorImpl						
sun.reflect.LangReflectAccess						
sun.reflect.MagicAccessorImpl						
sun.reflect.MethodAccessor						
sun.reflect.MethodAccessorImpl						
sun.reflect.misc.ReflectUtil						
sun.reflect.NativeConstructorAccessorImpl						
sun.reflect.Reflection						
sun.reflect.ReflectionFactory						
sun.reflect.ReflectionFactory\$1						

sun.reflect.ReflectionFactory \$GetReflectionFactoryAction						
sun.reflect.UnsafeFieldAccessorImpl						
sun.reflect.UnsafeStaticFieldAccessorImpl						
sun.security.action.GetPropertyAction						
sun.security.util.Debug						
sun.util.locale.BaseLocale		○		○		○
sun.util.locale.BaseLocale\$Cache		○		○		○
sun.util.locale.BaseLocale\$Key		○		○		○
sun.util.locale.LocaleObjectCache		○		○		○
sun.util.locale.LocaleObjectCache\$CacheEntry		○		○		○
sun.util.locale.LocaleUtils		○		○		○
sun.util.PreHashMap						

## A.1.9 メッセージ

ここでは、トレース情報に表示されるメッセージとその意味を説明します。

### ■出力形式

メッセージの出力形式は、以下のようになります。

```
FJTRACE:メッセージ番号: メッセージ内容
```

### ■メッセージとその意味

#### FJTRACE:100

**Unexpected OPCODE was detected. (OPCODE = %s [%n])**

[可変情報]

%s=認識できなかったオペレーションコード(ニーモニック)

%n=認識できなかったオペレーションコード(番号)

[意味]

exit情報の出力処理を埋め込もうとしたが、復帰値を取得できませんでした。

[処理]

メソッドトレースは通常通り実施します。

exit情報の復帰値として、このメッセージが出力されます。

#### FJTRACE:101

**[%s]**

[可変情報]

%s=発生したException、Errorクラス名

[意味]

enter情報に渡されたパラメタ、またはexit情報に渡された復帰値が、オブジェクトであった場合、トレース情報としてtoStringメソッドで得られる値を出力する。

そのtoStringメソッド内で、ExceptionやErrorが発生しました。

[処理]

メソッドトレースは通常通り実施します。

enter情報のパラメタ値、またはexitの復帰値として、このメッセージが出力されます。

#### FJTRACE:201

#### **FileNotFoundException.getMessage()の返却文字列**

[意味]

指定された制御ファイルが見つかりません。

[処理]

メソッドトレースを行わず、アプリケーションを実行します。

**FJTRACE:202**

#### **IOException.getMessage()の返却文字列**

[意味]

制御ファイルのオープンでエラーが発生しました。

[処理]

メソッドトレースを行わず、アプリケーションを実行します。

**FJTRACE:203**

#### **The minimum of "traceFileCount" is 1. Use the minimum value(traceFileCount = 1).**

[意味]

“traceFileCount”の指定に誤りがあります(0以下の数値が指定されています)。

[処理]

“traceFileCount”に1を設定します。

**FJTRACE:204**

#### **The maximum of "traceFileCount" is 10. Use the maximum value(traceFileCount = 10).**

[意味]

“traceFileCount”の指定に誤りがあります(指定できる最大値"10"を超えた数値が指定されています)。

[処理]

“traceFileCount”に10を設定します。

**FJTRACE:205**

#### **The specification error of "traceFileCount". Use the default value(traceFileCount = 2).**

[意味]

“traceFileCount”の指定に誤りがあります(数値以外の値が指定されています)。

[処理]

“traceFileCount”に2を設定します。

**FJTRACE:206**

#### **The specification error of "traceFileSize". Use the default value(traceFileSize = 1).**

[意味]

“traceFileSize”の指定に誤りがあります(0以下の数値が指定されています)。

[処理]

“traceFileSize”に1を設定します。

**FJTRACE:207**

#### **The maximum of "traceFileSize" is 2147483647. Use the default value(traceFileSize = 1).**

[意味]

“traceFileSize”の指定に誤りがあります(指定できる最大値"2147483647"を超えた数値が指定されています)。

[処理]

“traceFileSize”に1を設定します。

**FJTRACE:208**

#### **The specification error of "traceFileSize". Use the default value(traceFileSize = 1).**

[意味]

“traceFileSize”の指定に誤りがあります(数値以外の値が指定されています)。

[処理]

“traceFileSize”に1を設定します。

#### FJTRACE:209

##### **Unrecognized statement "%s".**

[可変情報]

%s=指定された行

[意味]

定義項目が“定義名=値”の形式で指定されていません。

または、指定された定義名に誤りがあります。

このメッセージは、制御ファイルに“statementStyle”の記述がある場合に出力されます(“statementStyle”の記述がない場合には出力されません)。

[処理]

指定された行は無効となります。

#### FJTRACE:210

##### **The specification error of "%s1". Use the default value(%s2).**

[可変情報]

%s1=指定された行

%s2=採用された省略値

[意味]

“statementStyle”、“valueLevel”および“timeStamp”に指定された値に誤りがあります。

このメッセージは、制御ファイルに“statementStyle”の記述がある場合に出力されます(“statementStyle”の記述がない場合には出力されません)。

[処理]

指定された行は無効となり、省略値が採用されます。

#### FJTRACE:211

##### **Two or more same control items were specified (%s).**

[可変情報]

%s=採用された行

[意味]

同じ定義項目が複数定義されています。

このメッセージは、制御ファイルに“statementStyle”の記述がある場合に出力されます(“statementStyle”の記述がない場合には出力されません)。

[処理]

最後に正しく指定された行が有効(採用された行)となります。また、採用された行以外は無効となります。

#### FJTRACE:212

##### **Both "traceFile" and "traceDirectory" are specified. "traceFile" is used.**

[意味]

“traceFile”、“traceDirectory”の両方の指定があります。“traceFile”の指定が有効となります。

このメッセージは、制御ファイルに“statementStyle”の記述がある場合に出力されます(“statementStyle”の記述がない場合には出力されません)。

[処理]

“traceDirectory”の指定は無効となります。

#### FJTRACE:213

##### **The control item "%s1" is specified. "%s2" is ignored.**

[可変情報]

%s1=優先された定義  
%s2=無効となった定義

[意味]

組み合わせ不可能な定義が指定されています。次の定義の組み合わせが考えられます。

- “statementStyle = BASIC”と解釈される定義があり、かつ、“classPath”の定義がある場合
- “statementStyle = BASIC”と解釈される定義があり、かつ、“not\_traceMethod”の定義がある場合
- “traceFile”の定義があり、かつ、“traceFileCount”の定義がある場合
- “traceFile”の定義があり、かつ、“traceFileSize”の定義がある場合

このメッセージは、制御ファイルに“statementStyle”の記述がある場合に出力されます(“statementStyle”の記述がない場合には出力されません)。

[処理]

組み合わせ不可能な定義(可変情報 “%s2”に示される定義)は、無効となります。

**FJTRACE:400**

**Syntax error of a regular expression. [%s]**

[可変情報]

%s=指定された行

[意味]

“traceMethod”、“not\_traceMethod”に指定された正規表現の構文に誤りがあります。

このメッセージは、制御ファイルに“statementStyle”の記述がある場合に出力されます(“statementStyle”の記述がない場合には出力されません)。

[処理]

指定された行は無効となります。

**FJTRACE:500**

**%s does not exist.**

[可変情報]

%s=ディレクトリ名

[意味]

“traceDirectory”で指定されたディレクトリが存在しません。

[処理]

メソッドトレースを行わず、アプリケーションを実行します。

**FJTRACE:501**

**%s is not a directory.**

[可変情報]

%s=ディレクトリ名

[意味]

“traceDirectory”で指定されたディレクトリはディレクトリではありません。

[処理]

メソッドトレースを行わず、アプリケーションを実行します。

**FJTRACE:502**

**%s is not a writable directory.**

[可変情報]

%s=ディレクトリ名

[意味]

“traceDirectory”で指定されたディレクトリに対して、書き込み権がありません。

[処理]

メソッドトレースを行わず、アプリケーションを実行します。

#### FJTRACE:503

**Cannot create logfile %s. Cannot continue logging.**

[可変情報]

%s=トレース情報出力ファイル名

[意味]

トレース情報出力ファイルを作成できません。

[処理]

これ以降のトレース出力は行いません。

#### FJTRACE:602

**A necessary module for the FJTRACE execution is not found.**

[意味]

メソッドトレース (FJTRACE) 実行に必要なモジュールが見つかりません。

fjtrace.dllまたはlibfjtrace.soが、必要な実行形式をしていない、またはmt.jarと同じディレクトリ内に配置されていない場合があります。

[処理]

これ以降のトレース出力は行いません。

#### FJTRACE:603

**API failure. (%s)**

[可変情報]

%s=ネイティブライブラリでの処理中にエラー復帰したAPI名

##### **Solaris32/64**

malloc, thr\_keycreateのいずれかになります。

##### **Linux32/64**

malloc, pthread\_key\_createのいずれかになります。

[意味]

メソッドトレース (FJTRACE) 実行時、ネイティブライブラリでの処理中にAPIが異常復帰しました。

原因として、メモリ不足を含む、リソースの不足が考えられます。

[処理]

下記の形式でメッセージを標準出力(または、使用方法記載のログファイル)に出力し、メソッドトレース、および、アプリケーションの実行を停止します。

```
FATAL ERROR in native method: FJTRACE:603: API failure. (API名)
```

```
以下、この事象発生時のスタックトレースの出力
```

たとえば、Solaris上でthr\_keycreateがメモリ不足が発生している場合は、次のように出力されます。

```
FATAL ERROR in native method: FJTRACE:603: API failure. (thr_keycreate)
```

```
at com.fujitsu.jvm.mt.ProcessInfo.getStatus(Native Method)
```

```
at com.fujitsu.jvm.mt.B2B.init(B2B.java:116)
```

```
at com.fujitsu.jvm.prof.FJProf2.prof_init(FJProf2.java:15)
```

## A.1.10 注意事項

ここでは、メソッドトレース機能の注意事項を説明します。

- メソッドトレース機能は、製品添付のJDK/JREでのみ動作可能です。
- メソッドトレース機能はJavaの提供するJVMTI機能とバイトコードの更新により実現しています。そのため、他のJVMTI機能の実装によるプロファイラやデバッガと同時に使用しないことをお勧めします。
- メソッドトレース機能を使用する場合、実行性能が劣化します。したがって、本運用ではトレースを行わないことをお勧めします。
- 使用するJDK/JREの違いによって、メソッドトレース機能を有効にする方法が異なります。詳細は、“A.1.5 使用方法”を参照してください。
- 任意のディレクトリを作成してメソッドトレース機能の動作に必要なモジュールをコピーして使用することもできます。その場合、作成するディレクトリのパスは必ず“tools¥fjtrace”(Windows(R)の場合)または“tools/fjtrace”(Solaris/Linuxの場合)で終わるものにしてください。また、環境変数もそれに合わせて設定してください。
- “traceFile”の指定は、デバッグ目的だけで使用し、アプリケーションを連続で実行する場合には使用しないでください。この指定ではファイルローテーションされず、常に1つのファイルにログが書き込まれるため、巨大なログファイルとなります。
- ファイルローテーションするように指定した場合、1つのファイルサイズが、“traceFileSize”に指定した値よりも大きくなる場合があります。これは、“traceFileSize”に指定した値の境界で、1行に表されるべき情報が複数のファイルに分割されないようにしているためです。
- 文字列の出力は、常に、JavaプロセスのOSのロケールに従った標準のエンコーディングで行われます。Javaプロセス内で、別のエンコーディングを使用している場合に、文字化けする可能性もあります。その場合は、“valueLevel=addHex”等で16進数表示するか、出力ファイルのコード変換などを適宜行ってください。
- 以下の場合、カレントディレクトリが移動します。そのため“traceFile”か“traceDirectory”を必ず指定してください。

- IJServerクラスタ



- IJServer

- ファイル出力中、出力先のディスク容量が不足した場合、それ以降はトレース出力されません。
- トレースするメソッドが存在しない場合、“traceFile”、または“traceDirectory”に指定したファイルは作成されません。
- 動的にCLASSPATHを決定するプログラムをトレースする場合、制御ファイルに“classPath”の設定が必要です。“classPath”の設定がない場合、標準出力(または、使用方法記載のログファイル)に“java.lang.ClassNotFoundException: Exception while looking for class xxxxx: java.io.IOException: Couldn't find: xxxxx”(xxxxxはクラスファイル名)というエラーメッセージが表示されることがあります。その場合、トレース処理は行いますが、トレース対象の決定は、クラス名、インターフェース名、スーパークラス名のみで判断し、メソッド名、シグネチャの指定は無効となります。
- サブプレットのメソッドトレースで、制御ファイルに“classPath”の設定がない場合、サブプレットがハングアップしたり、異常終了することがあります。これは、サブプレットのログファイルへの出力とメソッドトレースのエラーメッセージの出力が競合を起こし、デッドロックを起こすためです。この場合、“classPath”を制御ファイルに指定してください。指定されている場合、正しく設定されているか確認をしてください。
- メソッドトレース機能を利用した場合、標準出力(または、使用方法記載のログファイル)に“java.lang.ClassFormatError: LVTT entry for 'yyyy' in class file a does not match any LVT entry”(yyyyはプログラム中の変数名)というエラーメッセージが表示されることがあります。その場合、トレース対象のJava VMオプションに-noverifyオプションまたは、-Xverify:noneオプションを指定してください。ただし、この場合、クラスファイルの検証がされなくなりますので、このオプションの指定はメソッドトレース利用時のみに限定し、アプリケーションの実運用時には指定しないでください。

## A.2 jheap

Windows32/64 Solaris32/64 Linux32/64

## ■ jheapとは

jheapは、実行中のJavaプログラムのヒープ使用状況を出力するツールです。

## ■ 動作環境

サポート対象のJDK/JRE

製品添付のJDK/JRE

サポート対象のガーベジコレクション(GC)

FJVMがサポートする次のGC処理が動作対象のGC処理です。

- ・ シリアルGC
- ・ パラレルGC
- ・ CMS付きパラレルGC

上記以外のGC処理に対しては、jheapは動作しません。

FJVMがサポートするGC処理の詳細は、“チューニングガイド”の“JDK/JREのチューニング”の“FJVMでサポートされるガーベジコレクション処理”を参照してください。

## ■ 格納先

jheapは、コマンドプロンプトから起動するコマンドで、モジュール名はjheapです。格納先は次のとおりです。

### Windows32/64

以下、製品を“C:\Interstage”にインストールした場合を想定しています。

- ・ C:\Interstage\jdk6\tools\jheap (JDK 6)
- ・ C:\Interstage\jre6\tools\jheap (JRE 6)
- ・ C:\Interstage\jdk7\tools\jheap (JDK 7)
- ・ C:\Interstage\jre7\tools\jheap (JRE 7)

### Solaris32/64 Linux32/64

- ・ /opt/FJSVawjbc/jdk6/tools/jheap (JDK 6)
- ・ /opt/FJSVawjbc/jre6/tools/jheap (JRE 6)
- ・ /opt/FJSVawjbc/jdk7/tools/jheap (JDK 7)
- ・ /opt/FJSVawjbc/jre7/tools/jheap (JRE 7)

## ■ 形式

jheap [-i interval] [-c count] [-t] pid

## ■ 機能説明

pidで指定した動作中のJavaプログラムのヒープ使用状況を一定時間の間隔で標準出力に出力します。

pidには、プロセスIDを指定します。

Windows(R)の場合は、プロセスIDは“タスクマネージャ”で確認することができます。

Java EE機能の場合、IJSERVERクラスタのプロセスIDの確認方法は、“Java EE運用ガイド”を参照してください。



J2EE機能の場合、ワークユニットのプロセスIDは、Interstage管理コンソールのIJSERVERモニタ画面で参照することができます。

## ■ オプション

-i interval	測定の間隔を秒単位で指定します。省略可能です。 省略した場合は、5秒間隔で測定します。
-c count	出力する回数を指定します。省略可能です。 省略した場合は、指定したプロセスが終了するまで出力し続けます。
-t	タイムスタンプを出力する場合に指定します。省略可能です。
pid	プロセスIDを指定します。

プロセスID以外の、-i, -c, -t オプションの指定の順序は任意です。

また、-i, -c オプションを複数回指定した場合は、最後に指定したオプションが有効となります。

“-i”および“-c”オプションには、正の整数(1~2147483647)を指定することができます。

そのとき、オプションに指定された文字列を先頭から解析し、数字以外の文字が指定された時点で解析を終了します。たとえば、“12.345”は“12”と解釈します。

## ■使用方法

次のステップで、jheapを起動してください。

1. jheapの格納先ディレクトリを環境変数PATHに追加するか、環境変数PATHに追加されているディレクトリにjheapを格納してください。  
または、jheap.exeをフルパスで指定して実行してください。
2. コマンドプロンプトからjheapコマンドを起動します。

コマンド起動時の例：

```
-----
% jheap 123
<new> [used: 847728] [free: 1511568] [total: 2359296]
<old> [used: 33699152] [free: 2820784] [total: 36519936]
<perm> [used: 6460336] [free: 93264] [total: 6553600]
<new> [used: 847728] [free: 1511568] [total: 2359296]
<old> [used: 33699152] [free: 2820784] [total: 36519936]
<perm> [used: 6460336] [free: 93264] [total: 6553600]
<new> [used: 847728] [free: 1511568] [total: 2359296]
<old> [used: 33699152] [free: 2820784] [total: 36519936]
<perm> [used: 6460336] [free: 93264] [total: 6553600]
. . .
-----
```

- ・ 測定対象のJavaプログラムが終了すると、jheapも終了します。
- ・ jheapの実行を途中で終了するには、jheapを実行したターミナルで[Ctrl]+[C]を押下します。

## ■解説

Javaアプリケーションのメモリ使用状況には以下があります。

- ・ プロセス全体のメモリ使用状況
- ・ Javaのオブジェクトなどが格納されるJavaヒープの使用状況

プロセス全体のメモリ使用状況は、OSの機能(Windows(R)ではタスクマネージャやパフォーマンスモニタ、Solarisではpmap/prstatなど)で把握することができます。

一方、Javaヒープの使用状況は、以下で確認することができますが、プログラム起動時に指定するか、あらかじめソースコードに仕込む必要があります。

- ・ javaコマンドのオプション-verbose:gc
- ・ java.lang.Runtime.totalMemory() / java.lang.Runtime.freeMemory()

jheapは、JavaプログラムのJavaヒープの使用状況を、プログラムの起動後に調査することができます。

Javaヒープは、通常のオプションでは、New世代領域、Old世代領域、Permanent世代領域の3つに大別されます。jheapでは、以下のように、それぞれの領域の使用状況を検出することができます。

---

```
<new> [used: 847728] [free: 1511568] [total: 2359296]
<old> [used: 33699152] [free: 2820784] [total: 36519936]
<perm> [used: 6460336] [free: 93264] [total: 6553600]
```

used :現在の使用量 (byte)  
free :現在の空き容量 (byte)  
total :現在の領域長 (byte)

---

これら3つの領域は、newなどによるメモリ確保、ガーベジコレクションなどにより、増減します。totalは、現在メモリがマップされている量であり、最大値ではないことに注意が必要です。

new + oldで、-verbose:gcで表示される情報に相当します。

New世代領域、Old世代領域ともに、Javaクラスのインスタンスが格納される領域です。

New世代領域には、newされてからの時間が比較的短いインスタンスが格納されます。

Old世代領域には、newされてからの時間が比較的長いインスタンスがガーベジコレクションにより、New世代領域から移動されます。

Permanent世代領域は、標準のJavaの機能で測定することはできませんが、Javaのクラスや定数などの不変な情報が格納される領域です。Javaプログラムで使用するクラスの種類が多かったり、クラスのサイズが大きかったりする場合に不足する場合があります。

記録された値を元に、ヒープ域の使用状況と実際の処理の経過時間とを比較し、繁忙時のヒープの使用状況を確認できます。もし結果がヒープ域の最大値に近接した値であれば、ヒープ域の最大値を拡張し、余裕を持たせる必要があります。

## ■注意事項

### Windows32/64

ServletやEJBなどのサーバアプリケーションの測定をするためには、Administrator権限でjheapを実行する必要があります。

また、Windows(R) XP/Windows Server(R) 2003 以外の場合、Administratorグループのメンバーにより、管理者権限で実行したコマンドプロンプトからjheapを実行する必要があります。

管理者権限でコマンドプロンプトを実行するには、コマンドプロンプトの実行ファイルをエクスプローラー上で右クリックし、“管理者権限で実行する”を選択します。

### Solaris32/64 Linux32/64

測定対象のプロセスに対して、プロセスの読み込み権が必要です。

ServletやEJBなどのサーバアプリケーションの測定をするためには、通常rootでjheapを実行する必要があります。

## A.3 スレッドダンプツール

---

### Windows32/64

### ■スレッドダンプツールとは

スレッドダンプツールは、Java VMのスレッドダンプ(コマンドプロンプトから[Ctrl]キーを押しながら[Break]キーを押して採取したスレッドダンプ)と同じく、各スレッドの状態を出力し、プログラム全体や、各スレッドの実行状況を解析する助けとなるスレッドダンプを出力する機能です。

スレッドダンプツールは、Java VMのスレッドダンプ出力に以下の機能を追加しています。

- ・ ファイル出力  
オプション(-f)で指定したファイルに、スレッドダンプを出力できます。
- ・ ヘッダー、フッター出力  
標準出力や標準エラー出力にスレッドダンプを出力した場合、アプリケーションのログファイル(ServletやEJBの標準出力、あるいはJava VMの出力を格納するファイルなど)に、そのスレッドダンプが含まれることとなります。ヘッダーやフッターは、そのログファイルを参照する場合に、目印となる情報です。(-fオプションを指定した場合も出力します。)ヘッダーには、スレッドダンプの採取日時、コマンドラインおよび、メモリの使用状況の情報が含まれます。

なお、Java VMのスレッドダンプ出力の操作方法は、“チューニングガイド”の[JDK/JREのチューニング]>[チューニング/デバッグ技法]>[スレッドダンプ]を参照してください。

## ■動作環境

### サポート対象のJDK/JRE

製品添付のJDK/JRE

## ■2種類のコマンド

スレッドダンプツールには、以下の2種類のコマンドがあります。

- 通常のアプリケーション(ユーザが起動するプログラム)として動作するthdumpコマンド
- Windowsサービス(システム内に常駐して動作するプログラム)として動作するthdumpSVCコマンド

これら2種類のコマンドは、以下にしたがって使い分けてください。

- Javaプログラムが、通常のアプリケーションとして動作している場合は、thdumpコマンドを使用してください。
- Javaプログラムが、Windowsサービスとして動作している場合は、thdumpSVCコマンドを使用してください。

例えば、以下はWindowsサービスとして動作するので、スレッドダンプを採取するためにはthdumpSVCコマンドを使用します。

- Java EE実行環境
  - IJServerクラスタ
  - Interstage Java EE DASサービス
  - Interstage Java EE Node Agentサービス
  - Interstage JMX サービス
- J2EE実行環境
  - IJServerワークユニット

なお、各コマンドには、機能的な相違点はありません。

## ■格納先

格納先は次のとおりです(製品を“C:¥Interstage”にインストールした場合)。

- thdumpコマンド:
  - C:¥Interstage¥jdk6¥tools¥thdump (JDK 6)
  - C:¥Interstage¥jre6¥tools¥thdump (JRE 6)
  - C:¥Interstage¥jdk7¥tools¥thdump (JDK 7)
  - C:¥Interstage¥jre7¥tools¥thdump (JRE 7)
- thdumpSVCコマンド:
  - C:¥Interstage¥jdk6¥tools¥thdump¥thdumpSVC (JDK 6)
  - C:¥Interstage¥jre6¥tools¥thdump¥thdumpSVC (JRE 6)
  - C:¥Interstage¥jdk7¥tools¥thdump¥thdumpSVC (JDK 7)
  - C:¥Interstage¥jre7¥tools¥thdump¥thdumpSVC (JRE 7)

### A.3.1 thdumpコマンドの使用方法

---

thdump.exeの格納先ディレクトリを環境変数PATHに設定したうえで実行するか、thdump.exeをフルパスで指定して実行してください。

## ■形式

```
thdump.exe [-f logfile | -j] -p processid  
または  
thdump.exe [-f logfile | -j] program name with ".exe"
```

※各オプションの詳細は、“[A.3.3 オプション](#)”を参照してください。

## ■使用方法

- "java.exe"で起動したアプリケーションのスレッドダンプをアプリケーションの標準出力に出力する場合:

```
C:¥> thdump java.exe
```

- プロセスIDが123番のプロセスのスレッドダンプを"a.log"に出力する場合:

```
C:¥> thdump -f a.log -p 123
```

- Internet Explorer + JBKプラグインで動作しているアプレットのスレッドダンプを、JBKプラグインのエラーログ(jbktrace.0)に出力する場合:

```
jbkplugin.propertiesファイル
```

```
-----  
jbk.plugin.debug.tracedir=C:¥temp  
jbk.plugin.debug.showvmmsg=true  
-----
```

```
C:¥> thdump -j iexplore.exe
```

## A.3.2 thdumpSVCコマンドの使用法

---

thdumpSVCコマンドは、Windowsサービスとして動作するJavaプログラムのスレッドダンプを採取する場合に使用します。スレッドダンプの採取は、次の手順で操作します。

1. Windowsサービス登録
2. スレッドダンプツール開始
3. スレッドダンプツール停止
4. Windowsサービス削除

2と3の操作を繰り返すことにより、複数のスレッドダンプを採取することができます。スレッドダンプの採取が完了したら、4の“Windowsサービス削除”を行ってください。

- Windowsサービス登録/削除にはWindows(R)の“InstallUtil.exe”を内部的に使用します。
- スレッドダンプツールの開始および終了には、“sc.exe”を使用します。
- スレッドダンプツールの操作は、管理者権限で実行したコマンドプロンプトから操作する必要があります。管理者権限でコマンドプロンプトを実行するには、コマンドプロンプトの実行ファイルをエクスプローラー上で右クリックし、“管理者権限で実行する”を選択します。
- thdumpSVCコマンドを使用するためには、Microsoft(R) .NET Framework V2.0 を動作環境にインストールしてください。

## ■Windowsサービスへの登録方法

### 【コマンド形式】

```
thdumpSVC -Install
```

スレッドダンプツールをWindowsサービスとして登録するには、thdumpSVCコマンドの格納先をカレントディレクトリにするか、thdumpSVCコマンドをフルパスで指定して、上記コマンドを実行してください。

なお同じシステムで登録可能なスレッドダンプツールは1つだけです。2つ目以降を登録しようとした場合、コマンド実行時エラーとなります。

### 【確認方法】

Windowsサービスに登録できたかどうかは、Windows(R)の管理ツール“サービス”で確認できます。名前項目に“Thdump Service”が表示されていれば、登録に成功しています。

コマンド実行後に、画面にWindows(R)の制御によるメッセージが表示されます。正常に登録できた場合は、メッセージの最後に次の文字列が表示されます。

```
コミット段階が正常に終了しました。  
トランザクション インストールが完了しました。
```

異常が発生した場合は、メッセージの最後に次の文字列が表示されます。

```
トランザクション インストールが完了しました。  
インストールが失敗し、ロールバックが実行されました。
```

すでにスレッドダンプツールの登録が完了している場合には異常が発生しますので、Windows(R)の管理ツール“サービス”でスレッドダンプツールの有無を確認してください。

## ■スレッドダンプツールの開始方法

### 【コマンド形式】

```
sc start thdumpService [-f logfile | -j] -p processid  
または  
sc start thdumpService [-f logfile | -j] program name with ".exe"
```

※各オプションの詳細は、“[A.3.3 オプション](#)”を参照してください。

スレッドダンプツールを開始するには、上記コマンドを実行してください。コマンド実行後に、スレッドダンプ採取が行われます。

なお同じシステムで開始可能なスレッドダンプツールは1つだけです。すでにスレッドダンプツールが動作している場合に、上記コマンドを実行すると、コマンド実行時エラーとなります。

### 【確認方法】

スレッドダンプツールを開始できたかどうかは、Windows(R)の管理ツール“サービス”で確認できます。状態項目に“開始”が表示されていれば、開始に成功しています。

コマンド実行後に、画面にscコマンドの制御によるメッセージが表示されます。

異常が発生した場合には、次のどれかの文字列が表示されます。なお、異常時のメッセージ以外は、無視しても問題ありません。

- “[SC] StartService FAILED nnnn:”(nnnnは数字)
- “[SC] StartService: OpenService FAILED nnnn:”(nnnnは数字)

スレッドダンプツールが未登録の場合か、すでにスレッドダンプツールが開始している場合には異常が発生しますので、Windows(R)の管理ツール“サービス”でスレッドダンプツールの状況を確認してください。

異常時のメッセージの出力例を、次に示します。

- スレッドダンプツールが未登録の場合:  
“[SC] StartService: OpenService FAILED 1060:”
- すでにスレッドダンプツールが開始している場合:  
“[SC] StartService FAILED 1056:”

## ■スレッドダンプツールの停止方法

### 【コマンド形式】

```
sc stop thdumpService
```

スレッドダンプツールを停止するには、上記コマンドを実行してください。

### 【確認方法】

スレッドダンプツールを停止できたかどうかは、Windows(R)の管理ツール“サービス”で確認できます。状態項目が空白であれば、停止に成功しています。

コマンド実行後に、画面にscコマンドの制御によるメッセージが表示されます。

異常が発生した場合には、次のどれかの文字列が表示されます。なお、異常時のメッセージ以外は、無視しても問題ありません。

- “[SC] ControlService FAILED nnnn:”(nnnnは数字)
- “[SC] OpenService FAILED nnnn:”(nnnnは数字)

スレッドダンプツールが未登録の場合か、すでにスレッドダンプツールが停止している場合には異常が発生しますので、Windows(R)の管理ツール“サービス”でスレッドダンプツールの状況を確認してください。

異常時のメッセージの出力例を、次に示します。

- スレッドダンプツールが未登録の場合:  
“[SC] OpenService FAILED 1060:”
- すでにスレッドダンプツールが停止している場合:  
“[SC] ControlService FAILED 1062:”

## ■ Windowsサービスからの削除方法

### 【コマンド形式】

```
thdumpSVC -Install -u
```

スレッドダンプツールをWindowsサービスから削除するには、thdumpSVCコマンドの格納先をカレントディレクトリにするか、thdumpSVCコマンドをフルパスで指定して、上記コマンドを実行してください。

### 【確認方法】

Windowsサービスから削除できたかどうかは、Windows(R)の管理ツール“サービス”で確認できます。名前項目に“Thdump Service”が表示されていない場合は、削除に成功しています。

コマンド実行後に、画面にWindowsの制御によるメッセージが表示されます。

正常に削除できた場合は、メッセージの最後に次の文字列が表示されます。

サービス 'thdumpService' は正常にシステムから削除されました。

アンインストールが完了しました。

異常が発生した場合は、メッセージの最後に次の文字列が表示されます。

アンインストール中に例外が発生しました。この例外は無視され、アンインストール続行します。ただし、アンインストール完了後、アプリケーションは完全にアンインストールされない可能性があります。

すでにスレッドダンプツールが削除されている(未登録の場合)には異常が発生しますので、Windows(R)の管理ツール“サービス”でスレッドダンプツールの有無を確認してください。

## ■ エラーログファイル

Windowsサービスにはコンソールがありません。スレッドダンプ採取時にエラーが発生したかどうかを確認するには、エラーログファイルを参照してください。

エラーログファイルの格納先は、thdumpSVCコマンドの格納先ディレクトリです。thdumpSVCコマンドの格納先ディレクトリは書込み可能な状態にしてください。

エラーログファイルは、Windowsサービス登録時に作成され、Windowsサービス削除までの間に発生したエラーを記録します。問題なくスレッドダンプを取得できた場合は、エラーログファイルのサイズは0バイトになります。

## A.3.3 オプション

スレッドダンプツールの起動時(開始時)に指定するオプションを、下表に示します。

-f logfile	スレッドダンプの出力先を指定します。 省略した場合、標準出力または標準エラー出力に出力されます。 -jオプションと同時に指定できません。
-j	ヘッダーおよびフッターをJava VMの出力ルーチンを用いて出力する場合に指定します。 JNI(Java Native Interface)のJNI_CreateJavaVM()関数に、vfprintf_hookオプションを指定して起動したJava VMに対して指定します。JBKプラグインのjbkplugin.propertiesファイルに、"jbc.plugin.debug.showvmmsg=true"と指定した場合は、このオプションを使います。 -fオプションと同時に指定できません。
-p processid	対象とするプロセスをプロセスIDで指定します。 下の「program name with ".exe"」オプションと同時に指定できません。
program name with ".exe"	対象とするプロセスをjava.exeなど".exe"で終わるコマンド名で指定します。 コマンド名が指定されると、スレッドダンプツールは、現在システムで動作しているすべてのプロセスのうち、指定されたコマンド名により起動されたプロセスから1つを選択して処理します。同じコマンドから複数のプロセスを起動している場合は、-pオプションを使用して、プロセスIDを指定してください。 -pオプションと同時に指定できません。

「program name with ".exe"」以外の、-f、-j、-pオプションの指定の順序は任意です。  
また、-f、-pのオプションを複数回指定した場合は、最後に指定したものが有効となります。  
コマンドを実行すると、ヘッダー、スタックトレース、フッターの順に出力されます。

対象プロセスは、プロセスID、または".exe"で終わるプログラム名(java.exeなど)で指定します。プロセスIDはWindows(R)の「タスクマネージャ」で確認することができます。

Java EE機能の場合、IISServerクラスタのプロセスIDの確認方法は、「Java EE運用ガイド」を参照してください。



J2EE機能の場合、ワークユニットのプロセスIDは、Interstage管理コンソールのIIServerモニタ画面で参照することができます。

### A.3.4 サンプルプログラムと出力例

以下にデッドロックのサンプルプログラムと、スレッドダンプの出力例を示します。

スレッドダンプの解析方法は、「チューニングガイド」の「JDK/JREのチューニング」の「スレッドダンプ」を参照してください。

#### ■ サンプルプログラム

```
public class Deadlock implements Runnable {
    volatile boolean finished = false;
    public void run() {
        try {
            synchronized (this) {
                while (!finished) {
                    System.out.print(".");
                    Thread.sleep(500);
                }
            }
        }
        catch (InterruptedException ex) {
            ex.printStackTrace(System.out);
            Thread.currentThread().interrupt();
        }
    }
    public static void main(String args[]) {
        Object obj = new Object();
        Deadlock test = new Deadlock();
        Thread th = new Thread(test, "Child");
        try {
            th.start();
        }
    }
}
```

```

    Thread.sleep(5000);
    synchronized (test) {
        test.finished = true;
    }
}
catch (Exception ex) {
    ex.printStackTrace(System.out);
}
}
}
}

```

---

このサンプルプログラムは、run()メソッドで、DeadLockクラスのインスタンスに対して、synchronized句によりロックを取得しています。run()メソッドは、finishedフラグがfalseの間ループを繰り返します。一方、main()メソッドは、run()メソッド呼出し後に、DeadLockクラスのインスタンスに対してロックを獲得できたならば、finishedフラグをtrueに設定します。

この例の場合、run()メソッドが、DeadLockクラスのインスタンスに対してロックを獲得していますので、main()メソッドはロックを獲得できずに、ハングアップします。

## ■スレッドダンプの出力例

JDK6におけるスレッドダンプの出力例を次に示します。

---

```

*****
Java Thread dump Tool for WindowsNT/2000/XP
Copyright (c) 2002 FUJITSU LIMITED. All rights reserved.

start at Tue Jan 11 10:38:25 2011

cmd = C:\Interstage\JDK6\bin\java.exe Deadlock

Memory usage (Free/Total) = 7470040/7864320 (bytes)

(ProcessId = 0x754, Tool ThreadId = 0xe98)
*****

Full thread dump Java HotSpot(TM) Server VM (17.1.03_FUJITSU_MODIFIED-B11 mixed mode):
[pid=1876] Time=Tue Jan 11 10:38:25 2011

"Java Thread dump tool" prio=10 tid=0x03c0f800 nid=0xe98 waiting on condition [0x00000000]
  java.lang.Thread.State: RUNNABLE

"Child" prio=6 tid=0x03c0e000 nid=0xbec waiting on condition [0x045cf000]
  java.lang.Thread.State: TIMED_WAITING (sleeping)
    at java.lang.Thread.sleep(Native Method)
    at Deadlock.run(Deadlock.java:8)
    - locked <0x0fd98740> (a Deadlock)
    at java.lang.Thread.run(Thread.java:619)

"Low Memory Detector" daemon prio=6 tid=0x03bf9000 nid=0x918 runnable [0x00000000]
  java.lang.Thread.State: RUNNABLE

"CompilerThread1" daemon prio=10 tid=0x03c08800 nid=0x59c waiting on condition [0x00000000]
  java.lang.Thread.State: RUNNABLE

"CompilerThread0" daemon prio=10 tid=0x03c03800 nid=0xb5c waiting on condition [0x00000000]
  java.lang.Thread.State: RUNNABLE

"Attach Listener" daemon prio=10 tid=0x03e49800 nid=0x2b8 runnable [0x00000000]
  java.lang.Thread.State: RUNNABLE

"Signal Dispatcher" daemon prio=10 tid=0x03e5a800 nid=0x2fc runnable [0x00000000]

```

```

java.lang.Thread.State: RUNNABLE

"Finalizer" daemon prio=8 tid=0x03bea800 nid=0x170 in Object.wait() [0x03e3f000]
  java.lang.Thread.State: WAITING (on object monitor)
    at java.lang.Object.wait(Native Method)
    - waiting on <0x0fd60b18> (a java.lang.ref.ReferenceQueue$Lock)
    at java.lang.ref.ReferenceQueue.remove(ReferenceQueue.java:118)
    - locked <0x0fd60b18> (a java.lang.ref.ReferenceQueue$Lock)
    at java.lang.ref.ReferenceQueue.remove(ReferenceQueue.java:134)
    at java.lang.ref.Finalizer$FinalizerThread.run(Finalizer.java:159)

"Reference Handler" daemon prio=10 tid=0x03be5c00 nid=0xb04 in Object.wait() [0x03def000]
  java.lang.Thread.State: WAITING (on object monitor)
    at java.lang.Object.wait(Native Method)
    - waiting on <0x0fd60a20> (a java.lang.ref.Reference$Lock)
    at java.lang.Object.wait(Object.java:485)
    at java.lang.ref.Reference$ReferenceHandler.run(Reference.java:116)
    - locked <0x0fd60a20> (a java.lang.ref.Reference$Lock)

"main" prio=6 tid=0x00317000 nid=0xe80 waiting for monitor entry [0x0093f000]
  java.lang.Thread.State: BLOCKED (on object monitor)
    at Deadlock.main(Deadlock.java:25)
    - waiting to lock <0x0fd98740> (a Deadlock)

"VM Thread" prio=10 tid=0x03be2c00 nid=0xd34 runnable

"GC task thread#0 (ParallelGC)" prio=6 tid=0x0031e800 nid=0xc84 runnable

"GC task thread#1 (ParallelGC)" prio=6 tid=0x03b10400 nid=0xaa4 runnable

"GC task thread#2 (ParallelGC)" prio=6 tid=0x03b11400 nid=0xe7c runnable

"GC task thread#3 (ParallelGC)" prio=6 tid=0x03b12800 nid=0x480 runnable

"VM Periodic Task Thread" prio=10 tid=0x03bfac00 nid=0xe88 waiting on condition

"RAS Control Thread" prio=10 tid=0x03be2000 nid=0xb4c runnable
JNI global references: 854

```

```

Heap
PSYoungGen      total 3584K, used 385K [0x0fd60000, 0x10160000, 0x10470000)
  eden space 3072K, 12% used [0x0fd60000, 0x0fdc0428, 0x10060000)
  from space 512K, 0% used [0x100e0000, 0x100e0000, 0x10160000)
  to   space 512K, 0% used [0x10060000, 0x10060000, 0x100e0000)
PSOldGen        total 4096K, used 0K [0x0c470000, 0x0c870000, 0x0fd60000)
  object space 4096K, 0% used [0x0c470000, 0x0c470000, 0x0c870000)
PSPermGen       total 16384K, used 2012K [0x08470000, 0x09470000, 0x0c470000)
  object space 16384K, 12% used [0x08470000, 0x08667288, 0x09470000)

```

```

*****
Java Thread dump end
*****
-----

```

### A.3.5 注意事項

- Javaプログラム以外に対してスレッドダンプツールを使用しても、スレッドダンプやメッセージなどは何も出力されません。
- アプリケーションの不良(JNIによる領域破壊など)で、Java VM内の管理領域が破壊されていたり、または、不整合があったりする場合、スレッドダンプ採取対象のプロセスでアプリケーションエラーが発生する場合があります。

- `-f`オプションを使用してファイルにスレッドダンプを出力する際、スレッドダンプ採取処理の間に、アプリケーションが標準出力にログ等を出力した場合、それらが、`-f`オプションで指定したファイルに混入することがあります。アプリケーション側で、すでに標準出力、標準エラー出力をリダイレクトして、ログファイル等を作成している場合、アプリケーションの出力が、これらログファイルには出力されません。
- JNIのJNI\_CreateJavaVMを使って起動したアプリケーションで、Java VMのオプション`vfprintf_hook`を指定した場合は、`-j`オプションを使用してください。この場合、`-f`オプションは使用できません。`vfprintf_hook`で指定したフック関数の処理によりスレッドダンプが出力されます。
- Java VMで、VM Thread(Java VM内の管理スレッド)がハングアップしている場合は、スレッドダンプが出力できません。
- 動作環境がWindows Server(R) 2003、またはWindows(R) XP以前である場合、かつ物理コンソールからログオンしている場合は、`thdump`コマンド・`thdumpSVC`コマンドのどちらも使用することができます。
- スレッドダンプを取得しようとしているJavaプログラムに対し、スレッドダンプツールの実行をするユーザは、同等以上の権限をもっている必要があります。(Administratorが実行しているJavaプログラムに対し、一般ユーザがスレッドダンプツールを実行してもスレッドダンプの採取はできません)
- スレッドダンプを取得する場合、Administrator権限が必要です。  
また、Windows(R) XP/Windows Server(R) 2003 以外の場合、Administratorグループのメンバーにより、管理者権限で実行したコマンドプロンプトからスレッドダンプツールの操作をする必要があります。管理者権限でコマンドプロンプトを実行するには、コマンドプロンプトの実行ファイルをエクスプローラー上で右クリックし、“管理者権限で実行する”を選択します。
- サービスプログラムのスレッドダンプを取得する場合、Administrator権限が必要です。

## A.4 JDKに含まれるトラブルシューティングに役立つツール

---

JDKに含まれる以下のツールをトラブルシューティングツールとして使用可能です。

- `jconsole`
- `jps`
- `jstat`
- `jstatd`
- `jinfo`
- `jhat`
- `jmap`
- `jsadefbugd`
- `jstack`
- `jvisualvm`
- `jcmd`(JDK7だけ)

JDKツールの詳細については、JDKドキュメントに記載されています。

「チューニングガイド」-「JDK/JREのチューニング」-「基礎知識」-「JDK関連のドキュメント」を参照してください。

### A.4.1 注意事項

---

#### A.4.1.1 jinfo使用時の注意事項

- `jinfo`で`jsadefbugd`(リモートデバッグサーバー)に接続する場合、以下のオプションは使用できません。
  - `-flag` オプション

- `jinfo`で`core`ファイルを対象とする場合、以下のオプションは使用できません。
  - `-flag` オプション
- `jinfo`で表示される以下の情報は、サポート対象外です。
  - “オプションなし”で実行した場合に出力される“VM Flags”
  - “`-flags`”オプションで実行した場合に出力される“VM Flags”

#### A.4.1.2 `jmap`使用時の注意事項

- `jmap`で`jsadebugd`(リモートデバッグサーバー)に接続する場合、以下のオプションは使用できません。
  - オプションなし
  - `-histo:live` オプション
  - `-dump:live` オプション
- `jmap`で`core`ファイルを対象とする場合、以下のオプションは使用できません。
  - `-histo:live` オプション
  - `-dump:live` オプション
- JDK6とJDK7で、インターン(`intern`)された文字列情報を`jmap`で表示する場合のオプションが異なります。
  - JDK6: `-permstat` オプション
  - JDK7: `-heap` オプション



#### 参照

.....

インターンされた文字列に関しては、以下のOracleドキュメントを参照してください。

- JDK6の場合
    - <http://docs.oracle.com/javase/jp/6/api/java/lang/String.html>
    - メソッドの詳細 → `intern`
  - JDK7の場合
    - <http://docs.oracle.com/javase/7/docs/api/java/lang/String.html>
    - Method Detail → `intern`
- .....

#### A.4.1.3 `jstack`使用時の注意事項

**Windows32/64 Linux32/64**

- Windows/Linuxにおいて、`jstack`を使用する場合、以下のオプションは使用できません。
  - `-m` オプション

**Solaris32/64**

- Solarisにおいて、`jstack`で`jsadebugd`(リモートデバッグサーバー)に接続する場合、以下のオプションは使用できません。
  - `-m` オプション

#### A.4.1.4 `jstat`使用時の注意事項

- `jstat`を使用する場合、以下のオプションは使用できません。
  - `-version` オプション

- JavaプログラムがWindowsサービスとして動作している場合、使用できません。

#### A.4.1.5 jvisualvm使用時の注意事項

- jvisualvm を使用する場合、画面に全角文字が正しく表示されない場合があります。

その場合は、fontconfigによる以下のパターンマッチング結果で、一番最初に日本語フォントがリストされるように環境設定してください。

```
sans:weight=regular:slant=roman:lang=ja-JP
```



例

##### fc-matchコマンドによるパターンマッチング結果の確認方法

fc-matchコマンドで、パターンとのマッチング結果を検索し、日本語フォントが出力されることを確認します。fc-matchコマンドの詳細については、manコマンドでfc-match(1)を参照してください。

```
$fc-match sans:weight=regular:slant=roman:lang=ja-JP
```



例

##### 「.fonts.conf」ファイルによるfontconfigのパターンマッチング設定方法

日本語フォントがパターンマッチングの先頭になるように「.fonts.conf」ファイルを作成し、jvisualvmを起動するユーザのホームディレクトリに配置します。

なお、「.fonts.conf」ファイルは、jvisualvmだけでなく、fontconfigを使用するその他のアプリケーションにも影響しますので、注意してください。

.fonts.confの詳細については、manコマンドでfonts-conf(5)を参照してください。

##### 「.fonts.conf」ファイルの例

```
<?xml version="1.0"?>
<!DOCTYPE fontconfig SYSTEM "fonts.dtd">

<!-- ~/.fonts.conf for per-user font configuration -->

<fontconfig>
<match target="pattern">

  <!-- Condition -->
  <test name="family" compare="contains">
    <string>sans</string>
  </test>
  <test name="weight" compare="contains">
    <const>regular</const>
  </test>
  <test name="slant" compare="contains">
    <const>roman</const>
  </test>
  <test name="lang" compare="contains">
    <string>ja-JP</string>
  </test>

  <!-- Insert "ja" into list of "lang". -->
  <edit name="lang" mode="prepend_first">
    <string>ja</string>
```

```

</edit>

<!-- Insert font name explicitly into list of "family". -->
<edit name="family" mode="prepend_first">
  <string>VL PGothic</string>
</edit>

</match>

<match target="pattern">

  <!-- Condition -->
  <test name="family" compare="contains">
    <string>sans</string>
  </test>
  <test name="weight" compare="contains">
    <const>regular</const>
  </test>
  <test name="slant" compare="contains">
    <const>roman</const>
  </test>
  <test name="lang" compare="contains">
    <string>ja</string>
  </test>

  <!-- Insert "ja" into list of "lang". -->
  <edit name="lang" mode="prepend_first">
    <string>ja</string>
  </edit>

  <!-- Insert font name explicitly into list of "family". -->
  <edit name="family" mode="prepend_first">
    <string>VL PGothic</string>
  </edit>

</match>

</fontconfig>

```

**EE SJE**

- 以下の条件の場合、ローカルで動作するJavaプログラムにjvisualvmで接続しようとする、jvisualvmのウィンドウ右下に警告マークが表示され、Javaプログラムに接続できません。なお、警告マークをクリックし、警告の詳細を表示すると、“java.lang.ClassNotFoundException”が表示されます。

[条件]

- JDK6を使用している。
- J2EE互換機能を使用するために、以下のファイルを作成している。

**Windows32/64**

- [Interstageインストールフォルダ]¥jdk6¥jre¥lib¥orb.properties

**Solaris32/64 Linux32/64**

以下の“\$DIR”はインストール時に指定する相対ディレクトリ名です。“\$DIR”のシステム推奨名は“opt”です。

- /\$DIR/FJSVawjkb/jdk6/jre/lib/orb.properties

[対処]

ローカルで動作するJavaプログラムに対して、JMX接続を使用して接続してください。



## 例

### 対象JavaプログラムのJMXエージェントをポート番号50000番で有効にする場合

1. jvisualvm の接続対象のJavaプログラムを、JMXエージェントを有効にして起動します。

```
java -Dcom.sun.management.jmxremote.port=50000 -Dcom.sun.management.jmxremote.authenticate=false
-Dcom.sun.management.jmxremote.ssl=false <アプリケーション名>
```

※実際は改行せず、1行のコマンドとして実行してください。

2. jvisualvm を起動します。
3. [ファイル]メニューから[JMX接続を追加]を選択し、[JMX接続の追加]画面を表示します。
4. [接続]に以下を入力し、[了解]ボタンを押下します。

```
localhost:50000
```

5. アプリケーションウィンドウの[ローカル]に以下のJavaプログラムが表示されることを確認し、ダブルクリックします。

```
localhost:50000
```

- jvisualvm を使用する場合、以下の情報はサポート対象外です。
  - 概要タブの“JVMフラグ”
- 解析対象に巨大なサイズのヒープダンプを指定した場合、メモリ不足となる場合があります。
 

その場合、jvisualvmが使用するメモリ割り当てプールの最大値を調整してください。



## 例

### メモリ割り当てプールの最大値を1024MBに指定して、JDK7 jvisualvmを使用する場合

Windows32/64

```
[JDK7インストール先フォルダ名]¥bin¥jvisualvm.exe -J-Xmx1024m
```

Solaris32/64 Linux32/64

```
[JDK7インストール先フォルダ名]/bin/jvisualvm -J-Xmx1024m
```

- OQLコンソールは使用できません。
- リモートマシン上で動作するJavaプログラムに対して、メモリーサンブラは使用できません。

Windows32/64

- セッションの違うJavaプログラム、またはWindowsサービスとして動作するJavaプログラムに対して、以下の機能は使用できません。
  - プロファイラ
  - メモリーサンブラ

上記以外の機能を使用するためには、JMX接続を使用してJavaプログラムに接続してください。
- 外部のネットワークと分離された安全な環境を除いて、JMX通信にはSSLの使用を推奨します。
 

jvisualvmでSSLを有効化する場合、システムプロパティを設定してjvisualvmを起動します。

設定が必要なシステムプロパティや、詳細な設定方法に関しては、JDKドキュメントの「監視および管理」から「JMX APIを使用する監視と管理」を参照してください。

jvisualvmの起動時に、システムプロパティを設定する場合は、-Jオプションを使用します。



例

システムプロパティを指定して、JDK7 jvisualvmを起動する場合

Windows32/64

```
[JDK7インストール先フォルダ名]\bin\jvisualvm.exe -J-D[システムプロパティ]=[値]
```

Solaris32/64 Linux32/64

```
[JDK7インストール先フォルダ名]/bin/jvisualvm -J-D[システムプロパティ]=[値]
```

#### A.4.1.6 jconsole使用時の注意事項

EE SJE

- 以下の条件の場合、ローカルで動作するJavaプログラムにjconsoleで接続しようとする時、“java.lang.ClassNotFoundException”が表示され、ローカルで動作するJavaプログラムにjconsoleで接続できません。

[条件]

- JDK6を使用している。
- J2EE互換機能を使用するために、以下のファイルを作成している。

Windows32/64

```
[Interstageインストールフォルダ]\jdk6\jre\lib\orb.properties
```

Linux32/64 Solaris32/64

以下の“\$DIR”はインストール時に指定する相対ディレクトリ名です。“\$DIR”のシステム推奨名は“opt”です。

```
- /$DIR/FJ$Vawj$bk/jdk6/jre/lib/orb.properties
```

[対処]

ローカルで動作するJavaプログラムに対して、リモート監視を使用して接続してください。



例

対象JavaプロセスのJMXエージェントをポート番号50000番で有効にする場合

- jconsole の接続対象のJavaプログラムを、JMXエージェントを有効にして起動します。

```
java -Dcom.sun.management.jmxremote.port=50000 -Dcom.sun.management.jmxremote.authenticate=false  
-Dcom.sun.management.jmxremote.ssl=false <アプリケーション名>
```

※実際は改行せず、1行のコマンドとして実行してください。

- リモート監視を使用して、jconsole を起動します。

```
jconsole localhost:50000
```

- jconsole を使用する場合、以下の情報はサポート対象外です。
  - VMサマリータブの“VM引数”

#### Windows32/64

- ・ セッションの違うJavaプログラム、またはWindowsサービスとして動作するJavaプログラムに対して、jconsoleを使用する場合、リモート監視を使用してJavaプログラムに接続してください。
- ・ SSLを使用せずにJavaプログラムに対してjconsoleを接続した場合、以下のようなメッセージのダイアログが表示されます。

```
Secure connection failed. Retry insecurely?  
The connection to <接続対象名> could not be made using SSL.  
Would you like to try without SSL?  
(Username and password will be sent in plain text.)
```

#### 接続対象名

接続対象のJavaプログラムのプロセスID、ホスト名とポート番号などです。

ローカル監視を使用して接続する場合、または外部のネットワークと分離された安全な環境で使用する場合、[Insecure connection]ボタンを押下して、Javaプログラムに接続してください。

安全が保証されていない環境で使用する場合、パスワードやSSLの設定を推奨します。設定方法は、JDKドキュメントの「監視および管理」から「JMX APIを使用する監視と管理」を参照してください。

### A.4.1.7 jps使用時の注意事項

- ・ 以下のオプションは使用できません。
  - － -V オプション

#### Windows32/64

- ・ JavaプログラムがWindowsサービスとして動作している場合、使用できません。

### A.4.1.8 jcmd使用時の注意事項

- ・ 以下のオプションは使用できません。
  - － VM.flags オプション
  - － VM.flags -all オプション
  - － VM.command\_line オプション

#### Windows32/64

- ・ JavaプログラムがWindowsサービスとして動作している場合、jcmdのオプションとしてメインクラス名は使用できません。

### A.4.1.9 jhat使用時の注意事項

- ・ 解析対象に巨大なサイズのヒープダンプを指定した場合、OutOfMemoryErrorとなるか、または読み込みに時間がかかる場合があります。

その場合、jhatが使用するメモリ割り当てプールの最大値を調整してください。



例

メモリ割り当てプールの最大値を1024MBに指定して、JDK7 jhatを使用する場合

#### Windows32/64

```
[JDK7インストール先フォルダ名]¥bin¥jhat.exe -J-Xmx1024m <ヒープダンプファイル名>
```

#### Linux32/64 Solaris32/64

```
[JDK7インストール先フォルダ名]/bin/jhat -J-Xmx1024m <ヒープダンプファイル名>
```

## 注意

ヒープダンプのサイズによっては、メモリ割り当てプールの最大値を調整しても、長時間jhatから応答が返ってこない場合があります。

その場合は、jvisualvmを使用して解析してください。

## A.5 Java監視機能

Java監視機能を使用するには、JDKのインストールが必要です。

### ■Java監視機能とは

Java監視機能は、アプリケーションサーバ上で現在動作中のアプリケーション(※1)のJava VM(JVM)に関するパフォーマンスデータを収集、表示する機能です。

Java監視機能の利用者に対し、次の2つの機能を提供します。

1. パフォーマンスデータ収集開始、停止等の操作入力
2. 収集したパフォーマンスデータを視覚化

本機能は、Java SEで提供されるjconsoleと、富士通が拡張したjconsoleプラグインとで構成されます。

※1: Java監視機能が監視できるサーバ上のアプリケーションは、次の条件を満たすものです。

- Java SE 6/7 のJava VMにより動作

### ■収集可能なパフォーマンスデータについて

Java監視機能により収集できるパフォーマンスデータは、次のものです。

データ名	データの概要
メソッドサンプリング	一定の間隔で、メソッドの実行情報(どのメソッドが実行されているか)を示します。 収集の開始と終了、サンプリング間隔、およびスタックトレースの階層の指定は、Java監視機能の画面上より操作することができます。 収集したデータは、直接コンソールに表示します。
オブジェクトヒストグラム	Javaオブジェクトの使用状況を任意の二点間の差分にて示します。 Javaオブジェクトの使用状況とは、現在のヒープにおけるJavaオブジェクトのヒストグラム(どのクラスのインスタンスが、何個存在し、何バイト占めている)を示す情報です。 収集の開始のタイミングは、Java監視機能の画面上より操作することができます。収集したデータは、直接Java監視機能の画面上に表示します。
ヒープダンプ	現在のヒープダンプ(ヒープイメージ)を示します。 収集の開始のタイミングは、Java監視機能の画面上より操作することができます。 ヒープダンプは、Java監視機能の対象とするコンテナが動作しているサーバ上にファイルとして出力されます。 出力されたファイルは、ヒープダンプ表示プログラム(jhat)を使用することにより、ヒープの状況を観察することができます。
GC情報	Java VMのオプションを変更します。 操作可能なオプションは次の通りです。 <ul style="list-style-type: none"><li>• GC表示ポリシー</li><li>• System.gc()の抑止</li><li>• System.gc()の検出</li></ul>

データ名	データの概要
	<ul style="list-style-type: none"> <li>・ タイムスタンプ形式</li> </ul> <p>Java VMのオプションの変更をJava監視機能の画面上より操作することができます。 オプションの操作によりJava VMより得られる情報は、サーバ上の標準出力をロギングしているファイルに出力されます。</p>

## A.5.1 パフォーマンスデータの収集

Java VM(JVM)に関するパフォーマンスデータを収集します。

### ■操作方法

パフォーマンスデータを収集するには、Javaプロセスの起動時にオプションを設定します。

### ◆設定するオプション(単独で動作するJavaアプリケーションのデータを収集する場合)

パフォーマンスデータを収集するには、次のクラスパス、JVMオプションとシステムプロパティを追加してください。

項目		値
追加するクラスパス	Windowsの場合	次のいずれかの値 <ul style="list-style-type: none"> <li>・ [JDK インストール先フォルダ]¥jre¥lib¥fmoni.jar</li> <li>・ [JRE インストール先フォルダ]¥lib¥fmoni.jar</li> </ul>
	Oracle SolarisまたはLinuxの場合	次のいずれかの値 <ul style="list-style-type: none"> <li>・ [JDK インストール先フォルダ]/jre/lib/fmoni.jar</li> <li>・ [JRE インストール先フォルダ]/lib/fmoni.jar</li> </ul>
追加するJVMオプション		-agentlib:fjprof
追加するシステムプロパティ	ローカル環境のアプリケーションのパフォーマンスデータを収集する場合	<ul style="list-style-type: none"> <li>・ com.sun.management.agent.class=com.fujitsu.jvm.management.Agent</li> <li>・ com.sun.management.jmxremote</li> </ul>
	リモート環境のアプリケーションのパフォーマンスデータを収集する場合	<ul style="list-style-type: none"> <li>・ com.sun.management.agent.class=com.fujitsu.jvm.management.Agent</li> <li>・ com.sun.management.jmxremote.port=&lt;ポート番号&gt;</li> <li>・ com.sun.management.jmxremote.authenticate=&lt;trueまたはfalse&gt;</li> <li>・ com.sun.management.jmxremote.ssl=&lt;trueまたはfalse&gt;</li> </ul>

コマンドラインで指定する例を示します(実際は改行せず、1行のコマンドとして実行してください)。

```
java
-classpath C:¥Interstage¥jdk6¥jre¥lib¥fmoni.jar;%CLASSPATH%
-agentlib:fjprof
-Dcom.sun.management.agent.class=com.fujitsu.jvm.management.Agent
-Dcom.sun.management.jmxremote.port=<ポート番号>
-Dcom.sun.management.jmxremote.authenticate=false
-Dcom.sun.management.jmxremote.ssl=false
<アプリケーション名>
```



注意

システムプロパティ`com.sun.management.jmxremote.authenticate`を`true`に設定した場合、コンソール機能から接続する際に、パスワード認証が必要になります。

また、システムプロパティ`com.sun.management.jmxremote.ssl`を`true`に設定した場合、コンソール機能との通信にSSLが使用されます。

外部のネットワークと分離された安全な環境を除いて、コンソール機能との通信にはSSLの使用を推奨します。

これらのシステムプロパティの設定方法などについては、JDKドキュメントの「監視および管理」から「JMX を使用する監視と管理」を参照してください。これらのシステムプロパティの設定内容によっては、更にほかのシステムプロパティの指定が必要になることがあります。

#### ◆設定するオプション(Java EEで動作するプロセスのデータを収集する場合)

パフォーマンスデータの収集を可能とする場合、Java EE運用コマンドの`asadmin`コマンド等を用い、IIServerクラスタに以下のJVMオプションとシステムプロパティを設定してください。

設定	設定する値
JVMオプション	<code>-agentlib:fjprof</code>
システムプロパティ	<code>com.fujitsu.interstage.javaee.management.agent.start=true</code>

IIServerクラスタの起動手順や値の設定方法等の詳細な情報や注意点は、「Java EE運用ガイド」を参照してください。

#### ■Java監視機能で使用するポート番号の調べ方

Java監視機能で監視するJavaプロセスに接続するには、ポート番号を指定する必要があります。



ローカル環境で動作する、単独のJavaアプリケーションに接続する場合は、ポート番号を指定する必要はありません。

#### ◆単独で動作するJavaアプリケーションのポート番号の調べ方

`java`コマンドの起動時に指定した、`com.sun.management.jmxremote.port`システムプロパティの値です。

#### ◆Java EEで動作するプロセスのポート番号の調べ方

Java監視機能で監視するIIServerクラスタのプロセス(サーバーインスタンス)への接続に使うポート番号を調べるには、`asadmin`コマンドの`list-system-properties`サブコマンドを使い調べます。

```
asadmin list-system-properties <サーバインスタンス名>
```

このコマンドを実行すると次のような結果が表示されます。(以下、表示のポート番号は例として示しています)

```
HTTP_LISTENER_PORT=28090
IIOP_SSL_LISTENER_PORT=23820
JMS_PROVIDER_PORT=7676
IIOP_LISTENER_PORT=23700
JMX_SYSTEM_CONNECTOR_PORT=28686
IIOP_SSL_MUTUALAUTH_PORT=23920
Command list-system-properties executed successfully.
```

表示中の**JMX\_SYSTEM\_CONNECTOR\_PORT**が、接続に使用するポート番号です。

なお、ポート番号の調べ方に示した情報の詳細は、「Java EE運用ガイド」、「リファレンスマニュアル(コマンド編)」を参照してください。

## A.5.2 コンソール機能

---

パフォーマンスデータ収集に収集開始、停止等の操作入力や収集したパフォーマンスデータを視覚化する機能です。

### ■操作方法

コンソール機能は、インストールされたJDKに付属しているjconsoleを起動する際、提供するモジュール(fjconsole.jar)をプラグインとして指定することにより、jconsole内に画面が追加され使用できるようになります。

次の順番で操作を行います。

1. jconsoleの起動
2. 監視対象のコンテナへの接続
3. Java監視機能の操作
4. 接続解除
5. jconsoleの終了

### ■注意事項

- Java監視機能は、複数のコンソールから同じJavaプロセスに接続しないでください。  
複数のコンソール間で操作を共用することにより、コンソール間で想定した測定とは異なる監視を行ってしまう場合があります(たとえば、VMオプションの設定でAコンソールからはONにしたのに、BコンソールからはOFFにしてしまうような、操作の矛盾がでてしまうことです)。  
コンソールとコンテナは、一対一で接続してください。
- 複数のパフォーマンスデータを同時に採取する操作をしないでください。  
あるパフォーマンスデータを採取中(たとえば、メソッドサンプリング)、別のパフォーマンスデータの採取(たとえば、オブジェクトヒストグラム)を行う操作はしないでください。  
相互に影響して、正しいパフォーマンスデータの取得ができない場合があります。
- Java監視機能で監視できるアプリケーション。  
Java監視機能で監視できるアプリケーションは、Java SE 6/7 で動作するアプリケーションです。
- Java監視機能を終了しても、それまでに操作した状態は変更されません(終了時に元へ戻しません)。  
そのため、監視対象のアプリケーションを起動するときに設定していたオプションと、Java監視機能を終了させた時のオプションが、異なる状態を示す場合があります。
- Java監視機能は、Interstage Studioと同時に使用することはできません。
- その他の注意事項については“[A.4.1.6 jconsole使用時の注意事項](#)”を参照してください。

## A.5.3 jconsoleの起動

---

jconsoleの起動は、スタートメニューからの起動とコマンドプロンプトからの起動の2つがあります。

### a. スタートメニューからの起動(Windowsのみ)

Java監視機能のインストールが完了するとWindowsのスタートメニュー内にJava監視機能のメニュー項目が追加されます。

Java SE 6のJavaアプリケーションを監視する場合は[コンソール(JDK 6)]を、Java SE 7のJavaアプリケーションを監視する場合は[コンソール(JDK 7)]を選び、Java監視機能(jconsole)を起動してください。

Java監視機能のメニュー項目の表示方法は、オペレーティングシステムによって異なります。



例

### Windows 7 の場合

[スタート]>[すべてのプログラム]>[Interstage Application Server]>[Java監視機能]>[コンソール(JDK 6)]

[スタート]>[すべてのプログラム]>[Interstage Application Server]>[Java監視機能]>[コンソール(JDK 7)]

## Windows Server 2012 の場合

[スタート画面]で右クリック > [すべてのアプリ] > [コンソール(JDK 6)]

[スタート画面]で右クリック > [すべてのアプリ] > [コンソール(JDK 7)]

### b. コマンドプロンプトからの起動

コマンドプロンプト画面から、Java監視機能を起動することができます。

1. コマンドプロンプト画面を開きます。
2. 次のコマンドラインを入力します。

#### Windowsの場合

```
> [JDK6インストール先フォルダ名]\tools\%fjconsole%fjconsole.bat [オプション]
```

#### Oracle SolarisまたはLinuxの場合

```
$ sh [JDK6インストール先フォルダ名]/tools/fjconsole/fjconsole.sh [オプション]
```

Java監視機能の起動に使用するスクリプトにはオプションを指定することもできます。指定したオプションは、jconsoleのオプションとして使用されます。jconsoleで使用可能なオプションには、次のようなものがあります。

```
-interval=n   更新間隔をn秒に設定します（デフォルトは4秒）
-notice       初期状態のウィンドウをタイル状に並べません（接続が複数ある場合）
-J            jconsoleを実行するJava仮想マシンへの入力引数を指定します
```



#### コマンドプロンプト画面から起動する場合

注意

コマンドプロンプト画面から起動する場合、かならず、JDK 6インストール先のフォルダ名から始まるフルパスで指定してください。

## A.5.4 監視対象アプリケーションへの接続

jconsoleを起動できたら、監視対象のアプリケーションに接続します。

接続に必要な情報として、次の情報が必要です。

- 接続先サーバのホスト名（またはIPアドレス）
- アプリケーションに割り当てられたポート番号
- ユーザ名とパスワード



ローカル環境で動作する、単独のJavaアプリケーションに接続する場合

参考

接続先サーバのホスト名とポート番号の入力や、パスワード認証は必要ありません。[jconsole:新規接続]のダイアログに表示される[ローカルプロセス]のなかから、接続したいプロセスを選んで[接続]ボタンを押せば接続できます。



アプリケーションに割り当てられたポート番号の調べ方

注意

“◆Java EEで動作するプロセスのポート番号の調べ方”を参照してください。



単独で動作するJavaアプリケーションへの接続にユーザ名とパスワード

参考

監視対象のJavaプロセスでシステムプロパティ`com.sun.management.jmxremote.authenticate`を`true`に設定した場合、コンソール機能から接続する際に、パスワード認証が必要になります。

詳しくは、JDKドキュメントの「監視および管理」から「JMX を使用する監視と管理」を参照してください。



IJServerクラスタへの接続に必要なユーザ名とパスワード

参考

使用する“ユーザ名”と“パスワード”は、Interstage Java EE DASサービスへの認証で使用するユーザ名とパスワードを使用してください。

Interstage Java EE DASサービスへの認証で使用するユーザ名とパスワードの詳細は、“Java EE運用ガイド”を参照してください。

`jconsole`コマンドを実行すると[Jconsole:新規接続]というダイアログボックスが表示されます。

`jconsole`画面のメニューバーから[接続]>[新規接続]を選んでも同じダイアログボックスを表示できます。

ダイアログボックスに次の情報を入力します。

入力項目名	入力情報
リモートプロセス (R):	接続するリモートプロセスのアドレスです。指定の方法は2つあります。詳細は「リモートプロセスのアドレス指定方法」を参照してください。
ユーザ名 (U):	接続の際、認証の必要がある場合のユーザ名です。詳細は「使用するユーザ名とパスワード」を参照してください。
パスワード (P):	ユーザ名に対応するパスワードです。詳細は「使用するユーザ名とパスワード」を参照してください。



リモートプロセスのアドレス指定方法

参考

リモートプロセスのアドレス指定には、2つの方法があります。

#### 【ホスト名:ポート番号】

接続先サーバのホスト名と接続先コンテナに割り当てられたポート番号によるアドレスの指定方法です。

ホスト名“`www.my.host.com`”、ポート番号“`38686`”とした場合の例を示します。

```
www.my.host.com:38686
```

接続先サーバのホスト名の代わりにIPアドレスを使用することも可能です。

IPアドレス“`192.168.1.2`”、ポート番号“`38686`”とした場合の例を示します。

```
192.168.1.2:38686
```

#### 【JMX URL】

JMXエージェントを接続先として指定する方法です。

以下の形式で入力します。

```
service:jmx:rmi:///jndi/rmi://ホスト名:ポート番号/jmxrmi
```

ホスト名:ポート番号以外は固定です。

英字部分は、すべて小文字で入力する必要があります。

ホスト名:ポート番号は、【ホスト名:ポート番号】で説明した形式で指定します。



接続時、jconsoleの出力ウィンドウが表示された場合

注意

接続時、Java監視機能がエラーを検出するとjconsoleの出力ウィンドウを表示する場合があります。

jconsoleの出力ウィンドウには、エラーを検出した際のException名、エラーメッセージ、Exception発生時のスタックトレースを表示します。

メッセージの意味、必要な対処は「[A.5.12 メッセージ](#)」を参照してください。



Java EEで動作するプロセスに接続できない場合

注意

接続するポート番号を間違っている可能性があります。ポート番号を調べる際のコマンド指定に間違いがないか、確認してください。

コマンドの例については、「[A.5.1 パフォーマンスデータの収集](#)」の「[◆Java EEで動作するプロセスのポート番号の調べ方](#)」を参照してください。

特に、asadminコマンドのオペランドとして「サーバーインスタンス名」を指定していることを確認してください。「サーバーインスタンス名」以外（「IJServerクラスタ名」など）のオペランドを指定した結果表示される情報は、Java監視機能では使用しません。

IJServerクラスタに含まれるサーバーインスタンス名を調べるには、asadminコマンドのlist-instancesサブコマンドを使ってください。コマンドは、次のように実行してください。

```
asadmin list-instances <IJServer クラスタ名>
```



Linux上で動作するJavaプロセスに接続できない場合 [Linux32/64](#)

注意

Linux上で動作する監視対象のJavaプロセスへのリモート接続に失敗する場合、Linux上で次の確認を行ってください。

- ローカルホスト名からIPアドレスへの変換が正しく行われているか

ローカルホスト名からIPアドレスへの変換が正しく行われているかの確認は、例として、次のコマンドの実行結果から確認することができます。

```
hostname -i
```

コマンド実行の結果として“127.0.0.1”が返された場合、ローカルホスト名からIPアドレスへの変換が正しく行われていないため接続に失敗していると考えられます。

/etc/hostsファイルの編集やDNSサーバの設定等で、正しくIPアドレスが解決できるよう、システムの設定を行ってください。

- Java監視機能で使用するポートがアクセス許可されているか

ファイアウォール等でJava監視のために用いるポートが塞がれている（アクセス許可されていない）場合は、コンソール機能がそのポートにアクセスできるように許可してください。

ポートへのアクセスを許可は、例として、次のコマンドを実行することで行えます。

```
/usr/sbin/iptables -I INPUT -s jconsole-host -p tcp -d jmxremote-host -j ACCEPT
```

jconsole-host : コンソール機能が動作するホストのIPアドレスやホスト名

jmxremote-host : 監視対象のJavaプロセスが動作するホストのIPアドレスやホスト名

ただし、ポートを開ける際には、十分セキュリティ上のリスクを考慮してください。

## A.5.5 Java監視機能のタブ

jconsoleを起動し、測定対象のコンテナに接続が完了するとJava監視機能の操作画面が選択できるようになります。

Java監視機能としての操作画面は3つあり、それぞれに対応するタブを選択することで操作可能になります。

機能名称	タブ名
メソッドサンプリング機能	メソッドサンプリング
ヒープ分析機能	ヒープ分析
VMオプションの変更機能	VMオプション変更

図A.1 表示されたタブの例



jconsoleで標準に提供されるタブや画面、コマンドオプションについては、Oracle Corporationの提供するJDK6のjconsoleコマンドのマニュアルを参照してください。



jconsoleコマンド起動時の注意

注意

Java監視機能をjconsoleから操作できるようにするためには、かならず“[jconsoleの起動](#)”で示した方法で起動する必要があります。

起動方法が異なった場合、jconsoleが起動できない、Java監視機能のタブが表示されないなどの現象が起こります。

## A.5.6 メソッドサンプリング機能

### ■メソッドサンプリング機能とは

一定の間隔で、メソッドの実行情報(どのメソッドが実行されているか)を示す機能です。



メソッドサンプリング中の他パフォーマンスデータ採取の禁止

注意

メソッドサンプリング実行中にタブ操作により他のパフォーマンスデータ採取を行わないでください。

他のパフォーマンスデータ採取処理の発生により、正しいメソッドサンプリングの結果採取ができなくなります。

特にヒープ分析と同時に操作しないようにしてください。



メソッドサンプリング操作中にjconsoleの出力ウィンドウが表示された場合

注意

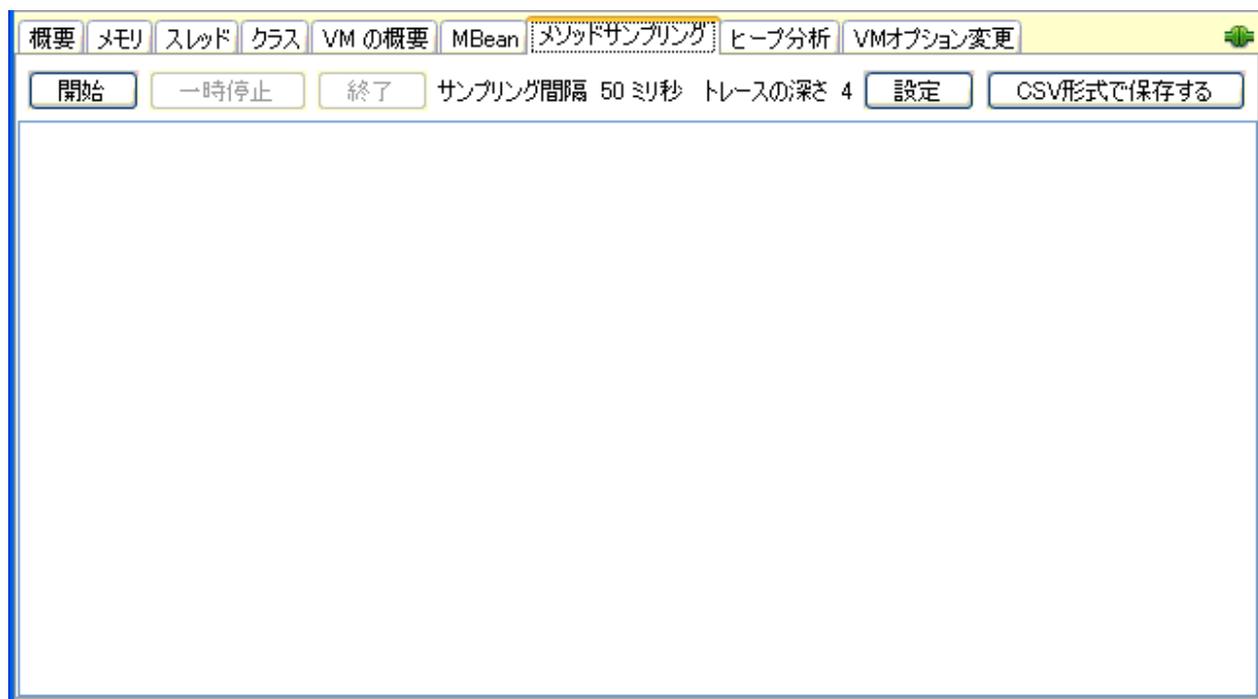
メソッドサンプリング操作時、Java監視機能がエラーを検出するとjconsoleの出力ウィンドウを表示する場合があります。

jconsoleの出力ウィンドウには、エラーを検出した際のException名、エラーメッセージ、Exceptionが発生した時のスタックトレースを表示します。

メッセージの意味、必要な対処は“[A.5.12 メッセージ](#)”を参照してください。

## ◆メソッドサンプリング機能の選択

「メソッドサンプリング」のタブを選択(クリック)することでメソッドサンプリングの画面が表示され、操作が可能な状態になります。



## ◆メソッドサンプリング機能でできること

メソッドサンプリングの画面からは、次の操作ができます。

- メソッドサンプリング結果の表示
- メソッドサンプリング結果の保存

## ■メソッドサンプリング結果の表示

1. メソッドサンプリングの間隔、トレースの深さを設定します。

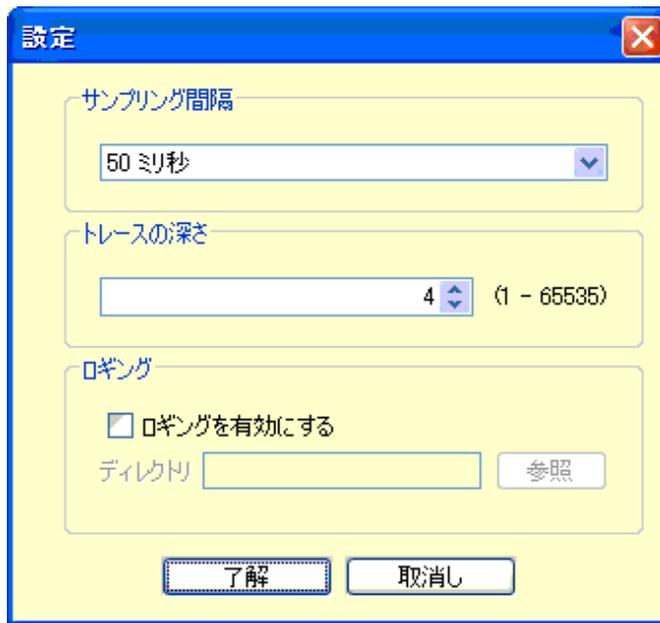
現在の設定値は、メソッドサンプリングの画面内に「サンプリング間隔」と「トレースの深さ」として表示しています。

値を変更する場合、次の手順で変更を行います。

- a. [設定]ボタンを押します。



- b. “設定”ダイアログボックスが表示されます。



※[設定]ダイアログボックスでは、画面上に表示されたメソッドサンプリング結果をファイルに出力するロギング機能の指定も行えますが、ここでは説明を省略します。“ロギング”機能に関しては、“[メソッドサンプリング結果の保存](#)”を参照してください。

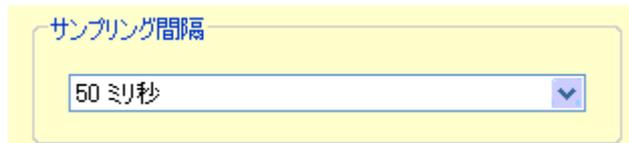
- c. “サンプリング間隔”を設定します。

変更可能なサンプリング間隔は10ミリ秒/50ミリ秒/100ミリ秒/500ミリ秒/1秒の5通りです。

サンプリング間隔のデフォルトは50ミリ秒です。

サンプリング間隔を変更する必要が無い場合、トレースの深さを設定します。

1. “サンプリング間隔”を操作します。



2. プルダウンの一覧内からサンプリング間隔とする時間を選びます。



- d. “トレースの深さ”を設定します。

変更可能なトレースの深さは1から65535です。

トレースの深さのデフォルトは4です。

トレースの深さを変更する必要が無い場合、メソッドサンプリングを開始します。

1. “トレースの深さ”を操作します。

2. トレースの深さを入力します。

キーボードから直接数値を入力するかスピンボックスを操作してください。



“設定”ダイアログボックスの操作はメソッドサンプリング開始前

注意

“設定”ダイアログボックスからサンプリング間隔、トレースの深さの変更は、メソッドサンプリングの開始前に行うことが可能です。

開始後は、[設定]ボタンがグレー表示され、ボタンを操作することができなくなります。



サンプリング間隔は表示の更新間隔ではありません

注意

メソッドサンプリング結果の表示は、一定の間隔で更新されます。この間隔は、サンプリング間隔に指定した時間ではありません。

メソッドサンプリング結果の表示の更新間隔は、jconsoleのintervalオプションで設定します。

intervalオプションのデフォルトは4秒間隔であり、Java監視機能も4秒間隔で更新されます。



“トレースの深さ”の指定は監視対象Javaプロセスのパフォーマンスに影響を与えます

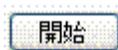
注意

トレースの深さは1～65535まで指定できますが、この数値を大きく(トレースを深く)指定するほど、監視対象のJavaプロセスのパフォーマンスに影響を与えます。

監視環境を検討し、過度な数値での測定を行わないようにしてください。

2. メソッドサンプリングを開始します。

- a. [開始]ボタンを押します。



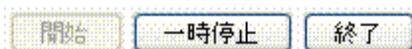
メソッドサンプリングを開始すると次のボタンの表示に変更されます。

ボタン名称	開始前	開始後
[開始]ボタン	活性状態(選択可能)	グレー表示(選択不可能)
[一時停止]ボタン	グレー表示(選択不可能)	活性状態(選択可能)
[終了]ボタン	グレー表示(選択不可能)	活性状態(選択可能)

【メソッドサンプリング開始前】



【メソッドサンプリング開始後】



- b. 表示領域にメソッドサンプリングの結果が表示されます。

図A.2 メソッドサンプリングの結果表示例

```
TRACE 300004:
    java.net.SocketInputStream.socketRead0(SocketInputStream.java:Unknown line)
    java.net.SocketInputStream.read(SocketInputStream.java:129)
    java.io.BufferedInputStream.fill(BufferedInputStream.java:218)
    java.io.BufferedInputStream.read(BufferedInputStream.java:235)
TRACE 300002:
    sun.nio.ch.DevPollArrayWrapper.poll0(DevPollArrayWrapper.java:Unknown line)
    sun.nio.ch.DevPollArrayWrapper.poll(DevPollArrayWrapper.java:164)
    sun.nio.ch.DevPollSelectorImpl.doSelect(DevPollSelectorImpl.java:68)
    sun.nio.ch.SelectorImpl.lockAndDoSelect(SelectorImpl.java:69)
TRACE 300001:
    java.net.PlainSocketImpl.socketAccept(PlainSocketImpl.java:Unknown line)
    java.net.PlainSocketImpl.accept(PlainSocketImpl.java:384)
    java.net.ServerSocket.implAccept(ServerSocket.java:450)
    java.net.ServerSocket.accept(ServerSocket.java:421)
TRACE 300003:
    java.net.PlainDatagramSocketImpl.receive0(PlainDatagramSocketImpl.java:Unknown line)
    java.net.PlainDatagramSocketImpl.receive(PlainDatagramSocketImpl.java:136)
    java.net.DatagramSocket.receive(DatagramSocket.java:712)
    net.jxta.impl.endpoint.mcast.McastTransport.run(McastTransport.java:579)
TRACE 300006:
    java.net.SocketInputStream.socketRead0(SocketInputStream.java:Unknown line)
    java.net.SocketInputStream.read(SocketInputStream.java:129)
    com.sun.net.ssl.internal.ssl.InputRecord.readFully(InputRecord.java:293)
    com.sun.net.ssl.internal.ssl.InputRecord.read(InputRecord.java:331)
CPU SAMPLES BEGIN (total = 9268) Tue Jul 29 11:35:17 2008
rank  self  accum  count  trace  method
  1 39.15% 39.15%   3628 300004 java.net.SocketInputStream.socketRead0
  2 25.47% 64.62%   2361 300002 sun.nio.ch.DevPollArrayWrapper.poll0
  3 25.47% 90.09%   2361 300001 java.net.PlainSocketImpl.socketAccept
  4  8.49% 98.59%    787 300003 java.net.PlainDatagramSocketImpl.receive0
  5  0.83% 99.42%    77 300006 java.net.SocketInputStream.socketRead0
CPU SAMPLES END
```



負荷によるメソッドサンプリング結果表示への影響

注意

メソッドサンプリングの結果の表示は、一定の間隔で更新処理を行っています。

しかし、測定対象としているコンテナ (Java VM) の負荷、コンテナとjconsole間の通信状態、あるいはjconsole自体の負荷に影響を受け一定間隔の更新が行われないことがあります。



同じトレースの同じメソッドが複数回表示される場合があります

注意

メソッドサンプリングの結果の表示を行っている時、同じトレースの同じメソッドが複数表示されることがあります。

これは、Javaソースとして一行であっても、バイトコード命令として複数行の命令になるためです。

### 3. メソッドサンプリングの表示更新を一時停止します。

今表示されているメソッドサンプリングの結果を画面上で分析している間、あるいはCSVファイルに出力させる時など、今の表示を保ちたいことがあります。そのようなときに表示更新を一時停止します。(表示更新を一時停止するだけであり、メソッドサンプリングの処理は停止しません。メソッドサンプリングの停止は[終了]ボタンを押すことで行ってください。)

なお、ロギング機能でロギングを行う設定をしている ([設定]ダイアログで“ロギングを有効にする”を選んでいる) 場合、メソッドサンプリングの表示更新を一時停止中は、ロギングが行われません。

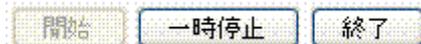
メソッドサンプリングの表示更新を一時停止するためには、[一時停止]ボタンを押すことで行ってください。

**一時停止**

[一時停止]ボタンを押すと次のボタンの表示に変更されます。

ボタン名称	一時停止前	一時停止後
[開始]ボタン	グレー表示(選択不可能)	グレー表示(選択不可能)
[一時停止]ボタン	活性状態(選択可能)	[再開]ボタンに変更され、活性状態(選択可能)
[終了]ボタン	活性状態(選択可能)	活性状態(選択可能)

【一時停止前】



【一時停止後】



4. メソッドサンプリングの表示更新を一時停止から再開します。

一時停止させているメソッドサンプリング結果の表示を再開させます。

再開には、[再開]ボタンを押します。

**再開**

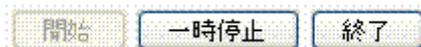
[再開]ボタンを押すと次のボタンの表示に変更されます。

ボタン名称	一時停止前	一時停止後
[開始]ボタン	グレー表示(選択不可能)	グレー表示(選択不可能)
[再開]ボタン	活性状態(選択可能)	[一時停止]ボタンに変更され、活性状態(選択可能)
[終了]ボタン	活性状態(選択可能)	活性状態(選択可能)

【再開前】

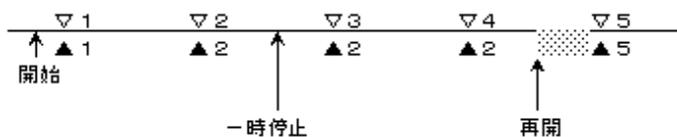


【再開後】



再開時、メソッドサンプリングの結果の表示およびロギングを行っている([設定]ダイアログで“ロギングを有効にする”を選んでいる)場合は、再開後最初のメソッドサンプリングが行われるまで変更されません。

メソッドサンプリング



メソッドサンプリング表示

▽N(Nは数字)はメソッドサンプリング採取のタイミング

▲N(Nは数字)はメソッドサンプリングの結果として表示されているメソッドサンプリングの回数

上の例では、一時停止後、メソッドサンプリングは2回(3と4)採取されていますが、表示は一時停止前(2)の内容がそのまま表示されます。

(メソッドサンプリング4と5の間で)再開を押しましたが、再開後、最初のメソッドサンプリングが実施されるまで(網掛けの間)、表示は一時停止中と同じ(2)表示が続きます。

再開後、最初のメソッドサンプリング(5)が実行されると表示も変更(5)されます。

#### 5. メソッドサンプリングを終了します。

メソッドサンプリングの採取を終了させ、結果の表示を停止します。

終了には、[終了]ボタンを押します。

終了

[終了]ボタンを押すと次のボタンの表示に変更されます。

ボタン名称	終了前	終了後
[開始]ボタン	グレー表示(選択不可能)	活性状態(選択可能)
[一時停止ボタン][再開]ボタン	活性状態(選択可能)	[一時停止]ボタンに変更され、グレー表示(選択不可能)
[終了]ボタン	活性状態(選択可能)	グレー表示(選択不可能)

#### 【終了前】

開始

一時停止

終了

(一時停止中)

開始

再開

終了

#### 【再開後】

開始

一時停止

終了

[終了]ボタンは、メソッドサンプリングの開始後、いつでも押すことができます。

また、[終了]ボタンを押さずにjconsoleを終了した場合、メソッドサンプリングは継続されます。



メソッドサンプリング中のjconsoleの再起動

注意

メソッドサンプリングを開始したまま(メソッドサンプリングの[終了]ボタンを押さずに)、jconsoleを終了させた場合、メソッドサンプリングの採取は継続されます。

この状態でjconsoleを再度起動し、同一の測定対象コンテナに接続した場合、メソッドサンプリングは継続しているためボタンは、以下の状態になります。

開始

一時停止

終了

また、メソッドサンプリング結果の再表示も行われます。

## ■メソッドサンプリング結果の保存

メソッドサンプリング結果を保存する方法には、次の2つの方法があります。

- ・ 現在表示されているメソッドサンプリングの結果を保存する
- ・ メソッドサンプリングの表示結果をロギングする

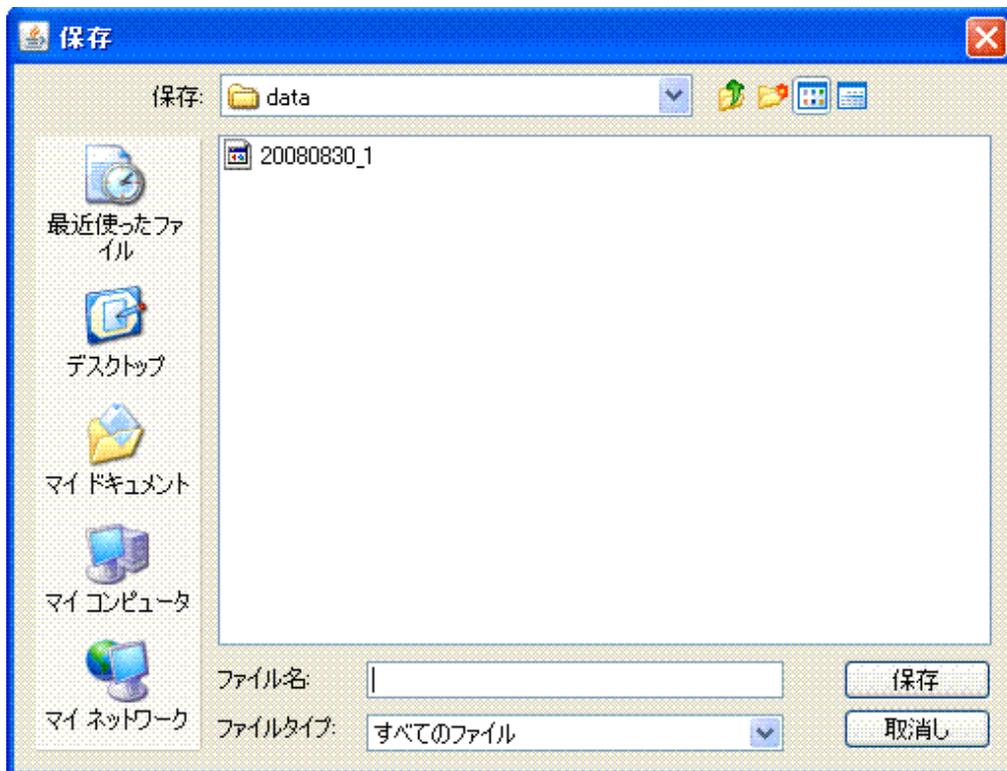
### ◆現在表示されているメソッドサンプリングの結果を保存する

現在表示されているメソッドサンプリングの結果をCSV形式でファイルに保存します。

保存したいメソッドサンプリングの結果が表示されている状態で、[CSV形式で保存する]ボタンを押します。

CSV形式で保存する

[CSV形式で保存する]ボタンを押すと“保存”ダイアログボックスが表示されます。



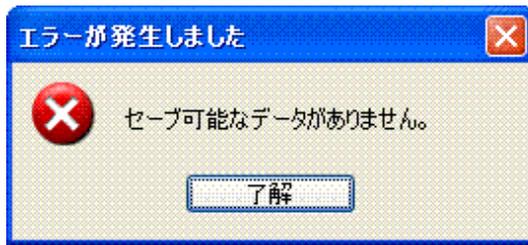
以下の手順で保存を行います。

1. データを保存するフォルダに移動
2. “ファイル名:”に保存するファイル名を入力
3. [保存]ボタンを押す

[取消し]ボタン、あるいは[×]ボタンを押すと保存処理を行わずに“保存”ダイアログボックスが終了します。

なお、保存したファイルのデータ項目の意味は、“◆ファイル保存されたCSV形式”を参照してください。

メソッドサンプリングの結果が表示されていない状態で、[CSV形式で保存する]ボタンを押すと以下のメッセージが表示され、保存処理を行わずに終了します。



メソッドサンプリング結果の保存は、表示更新を一時停止または中止してから

#### ポイント

[CSV形式で保存する]ボタンを押し“保存”ダイアログボックスを表示している間もメソッドサンプリングは継続されています。

そのため、“保存”ダイアログボックスの操作を行っている間にも、メソッドサンプリングの結果の表示は更新されてしまう可能性があります。

目的のメソッドサンプリング結果を確実に保存するために[一時停止]ボタンあるいは[終了]ボタンを押し、メソッドサンプリングの結果の表示が更新されないようにした後、[CSV形式で保存する]ボタンを押しメソッドサンプリング結果の保存操作を行ってください。

### ◆メソッドサンプリングの表示結果をロギングする

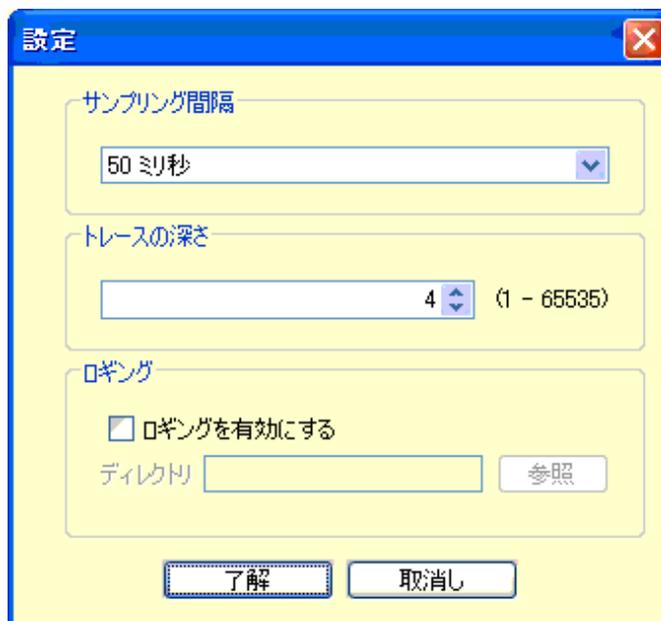
メソッドサンプリングの開始時から表示される結果を自動的にテキスト形式でファイルに保存します。

メソッドサンプリング開始前に、[設定]ダイアログからロギングの設定を行います。

1. [設定]ボタンを押します。



2. “設定”ダイアログボックスが表示されます。



3. “ロギング”を設定します。

- a. “ロギングを有効にする”チェックボックスを指定します。

“ロギングを有効にする”チェックボックスがチェックされていないと、“ディレクトリ”はグレー表示され、ディレクトリを入力することができません。

- b. “ディレクトリ”にロギングファイルを作成するディレクトリを指定します。

“ディレクトリ”の指定には、直接キーボードからディレクトリ名を入力する他、[参照]ボタンを押し、“開く”ダイアログボックスからディレクトリを選ぶことも可能です。

## A.5.7 ヒープ分析機能

---

### ■ヒープ分析機能とは

Javaヒープ領域でのオブジェクトの使用状況を参照する機能です。

次の2つのデータからJavaヒープの分析を行います。

- a. オブジェクトヒストグラム

Javaオブジェクトの使用状況を任意の二点間の差分にて示します。

Javaオブジェクトの使用状況とは、現在のヒープにおけるJavaオブジェクトのヒストグラム(どのクラスのインスタンスが、何個存在し、何バイト占めている)を示すものです。

- b. ヒープダンプ

現在のヒープダンプ(ヒープイメージ)を示します。



ヒープ分析時の留意事項

注意

オブジェクトヒストグラムやヒープダンプを採取した時、サーバ上のJava VMは一時的に停止状態になります。この停止状態の長さは、ヒープの状況(大きさや使用の複雑さ)に依存します。

この間ヒープ分析の画面上のボタンを押すと応答の待ちが発生し、一時的に操作ができないことがあります。



ヒープ分析中の他パフォーマンスデータ採取の禁止

注意

ヒープ分析実行中にタブ操作により他のパフォーマンスデータ採取を行わないでください。

他のパフォーマンスデータ採取処理の発生により、正しいヒープ分析の結果採取ができなくなります。

特にメソッドサンプリングと並行しないようにしてください。



ヒープ分析操作中にjconsoleの出力ウィンドウが表示された場合

注意

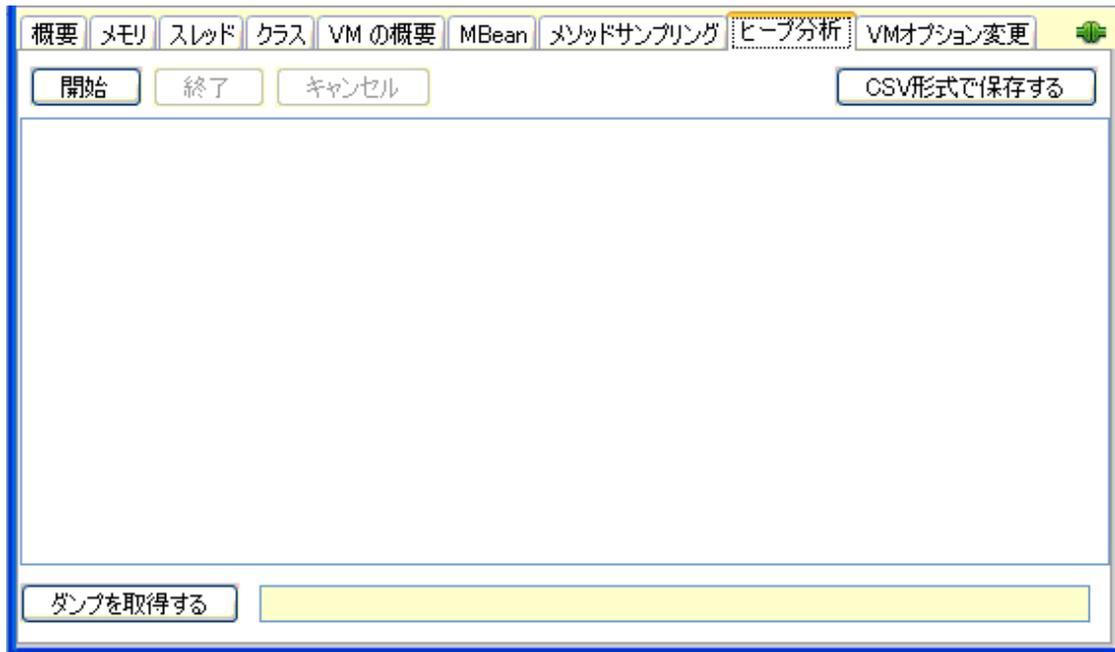
ヒープ分析操作時、Java監視機能がエラーを検出するとjconsoleの出力ウィンドウを表示する場合があります。

jconsoleの出力ウィンドウには、エラーを検出した際のException名、エラーメッセージ、Exception発生時のスタックトレースを表示します。

メッセージの意味、必要な対処は“[A.5.12 メッセージ](#)”を参照してください。

### ■ヒープ分析機能の選択

「ヒープ分析」のタブを選択(クリック)することでオブジェクトヒストグラムの画面が表示され、操作が可能な状態になります。



### ■ヒープ分析機能でできること

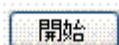
ヒープ分析機能から操作できることは、次に示す通りです。

- オブジェクトヒストグラムの表示
- オブジェクトヒストグラムデータの保存
- ヒープダンプの採取

### ■オブジェクトヒストグラムの表示

オブジェクトヒストグラムの表示は、次の手順で操作します。

1. [開始]ボタンを押します。



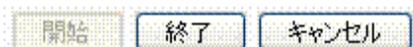
オブジェクトヒストグラムの採取を開始するとボタンの表示は次のように変更されます。

ボタン名称	開始前	開始後
[開始]ボタン	活性状態(選択可能)	グレー表示(選択不可能)
[終了]ボタン	グレー表示(選択不可能)	活性状態(選択可能)
[キャンセル]ボタン	グレー表示(選択不可能)	活性状態(選択可能)

【オブジェクトヒストグラムの採取開始前】



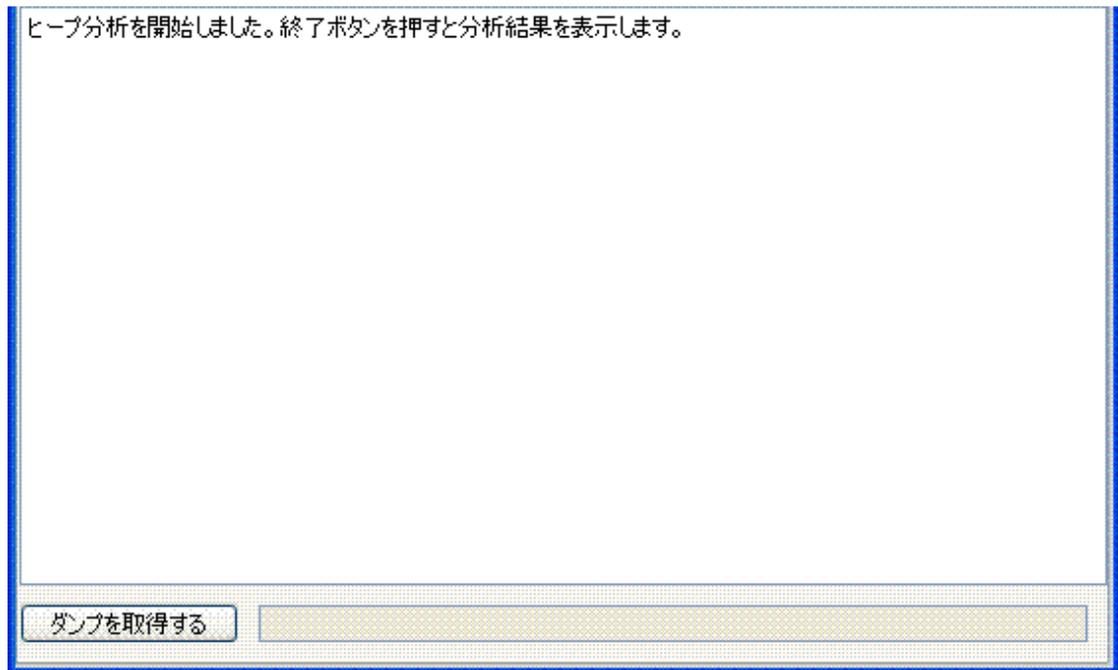
【オブジェクトヒストグラムの採取開始後】



この時点では、表示領域にオブジェクトヒストグラムが表示されることはありません。

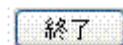
また、以前採取したオブジェクトヒストグラムが表示されていた場合、クリアします。

その後、ヒープ分析を開始した旨を伝えるメッセージが表示されます。



オブジェクトヒストグラムの採取を開始するときに、監視対象のJavaプロセスでは、GCが実行されます。GCの実行がアプリケーションに大きな影響を与える場合は、この操作を行わないでください。

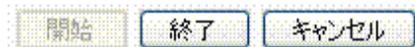
2. [終了]ボタンを押します。



オブジェクトヒストグラムの採取を終了するとボタンの表示は次のように変更されます。

ボタン名称	終了前	終了後
[開始]ボタン	グレー表示(選択不可能)	活性状態(選択可能)
[終了]ボタン	活性状態(選択可能)	グレー表示(選択不可能)
[キャンセル]ボタン	活性状態(選択可能)	グレー表示(選択不可能)

【オブジェクトヒストグラムの採取終了前】

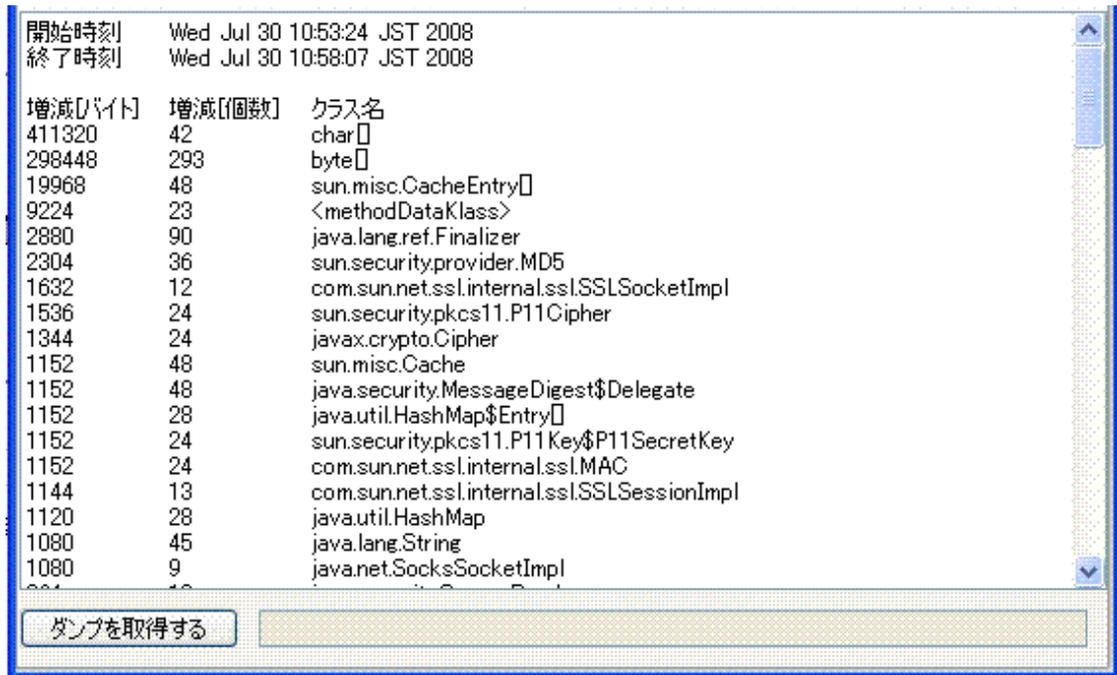


【オブジェクトヒストグラムの採取終了後】



[終了]ボタンが押されると表示領域にオブジェクトヒストグラムが表示されます。

図A.3 オブジェクトヒストグラムの表示例



なお、表示されたデータ項目の説明は、“◆画面表示されたデータ形式”を参照してください。



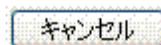
注意

オブジェクトヒストグラムの採取を終了するとき、監視対象のJavaプロセスでは、GCが実行されます。GCの実行がアプリケーションに大きな影響を与える場合は、この操作を行わないでください。

### 3. オブジェクトヒストグラムの採取をキャンセルします。

オブジェクトヒストグラムの採取をキャンセルすると採取のために実行されていたヒープの分析も終了します。

オブジェクトヒストグラムの採取をキャンセルするには[キャンセル]ボタンを押します。



オブジェクトヒストグラムの採取をキャンセルするとボタンの表示は次のように変更されます。

ボタン名称	キャンセル前	キャンセル後
[開始]ボタン	グレー表示(選択不可能)	活性状態(選択可能)
[終了]ボタン	活性状態(選択可能)	グレー表示(選択不可能)
[キャンセル]ボタン	活性状態(選択可能)	グレー表示(選択不可能)

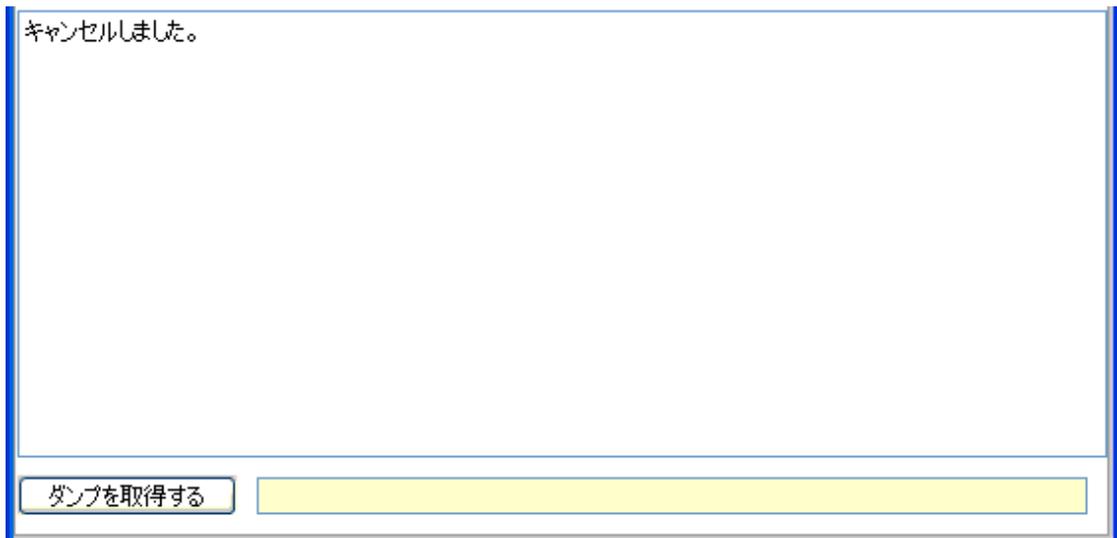
【オブジェクトヒストグラムの採取終了前】



【オブジェクトヒストグラムの採取終了後】



[キャンセル]ボタンが押されると表示領域に「キャンセルしました。」とメッセージが表示されます。

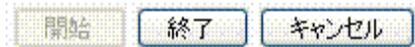


オブジェクトヒストグラム採取中のjconsoleの再起動

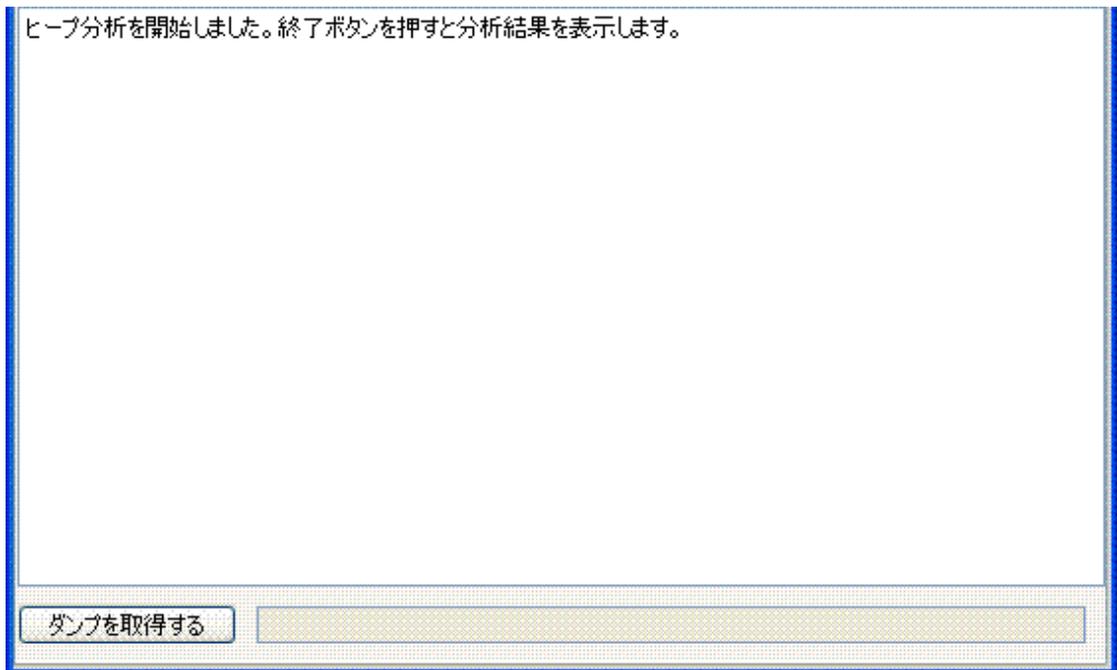
注意

オブジェクトヒストグラムを開始したまま(オブジェクトヒストグラムの[終了]ボタンあるいは[キャンセル]ボタンを押さずに)、jconsoleを終了させた場合、オブジェクトヒストグラムの採取は継続されます。

この状態でjconsoleを再度起動し、同一の測定対象コンテナに接続した場合、メソッドサンプリングは継続しているためボタンは、以下の状態になります。



また表示域は、以下の状態になります。



この状態から[終了]ボタンを押すことでjconsole終了前からの継続でオブジェクトヒストグラムの採取が完了し、オブジェクトヒストグラムの表示が行われます。また、[キャンセル]ボタンによりオブジェクトヒストグラムの採取をキャンセルすることができます。

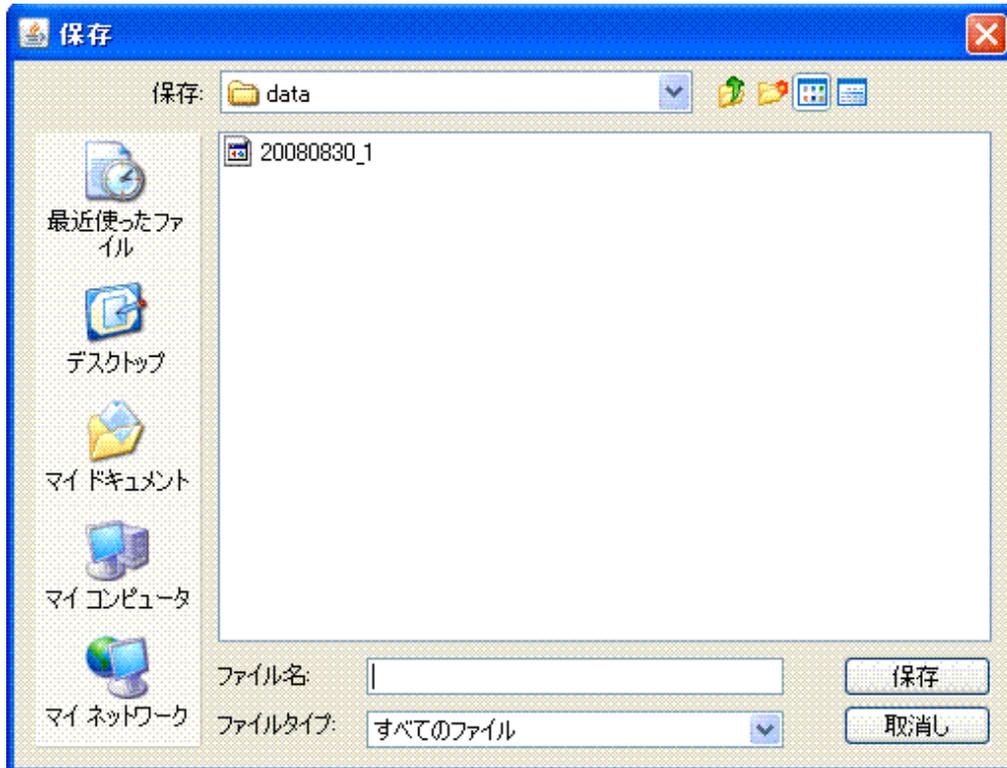
## ■オブジェクトヒストグラムデータの保存

現在表示されているオブジェクトヒストグラムの結果をCSV形式でファイルに保存します。

保存したいオブジェクトヒストグラムが表示されている状態で、[CSV形式で保存する]ボタンを押します。

CSV形式で保存する

[CSV形式で保存する]ボタンを押すと“保存”ダイアログボックスを表示します。



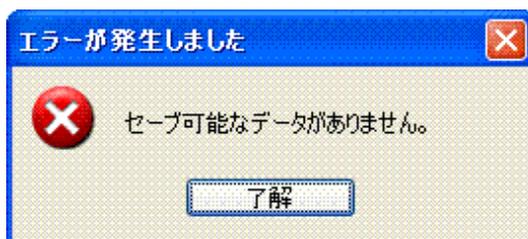
以下の手順で保存を行います。

1. データを保存するフォルダに移動
2. “ファイル名:”に保存するファイル名を入力
3. [保存]ボタンを押す

保存したファイルの説明は、“◆ファイル保存されたCSV形式”を参照してください。

[取消し]ボタン、あるいは[×]ボタンを押すと“保存”ダイアログボックスが閉じられ、保存処理は行われません。

オブジェクトヒストグラムの結果が表示されていない状態で、[CSV形式で保存する]ボタンを押した場合、以下のメッセージが表示され、保存処理は行われません。



## ■ヒープダンプの採取

現在のJavaヒープのイメージをヒープダンプといい、バイナリ形式でファイルに保存することができます。

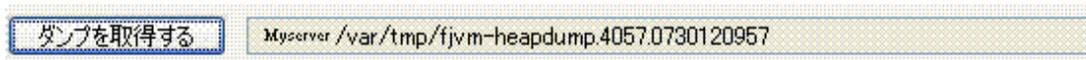
ヒープダンプの採取は、[ダンプを取得する]ボタンを押します。



ヒープダンプを採取するときに、監視対象のJavaプロセスでは、GCが実行されます。GCの実行がアプリケーションに大きな影響を与える場合は、この操作を行わないでください。

ヒープダンプは、Java監視機能の操作を行っているマシン上ではなく、測定対象としているコンテナが動作しているアプリケーションサーバ上に採取します。(Java監視機能の画面上で見ることができません。Java監視機能が動作している端末上にファイル転送することはありません。)

採取したファイルがどこに取られているかを[ダンプを取得する]ボタンの横の表示域に表示します。



ヒープダンプファイルが保存される場所およびファイル名については、“◆ヒープダンプ”を参照してください。

ヒープダンプの採取が何らかの原因により失敗した場合、[ダンプを取得する]ボタンの横の表示域には、「取得に失敗しました」のエラーメッセージが、詳細メッセージ(取得に失敗した原因を示す情報)と表示されます。

また、詳細メッセージは、測定対象Javaプロセスの標準出力にも出力されます。

採取したヒープダンプは、インストールされるJDK6に添付されるjhatコマンドで分析可能です。



採取したヒープダンプのアクセス権

採取したヒープダンプファイルのアクセス権は、ヒープダンプを採取した監視対象のプロセスの所有者のみに与えられます。



ヒープダンプ採取後のjconsoleの再起動

ヒープダンプを採取した後(ファイル名を表示域に表示した状態で)、jconsoleを一旦終了させ、再度起動の後、同一のコンテナ(サーバ上のプロセス)に再接続した場合、サーバ上に終了前に採取したヒープダンプファイルが存在するとヒープダンプファイル名を表示した状態になります。

ただし、サーバ上に終了前に採取したヒープダンプファイルが存在しない場合は、ヒープダンプファイル名は表示されません。



ヒープダンプ採取前のディスク容量確認

ヒープダンプファイルの出力ファイルサイズが大きくなり、ヒープダンプの採取によるサーバ側のディスク容量を圧迫することがあります。

ヒープダンプ採取の前には、出力先の空き容量が十分であることを確認してください。

## A.5.8 VMオプションの変更

### ■VMオプションの変更機能とは

通常Java VMの状態を見るためには、Java VM起動時にオプション(たとえばjavaコマンドのパラメタ)で設定する必要があります。また、設定したオプションはJava VMの起動から終了まで一意で、設定値を途中で変更することはできません。

Java監視機能のVMオプションの変更機能を使用すると、すでに起動中のJava VMのオプションを変更することができます。

ただし、変更できるオプションは、VMオプションの変更機能で規定する一部のオプションに限ります。



VMオプション操作により得られる結果の出力先  
**注意**

VMオプションの変更機能は、現在Java監視機能が監視しているJava VMに設定されているオプションの値を変更するだけです。

Java監視機能の画面上で結果を見ることや、Java監視機能が動作している端末上にファイルとして出力することはできません。

VMオプション操作により得られる結果は、Javaプロセスが動作しているサーバ上に出力されます。



Java監視機能が終了する時、変更したVMオプションの値はそのままになります  
**注意**

Java監視機能が終了した時、それまでに変更したVMオプションの値は継続されます。

(Java監視機能の終了とは、意図的なjconsoleの終了以外、Java監視機能やjconsoleの異常や接続の解除などを含みます。)

Java監視機能は、監視対象のJava VMを起動した時の値に戻す処理は行いません。



VMオプションの変更機能操作中にjconsoleの出力ウィンドウが表示された場合  
**注意**

VMオプションの変更機能操作の時、Java監視機能がエラーを検出するとjconsoleの出力ウィンドウを表示する場合があります。

jconsoleの出力ウィンドウには、エラーを検出した際のException名、エラーメッセージ、Exception発生時のスタックトレースを表示します。

メッセージの意味、必要な対処は“[A.5.12 メッセージ](#)”を参照してください。

### ■VMオプションの変更機能の選択

「VMオプション変更」のタブを選択(クリック)することでVMオプションの変更画面が表示され、操作が可能な状態になります。

概要	メモリ	スレッド	クラス	VMの概要	MBean	メソッドサンプリング	ヒープ分析	VMオプション変更
GC表示ポリシー:	表示なし	変更する						
System.gc()の抑止:	無効	変更する						
System.gc()の検出:	無効	変更する						
タイムスタンプ形式:	経過時間	変更する						

この時点で表示されている各オプションの値は、Java監視機能が、測定対象のJava VMに接続した時点のものです。

## ■VMオプションの変更機能で変更可能なオプション

変更できるオプションは次のものです。

- GC表示ポリシー
- System.gc()の抑止
- System.gc()の検出
- タイムスタンプ形式

## ■GC表示ポリシーの変更

GC表示ポリシーとは、Java VMで発生しているガーベジコレクション(GC)処理の情報を出力する形式を示すオプションです。

変更は次の手順で行います。

1. “GC表示ポリシー”の横の[変更する]ボタンを押します。



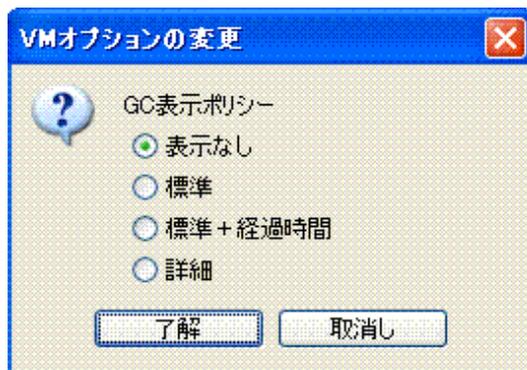
この例は、“GC表示ポリシー”の設定が行われていない(GCに関する情報出力の指定が行われていない)ことを示しています。

“GC表示ポリシー”には、次のものがあります。

GC表示ポリシー	意味
表示なし	GC情報の出力指定はありません。
標準	“-verbose:gc”指定に相当する情報の出力の指定です。
標準+経過時間	“-verbose:gc”指定に相当する情報にJava VM起動からの経過時間を付けた出力の指定です。
詳細	“-verbose:gc -XX:+UseFJverbose”指定に相当する情報の出力指定です。
未定義	“表示なし”、“標準”、“標準+経過時間”、“詳細”以外の指定がされている場合を示します。これは、非サポートオプションである「-XX:+PrintGCDetails」がJavaの起動オプションに含まれる場合などの、例外的な状態であることを示します。 変更ダイアログの選択肢としては、表示されません。例えば、「未定義」から「表示なし」に変更したあとで、「未定義」に戻すことはできません。

“-XX:+UseFJverbose”については、「チューニングガイド」の「ガーベジコレクション処理の結果ログ出力機能の強化」を参照してください。

2. GC表示ポリシーの変更ダイアログが表示されます。



3. 変更する形式のラジオボタンを選択します。



上の図は“標準+経過時間”を選んだ場合の例です。

このとき[取消し]ボタンまたは[×]ボタンを押すと、変更を行わずにダイアログを閉じます。

4. [了解]ボタンを押し、GC表示ポリシーが変更されます。

このとき“GC表示ポリシー”の表示も選んだものに変更されます。



GC情報の出力先

注意

“GC表示ポリシー”に「表示なし」以外を指定した場合に出力されるGC情報は、監視対象のJavaプロセスの標準出力に出力されます。

ただし、監視対象のJavaプロセス起動時に、オプションとして“-Xloggc:ファイル名”が指定されている場合、GC情報は“ファイル名”に指定したファイルに出力されます。



“-Xloggc:ファイル名”指定でのデフォルトGC表示ポリシー

注意

Javaプロセス起動時に、オプションとして“-Xloggc:ファイル名”が指定されている場合でかつ、GC情報の出力に関するオプションの指定が無い場合、“GC表示ポリシー”は「標準+経過時間」がデフォルトとして採用されます。

## ■System.gc()の抑止オプションの変更

Javaプログラム内でSystem.gc()メソッドまたはRuntime.gc()メソッドの呼び出しが行われた場合、この2つのメソッドによるGC処理を抑止するオプションです。

オプションの操作は、次の手順で行います。

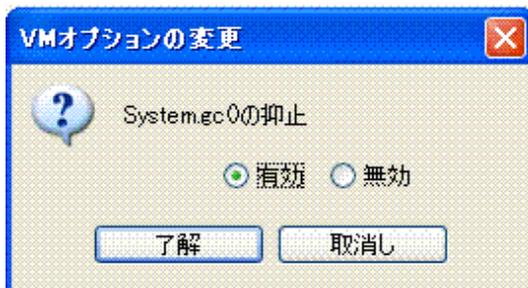
1. “System.gc()の抑止”の横の[変更する]ボタンを押します。



2. System.gc()の抑止の変更ダイアログが表示されます。



3. 変更するラジオボタンを選択します。



上の図は無効から有効に変更する場合の例です。

このとき[取消し]ボタンまたは[×]ボタンを押すと、変更を行わず、ダイアログを終了します。

4. [了解]ボタンを押し、System.gc()の抑止の選択状態が変更されます。

このとき“System.gc()の抑止”の表示も変更されます。



## ■System.gc()の検出オプションの変更

Javaプログラム内でSystem.gc()メソッドまたはRuntime.gc()メソッドの呼び出しが行われた場合、呼び出したJavaスレッドのスタックトレースを出力するオプションです。

オプションの操作は、次の手順で行います。

1. “System.gc()の検出”の横の[変更する]ボタンを押します。



2. System.gc()の検出の変更ダイアログが表示されます。



- 変更するラジオボタンを選択します。



上の図は無効から有効に変更する場合の例です。

このとき[取消し]ボタンまたは[×]ボタンを押すと、変更を行わず、ダイアログを終了します。

- [了解]ボタンを押し、System.gc()の検出の選択状態が変更されます。

このとき“System.gc()の検出”の状態の表示も変更されます。



注意

System.gc()検出時のスタックトレース出力先について

“System.gc()の検出”を「有効」に設定し、System.gc()メソッドまたはRuntime.gc()メソッドの呼び出しを検出した場合のスタックトレースの出力先は、監視対象のJavaプロセス起動時の“-XX:+PrintJavaStackAtSystemGC”オプションの指定の有無に影響を受けます。

“-XX:+PrintJavaStackAtSystemGC”オプションの指定	出力先
なし	標準出力(※)
あり	標準出力(※)およびFJVMログファイル

※:標準出力がロギングされている場合は、そのファイルへ出力されます。

“System.gc()の検出”を「無効」に設定した場合は、“-XX:+PrintJavaStackAtSystemGC”オプションの指定の有無に関係無くスタックトレースの出力は行われません。

“-XX:+PrintJavaStackAtSystemGC”オプションについては、「チューニングガイド」の「java.lang.System.gc()実行時におけるスタックトレース出力機能」を参照してください。

## ■タイムスタンプ形式オプションの変更

GC表示ポリシーで「詳細」を選んだときの出力情報や、動的コンパイル発生状況のログ出力に含まれる、タイムスタンプの形式を指定するオプションです。オプション“-XX:FJverboseTime”の指定に相当します。

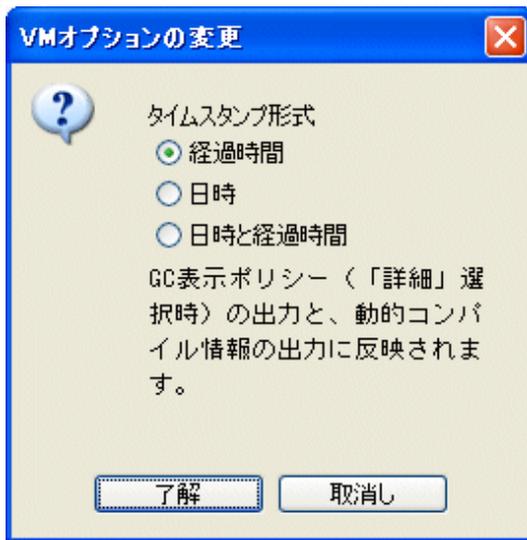
オプション“-XX:FJverboseTime”については、「チューニングガイド」の「ログ出力における時間情報のフォーマット指定機能」参照してください。

オプションの操作は、次の手順で行います。

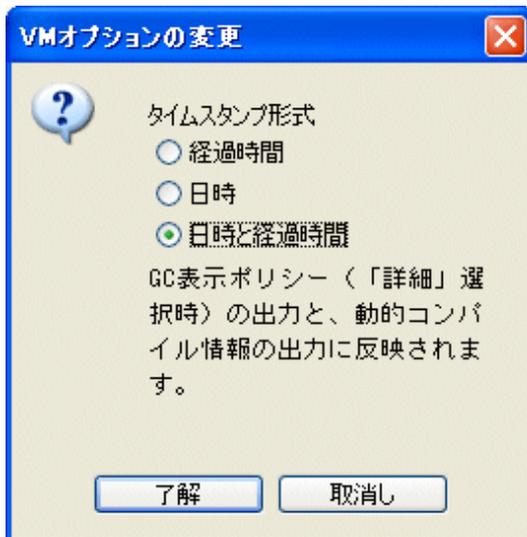
- “タイムスタンプ形式”の横の[変更する]ボタンを押します。



2. タイムスタンプ形式の変更ダイアログが表示されます。



3. 変更するラジオボタンを選択します。



上の図は、タイムスタンプの形式を「経過時間」から「日時と経過時間」に変更する場合の例です。

このとき[取消し]ボタンまたは[×]ボタンを押すと、オプションを変更せず、ダイアログを終了します。

4. [了解]ボタンを押すと、タイムスタンプ形式が変更されます。

このとき“タイムスタンプ形式”の表示も変更されます。



## A.5.9 接続の解除

Java監視機能と監視対象Javaプロセスの接続を切断することができます。

接続を切断すると、Java監視機能と監視対象Javaプロセスの間の、測定データの通信が切れます(この状態を「接続解除」といいます)。

## ■接続解除の手順

接続解除の順番は、次の通りです。

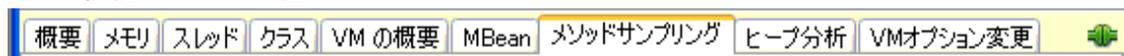
1. 接続アイコン()をクリックします。

図A.4 接続アイコン



接続アイコンは、タブの右横にあります。

図A.5 接続アイコンの配置



2. 「接続が失われました:再接続しますか?」のメッセージへの応答

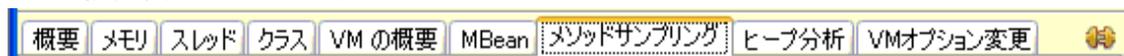
[取消し]ボタンと[再接続]ボタンが表示されます。

[取消し]ボタンを押すと再接続を行いません。

[再接続]ボタンを押すとこれまで接続していたホスト、ポートへの再接続が行われます。再接続については“[■再接続の手順](#)”を参照してください。

[取消し]ボタンを押すと接続アイコンが、接続解除アイコンに代わり、接続が切断された状態にあることを示します。

図A.6 接続解除アイコンの配置



図A.7 接続解除アイコン



## ■再接続の手順

再接続により接続解除した時と同じホスト、ポートへ接続することができます。

接続解除アイコンをクリックします。

図A.8 接続解除アイコン



接続解除アイコンは、タブの右横にあります。

図A.9 接続解除アイコンの配置



「接続しています」のメッセージが表示されます。

その後、接続解除アイコンが、接続アイコンに代わり、接続状態にあることを示します。

図A.10 接続アイコンの配置



## 図A.11 接続アイコン



意図しない「接続が失われました:再接続しますか?」のメッセージが表示された場合

**注意**

接続アイコンを操作していない(意図的な接続解除の操作を行っていない)場合にも「接続が失われました:再接続しますか?」のメッセージが表示されることがあります。

この場合、監視対象のJavaプロセスが終了している可能性があります。

そのため、再接続を行わないようにしてください。

メッセージに対し[取消し]ボタンを押してください。

## A.5.10 jconsoleの終了

---

Java監視機能を終了させるためにはjconsoleを終了させます。

jconsoleの終了は、[×]ボタンを押す、あるいはメニューから“終了”コマンドを実行します。

jconsoleの終了時にJava監視機能と監視対象Javaプロセスの接続が行われている場合、接続解除されます。

しかし、終了までにJava監視機能の操作した機能は、そのまま継続(保持)されます。

jconsoleを再起動後、同じ監視対象のコンテナに接続を行うとJava監視機能は、jconsole終了時の状態から再開されます。

## A.5.11 データ説明

---

Java監視機能が画面に表示、あるいはファイルに保存するデータの形式について説明します。

### ■メソッドサンプリング

メソッドサンプリング機能を操作した際に画面(結果の表示域)に表示されるデータの形式やファイル出力したCSV形式について説明します。

### ◆画面表示されたデータ形式

メソッドサンプリングの結果の表示例を以下に示します。

なお、ロギング機能の指定によりロギングファイルに採取された情報は、画面表示されたメソッドサンプリング結果と同じです。

ロギングファイルに採取された情報は、本説明を参照してください。

```

TRACE 300004:
    java.net.SocketInputStream.socketRead0(SocketInputStream.java:Unknown line)
    java.net.SocketInputStream.read(SocketInputStream.java:129)
    java.io.BufferedInputStream.fill(BufferedInputStream.java:218)
    java.io.BufferedInputStream.read(BufferedInputStream.java:235)
TRACE 300002:
    sun.nio.ch.DevPollArrayWrapper.poll0(DevPollArrayWrapper.java:Unknown line)
    sun.nio.ch.DevPollArrayWrapper.poll(DevPollArrayWrapper.java:164)
    sun.nio.ch.DevPollSelectorImpl.doSelect(DevPollSelectorImpl.java:68)
    sun.nio.ch.SelectorImpl.lockAndDoSelect(SelectorImpl.java:69)
TRACE 300001:
    java.net.PlainSocketImpl.socketAccept(PlainSocketImpl.java:Unknown line)
    java.net.PlainSocketImpl.accept(PlainSocketImpl.java:384)
    java.net.ServerSocket.implAccept(ServerSocket.java:450)
    java.net.ServerSocket.accept(ServerSocket.java:421)
TRACE 300003:
    java.net.PlainDatagramSocketImpl.receive0(PlainDatagramSocketImpl.java:Unknown line)
    java.net.PlainDatagramSocketImpl.receive(PlainDatagramSocketImpl.java:136)
    java.net.DatagramSocket.receive(DatagramSocket.java:712)
    net.jxta.impl.endpoint.mcast.McastTransport.run(McastTransport.java:579)
TRACE 300006:
    java.net.SocketInputStream.socketRead0(SocketInputStream.java:Unknown line)
    java.net.SocketInputStream.read(SocketInputStream.java:129)
    com.sun.net.ssl.internal.ssl.InputRecord.readFully(InputRecord.java:293)
    com.sun.net.ssl.internal.ssl.InputRecord.read(InputRecord.java:331)
CPU SAMPLES BEGIN (total = 9268) Tue Jul 29 11:35:17 2008
rank  self accum  count trace method
  1 39.15% 39.15%   3628 300004 java.net.SocketInputStream.socketRead0
  2 25.47% 64.62%   2361 300002 sun.nio.ch.DevPollArrayWrapper.poll0
  3 25.47% 90.09%   2361 300001 java.net.PlainSocketImpl.socketAccept
  4  8.49% 98.59%    787 300003 java.net.PlainDatagramSocketImpl.receive0
  5  0.83% 99.42%    77 300006 java.net.SocketInputStream.socketRead0
CPU SAMPLES END

```

メソッドサンプリングの結果は、[スタックトレース]と[サンプリング情報]の2種類のデータで構成されます。

- スタックトレース

例の中で、次の部分を示します。

```

TRACE 300004:
    java.net.SocketInputStream.socketRead0(SocketInputStream.java:Unknown line)
    java.net.SocketInputStream.read(SocketInputStream.java:129)
    java.io.BufferedInputStream.fill(BufferedInputStream.java:218)
    java.io.BufferedInputStream.read(BufferedInputStream.java:235)
TRACE 300002:
    sun.nio.ch.DevPollArrayWrapper.poll0(DevPollArrayWrapper.java:Unknown line)
    sun.nio.ch.DevPollArrayWrapper.poll(DevPollArrayWrapper.java:164)
    sun.nio.ch.DevPollSelectorImpl.doSelect(DevPollSelectorImpl.java:68)
    sun.nio.ch.SelectorImpl.lockAndDoSelect(SelectorImpl.java:69)
TRACE 300001:
    java.net.PlainSocketImpl.socketAccept(PlainSocketImpl.java:Unknown line)
    java.net.PlainSocketImpl.accept(PlainSocketImpl.java:384)
    java.net.ServerSocket.implAccept(ServerSocket.java:450)
    java.net.ServerSocket.accept(ServerSocket.java:421)
TRACE 300003:
    java.net.PlainDatagramSocketImpl.receive0(PlainDatagramSocketImpl.java:Unknown line)
    java.net.PlainDatagramSocketImpl.receive(PlainDatagramSocketImpl.java:136)
    java.net.DatagramSocket.receive(DatagramSocket.java:712)
    net.jxta.impl.endpoint.mcast.McastTransport.run(McastTransport.java:579)
TRACE 300006:
    java.net.SocketInputStream.socketRead0(SocketInputStream.java:Unknown line)
    java.net.SocketInputStream.read(SocketInputStream.java:129)
    com.sun.net.ssl.internal.ssl.InputRecord.readFully(InputRecord.java:293)
    com.sun.net.ssl.internal.ssl.InputRecord.read(InputRecord.java:331)

```

スタックトレースの形式は、以下の形式をしています。

[スタックトレースの形式]

スタックトレース番号：  
 メソッド名情報1（ソース名：行番号）  
 メソッド名情報2（ソース名：行番号）  
 メソッド名情報3（ソース名：行番号）  
 ……  
 メソッド名情報N（ソース名：行番号）

[スタックトレースを構成する各データの説明]

スタック トレース番号	スタックトレースごとに付けられた番号です。この番号は、サンプリング情報の“trace”項目と一致します。番号の前に“TRACE”という文字列が付きます。
メソッド名情 報1～N	スタックトレースを構成するメソッド名です。
ソース名	メソッド名に示されたメソッドがプログラミングされたjavaのソースコードファイル名です。ファイル名は、“SocketInputStream.java”などと示された部分です。
行番号	ソース名に示されたソースコードファイル中、そのメソッドが記述された行番号です。行番号は、ソース名の後ろで“:”で区切られています。例中の“129”などと示された部分です。行番号が不明な場合“Unknown line”と示されます。

・ サンプリング情報

例の中で、次の部分を示します。

```

CPU SAMPLES BEGIN (total = 9268) Tue Jul 29 11:35:17 2008
rank self accum count trace method
1 39.15% 39.15% 3628 300004 java.net.SocketInputStream.socketRead0
2 25.47% 64.62% 2361 300002 sun.nio.ch.DevPollArrayWrapper.pollIO
3 25.47% 90.09% 2361 300001 java.net.PlainSocketImpl.socketAccept
4 8.49% 98.59% 787 300003 java.net.PlainDatagramSocketImpl.receive0
5 0.83% 99.42% 77 300006 java.net.SocketInputStream.socketRead0
CPU SAMPLES END
  
```

サンプリング期間にメソッドがサンプリングされた回数のランクを示した情報です。

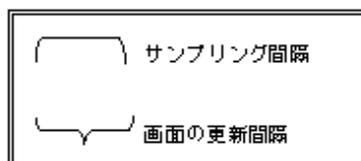
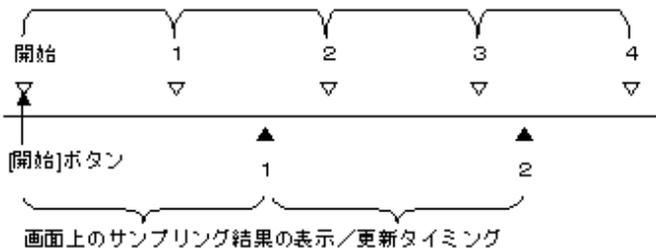
出力される情報は、サンプリング期間内において、サンプリング数の比率が0.1%以上のものです。



サンプリング期間について

サンプリング期間とは、メソッドサンプリングの結果として画面に表示される情報の対象範囲のことです。

メソッドサンプリング



上の図で、1回目(▲1)に画面に表示したデータのサンプリング期間は、メソッドサンプリング開始から1回目のサンプリング(▽1)までが相当します。

同様に、2回目(▲2)のサンプリング期間は、メソッドサンプリング開始から3回目のサンプリング(▽3)までが相当します。

サンプリング情報は、以下の形式をしています。

[サンプリング情報の形式]

開始見出し データ項目名 サンプリング情報1 サンプリング情報2 : サンプリング情報N 終了見出し
--

[サンプリング情報を構成する各データの説明]

開始見出し	サンプリング情報開始を示す行です。 “CPU SAMPLES BEGIN (total = %d) %DATE”の形式で示されます。 “CPU SAMPLES BEGIN”は固定の文字列です。 “%d”と“%DATE”は、可変の情報です。 <ul style="list-style-type: none"> <li>• “%d”: 総サンプリング数を示します。</li> <li>• “%DATE”: “Tue Jul 29 11:35:17 2008”のように“曜日 月 日 時刻 年”の形でサンプリングを行った日時の情報を示します。</li> </ul>
データ項目名	サンプリング情報のデータ項目の並びを示しています。 “rank”, “self”, “accum”, “count”, “trace”, “method” の順にサンプリング情報のデータが並んでいることを示しています。各データ項目の意味は、“データ項目の説明”を参照してください。
サンプリング情報1～N	サンプリングされたメソッドごとのサンプリング結果です。 各行のデータの並びは、データ項目名に示した順です。 各行は“count”項目の数値の大きい順(サンプリング期間内にサンプリングされた回数が多かった順)に示されます。
終了見出し	サンプリング情報の終わりを示す行です。 “CPU SAMPLES END”固定の文字列です。

[データ項目の説明]

データ項目名	意味
Rank	サンプリング期間内に該当メソッド(“method”項目が示すメソッド)がサンプリングされた回数の順位です。 サンプリングされた回数の多い順に付けられます。
Self	総サンプリング数に対する、該当メソッド(“method”項目が示すメソッド)のサンプリング数の比率です。
Accum	総サンプリング数に対する、サンプリング数の累積比率です。 該当メソッドの“self”の値と上位“rank”の“accum”の値の合計です。
Count	サンプリング期間内に該当メソッド(“method”項目が示すメソッド)がサンプリングされた回数です。
Trace	該当メソッド(“method”項目が示すメソッド)に対応するスタックトレースの番号です。
Method	サンプリングされたメソッド名です。

## ◆ファイル保存されたCSV形式

メソッドサンプリングの結果を[CSV形式で保存する]ボタンを押してファイルへ保存した、ファイル形式の説明をします。

メソッドサンプリングをファイルへ保存した場合は、ファイルの内容は画面に見えている現在のメソッドサンプリングの結果をカンマ(“,”)区切りしたもので、[スタックトレース]と[サンプリング情報]の2種類のデータで構成されます。

- スタックトレース

たとえば、画面上、次のスタックトレースが表示されていたとします。

```
TRACE 300004:
    java.net.SocketInputStream.socketRead0(SocketInputStream.java:Unknown line)
    java.net.SocketInputStream.read(SocketInputStream.java:129)
    java.io.BufferedInputStream.fill(BufferedInputStream.java:218)
    java.io.BufferedInputStream.read(BufferedInputStream.java:235)
TRACE 300002:
    sun.nio.ch.DevPollArrayWrapper.poll0(DevPollArrayWrapper.java:Unknown line)
    sun.nio.ch.DevPollArrayWrapper.poll(DevPollArrayWrapper.java:164)
    sun.nio.ch.DevPollSelectorImpl.doSelect(DevPollSelectorImpl.java:68)
    sun.nio.ch.SelectorImpl.lockAndDoSelect(SelectorImpl.java:69)
TRACE 300001:
    java.net.PlainSocketImpl.socketAccept(PlainSocketImpl.java:Unknown line)
    java.net.PlainSocketImpl.accept(PlainSocketImpl.java:384)
    java.net.ServerSocket.implAccept(ServerSocket.java:450)
    java.net.ServerSocket.accept(ServerSocket.java:421)
TRACE 300003:
    java.net.PlainDatagramSocketImpl.receive0(PlainDatagramSocketImpl.java:Unknown line)
    java.net.PlainDatagramSocketImpl.receive(PlainDatagramSocketImpl.java:136)
    java.net.DatagramSocket.receive(DatagramSocket.java:712)
    net.jxta.impl.endpoint.mcast.McastTransport.run(McastTransport.java:579)
TRACE 300006:
    java.net.SocketInputStream.socketRead0(SocketInputStream.java:Unknown line)
    java.net.SocketInputStream.read(SocketInputStream.java:129)
    com.sun.net.ssl.internal.ssl.InputRecord.readFully(InputRecord.java:293)
    com.sun.net.ssl.internal.ssl.InputRecord.read(InputRecord.java:331)
```

この場合、次のようなデータになります。

```
"TRACE", "300004:"
, "java.net.SocketInputStream.socketRead0(SocketInputStream.java:Unknown line)"
, "java.net.SocketInputStream.read(SocketInputStream.java:129)"
, "java.io.BufferedInputStream.fill(BufferedInputStream.java:218)"
, "java.io.BufferedInputStream.read(BufferedInputStream.java:235)"
"TRACE", "300002:"
, "sun.nio.ch.DevPollArrayWrapper.poll0(DevPollArrayWrapper.java:Unknown line)"
, "sun.nio.ch.DevPollArrayWrapper.poll(DevPollArrayWrapper.java:164)"
, "sun.nio.ch.DevPollSelectorImpl.doSelect(DevPollSelectorImpl.java:68)"
, "sun.nio.ch.SelectorImpl.lockAndDoSelect(SelectorImpl.java:69)"
"TRACE", "300001:"
, "java.net.PlainSocketImpl.socketAccept(PlainSocketImpl.java:Unknown line)"
, "java.net.PlainSocketImpl.accept(PlainSocketImpl.java:384)"
, "java.net.ServerSocket.implAccept(ServerSocket.java:450)"
, "java.net.ServerSocket.accept(ServerSocket.java:421)"
"TRACE", "300003:"
, "java.net.PlainDatagramSocketImpl.receive0(PlainDatagramSocketImpl.java:Unknown line)"
, "java.net.PlainDatagramSocketImpl.receive(PlainDatagramSocketImpl.java:136)"
, "java.net.DatagramSocket.receive(DatagramSocket.java:712)"
, "net.jxta.impl.endpoint.mcast.McastTransport.run(McastTransport.java:579)"
"TRACE", "300006:"
, "java.net.SocketInputStream.socketRead0(SocketInputStream.java:Unknown line)"
, "java.net.SocketInputStream.read(SocketInputStream.java:129)"
, "com.sun.net.ssl.internal.ssl.InputRecord.readFully(InputRecord.java:293)"
, "com.sun.net.ssl.internal.ssl.InputRecord.read(InputRecord.java:331)"
```

- ・ サンプルング情報

たとえば、画面上、次のサンプルング情報が表示されていたとします。

```
CPU SAMPLES BEGIN (total = 9268) Tue Jul 29 11:35:17 2008
rank self accum count trace method
 1 39.15% 39.15% 3628 300004 java.net.SocketInputStream.socketRead0
 2 25.47% 64.62% 2361 300002 sun.nio.ch.DevPollArrayWrapper.poll0
 3 25.47% 90.09% 2361 300001 java.net.PlainSocketImpl.socketAccept
 4 8.49% 98.59% 787 300003 java.net.PlainDatagramSocketImpl.receive0
 5 0.83% 99.42% 77 300006 java.net.SocketInputStream.socketRead0
CPU SAMPLES END
```

この場合、次のようなデータになります。

```
"CPU SAMPLES BEGIN (total = 9268) Tue Jul 29 11:35:17 2008"
"rank", "self", "accum", "count", "trace", "method"
"1", "39.15%", "39.15%", "3628", "300004", "java.net.SocketInputStream.socketRead0"
"2", "25.47%", "62.62%", "2361", "300002", "sun.nio.ch.DevPollArrayWrapper.poll0"
"3", "25.47%", "90.09%", "2361", "300001", "java.net.PlainSocketImpl.socketAccept"
"4", "8.49%", "98.59%", "787", "300003", "java.net.PlainDatagramSocketImpl.receive0"
"5", "0.83%", "99.42%", "77", "300006", "java.net.SocketInputStream.socketRead0"
"CPU SAMPLES END"
```

各データ項目の意味は、「◆画面表示されたデータ形式」を参照してください。

### ◆ロギングファイル

ロギング機能を利用してメソッドサンプルングの結果を保存した際のロギングファイルの説明をします。

ロギングファイルは、次のファイル名の命名規約により決定し、作成されます。

[ロギングファイル名]

```
${ログファイル格納ディレクトリ}¥sampling_${hostname}_${pid}_${date-time}.log
```

[ロギングファイル名の説明]

\${ログファイル格納ディレクトリ}	ログファイルが格納されるディレクトリです。 “設定”ダイアログボックスの“ロギング”>“ディレクトリ”で指定したディレクトリです。
sampling	メソッドサンプルングのロギングファイルであることを示す、接頭文字列です。 固定の文字列“sampling”を用います。
_	固有文字列とホスト名の上にセパレータ文字として“_”が入ります。
\${hostname}	監視対象としているJavaプロセスが動作しているホスト名です。 不明な場合は“unknown”と示されます。
_	ホスト名とプロセスIDの上にセパレータ文字として“_”が入ります。
\${pid}	監視対象としているJavaプロセスのプロセスIDです。
_	プロセスIDとロギングファイルの作成日の間にセパレータ文字として“_”が入ります。
\${data-time}	ロギングファイルの作成日時です。“yyyyMMDDhhmmss”の形式をしています。  yyyy-西暦(4桁で示します) MM-月(2桁で示します。1月から9月までは“08”のように0を入れます) DD-日(2桁で示します。1日から9日までは“04”のように0を入れます) hh-時(24時間系で示します。0時から9時までは“01”のように0を入れます) mm-分(2桁で示します。0分から9分までは“09”のように0を入れます) ss-秒(2桁で示します。0秒から9秒までは“07”のように0を入れます)
.log	固定文字列です。

ロギングファイルは、メソッドサンプリングの結果書き込みを行う前にファイルサイズの確認を行います。結果の書き込みによりファイルサイズが、2Gバイトを超えることを検出すると新たにファイルを作成し、新しいファイルに結果の書き込みを行います。(このファイル切り替えをファイルローテーションと呼びます)

ファイルローテーションが起こった場合、ロギングファイルの世代をファイル名中の“\${data-time}”の部分で知ることができます。



ロギングのタイミング

注意

メソッドサンプリング結果のロギングは、[開始]ボタンから[停止]ボタンまでの間、画面にメソッドサンプリング結果が表示されるごとにファイルへ追書きします。[一時停止]ボタンを押している間は、記録の追書きは行われません。

## ■オブジェクトヒストグラム

ヒープ分析のオブジェクトヒストグラム機能を操作した際に画面(結果の表示域)に表示されるデータの形式とファイル出力したCSV形式について説明します。

### ◆画面表示されたデータ形式

オブジェクトヒストグラムの結果の表示例を以下に示します。

増減[バイト]	増減[個数]	クラス名
開始時刻	Wed Jul 30 10:53:24 JST 2008	
終了時刻	Wed Jul 30 10:58:07 JST 2008	
411320	42	char[]
298448	293	byte[]
19968	48	sun.misc.CacheEntry[]
9224	23	<methodDataKlass>
2880	90	java.lang.ref.Finalizer
2304	36	sun.security.provider.MD5
1632	12	com.sun.net.ssl.internal.ssl.SSLSocketImpl
1536	24	sun.security.pkcs11.P11Cipher
1344	24	javax.crypto.Cipher
1152	48	sun.misc.Cache
1152	48	java.security.MessageDigest\$Delegate
1152	28	java.util.HashMap\$Entry[]
1152	24	sun.security.pkcs11.P11Key\$P11SecretKey
1152	24	com.sun.net.ssl.internal.ssl.MAC
1144	13	com.sun.net.ssl.internal.ssl.SSLSessionImpl
1120	28	java.util.HashMap
1080	45	java.lang.String
1080	9	java.net.SocksSocketImpl

ダンプを取得する

オブジェクトヒストグラムは、Javaヒープ上のオブジェクトを、クラス名ごとにサイズおよび個数の総和を開始点と終了点それぞれで求め、終了点の総和から開始点の総和を引いた値を示したものです。

開始点および終了点は、“開始時刻”([開始]ボタンを押したとき)と“終了時刻”([終了]ボタンを押したとき)に示された時刻になります。

オブジェクトヒストグラムの結果は、次の順番でソートされ表示されます。

1. 終了点の総和から開始点の総和を引いたオブジェクトサイズが大きなもの
2. 1の条件が同値のものが存在した場合、個数の大きなもの
3. 2の条件が同値のものが存在した場合、オブジェクトヒストグラム検出順

オブジェクトヒストグラムの結果は、次のデータで構成されます。

開始時刻	オブジェクトヒストグラムの情報収集を開始した時刻です。 “Wed Jul 30 10:53:24 JST 2008”のように“曜日 月 日 時刻 JST 年”の形でオブジェクトヒストグラムの情報収集を開始した日時の情報を示します。
終了時刻	オブジェクトヒストグラムの情報収集を終了した時刻です。 “Wed Jul 30 10:58:07 JST 2008”のように“曜日 月 日 時刻 JST 年”の形でオブジェクトヒストグラムの情報収集を終了した日時の情報を示します。
データ項目名	オブジェクトヒストグラムの結果に表示されているデータの項目名を示した行です。 データは、“増減[バイト]”、“増減[個数]”、“クラス名”の順であることを示しています。
結果1～N	終了点から開始点を引いた結果です。 サイズ(“増減[バイト]”)の大きなものから順に並べた結果です。

#### ◆ファイル保存されたCSV形式

オブジェクトヒストグラムの結果を[CSV形式で保存する]ボタンを押してファイルへ保存した、各行の説明をします。

オブジェクトヒストグラムをファイルへ保存した場合は、ファイルの内容は画面に見えている現在のオブジェクトヒストグラムの結果をカンマ(“,”)区切りしたものです。

たとえば、画面上、次のオブジェクトヒストグラムの結果が表示されていたとします。

開始時刻	Wed Jul 30 10:53:24 JST 2008	
終了時刻	Wed Jul 30 10:58:07 JST 2008	
増減[バイト]	増減[個数]	クラス名
411320	42	char[]
298448	293	byte[]
19968	48	sun.misc.CacheEntry[]
9224	23	<methodDataKlass>
2880	90	java.lang.ref.Finalizer
2304	36	sun.security.provider.MD5
1632	12	com.sun.net.ssl.internal.ssl.SSLSocketImpl
1536	24	sun.security.pkcs11.P11Cipher
1344	24	javax.crypto.Cipher
1152	48	sun.misc.Cache
1152	48	java.security.MessageDigest\$Delegate
1152	28	java.util.HashMap\$Entry[]
1152	24	sun.security.pkcs11.P11Key\$P11SecretKey
1152	24	com.sun.net.ssl.internal.ssl.MAC
1144	13	com.sun.net.ssl.internal.ssl.SSLSessionImpl
1120	28	java.util.HashMap
1080	45	java.lang.String
1080	9	java.net.SocksSocketImpl

ダンプを取得する

この場合、次のようなデータになります。

```

“開始時刻”, “Wed Jul 30 10:53:24 JST 2008”
“終了時刻”, “Wed Jul 30 10:58:07 JST 2008”
“増減[バイト]”, “増減[個数]”, “クラス名”
“411320”, “42”, “char []”
“298448”, “293”, “byte []”
“19968”, “48”, “sun. misc. CacheEntry []”
“9224”, “23”, “<methodDataKlass>”
“2880”, “90”, “java. lang. ref. Finalizer”
“2304”, “36”, “sun. security. provider. MD5”
“1632”, “12”, “com. sun. net. ssl. internal. ssl. SSLSocketImpl”
“1536”, “24”, “sun. security. pkcs11. P11Cipher”
“1344”, “24”, “javax. crypto. Cipher”
“1152”, “48”, “sun. misc. Cache”
“1152”, “48”, “java. security. MessageDigest$Delegate”

```

```

"1152", "28", "java.util.Hashtable$Entry[]"
"1152", "24", "sun.security.pkcs11.P11Key$P11SecretKey"
"1152", "24", "com.sun.net.ssl.internal.ssl.MAC"
"1144", "13", "com.sun.net.ssl.internal.ssl.SSLSessionImpl"
"1120", "28", "java.util.HashMap"
"1080", "45", "java.lang.String"
"1080", "9", "java.net.SocksSocketImpl"
:
[以下、省略]
:

```

各データ項目の意味は、“◆画面表示されたデータ形式”を参照してください。

## ◆ヒープダンプ

ヒープ分析の[ダンプを取得する]ボタンを押した場合、Java監視機能が測定対象としているコンテナが動作しているアプリケーションサーバ上に、ヒープダンプと呼ばれる現在のJavaヒープのイメージをファイルとして採取します。

このヒープダンプのファイルは、次のファイルパスおよびファイル名の命名規約により決定し、作成されます。

[ヒープダンプファイルのファイルパス]

```
/${tmpdir}/fjvm-heapdump.${pid}.${data-time}
```

[ヒープダンプファイル名の説明]

/\${tmpdir}	Java監視機能のヒープダンプ出力用フォルダです。 Javaのシステムプロパティ“java.io.tmpdir”に示されるフォルダが入ります。
fjvm-heapdump	ファイル名の接頭文字列です。“fjvm-heapdump”固有の文字列を用います。
.	固有文字列とPIDの間にセパレータ文字として“.”(ピリオド)が入ります。
\${pid}	測定対象のサーバ上でのプロセスIDです。
.	PIDとヒープダンプ採取日時の間にセパレータ文字として“.”(ピリオド)が入ります。
\${data-time}	ヒープダンプ採取の日時です。 “MMDDhhmmss”の形式をしています。 MM-月(2桁で示します。1月から9月までは“08”のように0を入れます) DD-日(2桁で示します。1日から9日までは“04”のように0を入れます) hh-時(24時間系で示します。0時から9時までは“01”のように0を入れます) mm-分(2桁で示します。0分から9分までは“09”のように0を入れます) ss-秒(2桁で示します。0秒から9秒までは“07”のように0を入れます)

ヒープダンプが採取された場合、ヒープ分析の画面上には以下のようにファイルパスが示されます。

**ダンプを取得する**      Myscerver /var/tmp/fjvm-heapdump.4057.0730120957



ヒープダンプ採取の留意事項

注意

“ヒープ分析”画面の[ダンプを採取する]ボタンは、ヒープダンプ採取の操作を提供します。

Java監視機能の画面(“ヒープ分析”)上でヒープダンプを解析することはできません。また、ヒープダンプを採取したサーバからJava監視機能が動作している端末上に、ヒープダンプのファイルをファイル転送することもできません。

## A.5.12 メッセージ

ここではJava監視機能で表示されるメッセージとその意味や対処方法を説明します。

なお、本章に記載の無いメッセージは、jconsole本体の出力するメッセージ(オリジナルのメッセージ)です。

jconsoleのオリジナルのメッセージについては、Oracle Corporationが提供するJDK6のjconsoleコマンドのマニュアルを参照してください。



jconsoleの出力ウィンドウへのエラーメッセージ表示

参考

Java監視機能がエラーを検出すると、jconsoleの出力ウィンドウにエラーの情報を出力する場合があります。

jconsoleの出力ウィンドウには、次の形式でエラーの情報が出力されます。

```
例外名 : メッセージ
         スタックトレース
         :
         :
```

例えば、次のように表示されます。

```
java.lang.IllegalStateException: Sampling process is already running
  at com.fujitsu.jvm.management.client.ProfileConsole$1.actionPerformed(ProfileConsole.java:266)
  at javax.swing.AbstractButton.fireActionPerformed(AbstractButton.java:1995)
  at javax.swing.AbstractButton$Handler.actionPerformed(AbstractButton.java:2318)
  at javax.swing.DefaultButtonModel.fireActionPerformed(DefaultButtonModel.java:387)
  at javax.swing.DefaultButtonModel.setPressed(DefaultButtonModel.java:242)
  :
  (以下省略)
```



jconsoleの出力ウィンドウ

注意

Windows環境では、jconsoleのメッセージは、jconsoleや、jconsoleを起動したコマンドプロンプトとは異なるウィンドウに出力されます(ウィンドウが存在しない場合は、メッセージ出力と同時にウィンドウが表示されます)。

Oracle SolarisやLinux環境では、jconsoleのメッセージは、jconsoleを起動したコマンドプロンプトに出力されます。

## ■Java監視機能の起動／接続時

**MBeanサーバーとの接続がありません。**

[状態]

異常

[意味]

Java監視機能(jconsole)と監視対象Javaプロセスの接続が切れている状態で、監視対象プロセスにアクセスしようとした(ボタン等を押しました)。

[表示場所]

エラーダイアログ

[利用者の対処]

Java監視機能(jconsole)と監視対象Javaプロセスを接続します。その後、エラーが発生した操作を行ってください。

**Failed to add "%s"**

[可変情報]

%s: プラグイン(Java監視機能)のタブ名

[状態]

異常

[意味]

Java監視機能(jconsoleプラグイン)は、jconsoleへの機能(タブ)の追加に失敗しました。

追加に失敗した機能(タブ)はjconsole上に表示されません。

[表示場所]

jconsoleの出力ウィンドウ

[利用者の対処]

監視対象に指定したJavaプロセスがJava監視機能の対象となっていないことが考えられます。

正しいプロセスを指定しなおしてください。

それでも改善されない場合は、監視対象とするJavaプロセスに正しくシステムプロパティやJVMオプションが設定されていないことも考えられます。(必要な設定については、“[A.5.1 パフォーマンスデータの収集](#)”を参照してください。)

監視対象のJavaプロセスが動作しているアプリケーションサーバの管理者に確認をしてください。

#### Failed to set a new connection for "%s"

[可変情報]

%s: プラグイン (Java監視機能) のタブ名

[状態]

異常

[意味]

jconsoleと監視対象Javaプロセスが接続した際、各機能(タブ)の内部データ更新に失敗しました。

メッセージと同時に例外発生時のスタックトレース情報を、jconsoleの出力ウィンドウに出力します。

[表示場所]

jconsoleの出力ウィンドウ

[利用者の対処]

Java監視機能や監視対象Javaプロセスの内部異常が発生し、動作が不安定になっていることが考えられます。Java監視機能、監視対象プロセスの両方を再起動してください。

## ■メソッドサンプリング

#### Cannot pause the sampling process when it is not running

[状態]

異常

[意味]

メソッドサンプリング機能処理中に問題が発生しました。

メッセージと同時に例外発生時のスタックトレース情報を、jconsoleの出力ウィンドウに出力します。

[表示場所]

jconsoleの出力ウィンドウ

[利用者の対処]

複数のjconsoleを利用している可能性があります。すべてのjconsoleを終了し、再度ひとつだけjconsoleを起動してください。

#### Cannot stop the sampling process when it is not running

[状態]

異常

[意味]

メソッドサンプリング機能処理中に問題が発生しました。

メッセージと同時に例外発生時のスタックトレース情報を、jconsoleの出力ウィンドウに出力します。

[表示場所]

jconsoleの出力ウィンドウ

[利用者の対処]

複数のjconsoleを利用している可能性があります。すべてのjconsoleを終了し、再度ひとつだけjconsoleを起動してください。

#### Exception while checking sampling state

[状態]

異常

[意味]

メソッドサンプリング機能処理中に問題が発生しました。

メッセージと同時に例外発生時のスタックトレース情報を、jconsoleの出力ウィンドウに出力します。

[表示場所]

jconsoleの出力ウィンドウ

[利用者の対処]

Java監視機能や監視対象Javaプロセスの内部異常が発生し、動作が不安定になっていることが考えられます。Java監視機能、監視対象Javaプロセスの両方を再起動してください。

#### Interval cannot be change while sampling process is running

[状態]

異常

[意味]

メソッドサンプリング機能のサンプリング間隔を変更処理中に、問題が発生しました。

メッセージと同時に例外発生時のスタックトレース情報を、jconsoleの出力ウィンドウに出力します。

[表示場所]

jconsoleの出力ウィンドウ

[利用者の対処]

複数のjconsoleを利用している可能性があります。すべてのjconsoleを終了し、再度ひとつだけjconsoleを起動してください。

#### Failed to set interval

[状態]

異常

[意味]

メソッドサンプリング機能のサンプリング間隔を変更する最中に、問題が発生しました。

メッセージと同時に例外発生時のスタックトレース情報を、jconsoleの出力ウィンドウに出力します。

[表示場所]

jconsoleの出力ウィンドウ

[利用者の対処]

Java監視機能や監視対象Javaプロセスの内部異常が発生し、動作が不安定になっていることが考えられます。Java監視機能、監視対象のJavaプロセスの両方を再起動してください。

#### Sampling process is already running

[状態]

異常

[意味]

メソッドサンプリング機能の開始処理中に問題が発生しました。

メッセージと同時に例外発生時のスタックトレース情報を、jconsoleの出力ウィンドウに出力します。

[表示場所]

jconsoleの出力ウィンドウ

[利用者の対処]

複数のjconsoleを利用している可能性があります。すべてのjconsoleを終了し、再度ひとつだけjconsoleを起動してください。

### Sampling process is not running

[状態]

異常

[意味]

メソッドサンプリング機能を停止している途中で問題が発生しました。

メッセージと同時に例外発生時のスタックトレース情報を、jconsoleの出力ウィンドウに出力します。

[表示場所]

jconsoleの出力ウィンドウ

[利用者の対処]

複数のjconsoleを利用している可能性があります。すべてのjconsoleを終了し、再度ひとつだけjconsoleを起動してください。

### Connection is not established

[状態]

異常

[意味]

メソッドサンプリング機能の処理中に問題が発生しました。

メッセージと同時に例外発生時のスタックトレース情報を、jconsoleの出力ウィンドウに出力します。

[表示場所]

jconsoleの出力ウィンドウ

[利用者の対処]

Java監視機能や監視対象のJavaプロセスに内部異常が発生し、動作が不安定になっていることが考えられます。Java監視機能、監視対象のJavaプロセスの両方を再起動してください。

### Failed to set stack trace depth

[状態]

異常

[意味]

スタックトレースの深さを変更する処理中に問題が発生しました。

メッセージと同時に例外発生時のスタックトレース情報を、jconsoleの出力ウィンドウに出力します。

[表示場所]

jconsoleの出力ウィンドウ

[利用者の対処]

Java監視機能や監視対象のJavaプロセスに内部異常が発生し、動作が不安定になっていることが考えられます。Java監視機能、監視対象のJavaプロセスの両方を再起動してください。

## StackTraceDepth cannot be changed while sampling process is running

[状態]

異常

[意味]

メソッドサンプリング機能の処理中に問題が発生しました。

メッセージと同時に例外発生時のスタックトレース情報を、jconsoleの出力ウィンドウに出力します。

[表示場所]

jconsoleの出力ウィンドウ

[利用者の対処]

Java監視機能や監視対象のJavaプロセスで内部異常が発生し、動作が不安定になっていることが考えられます。Java監視機能、監視対象のJavaプロセスの両方を再起動してください。

## ログファイル名が競合しました。

[状態]

異常

[意味]

ログファイル名の競合が発生し、ファイルを生成することができませんでした。

次のいずれかの状態になっていることが考えられます。

- ログギングを有効にした状態で、メソッドサンプリングの開始ボタンを押しました。
- ログギングの最中に、ログファイルのサイズが規定値をオーバーしたため、Java監視機能が新しいログファイルを生成しようとした。

[表示場所]

jconsoleの出力ウィンドウ

[利用者の対処]

メソッドサンプリングの開始時に本メッセージが出力された場合、メソッドサンプリングは開始されますが、ログギングは行われません。またメソッドサンプリング中に本メッセージが出力された場合、メソッドサンプリングは継続しますが、ログギングは停止します。ログファイルを出力するディレクトリを見直してください。

## ログファイルの生成に失敗しました。

[状態]

異常

[意味]

ログファイル名の競合が発生し、ファイルを生成することができませんでした。

次のいずれかの状態になっていることが考えられます。

- ログギングを有効にした状態で、メソッドサンプリングの開始ボタンを押しました。
- ログギングの最中に、ログファイルのサイズが規定値をオーバーしたため、Java監視機能が新しいログファイルを生成しようとした。

[表示場所]

jconsoleの出力ウィンドウ

[利用者の対処]

メソッドサンプリングの開始時に本メッセージが出力された場合、メソッドサンプリングは開始されますが、ログギングは行われません。またメソッドサンプリング中に本メッセージが出力された場合、メソッドサンプリングは継続しますが、ログギングは停止します。ログファイルを出力するディレクトリを見直してください。

#### サンプリングデータの出力に失敗しました。

[状態]

異常

[意味]

ログファイルにデータを書き込み中、問題(ディスク容量の不足など)が発生しました。

[表示場所]

jconsoleの出力ウィンドウ

[利用者の対処]

メソッドサンプリングは継続しますが、ロギングは停止します。jconsoleの出力ウィンドウに表示される例外の情報を参照し、問題の原因を取り除いてください。

#### サンプリング中に、その操作は実行できません。

[状態]

異常

[意味]

メソッドサンプリングの動作中に許可されていない操作を実行しました。

[表示場所]

jconsoleの出力ウィンドウ

[利用者の対処]

複数のjconsoleを利用している可能性があります。すべてのjconsoleを終了し、ひとつだけjconsoleを再起動してください。

#### 入力値が範囲を超えています。値は、1以上65535以下でなければなりません。

[状態]

異常

[意味]

“設定”ダイアログボックスの“トレースの深さ”に入力したスタックトレースの深さが規定値の範囲外です。

[表示場所]

エラーダイアログ

[利用者の対処]

スタックトレースの深さを、1以上65535以下の整数値を入力してください。

#### 入力値が不正です。

[状態]

異常

[意味]

“設定”ダイアログボックスの“トレースの深さ”に入力したスタックトレースの深さが規定値の範囲外です。

[表示場所]

エラーダイアログ

[利用者の対処]

スタックトレースの深さを、1以上65535以下の整数値で入力してください。

#### 入力値が空です。

[状態]

異常

[意味]

“設定”ダイアログボックスの“ディレクトリ”の指定がありません。

[表示場所]

エラーダイアログ

[利用者の対処]

ロギングを有効にするときは、ログファイルを生成するディレクトリも指定する必要があります。“設定”ダイアログボックスの“ディレクトリ”にログファイルを作成するディレクトリを指定してください。

#### ディレクトリの作成に失敗しました。

[状態]

異常

[意味]

“設定”ダイアログボックスの“ディレクトリ”に指定された、ディレクトリの作成に失敗しました。

[表示場所]

エラーダイアログ

[利用者の対処]

不明なドライブの指定、アクセス権を認められていないディレクトリの指定などが考えられます。“設定”ダイアログボックスの“ディレクトリ”に指定したディレクトリを見直してください。

#### サンプリングを開始できませんでした。

[状態]

異常

[意味]

メソッドサンプリングを開始することができませんでした。

[表示場所]

エラーダイアログ

[利用者の対処]

Java監視機能や監視対象のJavaプロセスの内部異常が発生し、動作が不安定になっていることが考えられます。Java監視機能、監視対象のJavaプロセスの両方を再起動してください。

#### サンプリングを終了できませんでした。

[状態]

異常

[意味]

メソッドサンプリングを終了することができませんでした。

[表示場所]

エラーダイアログ

[利用者の対処]

Java監視機能や監視対象のJavaプロセスの内部異常が発生し、動作が不安定になっていることが考えられます。Java監視機能、監視対象のJavaプロセスの両方を再起動してください。

## ■ヒープ分析

ヒープ分析を開始しました。終了ボタンを押すと分析結果を表示します。

[状態]

正常

[意味]

オブジェクトヒストグラム(ヒープ分析)を開始しました。

[表示場所]

オブジェクトヒストグラムの表示域

[利用者の対処]

正常にオブジェクトヒストグラムの採取が開始されました。任意の時点で[終了]ボタンを押し、ヒープ分析を終了してください。

#### キャンセルしました。

[状態]

正常

[意味]

オブジェクトヒストグラム(ヒープ分析)の取得をキャンセル(中止)しました。

[表示場所]

オブジェクトヒストグラムの表示域

[利用者の対処]

正常にオブジェクトヒストグラムのキャンセル処理がされました。

#### ヒストグラムの取得に失敗しました。

[状態]

異常

[意味]

オブジェクトヒストグラムの採取に失敗しました。

開始または終了の処理で失敗したことが考えられます。

[表示場所]

オブジェクトヒストグラムの表示域

[利用者の対処]

再度オブジェクトヒストグラムの採取を行ってください。

#### ヒストグラムの取得に失敗しました。( %s)

[可変情報]

%s: 詳細メッセージ

[状態]

異常

[意味]

オブジェクトヒストグラムの採取に失敗しました。

開始または終了の処理で失敗したことが考えられます。

[表示場所]

オブジェクトヒストグラムの表示域

[利用者の対処]

再度オブジェクトヒストグラムの採取を行ってください。

#### 取得に失敗しました。( %s)

##### [可変情報]

%s: 失敗した原因を示す詳細メッセージ

##### [状態]

異常

##### [意味]

ヒープダンプの採取に失敗しました。

##### [表示場所]

ヒープダンプファイル名の表示域([ダンプを取得する]ボタンの横)

##### [利用者の対処]

詳細メッセージを参照し、失敗した原因を取り除いた後、再度ヒープダンプの採取を行ってください。

詳細メッセージに「詳細は、サーバのログで確認してください。」と表示されている場合は、Java監視機能が何らかの理由で詳細メッセージを取得できませんでした。この場合、サーバのコンテナログに採取された詳細メッセージを参照してください。

#### ダンプファイルが見つかりません。

##### [状態]

異常

##### [意味]

ヒープダンプの採取には成功しましたが、結果を保存したファイルの確認でそのファイルを見つけることができませんでした。

アプリケーションサーバ側に格納したヒープダンプファイルが削除されているか、格納先の空き領域が無い場合が考えられます。

##### [表示場所]

ヒープダンプファイル名の表示域([ダンプを取得する]ボタンの横)

##### [利用者の対処]

再度ヒープダンプの採取を行ってください。

#### Failed to get the last dumpfile name: %s

##### [可変情報]

%s: 例外のメッセージ

##### [状態]

異常

##### [意味]

ヒープダンプの採取には成功しましたが、結果を保存したファイルの確認でそのファイルを見つけることができませんでした。

アプリケーションサーバ側に格納したヒープダンプファイルが削除されているか、格納先の空き領域が無い場合が考えられます。

##### [表示場所]

jconsoleの出力ウィンドウ

##### [利用者の対処]

再度ヒープダンプの採取を行ってください。

#### Failed to retrieve the cached histogram data

##### [状態]

異常

[意味]

監視対象Javaプロセスから、ヒープ分析のデータを取り出す中に問題が発生しました。

メッセージと同時に例外発生時のスタックトレース情報を、jconsoleの出力ウィンドウに出力します。

[表示場所]

jconsoleの出力ウィンドウ

[利用者の対処]

再度ヒープダンプの採取を行ってください。

#### Failed to get the time when histogram data had been cached

[状態]

異常

[意味]

監視対象Javaプロセスから、ヒープ分析のデータを取り出す中に問題が発生しました。

メッセージと同時に例外発生時のスタックトレース情報を、jconsoleの出力ウィンドウに出力します。

[表示場所]

jconsoleの出力ウィンドウ

[利用者の対処]

Java監視機能や監視対象のJavaプロセスで内部異常が発生し、動作が不安定になっていることが考えられます。Java監視機能、監視対象のプロセス両方を再起動してください。

#### Failed to clear the cached data

[状態]

異常

[意味]

ヒープ分析を終了またはキャンセルする際、内部的に一時保存しておいたデータの破棄に失敗しました。

メッセージと同時に例外発生時のスタックトレース情報を、jconsoleの出力ウィンドウに出力します。

[表示場所]

jconsoleの出力ウィンドウ

[利用者の対処]

Java監視機能や監視対象のJavaプロセスで内部異常が発生し、動作が不安定になっていることが考えられます。Java監視機能、監視対象のJavaプロセスの両方を再起動してください。

#### Cache is null

[状態]

異常

[意味]

ヒープ分析のデータを取得する際に問題が発生しました。

メッセージと同時に例外発生時のスタックトレース情報を、jconsoleの出力ウィンドウに出力します。

[表示場所]

jconsoleの出力ウィンドウ

[利用者の対処]

複数のjconsoleを利用している可能性があります。すべてのjconsoleを終了し、再度ひとつだけjconsoleを起動してください。

### Negative value cannot be used for calculation

[状態]

異常

[意味]

ヒープ分析のデータを取得する際に問題が発生しました。

メッセージと同時に例外発生時のスタックトレース情報を、jconsoleの出力ウィンドウに出力します。

[表示場所]

jconsoleの出力ウィンドウ

[利用者の対処]

ヒープ分析のため取得したデータ内に不正なデータを検出しました。監視対象のJavaプロセスで内部矛盾が発生している可能性があります。監視対象のJavaプロセスを再起動してください。

### Cache does not exist

[状態]

異常

[意味]

ヒープ分析のデータを取得する際に問題が発生しました。

メッセージと同時に例外発生時のスタックトレース情報を、jconsoleの出力ウィンドウに出力します。

[表示場所]

jconsoleの出力ウィンドウ

[利用者の対処]

複数のjconsoleを利用している可能性があります。すべてのjconsoleを終了し、再度ひとつだけjconsoleを起動してください。

## ■メソッドサンプリング/ヒープ分析共通

### セーブ可能なデータがありません。

[状態]

異常

[意味]

メソッドサンプリング結果(メソッドサンプリング画面)またはオブジェクトヒストグラム(ヒープ分析画面)のデータ表示がされていない状態で[CSV形式で保存する]ボタンを押しました。

[表示場所]

エラーダイアログ

[利用者の対処]

メソッドサンプリング結果(メソッドサンプリング画面)またはオブジェクトヒストグラム(ヒープ分析画面)を採取し、画面上にデータ表示された状態で[CSV形式で保存する]ボタンを押してください。

### ファイルに上書きできません。

[状態]

異常

[意味]

現在のJava監視機能の利用者に書き込み権限のないファイルに対し、上書きをしようとした。

[表示場所]

エラーダイアログ

[利用者の対処]

Java監視機能の利用者に書き込み権限のあるファイルを指定する、または新しいファイルに保存するようにしてください。

#### ファイルの書き込みに失敗しました。

[状態]

異常

[意味]

ファイルへの書き込み中に、問題が発生しました。

ファイルアクセスの失敗が考えられます。

Java監視機能は、指定されたファイルへの出力を中断しています。

[表示場所]

エラーダイアログ

[利用者の対処]

ファイルアクセスの異常の原因を取り除き、再度保存処理を実行してください。

または、失敗したディレクトリとは別のディレクトリに新しいファイルとして保存するようにしてください。

## ■VMオプション変更

### Failed to get value of "%s"

[可変情報]

%s: VMオプション名

[状態]

通知/異常

[意味]

VMオプションの値の表示を定期更新する際、VMオプションの値取得のタイミングと、jconsoleと監視対象のJavaプロセスの接続が切れるタイミングが重なったことを通知しています。

現象が何度も発生する場合は、Java監視機能や監視対象のJavaプロセスの内部異常が発生している可能性があります。

メッセージと同時に例外発生時のスタックトレース情報を、jconsoleの出力ウィンドウに出力します。

[表示場所]

jconsoleの出力ウィンドウ

[利用者の対処]

一度だけ表示された場合は、単なる通知です。再度、jconsoleと監視対象のJavaプロセスを接続してください。

その後、現象が再現する場合は、Java監視機能や監視対象のJavaプロセスで内部異常が発生し、動作が不安定になっていることが考えられます。Java監視機能、監視対象のJavaプロセスの両方を再起動してください。

### Failed to set value for "%s"

[可変情報]

%s: VMオプション名

[状態]

異常

[意味]

VMオプションの値を変更しようとした際、異常が発生したことを示しています。

メッセージと同時に例外発生時のスタックトレース情報を、jconsoleの出力ウィンドウに出力します。

[表示場所]

jconsoleの出力ウィンドウ

[利用者の対処]

Java監視機能や監視対象のJavaプロセスで内部異常が発生し、動作が不安定になっていることが考えられます。Java監視機能、監視対象のJavaプロセスの両方を再起動してください。

#### Failed to add panel for "%s"

[可変情報]

%s: VMオプションを表示・変更するパネルの名称

[状態]

異常

[意味]

VMオプションを表示・変更するパネルを生成できなかったことを示しています。

メッセージと同時に、例外発生時のスタックトレース情報を、jconsoleの出力ウィンドウに出力します。

[表示場所]

jconsoleの出力ウィンドウ

[利用者の対処]

Java監視機能や監視対象のJavaプロセスで内部異常が発生し、動作が不安定になっていることが考えられます。Java監視機能、監視対象のJavaプロセスの両方を再起動してください。

#### Failed to get print GC options type

[状態]

通知/異常

[意味]

VMオプションの値の表示を定期更新する際、VMオプションの値取得のタイミングと、jconsoleと監視対象のJavaプロセスの接続が切れるタイミングが重なったことを通知しています。

現象が何度も発生する場合は、Java監視機能や監視対象のJavaプロセスの内部異常が発生している可能性があります。

メッセージと同時に、例外発生時のスタックトレース情報を、jconsoleの出力ウィンドウに出力します。

[表示場所]

jconsoleの出力ウィンドウ

[利用者の対処]

一度だけ表示された場合は、単なる通知です。再度、jconsoleと監視対象のJavaプロセスを接続してください。

その後、現象が再現する場合は、Java監視機能や監視対象のJavaプロセスで内部異常が発生し、動作が不安定になっていることが考えられます。Java監視機能、監視対象のJavaプロセスの両方を再起動してください。

#### Failed to change GC options

[状態]

異常

[意味]

VMオプションの値を変更しようとした際、異常が発生したことを示しています。

メッセージと同時に、例外発生時のスタックトレース情報を、jconsoleの出力ウィンドウに出力します。

[表示場所]

jconsoleの出力ウィンドウ

[利用者の対処]

Java監視機能や監視対象のJavaプロセスで内部異常が発生し、動作が不安定になっていることが考えられます。Java監視機能、監視対象のJavaプロセスの両方を再起動してください。

#### Failed to set value of "%s"

[可変情報]

%s: VMオプション名

[状態]

異常

[意味]

VMオプションの値を変更しようとした際、異常が発生したことを示しています。メッセージと同時に、例外発生時のスタックトレース情報を、jconsoleの出力ウィンドウに出力します。

[表示場所]

jconsoleの出力ウィンドウ

[利用者の対処]

Java監視機能や監視対象のJavaプロセスで内部異常が発生し、動作が不安定になっていることが考えられます。Java監視機能、監視対象のJavaプロセスの両方を再起動してください。



## A.5.13 IJServer(J2EE)を使用する場合

IJServer(J2EE)でJava監視機能を使用する場合、以下の手順を実行します。

1. IJServerの環境設定
2. IJServerの起動
3. jconsoleからの接続

### IJServerの環境設定

#### Java VMオプションの設定

IJServerのJava VMオプションに次のオプションを設定します。WebアプリケーションとEJBアプリケーションを別JavaVMで運用するIJServer(Web + EJB[別VM])を使用している場合、Java監視機能を有効にするコンテナに対してJava VMオプションを設定してください。Servletコンテナ、EJBコンテナの双方でJava監視機能を有効にする場合、両コンテナのJava VMオプションの設定が必要です。

#### — 必須のJava VMオプション

Java監視機能を使用する場合、次の3つのJava VMオプションを設定する必要があります。

- -Dcom.fujitsu.interstage.j2ee.management.agent.start=true
- -agentlib:fjprof
- -Dcom.fujitsu.interstage.j2ee.ijserver.fjprof.port.base=<ポート番号>

jconsoleがIJServerと接続するために必要な、JMX RMI接続用のポート番号を指定します。指定できる値は、1以上65535以下の整数です。

- プロセス多重度が2以上の場合は、指定したポート番号からの連番が、各プロセスのポート番号に使用されます。
- IJServerの各プロセスに割り当てられたポート番号は、コンテナログに出力されたIJServer30008を確認してください。
- IJServerの各プロセスに割り当てられたポート番号が、1つでも他のシステム等ですでに使用されている場合、IJServerの起動に失敗します。未使用のポート番号がIJServerの各プロセスに連番で割り振られるように、値を設定しなおしてください。

- WebアプリケーションとEJBアプリケーションを別JavaVMで運用するIIServer(Web + EJB[別VM])を使用している場合、各コンテナに設定するポート番号が重複しないように注意してください。

#### 一 任意のJava VMオプション

リモート環境のIIServer(J2EE)に対してJava監視機能を有効にする場合、パスワード認証を使用することをお勧めします。パスワード認証については、“[パスワード認証を使用する場合の注意事項](#)”を参照してください。

パスワード認証は、デフォルトで有効になっています。次のシステムプロパティをIIServerのJava VMオプションに設定することで、パスワード認証の有効・無効を変更することができます。

- `-Dcom.fujitsu.interstage.j2ee.management.authenticate=<trueまたはfalse>`  
`true`: パスワード認証を有効にする (省略値)  
`false`: パスワード認証を無効にする

IIServer(J2EE)でパスワード認証を使用する場合は、次のシステムプロパティをIIServerのJava VMオプションに設定してください。

- `-Dcom.fujitsu.interstage.j2ee.management.access.file=<アクセスファイルの絶対パス>`
- `-Dcom.fujitsu.interstage.j2ee.management.password.file=<パスワードファイルの絶対パス>`

#### 一 定義例

IIServerに追加するJava VMオプションの定義例を示します。(実際は、改行せずに1行で記載してください)

##### 定義例:

###### Windows32/64

```
-Dcom.fujitsu.interstage.j2ee.management.agent.start=true
-agentlib:fjprof
-Dcom.fujitsu.interstage.j2ee.ijserver.fjprof.port.base=20000
-Dcom.fujitsu.interstage.j2ee.management.access.file=C:\jmxremote.access
-Dcom.fujitsu.interstage.j2ee.management.password.file=C:\jmxremote.password
```

###### Solaris32/64 Linux32/64

```
-Dcom.fujitsu.interstage.j2ee.management.agent.start=true
-agentlib:fjprof
-Dcom.fujitsu.interstage.j2ee.ijserver.fjprof.port.base=20000
-Dcom.fujitsu.interstage.j2ee.management.access.file=/home/ijserver/jmxremote.access
-Dcom.fujitsu.interstage.j2ee.management.password.file=/home/ijserver/jmxremote.password
```

#### パスワード認証を使用する場合の注意事項

IIServer(J2EE)でパスワード認証を使用する際に必要なアクセスファイル、パスワードファイルの注意事項を説明します。

##### 一 パスワードファイル

パスワードファイルは、jconsoleで接続を許可するユーザIDとパスワードを半角空白で区切って定義します。使用可能な文字は、Javaの標準のプロパティファイルのフォーマット (JavaのPropertiesクラスの仕様) に従います。

##### 定義例:

```
userA password00
userB password99
```

## 注意

- パスワードファイルは、**Interstage**が使用するディレクトリ(**Interstage**のインストールディレクトリ、**J2EE**共通ディレクトリなど)以外のディレクトリ配下に作成するようにしてください。これらのディレクトリ配下に置いた場合、資源のバックアップやリストア、一括情報採取ツールの実行に失敗する可能性があります。
- パスワードファイルは、アクセス権を制限する必要があります。アクセス権が正しくない場合、**IJServer**の起動ができません。次の点に注意してください。

### Windows32/64

パスワードファイルの所有者を**SYSTEM**に設定し、**SYSTEM**のみがリード権を持つようにしてください。なお、“**J2EE ユーザーズガイド**”の“ネットワーク上の共有資源へアクセスする場合の環境設定”で示している手順を実行している場合、そのとき設定したユーザを所有者に設定してください。これらの設定は、**Administrators**権限の管理者が、ファイルのプロパティのセキュリティタブから行ってください。また、以下のドキュメントを参考にしてください。

<http://docs.oracle.com/javase/jp/6/technotes/guides/management/security-windows.html>

### Solaris32/64 Linux32/64

ワークユニットの起動ユーザのみがリード権を持つように設定してください。なお**IJServer**の起動方法によって、ワークユニットの起動ユーザが異なります。

- **isstartwu**コマンドで起動する場合、コマンド投入者
- **Interstage**管理コンソールで起動する場合、**Interstage**管理コンソールのログインユーザ
- ワークユニット自動起動を有効にした**IJServer**が**Interstage**の起動に伴って起動する場合、設定した起動ユーザ

- パスワードファイルは、資源のバックアップの対象外です。資源の移行を行う場合、別途用意してください。

## 一 アクセスファイル

アクセスファイルには、**jconsole**で接続するユーザ**ID**に対するアクセス権を定義します。ユーザ**ID**とアクセス権限は半角空白で区切って定義します。ユーザ**ID**は、パスワードファイルの記述と同じ値を使用してください。アクセス権限には、以下の値を指定してください。

**readwrite**: 属性の読み取りおよび書き込み、操作の呼び出しを許可します。

定義例:

```
userA readwrite
userB readwrite
```

## 注意

- アクセスファイルは、**Interstage**が使用するディレクトリ(**Interstage**のインストールディレクトリ、**J2EE**共通ディレクトリなど)以外のディレクトリ配下に作成するようにしてください。これらのディレクトリ配下に置いた場合、資源のバックアップやリストア、一括情報採取ツールの実行に失敗する可能性があります。
- アクセスファイルは、ワークユニット起動者がリード権を持つようにしてください。パスワードファイルのように、ワークユニット起動者のみにリード権を限定する必要はありません。

### Windows32/64

**SYSTEM**がリード権を持つようにしてください。“**J2EE ユーザーズガイド**”の“ネットワーク上の共有資源へアクセスする場合の環境設定”で示している手順を実行している場合、そのとき設定したユーザがリード権を持つように設定してください。

### Solaris32/64 Linux32/64

ワークユニット起動者がリード権を持つようにしてください。

- アクセスファイルは、資源のバックアップの対象外です。資源の移行を行う場合、別途用意してください。

## IJServerの起動

Java VMオプションの設定や、パスワード認証を行う場合のファイルの準備が終了したら、IJServerを起動します。Java監視機能が有効になった場合、コンテナログにIJServer30007、IJServer30008のメッセージが出力されます。

Java監視機能の設定に誤りがある場合、IJServerの起動に失敗することがあります。また、コンテナログのIJServer30007の次の行にIJServer30008が出力されず、代わりにエラーメッセージが出力されます。エラーメッセージの内容から以下に示す原因を特定し、Java監視機能の設定を見直してください。

- Java VMオプションの設定に誤りがある
- ポート番号が重複している
- パスワード認証を使用する場合
  - パスワードファイルやアクセスファイルが存在しない、または読み取れない
  - パスワードファイルのアクセス権限が正しくない
  - パスワードファイルやアクセスファイルのフォーマットが正しくない

## jconsoleからの接続

“A.5.3 jconsoleの起動”以降で説明している手順でjconsoleを起動し、IJServerに接続してください。

### IJServer(J2EE)のポート番号の調べ方

Java監視機能で監視するIJServer(J2EE)のプロセスに接続する際のポート番号は、以下の方法で確認してください。

- IJServerの起動時にコンテナログに出力される、IJServer30008のメッセージに出力されています。
- IJServerのJava VMオプションの-Dcom.fujitsu.interstage.j2ee.ijserver.fjprof.port.baseの値に設定されています。プロセス多重度が2以上の場合は、指定されているポート番号からの連番が、各プロセスのポート番号に使用されています。

WebアプリケーションとEJBアプリケーションを別JavaVMで運用するIJServer(Web + EJB[別VM])を使用している場合、Servletコンテナ、EJBコンテナのそれぞれのJava VMオプション-Dcom.fujitsu.interstage.j2ee.ijserver.fjprof.port.baseで指定した値から連番で、各プロセスにポート番号が振られます。

たとえば、Webコンテナのプロセス多重度が2、Java VM オプションに

-Dcom.fujitsu.interstage.j2ee.ijserver.fjprof.port.base=20000を指定、EJBコンテナのプロセス多重度が3、Java VM オプションに-Dcom.fujitsu.interstage.j2ee.ijserver.fjprof.port.base=21000を指定した場合、次のようにポート番号が割り振られます。

プロセス通番	コンテナタイプ	ポート番号
1	WEB	20000
2	WEB	20001
3	EJB	21000
4	EJB	21001
5	EJB	21002

## A.6 チュートリアル

Javaアプリケーションのトラブルシューティングに関して、手順やポイントを説明します。

代表的な例として、性能ボトルネックの調査、メモリーークの分析、ハングアップやスローダウンの原因分析を取り上げます。また、Java実行環境の特徴のひとつである、ガーベジ・コレクションの分析方法についても説明します。

## A.6.1 前提となる知識

Java言語、Java VM、Java APIなど、Java関連技術の仕様についての知識を前提としています。それらに関する詳しい説明は、公開されている情報や、マニュアルなどを参照してください。

また、使用するツールの使用方法などについても、詳しく説明していません。JDKに付属しているツールの詳細については、JDKドキュメントに記載されています。「チューニングガイド」-「JDK/JREのチューニング」-「基礎知識」-「JDK関連のドキュメント」を参照してください。

Java VisualVMの詳細については、JDKドキュメントや、Java VisualVM開発者のサイトを参照してください。Java VisualVM開発者のサイトは、次のURLで公開されています。

<http://visualvm.java.net/index.html>

そのほかのツールについては、「付録A Javaツール機能」を参照してください。

## A.6.2 注意事項

本書説明では、主にJDK 6を使用しています。説明中のログや画面、アプリケーションの実行結果などは、利用者のOSやハードウェア環境、JDKのバージョンなどにより、異なることがあります。

## A.6.3 ガーベジ・コレクション(GC)について分析する

ガーベジ・コレクション(GC)に関するデータを分析して、GCにかかる時間や、Javaヒープの使用量などを調べる方法について説明します。

### A.6.3.1 GCログの採取

GCに関する重要なデータは、GCログです。GCログを参照すると、各GCの処理時間、GCの発生頻度、Javaヒープ使用量の変化などを調べることができます。GCログを出力するには、Javaの起動オプションに、次のオプションを追加してください。

```
-verbose:gc -XX:+UseFJverbose
```

図 1 GCログを出力するためのオプション

オプションを指定すると、次のようなGC発生時の情報が、標準出力に出力されます。

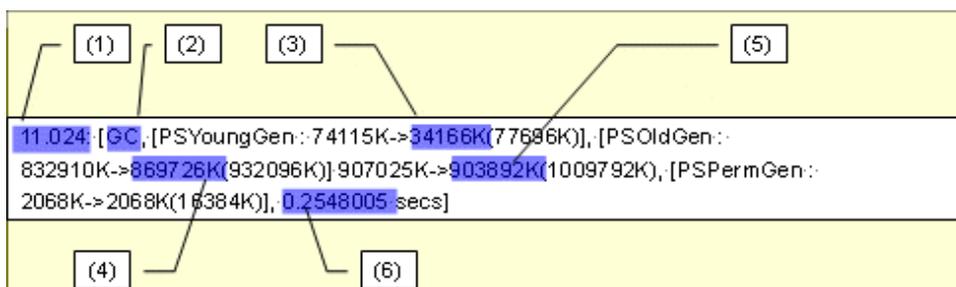


図 2 GCログの例(パラレルGCの場合)

GCログの解析で特に注目するのは、図2に示す6個の情報です。

- (1) Java VMが起動されてからの経過時間。単位は秒
- (2) GCの種別
- (3) GC処理実行後のNew世代領域に存在するオブジェクトの総量。単位はバイト
- (4) GC処理実行後のOld世代領域に存在するオブジェクトの総量。単位はバイト

(5) (3)と(4)の合計

(6) GC処理に要した時間。単位は秒



- ・ オプション-XX:FJverboseTimeを使うと、(1)の情報に日時を含めることもできます。詳しくは、チューニングガイドの「ログ出力における時間情報のフォーマット指定機能」を参照してください。
- ・ GCログは、アプリケーションの開発中や運用中を問わず、トラブル発生時の重要な情報です。上記オプションを常に設定することをお勧めします。  
GCログを出力することによって、パフォーマンスが低下することは、ほとんどありません。
- ・ Java監視機能を有効にしている場合は、アプリケーションが起動したあとでも、GCログの出力設定を変更できます。詳しくは、「A.5.8 VMオプションの変更」を参照してください。

### A.6.3.2 GCログ分析のポイント

まず、処理時間(図2の(6))が長いGCがあるかどうかを確認します(以降の説明では、処理時間が長いGCのことを「長時間GC」と表現します)。



GCの処理時間として許容される値はアプリケーションによって異なりますが、1秒以上かかるようなGCは、アプリケーションの性能に影響している可能性が考えられます。

アプリケーションに、レスポンス遅延などの問題が発生している場合は、長時間GCの発生時刻と問題の発生時間を照らし合わせます。それらが一致する場合、GCによってアプリケーションが長時間停止し、性能問題を引き起こした可能性が考えられます。

そのような場合は、長時間GCの発生を防ぐように、Javaヒープをチューニングしてください。チューニングに関する詳しい説明は、「チューニングガイド」-「JDK/JREのチューニング」-「チューニング方法」-「Javaヒープのチューニング」を参照してください。

アプリケーションの性能に影響するような長時間GCが存在しない場合は、生存オブジェクト量の推移を確認します。ここでは、GC処理実行後に存在するオブジェクトのことを、便宜的に「生存オブジェクト」と表現します。

生存オブジェクトの量は、GCログ(図2)の(5)に当たります。この値をプロットしてグラフを作成すると、生存オブジェクトの量が増加傾向にあるかが分かります。

図3は、生存オブジェクトの量をプロットしてグラフを作成した例です。

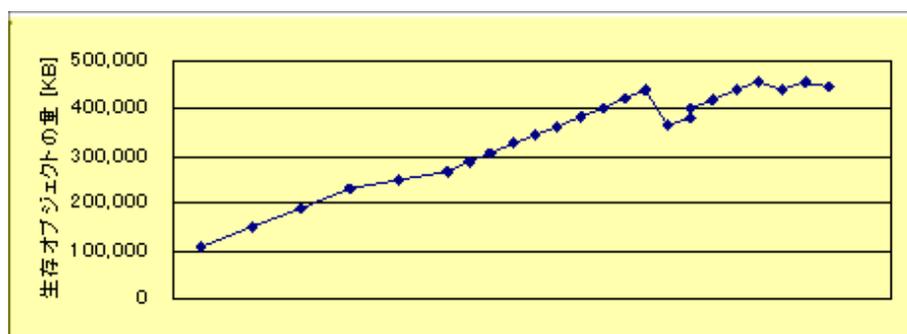


図 3 生存オブジェクト量が増加傾向にある例

生存オブジェクトの量が増え続けている場合は、メモリークが発生している可能性があります。そのような場合は、「A.6.5 メモリークを分析する」を参照して、メモリークが発生しているかどうかを調べてください。

### A.6.3.3 jstatコマンドを使った調査方法

動作中のプロセスに対してjstatコマンドを使うと、Javaヒープの使用状況を分析できます。jstatコマンドは、JDKに付属しています。

ただし、jstatは、定期的にJavaプロセスの情報を採取するツールです。GCログのように、GC処理実行後のJavaヒープの使用量(生存オブジェクトの量)を正確に調べるのには向いていませんので、注意してください。

jstatコマンドでJavaヒープの使用状況を調べるには、次のようにコマンドを実行します。

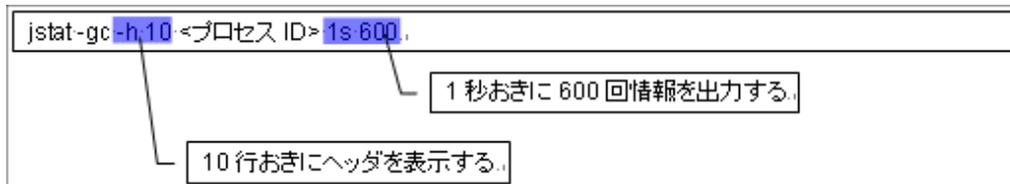


図 4 jstatコマンド実行の例



#### ポイント

「-h」オプションを使うと、定期的にヘッダが表示されるので、出力が長くなっても、どの列が何を示すのか分かります。

図4のようにjstatコマンドを実行すると、次のような結果を得られます。ここで注目すべきは、OUの列です。OUは、Old世代領域の使用量を示しています。

S0C	S1C	S0U	S1U	EC	EU	OC	OU	PC	PU	YGC	YGCT	FGC	FGCT	GCT
7232.0	7232.0	0.0	0.0	43776.0	40504.8	466048.0	0.0	16384.0	2716.6	0	0.000	0	0.000	0.000
7232.0	7232.0	0.0	0.0	43776.0	40504.8	466048.0	0.0	16384.0	2716.6	0	0.000	0	0.000	0.000
7232.0	7232.0	0.0	0.0	43776.0	42359.0	466048.0	0.0	16384.0	2716.6	0	0.000	0	0.000	0.000
7232.0	7232.0	0.0	0.0	43776.0	43235.6	466048.0	0.0	16384.0	2716.6	0	0.000	0	0.000	0.000
7232.0	7232.0	0.0	0.0	43776.0	43776.0	466048.0	0.0	16384.0	2716.6	0	0.000	0	0.000	0.000
:														
S0C	S1C	S0U	S1U	EC	EU	OC	OU	PC	PU	YGC	YGCT	FGC	FGCT	GCT
19392.0	19392.0	0.0	19383.7	19456.0	15729.9	466048.0	230372.1	16384.0	2727.2	7	0.342	0	0.000	0.342
19392.0	19392.0	0.0	19383.7	19456.0	16113.0	466048.0	230372.1	16384.0	2727.2	7	0.342	0	0.000	0.342
19392.0	19392.0	0.0	19383.7	19456.0	16879.4	466048.0	230372.1	16384.0	2727.2	7	0.342	0	0.000	0.342
19392.0	19392.0	0.0	19383.7	19456.0	17645.7	466048.0	230372.1	16384.0	2727.2	7	0.342	0	0.000	0.342
19392.0	19392.0	0.0	19383.7	19456.0	18412.0	466048.0	230372.1	16384.0	2727.2	7	0.342	0	0.000	0.342
:														
S0C	S1C	S0U	S1U	EC	EU	OC	OU	PC	PU	YGC	YGCT	FGC	FGCT	GCT
19392.0	19392.0	15344.0	0.0	19456.0	1913.6	466048.0	406129.0	16384.0	2729.2	16	0.623	0	0.000	0.623
19392.0	19392.0	15344.0	0.0	19456.0	2296.3	466048.0	406129.0	16384.0	2729.2	16	0.623	0	0.000	0.623
19392.0	19392.0	15344.0	0.0	19456.0	3061.7	466048.0	406129.0	16384.0	2729.2	16	0.623	0	0.000	0.623
19392.0	19392.0	15344.0	0.0	19456.0	3827.1	466048.0	406129.0	16384.0	2729.2	16	0.623	0	0.000	0.623
19392.0	19392.0	15344.0	0.0	19456.0	4592.5	466048.0	406129.0	16384.0	2729.2	16	0.623	0	0.000	0.623

図 5 jstatコマンドの実行結果の例

図5の例では、Old世代領域の使用量が、増加しつづけています。これは、生存オブジェクトが増え続けていることを意味しています。つまり、メモリリークが発生している可能性が考えられます。

## A.6.4 アプリケーションの性能ボトルネックを調べる

ここでは、次のように記述したメソッドを持つプログラムを例にして、アプリケーションの性能ボトルネックを調べる方法について説明します。

```
private static String doIt(final int nLoops) {
    String ret = "";
    String s = "Test";

    for (int i = 0; i < nLoops; i++) {
        ret += s;
    }

    return ret;
}
```

図 6 性能ボトルネックが存在するメソッドの例

メソッドdoIt()は、Stringオブジェクトに文字列”Test”を連結します。連結する回数は、メソッドの引数で指定します。

例えば、20万回の連結を実行するのに、300秒以上かかるとします(図7は、プログラムでメソッドdoIt()の実行時間を表示した例です。)

今回は、このプログラムの性能ボトルネックを調べていきます。

```
339328 milliseconds
```

図 7 性能ボトルネックが存在するメソッドの実行時間の例

アプリケーションの性能に関する情報を採集するには、Java監視機能を使用します。まず、Java監視機能の[概要]タブを開き、アプリケーションのCPU使用状況を確認してください。CPU使用率が高い場合は、アプリケーションが、CPUに負荷がかかる処理を実行している可能性があります。



### ポイント

CPU使用率が低い場合は、ロックの取得待ちや、I/O待ちなどによって、アプリケーションがスローダウンしている可能性が考えられます。「[A.6.6 ハングアップやスローダウンの原因を調べる](#)」を参照してください。

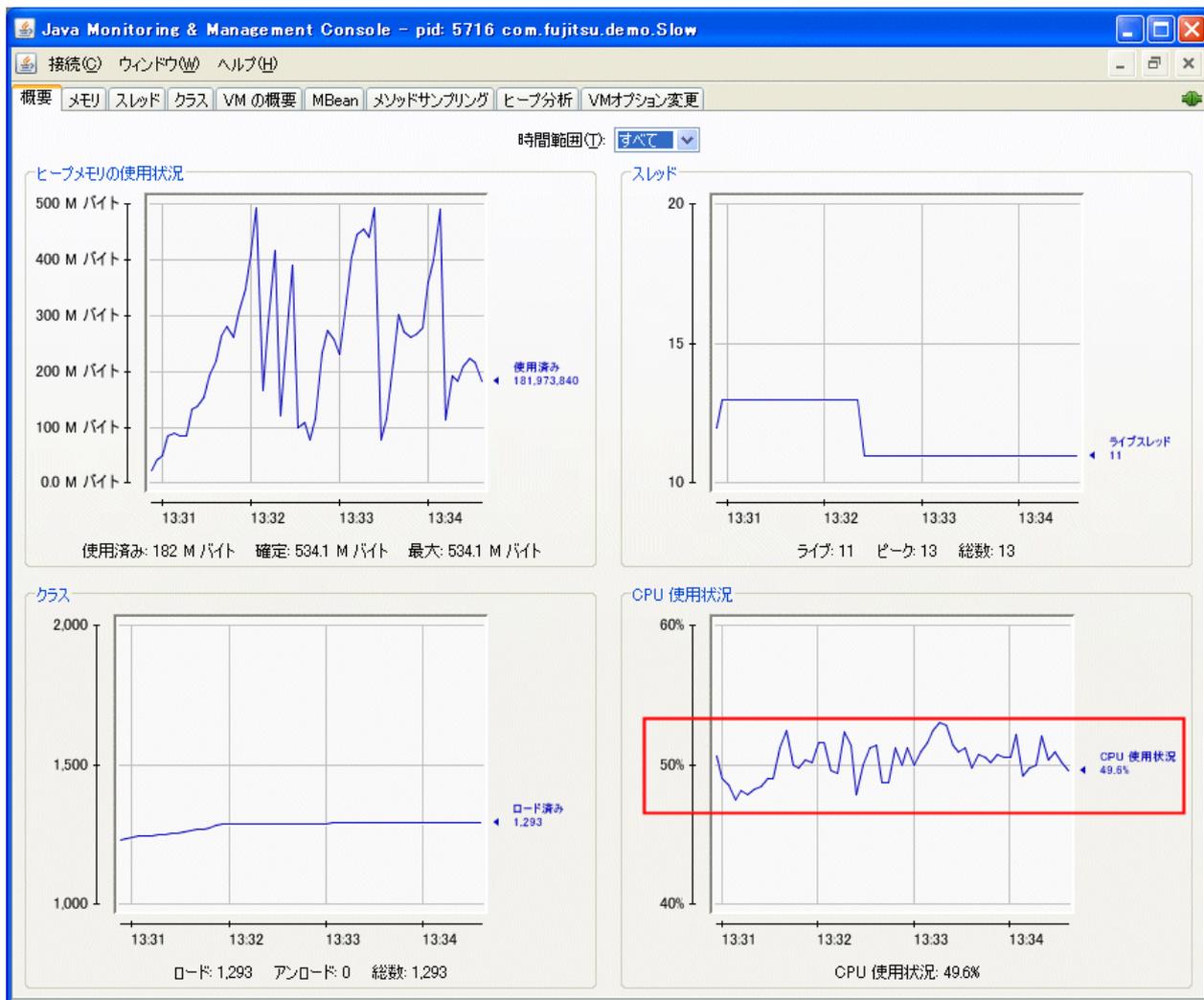


図 8 Java監視機能による、CPU使用率の確認

このアプリケーションでは、CPU使用率が50%前後で推移しています。今回アプリケーションを実行したマシンのCPUは、デュアルコアでした。つまり、1個のコアを常に使い続けていることが分かります。すなわち、CPUに負荷がかかるアプリケーションだと言えます。

そこで、アプリケーションが頻繁に実行しているメソッドを特定します。[メソッドサンプリング]タブを開いて、[開始]ボタンを押します。ここでは、サンプリング間隔を、デフォルト(50ミリ秒)のままとしています。



図 9 Java監視機能による、メソッドサンプリングの開始方法

ある程度の時間を置いて、[終了]ボタンを押します。次の画面は、サンプリングの結果です。

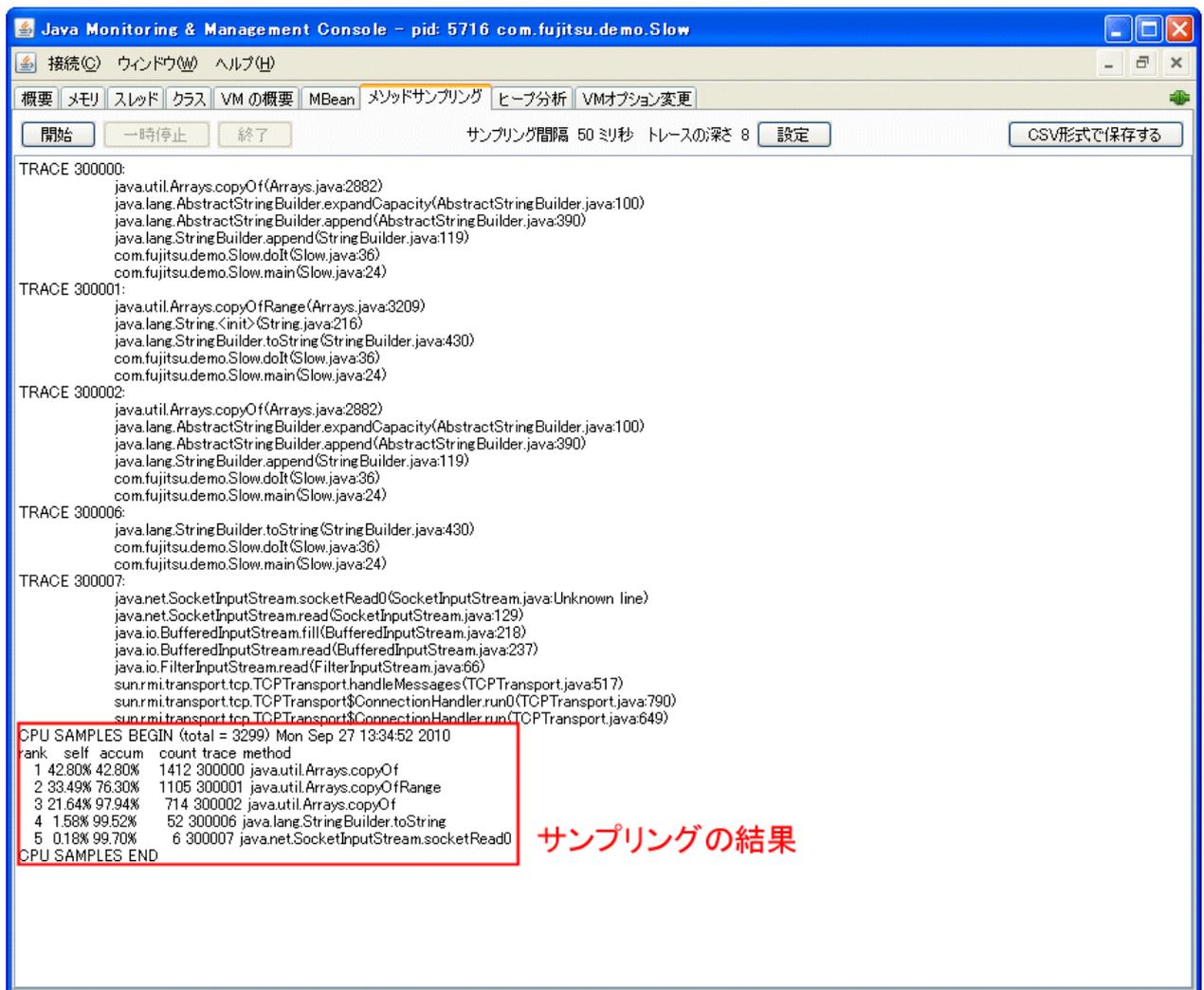


図 10 Java監視機能による、メソッドサンプリングの結果

サンプリングの結果は、一番下に出力されています。ここで上位に表示されているものは、測定期間中にサンプリングされた回数が多いメソッドを示しています。

rank	self	accum	count	trace	method
1	42.80%	42.80%	1412	300000	java.util.Arrays.copyOf
2	33.49%	76.30%	1105	300001	java.util.Arrays.copyOfRange
3	21.64%	97.94%	714	300002	java.util.Arrays.copyOf
4	1.58%	99.52%	52	300006	java.lang.StringBuilder.toString
5	0.18%	99.70%	6	300007	java.net.SocketInputStream.socketRead0

上位3つで95%以上を占める

図 11 Java監視機能による、メソッドサンプリングの結果(抜粋)

この結果から、java.util.Arrays#copyOf()メソッドと、java.util.Arrays#copyOfRange()メソッドのサンプリング回数が、全体の95%以上を占めていることが分かります。

次に、これらのメソッドの呼び出し関係(スタックトレース)を調べます。ここでは、スタックトレース「300000」「300001」「300002」を参照します。

```
TRACE 300000:
  java.util.Arrays.copyOf(Arrays.java:2882)
  java.lang.AbstractStringBuilder.expandCapacity(AbstractStringBuilder.java:100)
  java.lang.AbstractStringBuilder.append(AbstractStringBuilder.java:390)
  java.lang.StringBuilder.append(StringBuilder.java:119)
  com.fujitsu.demo.Slow.doIt(Slow.java:36)
  com.fujitsu.demo.Slow.main(Slow.java:24)
```

図 12 TRACE 300000のスタックトレース

```
TRACE 300001:
  java.util.Arrays.copyOfRange(Arrays.java:3209)
  java.lang.String.<init>(String.java:216)
  java.lang.StringBuilder.toString(StringBuilder.java:430)
  com.fujitsu.demo.Slow.doIt(Slow.java:36)
  com.fujitsu.demo.Slow.main(Slow.java:24)
```

図 13 TRACE 300001のスタックトレース

```
TRACE 300002:
  java.util.Arrays.copyOf(Arrays.java:2882)
  java.lang.AbstractStringBuilder.expandCapacity(AbstractStringBuilder.java:100)
  java.lang.AbstractStringBuilder.append(AbstractStringBuilder.java:390)
  java.lang.StringBuilder.append(StringBuilder.java:119)
  com.fujitsu.demo.Slow.doIt(Slow.java:36)
  com.fujitsu.demo.Slow.main(Slow.java:24)
```

図 14 TRACE 300002のスタックトレース



TRACE 300000とTRACE 300002は、スタックトレースに含まれるメソッドの順序が同じですが、別のトレースとして記録されています。

これは、Java監視機能のメソッドサンプリング機能が、バイトコードのレベルで、メソッドの実行箇所に関する情報を採取するからです。Javaプログラムの行数が同じでも、サンプリング時に実行中のバイトコードが異なれば、違うスタックトレースとして回数を記録します。

これらのスタックトレースを見ると、サンプリング回数が多かった直接のメソッドはjava.util.Arrays#copyOf()とjava.util.Arrays#copyOfRange()です。しかし、これらを引き出すきっかけになったのは、メソッドcom.fujitsu.demo.Slow#doIt()(プログラムの36行目)で使用している、java.lang.StringBuilderクラスのメソッドだと分かりません。

つまり、メソッドcom.fujitsu.demo.Slow#doIt()におけるjava.lang.StringBuilderクラスの使用が、性能負荷の要因だと考えることができます。

メソッドcom.fujitsu.demo.Slow#doIt()には、StringBuilderクラスを使用している箇所はないように見えます。しかし、次の結合演算が、Javaコンパイラによって、メソッドStringBuilder#append()を使う処理に変換されています。そのため、文字列を結合するたびに、StringBuilder#append()が呼び出されます。

```

private static String doIt(final int nLoops) {
    :
    for (int i = 0; i < nLoops; i++) {
        ret += s;
    }
    :
}

```

Java コンパイラが、この結合演算を `StringBuilder#append()` に置き換える

図 15 性能ボトルネックの箇所

Java SEのAPI仕様書を見ても分かるように、`StringBuilder()`コンストラクタは、初期容量が16文字の文字列バッファを構築します。`append()`の呼び出しを繰り返すうちにバッファが足りなくなると、バッファは自動的に拡張されます。

今回採取したメソッドサンプリングの情報から、`java.util.Arrays`クラスのメソッド`copyOf()`と`copyOfRange()`の実行に、時間がかかっていることが分かります。つまり、`StringBuilder#append()`によるバッファの拡張処理に、時間がかかっていると考えられます。

そこで、メソッド`com.fujitsu.demo.Slow#doIt()`を次のように修正し、`StringBuilder`のバッファを、あらかじめ大きく確保するようにします。

```

private static String doIt(final int nLoops) {
    String ret = "";
    String s = "Test";

    StringBuilder sb = new StringBuilder(s.length() * nLoops);

    for (int i = 0; i < nLoops; i++) {
        sb.append(s);
    }

    return sb.toString();
}

```

必要なバッファを、あらかじめ確保

図 16 性能ボトルネック箇所の修正例

プログラムをこのように修正すると、`StringBuilder`のバッファ拡張を防ぎ、メソッドの実行時間を飛躍的に短縮できました。図17は、修正後のプログラムを実行して、メソッド`doIt()`の実行時間を出力した例です。

```

16 milliseconds

```

図 17 性能ボトルネックを修正したメソッドの実行時間の例

#### A.6.4.1 Java VisualVMを使った調査

性能ボトルネックは、Java VisualVMを使って調べることもできます。詳しくは、JDKドキュメントを参照してください。

## A.6.5 メモリリークを分析する

ここでは、メモリリークの分析方法を説明します。  
まず、リークしているクラス(オブジェクトの種類)を特定します。リークしているクラスが分かったら、ヒープダンプを採取して、オブジェクトの参照関係を調べ、リークの原因となっているオブジェクトを特定します。

### A.6.5.1 リークしているクラスの特定制

リークしているクラスを特定する方法について説明します。  
まず、Java監視機能の[ヒープ分析]タブを開いて[開始]ボタンを押し、分析を始めてください。

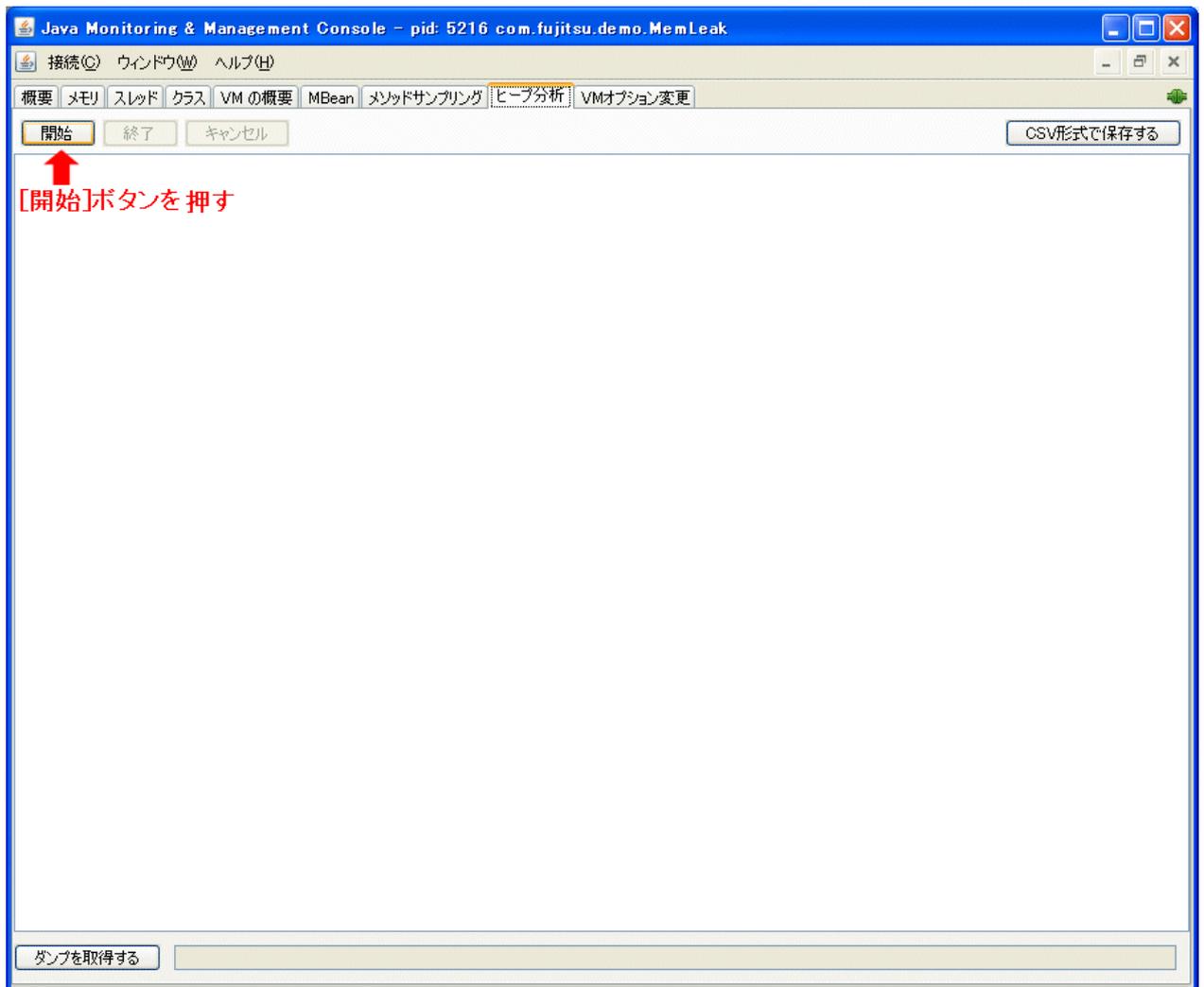


図 18 Java監視機能による、メモリリーク調査(分析の開始)

分析が開始されると、次のようなメッセージが表示されます。

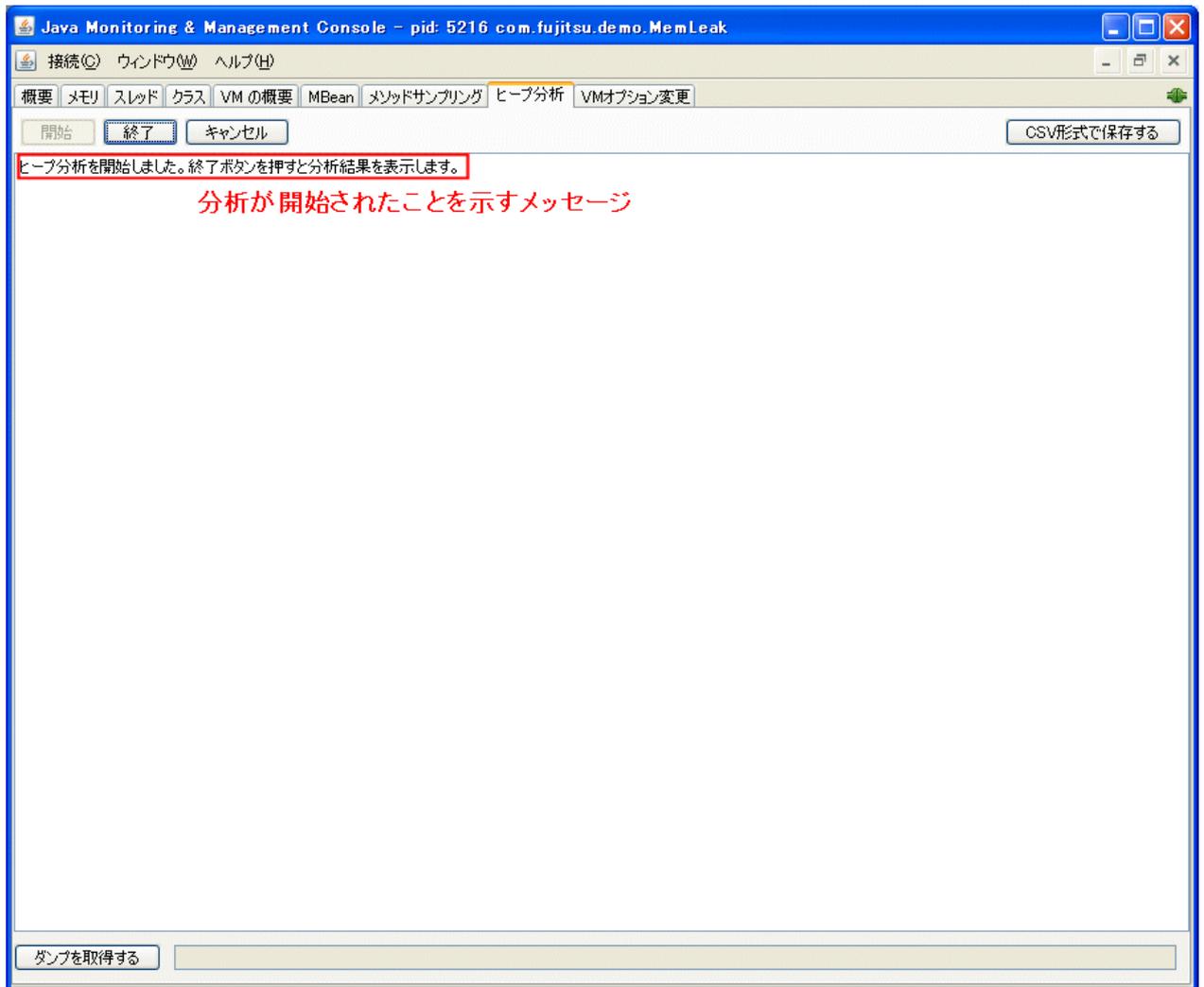


図 19 Java監視機能による、メモリーク調査(分析開始後)

[終了]ボタンを押すと、分析が終了します。開始から終了までの時間は、十分とってください。終了のタイミングを計るには、[メモリ]タブを開いて、Javaヒープ使用量の推移を確認します。

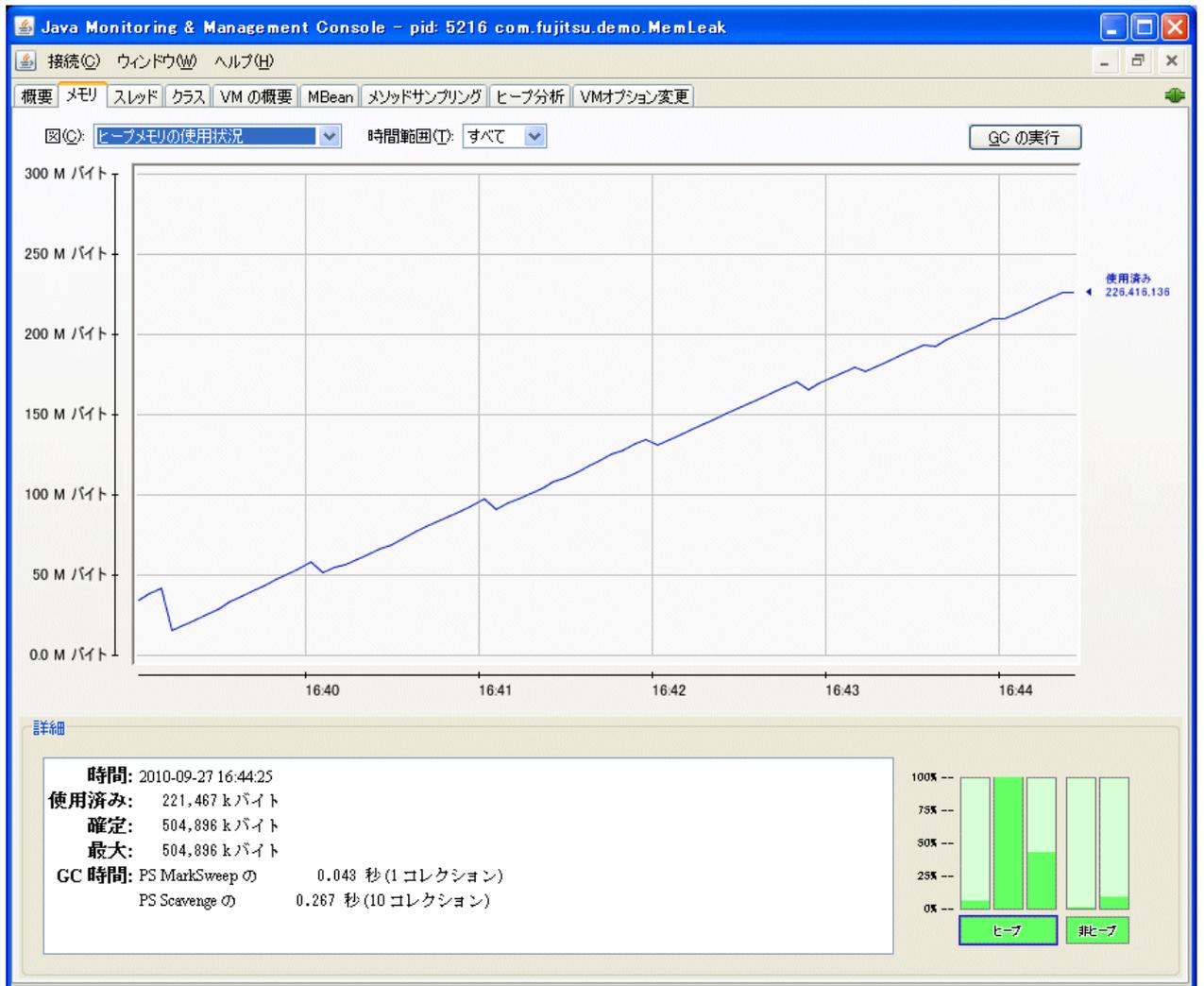


図 20 Java監視機能による、メモリーク調査 (Javaヒープ使用量の確認)

この例では、Javaヒープの使用量が、上昇し続けていることが分かります。  
Javaヒープの使用量が増加し続けることが確認できたら、[ヒープ分析]の画面で[終了]ボタンを押します。

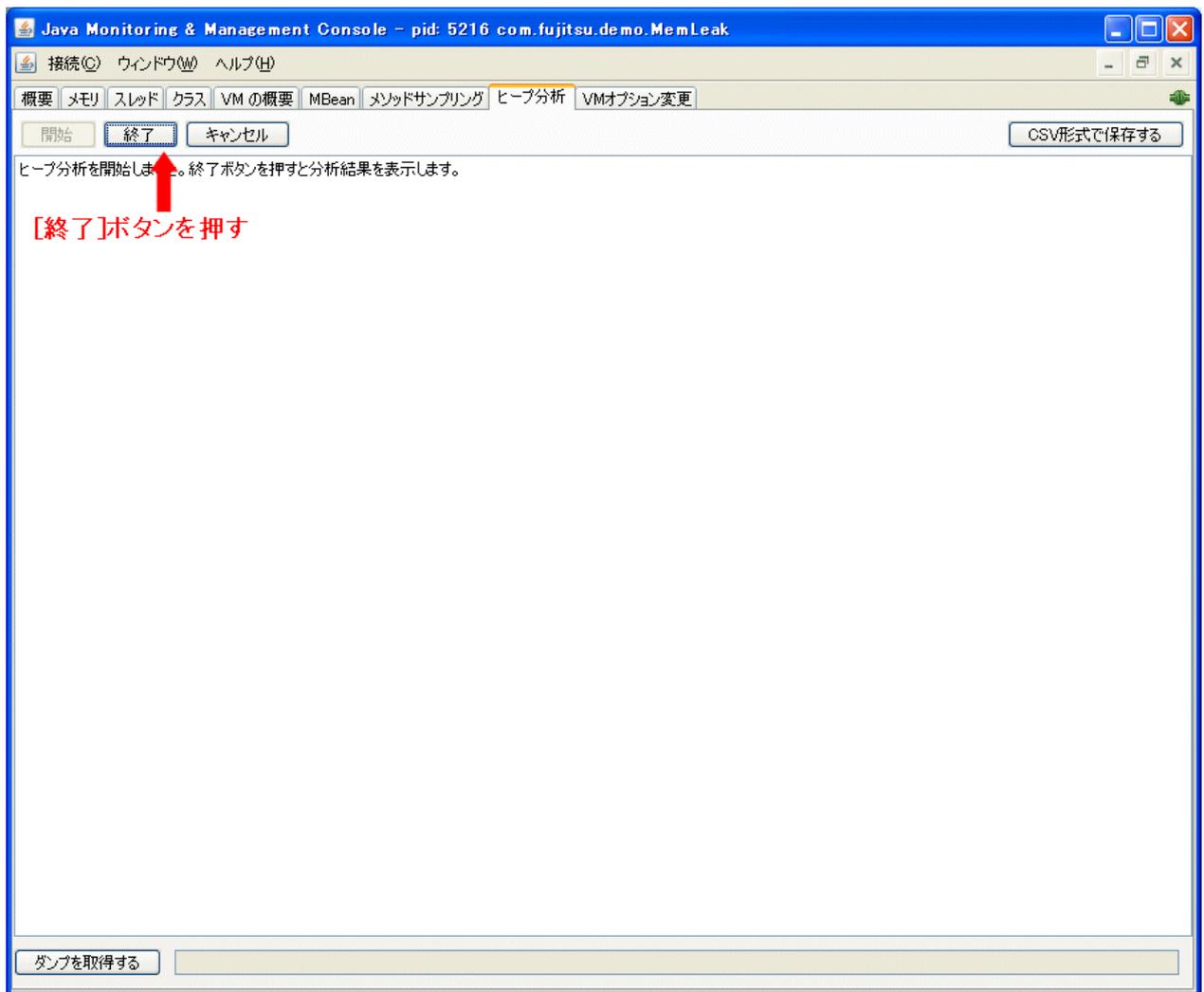


図 21 Java監視機能による、メモリーク調査(分析の終了方法)

分析が終了すると、オブジェクトヒストグラムが表示されます。測定期間中の増加量(バイト数)が多い順に、クラスが表示されます。

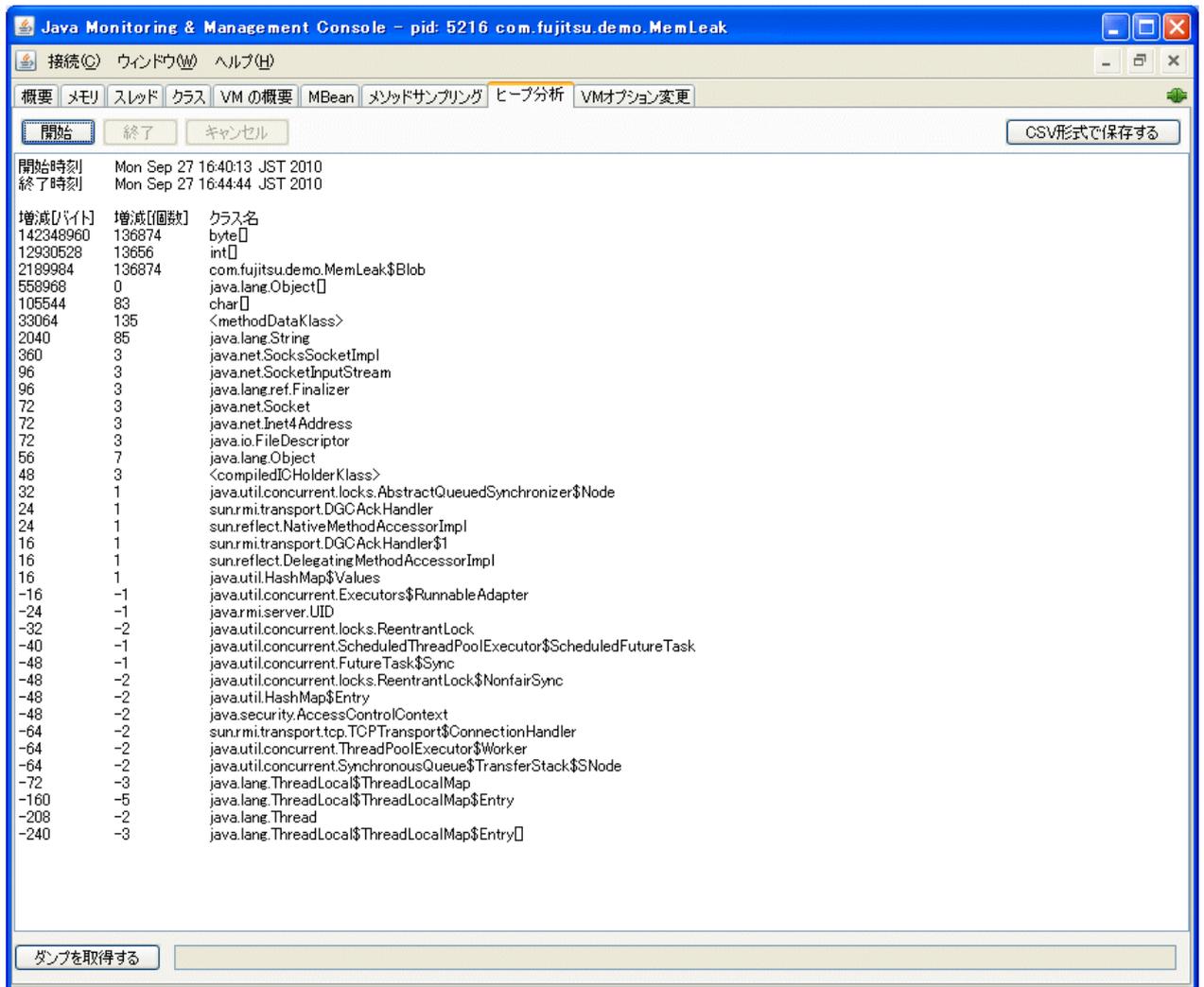


図 22 Java監視機能による、メモリーーク調査(分析結果の表示)

オブジェクトヒストグラムを見て、大量に増加しているクラスを特定します。

増減[バイト]	増減[個数]	クラス名
142348960	136874	byte[]
12930528	13656	int[]
2189984	136874	com.fujitsu.demo.MemLeak\$Blob
558968	0	java.lang.Object[]
105544	83	char[]
33064	135	<methodDataKlass>
2040	85	java.lang.String

図 23 Java監視機能による、メモリーーク調査(分析結果の抜粋)

まず、アプリケーションで作成したクラスが増加しているかどうかを確認します。この例では、com.fujitsu.demo.MemLeak\$Blobクラスのインスタンスが、13万個以上増加していることが分かります。つまり、このクラスのインスタンスが、メモリーークしている可能性が考えられます。



## ポイント

この例では、byteやintの配列も、大量に増加しています。しかし、プリミティブ型の配列は、リークとは関係なく増加している可能性も考えられます。

そのため、まずはアプリケーションで作成したクラスの増加量に着目するのが、リーク調査を効率的に行うコツです。アプリケーションで作成したクラスの増加量が小さい場合は、プリミティブ型の配列や、コアライブラリのクラスの増加量に着目します。

ここまでの調査で、com.fujitsu.demo.MemLeak\$Blobクラスのインスタンスがリークしている可能性が大きいことが分かりました。

次に、ヒープダンプを採取してオブジェクトの参照関係を調べ、リークの原因となっているオブジェクトを特定します。

### A.6.5.2 jmapコマンドを使った調査方法

増加しているクラスの種類の、jmapコマンドを使って調べることもできます。jmapコマンドを次のように実行します。実際には、標準出力をリダイレクトするなどして、結果をファイルなどに保存してください。

```
jmap -histo:live <プロセスID>
```

図 24 jmapコマンドの実行例

このとき、「-histo」オプションに「live」サブオプションを付けてください。「:live」を付けると、ヒストグラムを採取する前にGCを実行します。つまり、GC直後の生存オブジェクトだけを含むヒストグラムを得られます。これは、メモリークの調査において重要な点です。

jmapコマンドを複数回実行したあとで、採取したヒストグラムを比較してください。オブジェクトが増加しているクラスがあれば、メモリークしている可能性が考えられます。

例えば、図25でそれぞれ15,923個、15,342個だったbyte配列とcom.fujitsu.demo.MemLeak\$Blobが、図26の時点では184,406個と183,823個に増加しています。

num	#instances	#bytes	class name
1:	15923	16062672	[B
2:	3460	2772800	[I
3:	6393	673312	<constMethodKlass>
4:	6393	514456	<methodKlass>
5:	10802	481192	<symbolKlass>
6:	495	271336	<constantPoolKlass>
7:	15342	245472	com.fujitsu.demo.MemLeak\$Blob
8:	2245	230552	[C
9:	495	195784	<instanceKlassKlass>
10:	469	182432	<constantPoolCacheKlass>
:			

図 25 jmapで採取したヒストグラムの例(1)

このような場合は、これらのオブジェクトについて、メモリークが発生しているかどうか調べていきます。

num	#instances	#bytes	class name
1:	184406	191283136	[B
2:	26142	23888848	[I
3:	183823	2941168	com.fujitsu.demo.MemLeak\$Blob
4:	322	807792	[Ljava.lang.Object;
5:	6413	674944	<constMethodKlass>
6:	6413	516056	<methodKlass>
7:	10825	482120	<symbolKlass>
8:	497	272152	<constantPoolKlass>

9:	2257	231064	[C
10:	497	196504	<instanceKlassKlass>

図 26 jmapで採取したヒストグラムの例(2)

### A.6.5.3 ヒープダンプの採取と分析

ヒープダンプを採取して、com.fujitsu.demo.MemLeak\$Blobクラスの参照関係を調べます。Java監視機能の[ヒープ分析]タブで[ダンプを取得する]ボタンを押して、ヒープダンプを採取してください。



図 27 Java監視機能による、メモリリーク調査(ヒープダンプ採取)

ヒープダンプが出力されたファイル名は、ボタンの右側に表示されます。このファイルを、jhatで解析します。



ポイント

ヒープダンプの出力(ファイル)は、アプリケーションが動作しているマシンに生成されます。



ポイント

ヒープダンプは、jmapコマンドを使って採取することもできます。その場合、「-dump」オプションに「live」サブオプションを付けて、生存オブジェクトだけを含むヒープダンプを採取してください。

次の図は、ローカルマシンにコピーしたヒープダンプを、jhatを起動する例です。

```
> jhat fjvm-heapdump.5216.0927164520
```

図 28 jhatの起動例

jhatを起動したマシンに、インターネットブラウザでアクセスします。アクセスするポート番号は、デフォルトでは7000番です。

```
http://localhost:7000/object/0x10553ce0
```

図 29 インターネットブラウザでアクセスするURLの例

次の図は、ローカルマシンで動作しているjhatに対して、Firefoxでアクセスしたときの画面の例です。

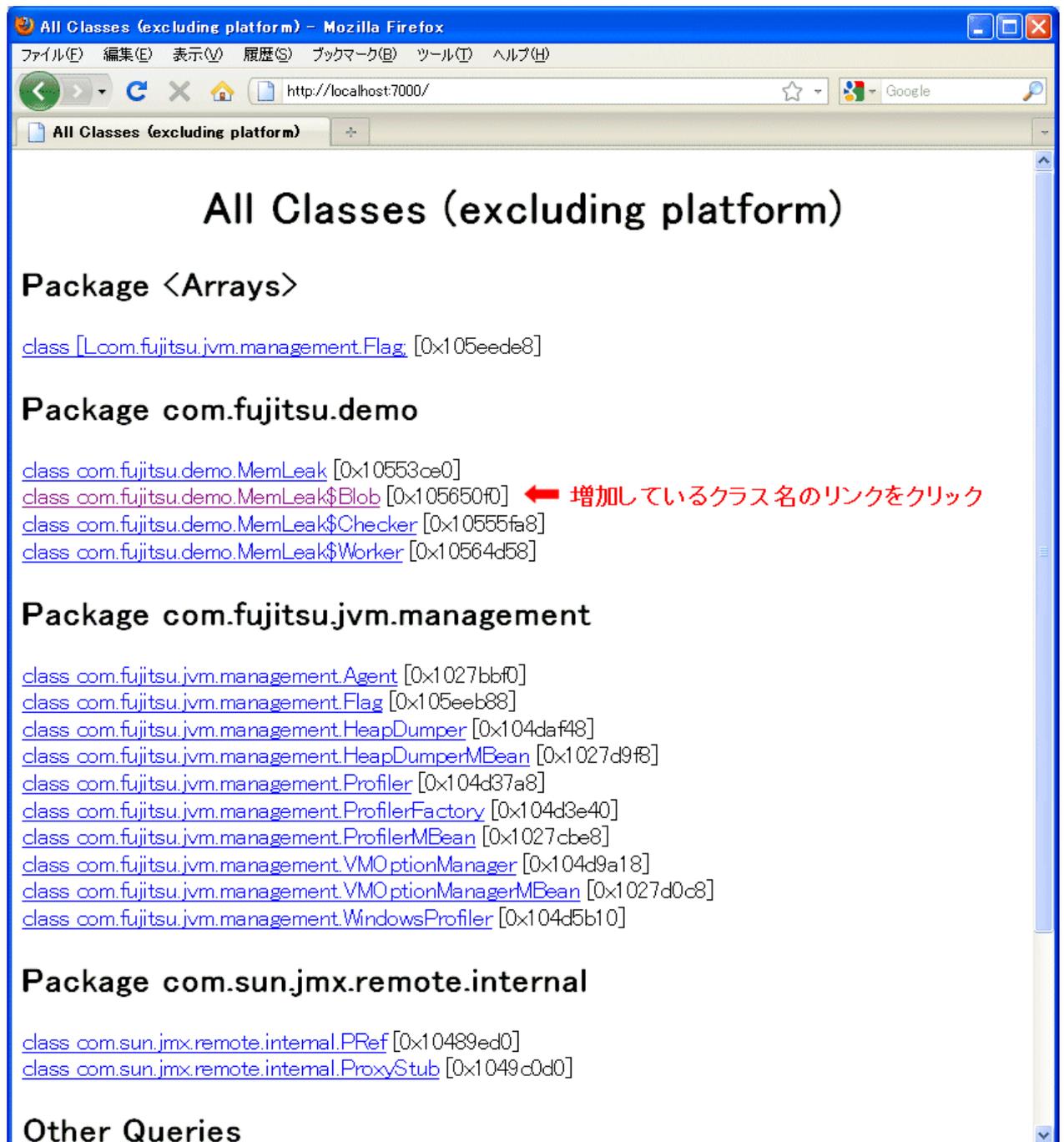


図 30 jhatによる調査(jhatのトップ画面)

増加しているcom.fujitsu.demo.MemLeak\$Blobクラスのリンクをクリックすると、そのクラスに関する情報が表示されます。

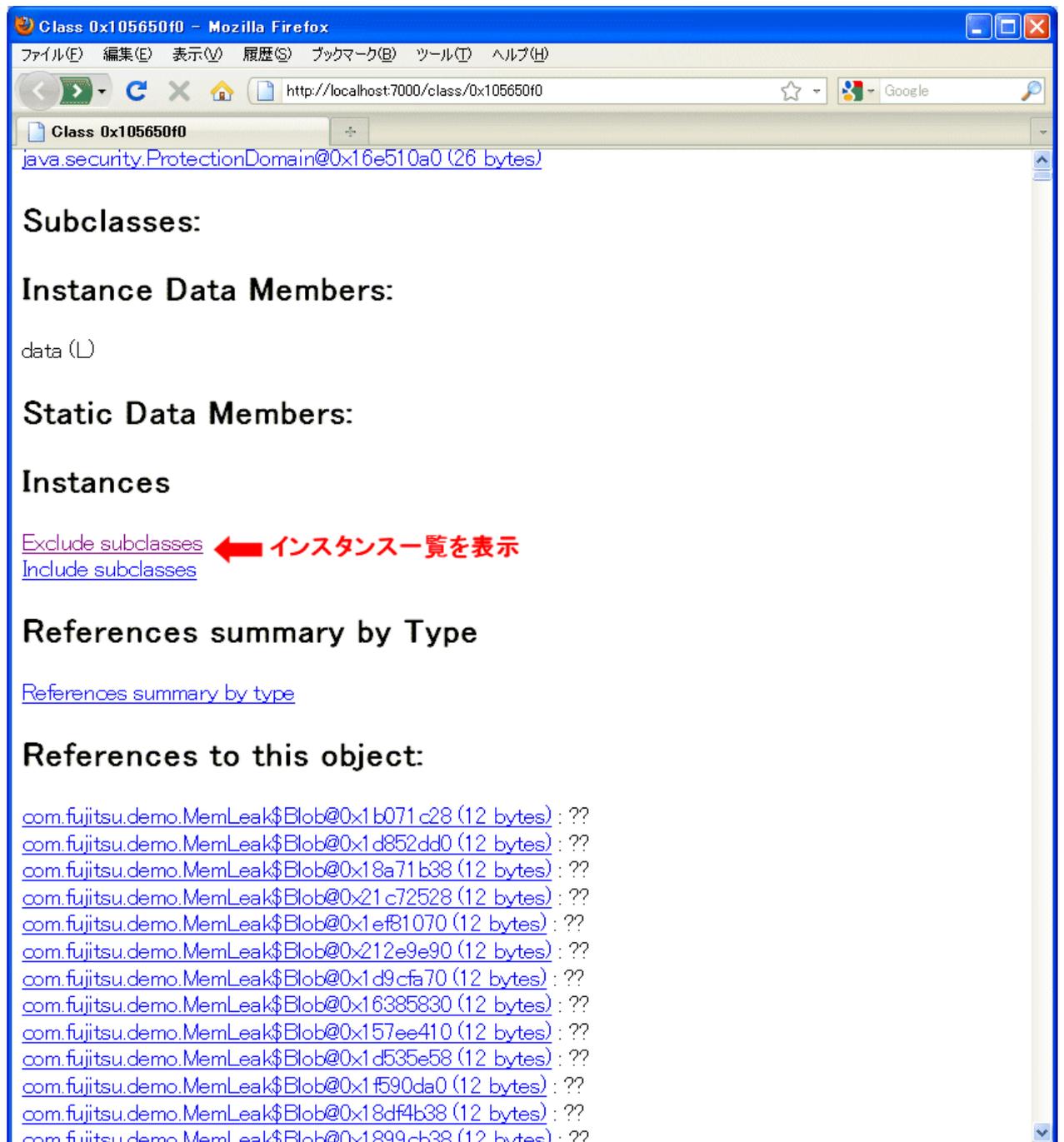


図 31 jhatによる調査(増加量が多いクラスの情報)

次に、このクラスのインスタンスを表示するため、「Instances」の「Exclude subclasses」をクリックします。

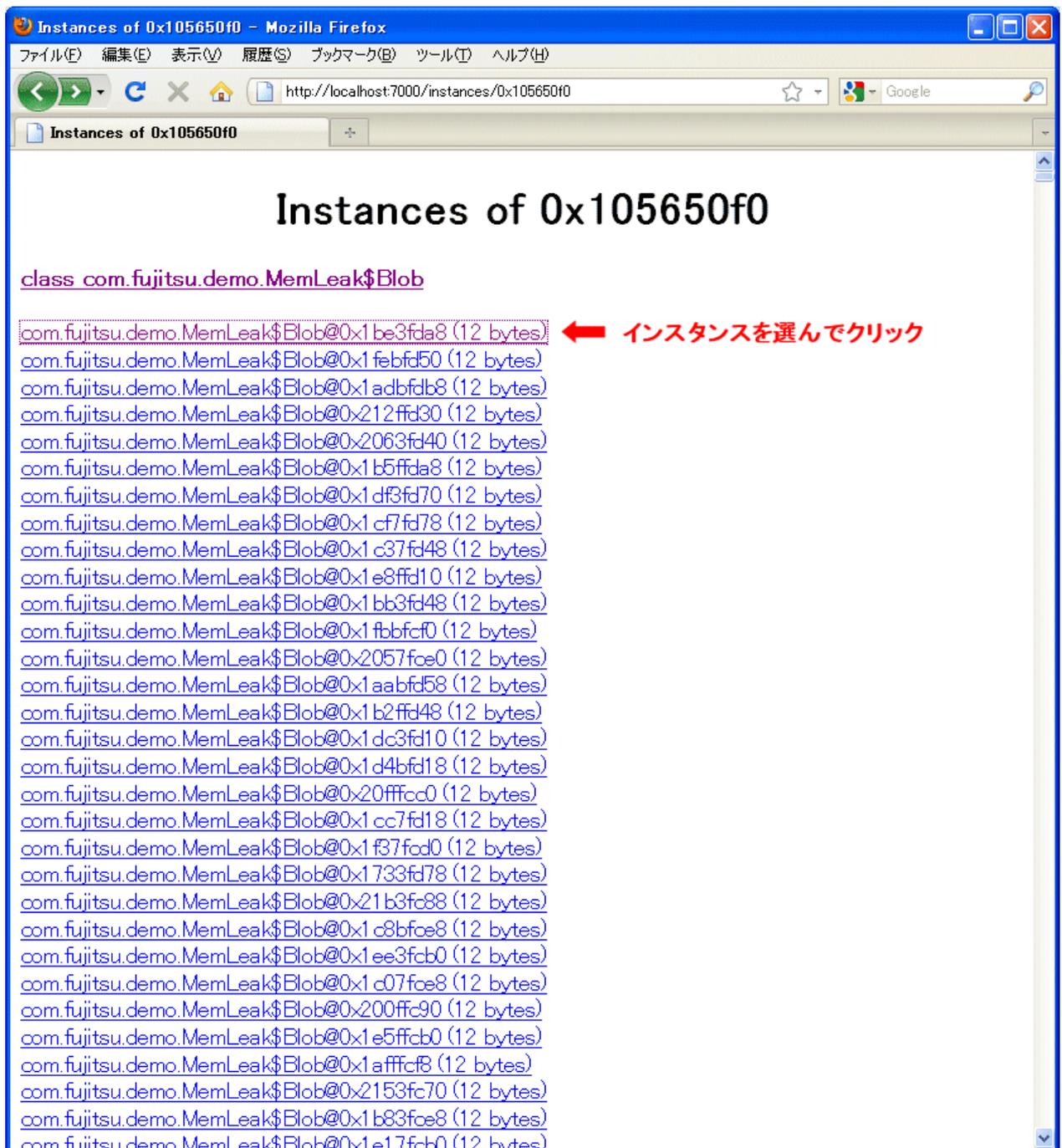


図 32 jhatによる調査(増加量が多いクラスのインスタンス一覧)

com.fujitsu.demo.MemLeak\$Blobクラスのインスタンスの一覧が表示されました。一覧から、適当なインスタンスのリンクをクリックします。

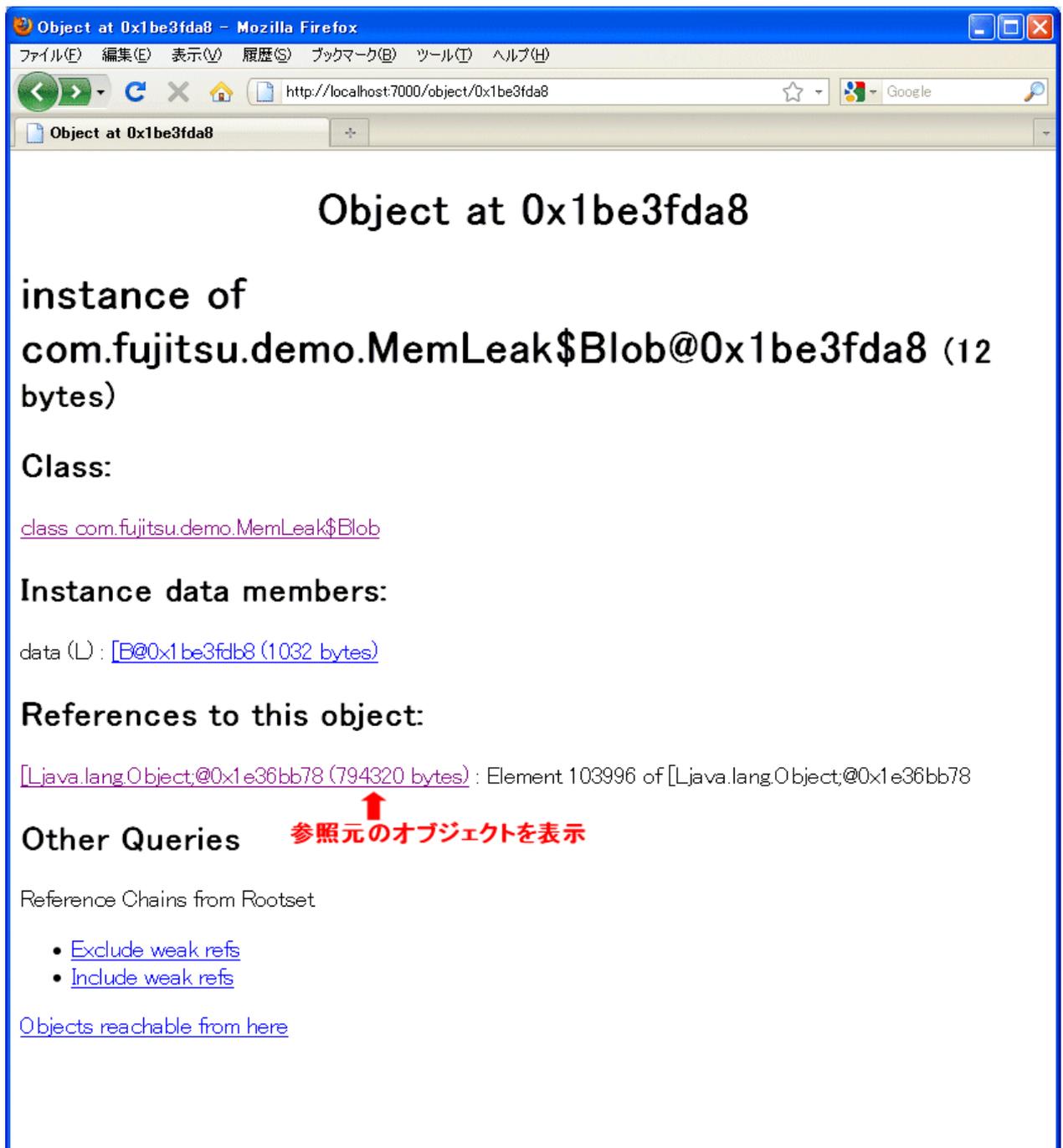


図 33 jhatによる調査(オブジェクトの参照元の調査)

インスタンスに関する情報が表示されました。

com.fujitsu.demo.MemLeak\$Blobオブジェクトを参照しているオブジェクトを特定するため、「References to this object」に表示されているリンクをクリックします。

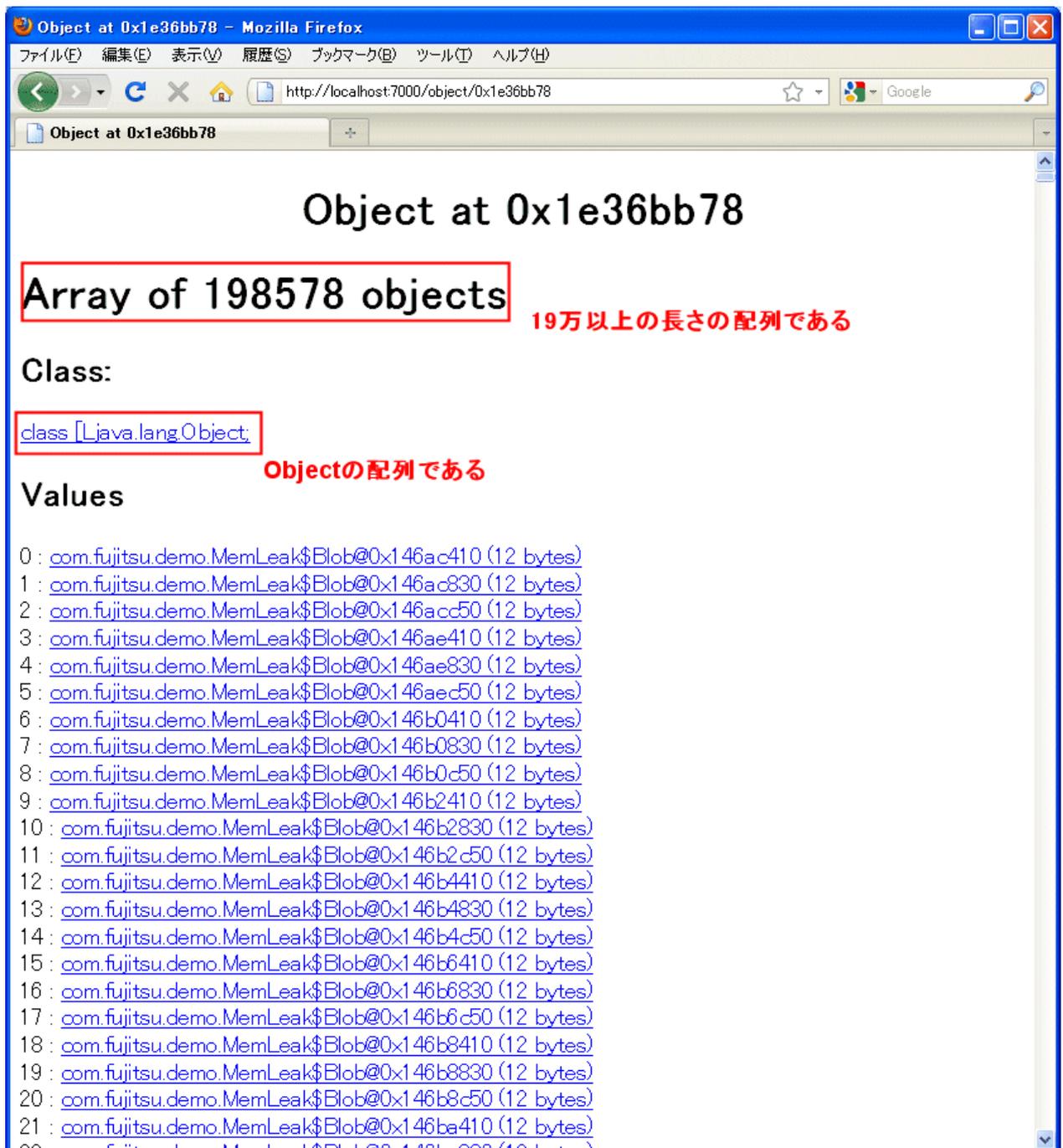


図 34 jhatによる調査(オブジェクトの参照元の情報)

com.fujitsu.demo.MemLeak\$Blobオブジェクトの参照元は、巨大な配列であることが分かりました。ブラウザのスクロールバーを下に移動して「References to this object」を見ると、更に参照元があることが分かります。

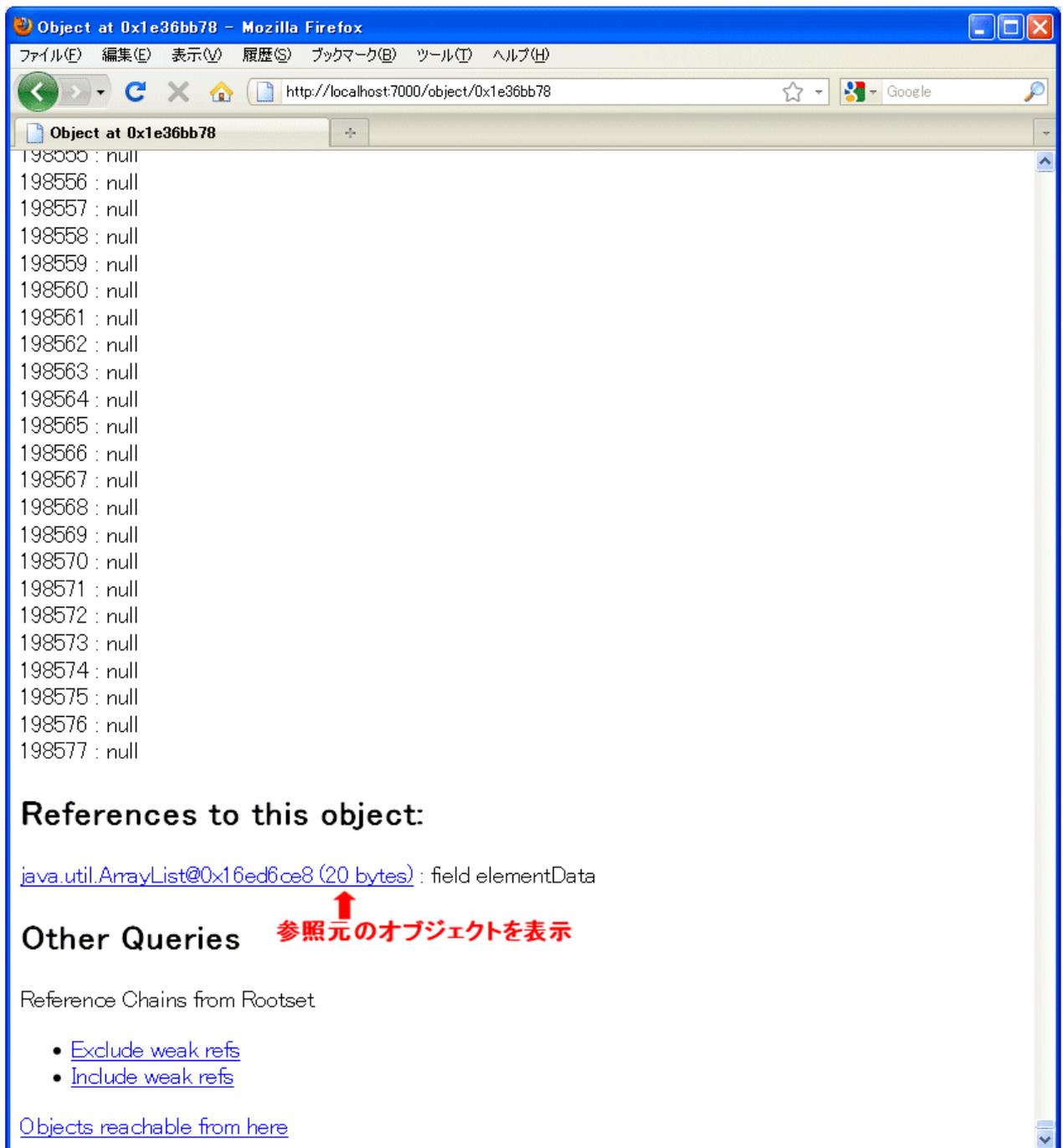


図 35 jhatによる調査(配列の参照元の調査)

このようにして「References to this object」のリンクを辿っていくと、com.fujitsu.demo.MemLeakのstatic変数blobsから参照していることが分かりました。

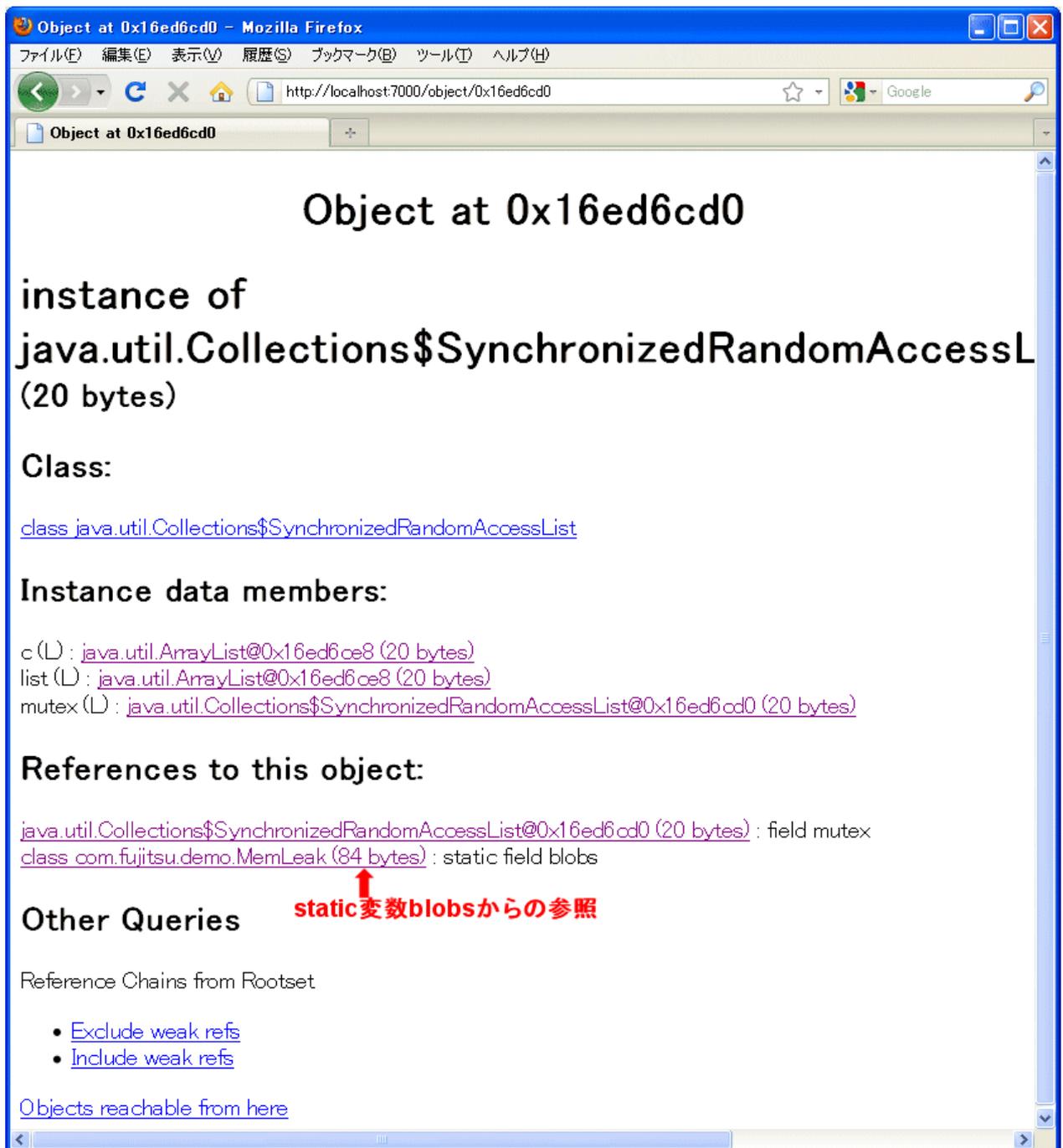


図 36 jhatによる調査(オブジェクトの参照元の調査)

com.fujitsu.demo.MemLeakクラスのリンクをクリックして、クラスの情報を表示してみると、確かにstatic変数blobsが存在しました。この変数は、java.util.Listインタフェースを実装したクラスのインスタンスを指しています。

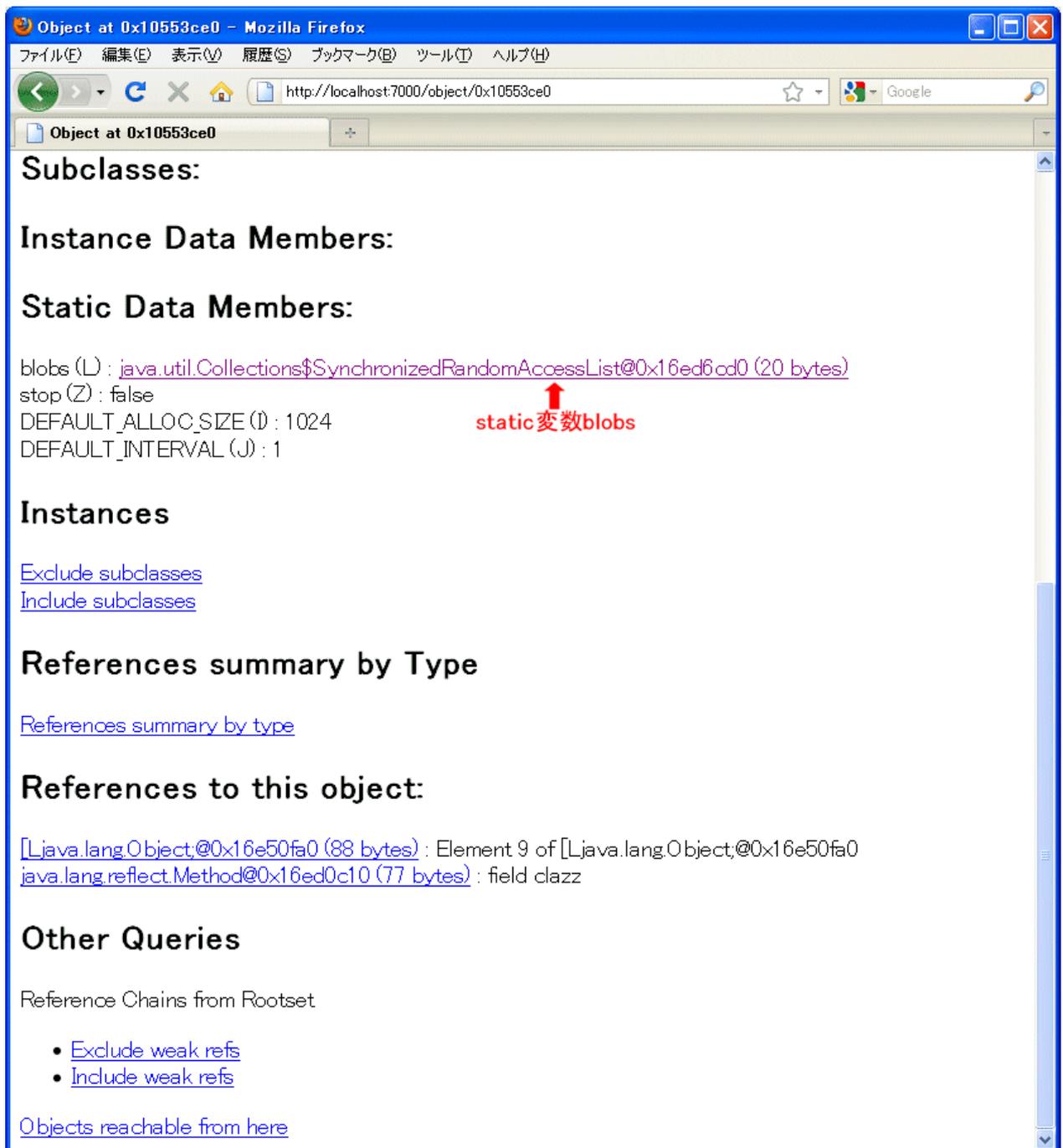


図 37 jhatによる調査(メモリークの原因となっているstatic変数の例)

つまり、このblobsにcom.fujitsu.demo.MemLeak\$Blobを追加したあと、不要になった時点でオブジェクトをリストから削除しているかどうか、プログラムを確認する必要があります。

#### A.6.5.4 Java VisualVMを使った調査

メモリークは、Java VisualVMを使って調べることもできます。詳しくは、JDKドキュメントを参照してください。

#### A.6.6 ハングアップやスローダウンの原因を調べる

アプリケーションのハングアップや、スローダウンといった現象が発生した場合は、アプリケーションがデッドロックしたり、I/O待ちで止まったりしている可能性が考えられます。

まず、デッドロックが発生しているかどうかを確認します。Java VMには、デッドロックを検出する機能が備わっていますので、それを利用します。

デッドロックが発生していない場合は、フルスレッドダンプを分析して、ハングアップやスローダウンの原因を特定します。

### A.6.6.1 デッドロックの検出

Java VMには、フルスレッドダンプを出力する機能が備わっています。その機能の一部として、デッドロックを検出する機能が実装されています。

フルスレッドダンプを出力する方法には、以下の3つがあります。

- シグナルQUITを送信する
  - LinuxまたはOracle Solarisの場合
  - 情報は、対象のJavaプロセスの標準出力に出力される
- Ctrl-Breakを送信する、またはthdumpコマンドを使う
  - Windowsの場合
  - 情報は、対象のJavaプロセスの標準出力に出力される
- jstackコマンドを使う
  - 情報は、jstackコマンドの標準出力に出力される(動作中に出力される一部のトレース情報は、標準エラー出力に出力されることがある)

フルスレッドダンプを出力する方法の例を示します。

```
$ kill -QUIT <プロセスID>
```

図 38 シグナルQUITを送信する例

```
> thdump -p <プロセスID>
```

図 39 thdumpコマンドを使う例

```
> jstack <プロセスID>
```

図 40 jstackコマンドを使う例

いずれかの方法でフルスレッドダンプを出力した場合に、デッドロックを検出すると、次のような情報が出力されます。

```
Found one Java-level deadlock:
=====
"Thread-2":
  waiting to lock monitor 0x0599d364 (object 0x0fd1f2f8, a java.lang.Object),
  which is held by "Thread-0"
"Thread-0":
  waiting to lock monitor 0x065338fc (object 0x0fd1f300, a java.lang.Object),
  which is held by "Thread-1"
"Thread-1":
  waiting to lock monitor 0x0599df94 (object 0x0fd1f308, a java.lang.Object),
  which is held by "Thread-2"

Java stack information for the threads listed above:
=====
"Thread-2":
  at com.fujitsu.demo.DeadLocking$ThreadC.run(DeadLocking.java:57)
  - waiting to lock <0x0fd1f2f8> (a java.lang.Object)
```

```

- locked <0x0fd1f308> (a java.lang.Object)
"Thread-0":
  at com.fujitsu.demo.DeadLocking$ThreadA.run(DeadLocking.java:29)
  - waiting to lock <0x0fd1f300> (a java.lang.Object)
  - locked <0x0fd1f2f8> (a java.lang.Object)
"Thread-1":
  at com.fujitsu.demo.DeadLocking$ThreadB.run(DeadLocking.java:43)
  - waiting to lock <0x0fd1f308> (a java.lang.Object)
  - locked <0x0fd1f300> (a java.lang.Object)

Found 1 deadlock.

```

図 41 デッドロック検出の例 (jstack)

この情報は、3つスレッド「Thread-0」「Thread-1」「Thread-2」のあいだでデッドロックが発生していることを示しています。それぞれのスレッドは、オブジェクトのロックを獲得した状態で、ほかのスレッドが保持しているロックを獲得しようとして、処理が止まっています。つまり、次の図のような三すくみ状態に陥っていることが分かります。

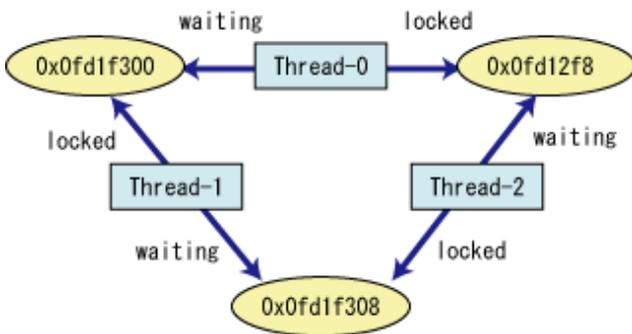


図 42 デッドロックの状態

そのため、これらのスレッドの処理を見直して、デッドロックが発生しないように、プログラムを修正する必要があります。デッドロックが検出されなかった場合は、どのメソッドが原因でハングアップやスローダウンが発生しているか、フルスレッドダンプの情報から確認します。

### A.6.6.2 ハングアップやスローダウンの調査

ハングアップやスローダウンが発生しているスレッドを特定するために、フルスレッドダンプを複数回採取します(目安は3回以上)。そして、採取したフルスレッドダンプの情報を比較して、ハングアップやスローダウンの発生箇所を調べます。フルスレッドダンプの採取方法については、「A.6.6.1 デッドロックの検出」を参照してください。

例えば、jstackコマンドを使って、次の例のようにフルスレッドダンプ情報を別々のファイルに保存します。そして、diffなどのOS付属ツールを使ってファイル同士を比較すると、フルスレッドダンプを簡単に比較できます。

```

jstack 5732 > threaddump-1.txt
jstack 5732 > threaddump-2.txt
jstack 5732 > threaddump-3.txt

```

図 43 フルスレッドダンプの採取

シグナルやthdumpコマンドを使ってフルスレッドダンプを採取した場合は、1回ごとの出力結果を、それぞれ別のファイルにコピーすると、同様に比較できます。

ここでは、図43のように、3回分のフルスレッドダンプを、それぞれ別のファイルに保存しているものとして説明します。

diffなどのOS付属ツールを使って、フルスレッドダンプの情報を、時系列で比較してください。

```
diff -u threaddump-1.txt threaddump-2.txt
```

図 44 フルスレッドダンプを比較するコマンドの例

比較の結果、変化が見られない場合は、ソケット接続の待ち受け(accept)、I/O待ち(readなど)、ロックの取得待ちなどで止まっているスレッドを探します。それらのスレッドが、アプリケーションのハングアップやスローダウンを引き起こしている可能性があります。

次の例では、「Thread-0」で呼び出したメソッドjava.net.ServerSocket#accept()が、ソケットの接続を待っている状態で停止していることが分かります。この待ち受け処理が適切かどうかなど、プログラムを見直す必要があります。

```
"Thread-0" prio=6 tid=0x0656dc00 nid=0x115c runnable [0x065df000..0x065dfd94]
  java.lang.Thread.State: RUNNABLE
    at java.net.PlainSocketImpl.socketAccept(Native Method)
    at java.net.PlainSocketImpl.accept(PlainSocketImpl.java:384)
    - locked <0x0fd28c48> (a java.net.SocksSocketImpl)
    at java.net.ServerSocket.implAccept(ServerSocket.java:453)
    at java.net.ServerSocket.accept(ServerSocket.java:421)
    at com.fujitsu.demo.Hangup$Worker.run(Hangup.java:36)
```

図 45 ソケット接続を待っているスレッドの例



#### ポイント

ネイティブメソッドを実行中のスレッドの状態は、「runnable」と表示されます。ネイティブメソッドの先で処理が遅延したり、停止したりしていても、Java VMはそれを検知できないからです。

「runnable」と表示されているスレッドであっても、ネイティブメソッドを実行中の場合は、ハングアップやスローダウンの可能性を検討してください。

## Java VisualVMを使った調査

フルスレッドダンプは、Java VisualVMを使って採取することもできます。詳しくは、JDKドキュメントを参照してください。

# 索引

<b>[A]</b>	
accesslog.....	16
access_log_levelパラメタ.....	21
access_log_level.....	28
access_log_policy.....	28
access_log_size.....	28
Active Directoryとの通信に関するトラブル.....	208
<b>[C]</b>	
cn_userexception_log.....	39
cn_userexception_log_size.....	41
cn_userexception_log_use.....	41
config.....	11,140,142
CORBAサービス運用中のシステム例外発生.....	140
CORBAサービス使用時の異常.....	140
CORBAサービスのIPCログの採取.....	36
CORBAサービスのスナップショットの採取.....	30
CORBAサービスのトレース情報の採取.....	4
CORBAサービスのネーミングサービスのユーザ例外ログの採取.....	39
CORBAサービスのログ情報の採取.....	15
CORBAワークユニットの標準出力/標準エラー出力がファイルに出力されない.....	145
<b>[D]</b>	
Dependency Injection失敗時の異常.....	114
<b>[E]</b>	
EJBアプリケーションの開発・運用時の異常.....	123
errlog.....	20
error_log_policy.....	28
error_log_size.....	29
EXTPのアンインストールが失敗する.....	159
<b>[F]</b>	
FJVMのトラブルシューティング機能.....	209
Fujitsu Enabler使用時の異常.....	210
<b>[G]</b>	
GCログの採取.....	321
GCログ分析のポイント.....	322
<b>[I]</b>	
IDLコンパイル.....	10,26
IIServer(J2EE)を使用する場合.....	317
infolog.....	20
info_log_policy.....	29
info_log_size.....	29
Interstage/ワークユニット停止時の調査資料の自動採取.....	44
Interstage HTTP Server運用時の異常.....	92
Interstage Java EE管理コンソール起動時の異常.....	123
Interstage Java EE管理コンソール操作中の異常.....	123
Interstage運用APIの異常発生時の対処.....	158
Interstage運用時の異常.....	72
Interstage管理コンソール起動時の異常.....	54
Interstage管理コンソール使用時の異常.....	168
Interstage管理コンソール操作中の異常.....	54
<b>[J]</b>	
J2EEアプリケーション開発・運用時の異常.....	101
Java EEアプリケーションで文字化けが発生する場合の対処.....	111
Java EE機能/Webサーバの使用時.....	75
Java EE使用時の異常.....	102
Java EEセキュリティの異常.....	138
Java VisualVMを使った調査.....	329,344,347
Java監視機能.....	265
Java監視機能のタブ.....	271
Java実行環境運用時の異常.....	209
Javaツール機能.....	211
jconsoleの終了.....	295
jconsole使用時の注意事項.....	263
jconsoleの起動.....	268
JDKに含まれるトラブルシューティングに役立つツール.....	258
jersey利用時の異常.....	119
jheap.....	247
jinfo使用時の注意事項.....	258,264
jmapコマンドを使った調査方法.....	335
jmap使用時の注意事項.....	259
JMSの開発・運用時の異常.....	131
JNDIアプリケーションで例外が発生する.....	201
jstack使用時の注意事項.....	259
jstatコマンドを使った調査方法.....	323
jstat使用時の注意事項.....	259
jvisualvm使用時の注意事項.....	260
<b>[L]</b>	
LDAPコマンドが正常に動作しない.....	199
Linuxにおけるコアダンプの採取.....	43
log_file_path.....	27
<b>[M]</b>	
MessageQueueDirector運用時の異常.....	193
Microsoft(R) Internet Information Servicesに関するトラブル.....	187
<b>[O]</b>	
odcntllog.....	27,29
odcvtrac.....	5
odformsnap.....	32
odformtrace.....	5,12
odfreesnap.....	32
odprthdrtrace.....	5



スキーマ拡張に関する異常.....	207
スナップ情報採取状況の確認.....	36
スナップ情報の採取.....	32
スナップ情報の出力形式.....	33
スナップ情報の分析.....	34
スナップショット機能.....	30
スレッドダンプが出力された場合の対処.....	134
スレッドダンプツール.....	250
制御ファイルの作成方法.....	213
性能監視ツール使用時の異常発生時の対処.....	156
性能情報が採取できない.....	156
性能情報のリアルタイム監視ができない.....	158
セッション情報管理機能を使用した場合の異常への対処.....	159
接続の解除.....	293
前提となる知識.....	321
その他の異常.....	63,129

[た]

大規模ファイル使用時の異常.....	116
注意事項.....	246,257,258,321
チュートリアル.....	320
追加プロパティに関する異常.....	118
デスクトップヒープが枯渇した場合.....	80
デッドロックの検出.....	345
データストアのイベントログにエラーメッセージが出力される.....	210
データ説明.....	295
データファイルを読めない.....	203
データベース連携サービス使用時の異常.....	160
テーブル作成コマンド実行時にエラーが発生する.....	204
テーブル作成コマンド実行時にデータベースのエラーが発生する.....	204
テーブル作成コマンドの応答がない.....	206
統合Windows認証に関するトラブル.....	182
動作原理.....	211
登録データが文字化けしている.....	202
トップエントリを操作できない.....	201
トラブル解決のための調査方法.....	81
トラブル事例.....	177
トラブル発生から対処までの流れ.....	175
トラブル発生時の対処.....	175
トレース情報採取の前準備.....	10
トレース情報出力の形式.....	227
トレース情報のテキスト出力.....	12
トレース情報の内容と採取コマンド.....	4
トレース情報の分析.....	14,230
トレース不可のクラス.....	232

[な]

認証サーバ間連携サービスに関するトラブル.....	187
認証に関するトラブル.....	178
認証に失敗する旨のメッセージが出力される場合.....	117
ネットワークエラー.....	141

[は]

パフォーマンスデータの収集.....	266
ハングアップやスローダウンの原因を調べる.....	344
ハングアップやスローダウンの調査.....	346
ヒープダンプの採取と分析.....	336

ヒープ分析機能.....	281
プロセスログ.....	16
プロセスログのデータ.....	17

[ま]

マシンのシステムダウン.....	162
マルチ言語サービス/J2EE互換機能の使用時.....	75
マルチサーバ管理の異常.....	81
メソッドサンプリング機能.....	272
メソッドトレース機能.....	211
メッセージ.....	242,303
メッセージの見方.....	81
メモリアークを分析する.....	330

[や]

ユーザ情報を登録するディレクトリサービスにActive Directoryを使用する場合のトラブル.....	185
よくある質問とその対処方法.....	98

[ら]

リソース管理プログラムの異常.....	164
リポジトリサーバに関するトラブル.....	184
リポジトリの応答が遅い.....	200
リポジトリの応答がない.....	201
リポジトリを起動できない.....	199
リークしているクラスの特定制定.....	330
ログイン時の異常.....	57,125
ログ採取環境(configファイルの設定).....	27
ログ採取環境(nsconfigファイルの設定).....	41
ログ採取環境の動的変更(odcntllogコマンド).....	29
ログ採取のためのアプリケーション開発.....	26
ログの異常.....	138
ログの出力先.....	176
ログのデータ.....	39

[わ]

ワークユニットが異常終了.....	152
ワークユニット起動コマンド無応答.....	151
ワークユニット起動失敗.....	149
ワークユニット使用時の異常発生時の対処.....	149
ワークユニット停止コマンド無応答.....	152