

# Cosminexus システム運用ガイド

手引・操作書

3020-3-M07-60

## マニュアルの購入方法

このマニュアル，および関連するマニュアルをご購入の際は，  
巻末の「ソフトウェアマニュアルのサービス ご案内」をご参  
照ください。

## 対象製品

適用 OS : Windows Server 2003 , Windows Server 2003 R2 , Windows Server 2003 ( x64 ) , Windows Server 2003 R2 ( x64 )

P-2443-7D74 uCosminexus Application Server Standard 07-60

P-2443-7K74 uCosminexus Application Server Enterprise 07-60

P-2443-7M74 uCosminexus Web Redirector 07-60

P-2443-7S74 uCosminexus Service Platform 07-60

適用 OS : Windows Server 2003 , Windows Server 2003 R2 , Windows Vista , Windows XP

P-2443-7E74 uCosminexus Developer Standard 07-60

P-2443-7F74 uCosminexus Developer Professional 07-60

P-2443-7T74 uCosminexus Service Architect 07-60

P-2443-7U74 uCosminexus Operator 07-60

適用 OS : Windows Server 2003 , Windows Server 2003 R2 , Windows Server 2003 ( x64 ) , Windows Server 2003 R2 ( x64 ) , Windows Vista , Windows XP

P-2443-7H74 uCosminexus Client 07-60

適用 OS : AIX 5L V5.2 , AIX 5L V5.3

P-1M43-7D71 uCosminexus Application Server Standard 07-60

P-1M43-7K71 uCosminexus Application Server Enterprise 07-60

P-1M43-7M71 uCosminexus Web Redirector 07-60

P-1M43-7S71 uCosminexus Service Platform 07-60

適用 OS : HP-UX 11i V2 ( IPF ) , HP-UX 11i V3 ( IPF )

P-1J43-7D71 uCosminexus Application Server Standard 07-60

P-1J43-7K71 uCosminexus Application Server Enterprise 07-60

P-1J43-7M71 uCosminexus Web Redirector 07-60

適用 OS : Red Hat Enterprise Linux AS 3 ( x86 ) , Red Hat Enterprise Linux AS 4 ( x86 ) , Red Hat Enterprise Linux ES 3 ( x86 ) , Red Hat Enterprise Linux ES 4 ( x86 ) , Red Hat Enterprise Linux AS 3 ( AMD64 & Intel EM64T ) , Red Hat Enterprise Linux AS 4 ( AMD64 & Intel EM64T ) , Red Hat Enterprise Linux ES 3 ( AMD64 & Intel EM64T ) , Red Hat Enterprise Linux ES 4 ( AMD64 & Intel EM64T )

P-9S43-7D71 uCosminexus Application Server Standard 07-60

P-9S43-7K71 uCosminexus Application Server Enterprise 07-60

P-9S43-7M71 uCosminexus Web Redirector 07-60

適用 OS : Red Hat Enterprise Linux AS 3 ( x86 ) , Red Hat Enterprise Linux AS 4 ( x86 ) , Red Hat Enterprise Linux 5 Advanced Platform ( x86 ) , Red Hat Enterprise Linux ES 3 ( x86 ) , Red Hat Enterprise Linux ES 4 ( x86 ) , Red Hat Enterprise Linux 5 ( x86 ) , Red Hat Enterprise Linux AS 3 ( AMD64 & Intel EM64T ) , Red Hat Enterprise Linux AS 4 ( AMD64 & Intel EM64T ) , Red Hat Enterprise Linux 5 Advanced Platform ( AMD/Intel 64 ) , Red Hat Enterprise Linux ES 3 ( AMD64 & Intel EM64T ) , Red Hat Enterprise Linux ES 4 ( AMD64 & Intel EM64T ) , Red Hat Enterprise Linux 5 ( AMD/Intel 64 )

P-9S43-7S71 uCosminexus Service Platform 07-60

適用 OS : Red Hat Enterprise Linux AS 3 ( IPF ) , Red Hat Enterprise Linux AS 4 ( IPF ) , Red Hat Enterprise Linux 5 Advanced Platform ( Intel Itanium )

P-9V43-7D71 uCosminexus Application Server Standard 07-60

P-9V43-7K71 uCosminexus Application Server Enterprise 07-60

P-9V43-7M71 uCosminexus Web Redirector 07-60

適用 OS : Solaris 9 , Solaris 10

P-9D43-7D71 uCosminexus Application Server Standard 07-60

P-9D43-7K71 uCosminexus Application Server Enterprise 07-60

P-9D43-7M71 uCosminexus Web Redirector 07-60

P-9D43-7S71 uCosminexus Service Platform 07-60

印の製品については、サポート時期をご確認ください。

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

本製品では日立トレース共通ライブラリをインストールします。

## 輸出時の注意

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

なお、ご不明な場合は、弊社担当営業にお問い合わせください。

## 商標類

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

AMD は、Advanced Micro Devices, Inc. の商標です。

cBank, cFramework, cContentsTemplate, cInputCheck, cLogWrapper, cPersistence, cSQLBuilder, cValidator, Eclipse Plugin For cFramework は、株式会社イーシー・ワンの日本における商標または登録商標です。

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

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

IIOP は、OMG 仕様による ORB(Object Request Broker) 間通信のネットワークプロトコルの名称です。

Intel は、Intel Corporation の会社名です。

Internet Explorer は、米国 Microsoft Corporation の米国及びその他の国における登録商標または商標です。

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

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

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

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

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

Microsoft Internet Information Services は、米国 Microsoft Corp. の商品名称です。

Microsoft Office Excel は、米国 Microsoft Corporation の商品名称です。

OMG, CORBA, IIOP, UML, Unified Modeling Language, MDA, Model Driven Architecture は、Object Management Group, Inc. の米国及びその他の国における登録商標または商標です。

ORACLE は、米国 Oracle Corporation の登録商標です。

Oracle は、米国 Oracle Corporation 及びその子会社、関連会社の登録商標です。

Oracle8i は、米国 Oracle Corporation の商標です。

Oracle9i は、米国 Oracle Corporation の商標です。

Oracle 10g は、米国 Oracle Corporation の商標です。

PA-RISC は、米国 Hewlett-Packard Company の商標です。

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

SOAP ( Simple Object Access Protocol ) は、分散ネットワーク環境において XML ベースの情報を交換するための通信プロトコルの名称です。

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

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

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

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

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

Windows Server は、米国 Microsoft Corporation の米国及びその他の国における登録商標です。

Windows Vista は、米国 Microsoft Corporation の米国及びその他の国における登録商標です。

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

Eclipse は、開発ツールプロバイダのオープンコミュニティである Eclipse Foundation, Inc. により構築された開発ツール統合のためのオープンプラットフォームです。

This product includes software developed by the Apache Software Foundation (<http://www.apache.org/>).

プログラムプロダクト「P-9D43-7D71, P-9D43-7K71, P-9D43-7M71, P-9D43-7S71」には、米国 Sun Microsystems, Inc. が著作権を有している部分が含まれています。

プログラムプロダクト「P-9D43-7D71, P-9D43-7K71, P-9D43-7M71, P-9D43-7S71」には、UNIX System Laboratories, Inc. が著作権を有している部分が含まれています。

## 発行

2006 年 4 月 (第 1 版) 3020-3-M07

2007 年 12 月 (第 4 版) 3020-3-M07-60

## 著作権

All Rights Reserved. Copyright (C) 2006, 2007, Hitachi, Ltd.

## 変更内容

変更内容 ( 3020-3-M07-60 ) uCosminexus Application Server Standard 07-60 , uCosminexus Developer Standard 07-60 , uCosminexus Developer Professional 07-60 , uCosminexus Client 07-60 , uCosminexus Application Server Enterprise 07-60 , uCosminexus Web Redirector 07-60 , uCosminexus Service Platform 07-60 , uCosminexus Service Architect 07-60 , uCosminexus Operator 07-60

追加・変更内容	変更箇所
Application Server で、バッチ処理を実行するアプリケーションの実行基盤をサポートした。これに伴い、バッチアプリケーションおよびバッチサーバに関する説明を追加した。	1.2 , 1.3.2 , 1.5 , 1.5.1 , 1.7.2 , 1.8.1 , 2.1 , 2.4 , 2.4.1 , 2.4.2 , 3.1 , 3.2.2 , 3.3.4 , 3.3.6 , 3.3.8 , 5.4 , 5.4.2 , 5.5.3 , 5.5.5 , 5.5.10 , 6.2.3 , 6.3.2 , 6.4.2 , 7.2.1 , 8.1 , 8.1.4 , 9.1 , 9.1.4 , 9.2 , 9.2.4 , 付録 A , 付録 A.15 , 付録 B.1
Solaris の記述を追加した。	2.3.3(1) , 2.5.2(2) , 5.4.1(1)(a) , 5.4.2(1)(a) , 5.4.7(2)(b) , 5.4.10(1)(b) , 5.4.10(2)(b) , 5.4.11(2) , 5.5.8(2)(a) , 5.6.5(1) , 9.1 , 9.2 , 9.3 , 付録 G.9(5)(b)
Connector 1.5 仕様に準拠したリソースアダプタを使用した場合の稼働情報ファイルでの稼働情報の監視について、情報を追加した。	2.3.3(7) , 2.4.2
メモリ枯渇監視情報の計算式を変更した。	2.5.2(1)
トレース情報を出力する次の機能レイヤを追加した。 <ul style="list-style-type: none"> <li>バッチアプリケーション実行機能</li> <li>uCosminexus Service Coordinator Interactive Workflow</li> <li>uCosminexus Service Adapter for Flat Files</li> </ul>	3.1 , 付録 A
トラブルシューティングに必要な資料として、運用管理エージェントで行うRMI 処理での保守ログを追加した。	5.4.1(1)
セッションフェイルオーバ機能のログに出力される内容を追加した。	5.6.3(3)
Web コンテナのトレース取得ポイント (セッションフェイルオーバ機能のトレース) を追加した。	付録 A.6
Message-driven Bean ( EJB2.1 ) のトレース取得ポイントを追加した。	付録 A.7(3)
コネクション自動クローズ機能を使用する場合のトレース取得ポイントを追加した。	付録 A.10(4)
Connector1.5 に準拠したリソースアダプタでワーク管理を使用する場合のトレース取得ポイントを追加した。	付録 A.10(6)
snapshot ログ収集対象ディレクトリパスを追加した。	付録 B.1 , 付録 B.6

追加・変更内容	変更箇所
次の製品の適用 OS に、Red Hat Enterprise Linux 5 Advanced Platform ( x86 ), Red Hat Enterprise Linux 5 ( x86 ), Red Hat Enterprise Linux 5 Advanced Platform ( AMD/Intel 64 ), および Red Hat Enterprise Linux 5 ( AMD/Intel 64 ) を追加した。 <ul style="list-style-type: none"> <li>• uCosminexus Service Platform</li> </ul>	-
次の製品の適用 OS に Linux ( IPF ) を追加した。 <ul style="list-style-type: none"> <li>• uCosminexus Application Server Standard</li> <li>• uCosminexus Application Server Enterprise</li> <li>• uCosminexus Web Redirector</li> </ul>	-
次の製品の適用 OS に Solaris を追加した。 <ul style="list-style-type: none"> <li>• uCosminexus Application Server Standard</li> <li>• uCosminexus Application Server Enterprise</li> <li>• uCosminexus Web Redirector</li> <li>• uCosminexus Service Platform</li> </ul>	-
HP-UX ( PA-RISC ) のサポート中止に伴い、この OS に適応していた次の製品の記述を削除した。 <ul style="list-style-type: none"> <li>• uCosminexus Application Server Standard</li> <li>• uCosminexus Application Server Enterprise</li> <li>• uCosminexus Web Redirector</li> </ul>	-
次の製品の適用 OS から HP-UX ( IPF ) を削除した。 <ul style="list-style-type: none"> <li>• uCosminexus Service Platform</li> </ul>	-

単なる誤字・脱字などはお断りなく訂正しました。

変更内容 ( 3020-3-M07-40 ) uCosminexus Application Server Standard 07-50 , uCosminexus Developer Standard 07-50 , uCosminexus Developer Professional 07-50 , uCosminexus Client 07-50 , uCosminexus Application Server Enterprise 07-50 , uCosminexus Web Redirector 07-50 , uCosminexus Service Platform 07-50 , uCosminexus Service Architect 07-50 , uCosminexus Operator 07-50

追加・変更内容
Management Server の運用管理ポータルを使用して構築したシステム、またはファイルの編集とコマンドの実行で構築したシステムに関する記述を削除した。
統合ユーザ管理に関する記述を削除した。
システムの監査を支援する機能を追加した。
運用管理ポータルでの運用作業に関する記述を削除した。
CTM のスケジュールキューの滞留監視に関する記述を削除した。
PRF トレース取得レベルの確認、および変更に関する記述を削除した。
業務アプリケーションの閉塞のうち、次の方法に関する記述を削除した。 <ul style="list-style-type: none"> <li>• Web サーバのサービス閉塞</li> <li>• J2EE サーバの停止によるサービス閉塞</li> </ul>
業務アプリケーションの閉塞解除に関する記述を削除した。

---

追加・変更内容

---

Web アプリケーションのサービス部分閉塞を利用した業務アプリケーションの入れ替えに関する記述を削除した。

---

Windows Vista を使用している場合の注意事項を追加した。

---

コマンド実行時に指定する Web システム名、サービスユニット名、およびサーバ名に関する注意事項を追加した。

---

CTM コマンドを使用した、CTM の稼働統計情報の確認に関する記述を削除した。

---

リソースアダプタを J2EE アプリケーションに含めてデPLOY する機能を追加した。

---

PRF コマンドを使用した性能解析トレースファイルの出力 / 収集に関する記述を削除した。

---

セキュリティアナライザ、およびセキュリティフィルタに関する記述を削除した。

---

CTM のスケジュールキューの稼働状況の確認と変更の手段、手順に関する記述を追加・変更した。

---

CTM を利用したサービス閉塞の手段のうち、CTM コマンドでの手段に関する記述を削除した。

---

業務アプリケーションの停止の手段のうち、運用管理コマンド、運用管理ポータルでの手段に関する記述を削除した。

---

CTM を利用したオンライン状態での業務アプリケーションの入れ替えの手段のうち、CTM のコマンドでの手段に関する記述を削除した。

---

snapshot ログの取得の手段のうち、運用管理ポータルおよび snapshotlog コマンドでの手段に関する記述を削除した。

---

usrconfig.properties, usrconfig.cfg で指定するキーを簡易構築定義ファイルのパラメタに置き換えた。

---

Cosminexus のログに次のログを追加した。

- J2EE サーバに直接デPLOY したリソースアダプタの稼働ログ
  - アプリケーションに含めてデPLOY したリソースアダプタの稼働ログ
  - 監査ログのメッセージログ
  - 監査ログの例外ログ
  - mngenvsetup コマンドの実行ログ
  - cosmienv コマンドのログ
- 

PRF トレースを出力する関連プログラムに、uCosminexus Service Adapter for Object Access を追加した。

---

セッショントレースの場合の Web コンテナのトレース取得ポイント、取得レベル、および取得できるトレース情報を追加、修正した。

---

DB Connector RM と連携する場合に、次のイベント ID でコネクション ID を取得できるようにした。

- 0x8D61
  - 0x8D63
  - 0x8D80
- 

Cosminexus システムで実行できる運用作業の一覧を追加した。

---

JP1/PAM, JP1/Cm2/NNM, および JP1/PFM/SSO と連携する機能を削除した。

---

AIX 5L V5.1 のサポート中止に伴い、この OS に適応していた次の製品の記述を削除した。

- uCosminexus Application Server Enterprise
  - uCosminexus Application Server Standard
  - uCosminexus Service Platform
  - uCosminexus Web Redirector
-

---

## 追加・変更内容

---

Microsoft(R) Windows(R) 2000 Advanced Server Operating System , Microsoft(R) Windows(R) 2000 Datacenter Server Operating System , および Microsoft(R) Windows(R) 2000 Server Operating System のサポート中止に伴い、これらの OS に適応していた次の製品の記述を削除した。

- uCosminexus Application Server Enterprise
- uCosminexus Application Server Standard
- uCosminexus Client
- uCosminexus Developer Professional
- uCosminexus Developer Standard
- uCosminexus Operator
- uCosminexus Service Architect
- uCosminexus Service Platform
- uCosminexus Web Redirector

---

Microsoft(R) Windows(R) 2000 Professional Operating System のサポート中止に伴い、この OS に適応していた次の製品の記述を削除した。

- uCosminexus Client
- uCosminexus Developer Professional
- uCosminexus Developer Standard
- uCosminexus Operator
- uCosminexus Service Architect

---

次の製品の適用 OS に、HP-UX ( IPF ) を追加した。

- uCosminexus Service Platform

---

次の製品の適用 OS に、HP-UX 11i V3 ( IPF ) を追加した。

- uCosminexus Application Server Enterprise
- uCosminexus Application Server Standard
- uCosminexus Web Redirector

---

次の製品の適用 OS に、Red Hat Enterprise Linux ES 3 ( AMD64 & Intel EM64T ) , および Red Hat Enterprise Linux ES 4 ( AMD64 & Intel EM64T ) を追加した。

- uCosminexus Application Server Enterprise
- uCosminexus Application Server Standard
- uCosminexus Service Platform
- uCosminexus Web Redirector

---

次の製品の適用 OS に、Windows Vista を追加した。

- uCosminexus Client
  - uCosminexus Developer Professional
  - uCosminexus Developer Standard
  - uCosminexus Operator
  - uCosminexus Service Architect
- 

第 3 版では、第 2 版の内容の一部を、マニュアル「Cosminexus 運用管理操作ガイド」または「Cosminexus リファレンス コマンド編」に移動し、目次の構成を変更しました。第 2 版との対応は次のようになっています。

---

第 2 版	第 3 版
1. システム運用の概要	1. システム運用の概要
2. システムの運用監視	2. システムの運用監視

---



第 2 版	第 3 版
3. システムの処理性能の解析	3. システムの処理性能の解析
4. 統合ユーザ管理のリソースの監視	マニュアル「Cosminexus 運用管理操作ガイド」に移動
5. 業務アプリケーションとリソースの管理	4. 業務アプリケーションとリソースの管理
6. トラブルシューティング	5. トラブルシューティング
-	6. 監査ログを使用したシステムの監査
-	7. データベース監査証跡と連携した運用
7. 運用管理ポータルで構築したシステムの起動と停止（Windows の場合）	マニュアル「Cosminexus 運用管理操作ガイド」に移動
8. 運用管理ポータルで構築したシステムの起動と停止（UNIX の場合）	マニュアル「Cosminexus 運用管理操作ガイド」に移動
9. ファイル編集とコマンド実行で構築したシステムの起動と停止（Windows の場合）	マニュアル「Cosminexus リファレンス コマンド編」に移動
10. ファイル編集とコマンド実行で構築したシステムの起動と停止（UNIX の場合）	マニュアル「Cosminexus リファレンス コマンド編」に移動
11. クラスタソフトウェアを使用したシステムの起動と停止（Windows の場合）	8. クラスタソフトウェアを使用したシステムの起動と停止（Windows の場合）
12. クラスタソフトウェアを使用したシステムの起動と停止（UNIX の場合）	9. クラスタソフトウェアを使用したシステムの起動と停止（UNIX の場合）
付録 A トレース取得ポイントと PRF トレース取得レベル	付録 A トレース取得ポイントと PRF トレース取得レベル
付録 B snapshot ログの収集対象一覧	付録 B snapshot ログの収集対象一覧
付録 C データベースと接続中にトラブルが発生したコネクションの特定	付録 C データベースと接続中にトラブルが発生したコネクションの特定
付録 D 機能ごとに出力されるログ情報	付録 D 機能ごとに出力されるログ情報
付録 E 障害発生時の CMR 用の表の回復	付録 E 障害発生時の CMR 用の表の回復
付録 F ベーシックモードの利用（互換機能）	付録 F ベーシックモードの利用（互換機能）
付録 G サーブレットエンジンモードでのシステム運用	付録 G サーブレットエンジンモードでのシステム運用
-	付録 H 代替手段で実行できる運用作業
付録 H 用語解説	付録 I 用語解説

変更内容（3020-3-M07-20）uCosminexus Application Server Standard 07-10，uCosminexus Application Server Enterprise 07-10，uCosminexus Developer Standard 07-10，uCosminexus Developer Professional 07-10，uCosminexus Client 07-10，uCosminexus Web Redirector 07-10，

uCosminexus Service Platform 07-10 , uCosminexus Service Architect 07-10 , uCosminexus Operator 07-10

---

追加・変更機能

---

タイムアウト発生時に、メッセージ「KDJE50080-W」が出力される機能を追加した。

---

性能解析トレースの取得ポイントに、イベント ID「0x8C41」を追加した。

---

業務アプリケーションでスローダウンやハングアップが発生したときに、性能解析トレースファイルとスレッドダンプを対応づけて問題個所を調査する方法を追加した。

---

Server Plug-in を使用して業務アプリケーションの名称を変更する機能を追加した。

---

運用管理エージェントが出力するログに、「運用管理エージェントの標準エラー出力コマンドライン」を追加した。

---

EJB クライアントアプリケーションが出力するログに、次のログを追加した。

- ユーザ出力ログ
  - ユーザエラーログ
  - JavaVM の保守情報、ガーベージコレクションのログ
  - EJB コンテナの保守情報
  - 起動プロセス標準出力情報
  - 起動プロセス標準エラー情報
- 

EJB クライアントアプリケーションのスレッドダンプを取得できる機能を追加した。

---

EJB クライアントアプリケーションのログの種別についての記述を追加した。

---

性能解析トレースの取得ポイントに、次のイベント ID を追加した。

- 0x8490
  - 0x8491
- 

uCosminexus Application Server Standard , uCosminexus Application Server Enterprise および uCosminexus Web Redirector の適用 OS に、HP-UX ( PA-RISC ) を追加した。

---

uCosminexus Application Server Standard , uCosminexus Application Server Enterprise および uCosminexus Web Redirector の適用 OS に、Linux ( IPF ) を追加した。

---

uCosminexus Service Platform の適用 OS に、AIX を追加した。

---

# はじめに

---

このマニュアルは、Cosminexus（コズミネクサス）を使用して構築したアプリケーションサーバの運用のうち、Smart Composer 機能を使用する運用以外の方法について説明したものです。Cosminexus では、次に示すプログラムプロダクトを使用してアプリケーションサーバを運用します。なお、これらのプログラムプロダクトを使用して構築したシステムを、Cosminexus システムといいます。

- P-1J43-7D71 uCosminexus Application Server Standard
- P-1J43-7K71 uCosminexus Application Server Enterprise
- P-1J43-7M71 uCosminexus Web Redirector
- P-1M43-7D71 uCosminexus Application Server Standard
- P-1M43-7K71 uCosminexus Application Server Enterprise
- P-1M43-7M71 uCosminexus Web Redirector
- P-1M43-7S71 uCosminexus Service Platform
- P-2443-7D74 uCosminexus Application Server Standard
- P-2443-7E74 uCosminexus Developer Standard
- P-2443-7F74 uCosminexus Developer Professional
- P-2443-7H74 uCosminexus Client
- P-2443-7K74 uCosminexus Application Server Enterprise
- P-2443-7M74 uCosminexus Web Redirector
- P-2443-7S74 uCosminexus Service Platform
- P-2443-7T74 uCosminexus Service Architect
- P-2443-7U74 uCosminexus Operator
- P-9D43-7D71 uCosminexus Application Server Standard
- P-9D43-7K71 uCosminexus Application Server Enterprise
- P-9D43-7M71 uCosminexus Web Redirector
- P-9D43-7S71 uCosminexus Service Platform
- P-9S43-7D71 uCosminexus Application Server Standard
- P-9S43-7K71 uCosminexus Application Server Enterprise
- P-9S43-7M71 uCosminexus Web Redirector
- P-9S43-7S71 uCosminexus Service Platform
- P-9V43-7D71 uCosminexus Application Server Standard
- P-9V43-7K71 uCosminexus Application Server Enterprise
- P-9V43-7M71 uCosminexus Web Redirector

このマニュアルでは、これらのプログラムプロダクトの構成ソフトウェアのうち、次に示す構成ソフトウェアについて説明しています。

- Cosminexus Component Container
- Cosminexus Component Container - Client
- Cosminexus Operator Plug-in

はじめに

- Cosminexus Component Container - Redirector
- Cosminexus Component Transaction Monitor
- Cosminexus DABroker Library
- Cosminexus Developer's Kit for Java
- Cosminexus Performance Tracer
- Cosminexus TPBroker

なお、オペレーティングシステム（OS）の種類によって、機能が異なる場合があります。OSごとの違いがある場合の表記方法については、「適用 OS の違いによる機能相違点の表記」を参照してください。

## 対象読者

このマニュアルは、Cosminexus のシステムを運用する方を対象にしています。

次の内容を理解されていることを前提としています。

- Windows または UNIX のシステム構築および運用に関する知識
- J2EE に関する知識
- SQL およびリレーショナルデータベースに関する基本的な知識
- CORBA に関する基本的な知識

JP1 連携機能を使用する場合は、次の内容も理解されていることを前提とします。

- JP1 の統合管理、ジョブ管理、ネットワーク管理およびアベイラビリティ管理に関する基本的な知識

Microsoft Cluster Service 連携機能を使用する場合は、次の内容も理解されていることを前提とします。

- Microsoft Cluster Service を使用したクラスタ構成に関する基本的な知識

HA モニタ連携機能を使用する場合は、次の内容も理解されていることを前提とします。

- HA モニタを使用したクラスタ構成に関する基本的な知識

また、このマニュアルは、マニュアル「Cosminexus 機能解説」、マニュアル「Cosminexus システム設計ガイド」を理解していることを前提としていますので、あらかじめお読みいただくことをお勧めします。

## マニュアルの構成

このマニュアルは、次に示す編と付録から構成されています。

### 第 1 編 概要

Cosminexus システムの運用方法の種類や運用サイクルなど、システム運用の概要について説明しています。

### 第 2 編 システムの監視

システム監視に関する作業について説明しています。

### 第 3 編 システムの保守

システム保守に関する作業について説明しています。

### 第 4 編 システムの監査

システム監査に関する作業について説明しています。

### 第 5 編 システムの起動と停止

クライアントソフトウェアを使用したシステムの起動方法と停止方法について説明しています。

### 付録 A トレース取得ポイントと PRF トレース取得レベル

トレース情報を取得できる機能レイヤとトレース取得ポイント、および PRF トレース取得レベルについて説明しています。

### 付録 B snapshot ログの収集対象一覧

snapshot ログの収集対象のディレクトリについて説明しています。

### 付録 C データベースと接続中にトラブルが発生した接続の特定

データベース (HiRDB, または Oracle) との接続中にトラブルが発生した場合に、性能解析トレースファイルを利用して、トラブルが発生した接続を特定する方法について説明しています。

### 付録 D 機能ごとに出力されるログ情報

特定の機能を利用したときに出力されるログ情報について説明しています。

### 付録 E 障害発生時の CMR 用の表の回復

CMR を含むアプリケーションの回復方法について説明しています。

### 付録 F ベーシックモードの利用 (互換機能)

ベーシックモードを使用した場合のシステムの運用方法について説明しています。

### 付録 G サーブレットエンジンモードでのシステム運用

サーブレットエンジンモードを使用した場合のシステムの運用方法について説明しています。

### 付録 H 代替手段で実行できる運用作業

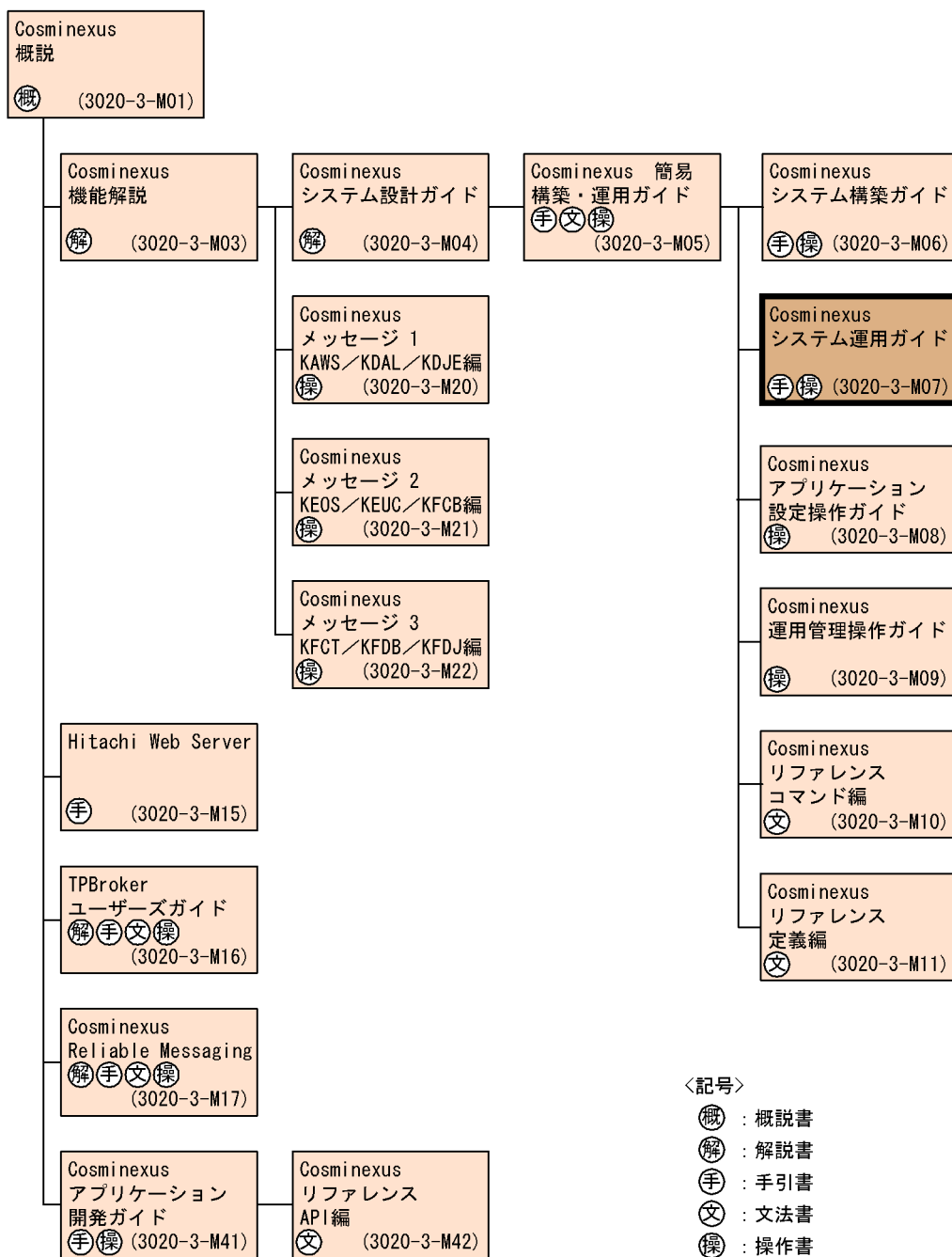
代替手段を利用して実行できる運用作業について説明しています。

### 付録 I 用語解説

このマニュアルで使用している用語の意味について説明しています。

## 関連マニュアル

Cosminexus のマニュアル体系について、次の図に示します。



マニュアル体系図で示した関連マニュアルについて、それぞれの位置づけを次に示します。

### Cosminexus 概説

Cosminexus の製品概要について説明しています。

### Cosminexus 機能解説

Cosminexus Component Container の機能を中心に、uCosminexus Application Server の概要と提供する機能について説明しています。

### Cosminexus システム設計ガイド

システム設計時に、システムの目的に応じたシステム構成や運用方法を検討するための指針について説明しています。また、チューニングの方法についても説明しています。

### Cosminexus 簡易構築・運用ガイド

セットアップウィザードおよび Smart Composer 機能を使用して、システムを構築・運用する手順について説明しています。また、セットアップウィザードおよび Smart Composer 機能が提供するコマンドやファイルについても説明しています。

### Cosminexus システム構築ガイド

システム構築時に必要な機能の設定方法について説明しています。

### Cosminexus アプリケーション設定操作ガイド

Cosminexus Component Container のサーバ管理コマンド、および Server Plug-in を使用した操作について説明しています。

### Cosminexus 運用管理操作ガイド

Cosminexus Component Container の運用管理ポータルの使用方法について説明しています。

### Cosminexus リファレンス コマンド編

Cosminexus のシステムを構築・運用するときに使用するコマンドについて説明しています。

### Cosminexus リファレンス 定義編

Cosminexus のシステムを構築・運用するとき、またはアプリケーションを開発するとき、使用するファイルの形式について説明しています。

### Cosminexus メッセージ 1 KAWS / KDAL / KDJE 編, Cosminexus メッセージ 2 KEOS / KEUC / KFCB 編, Cosminexus メッセージ 3 KFCT / KFDB / KFDJ 編

Cosminexus で出力されるメッセージについて説明しています。

### Hitachi Web Server

Hitachi Web Server (Web サーバ) の構築、管理方法について説明しています。

### TPBroker ユーザーズガイド

Cosminexus TPBroker の概要、機能、運用方法について説明しています。

### Cosminexus Reliable Messaging

Cosminexus RM を使用したメッセージの非同期通信によるアプリケーションの連携方法について説明しています。

### Cosminexus アプリケーション開発ガイド

構築した Cosminexus のシステムで動作させる、アプリケーションの開発方法について説

明しています。

#### Cosminexus リファレンス API 編

アプリケーションの開発で使用する API およびタグについて説明しています。

また、マニュアル体系図に示したマニュアル以外で、このマニュアルと関連するマニュアルを次に示します。必要に応じてお読みください。

- Cosminexus SOAP アプリケーション開発ガイド (3020-3-M47)
- Cosminexus Web サービスセキュリティ 使用の手引 (3020-3-M48)
- スケーラブルデータベースサーバ HiRDB Version 7 コマンドリファレンス (UNIX(R) 用) (3000-6-275)
- スケーラブルデータベースサーバ HiRDB Version 7 UAP 開発ガイド (UNIX(R)/Windows(R) 用) (3000-6-276)
- スケーラブルデータベースサーバ HiRDB Version 8 システム運用ガイド (UNIX(R) 用) (3000-6-354)
- スケーラブルデータベースサーバ HiRDB Version 8 コマンドリファレンス (UNIX(R) 用) (3000-6-355)
- スケーラブルデータベースサーバ HiRDB Version 7 コマンドリファレンス (Windows(R) 用) (3020-6-275)
- スケーラブルデータベースサーバ HiRDB Version 7 UAP 開発ガイド (Windows(R) 用) (3020-6-276)
- スケーラブルデータベースサーバ HiRDB Version 8 システム運用ガイド (Windows(R) 用) (3020-6-354)
- スケーラブルデータベースサーバ HiRDB Version 8 コマンドリファレンス (Windows(R) 用) (3020-6-355)
- スケーラブルデータベースサーバ HiRDB Version 8 UAP 開発ガイド (3020-6-356)
- JP1 Version 7i JP1/Base (3020-3-F04)
- JP1 Version 7i JP1/Automatic Job Management System 2 操作ガイド (3020-3-F09)
- JP1 Version 8 JP1/Automatic Job Management System 2 操作ガイド (3020-3-K24)
- 高信頼化システム監視機能 HA モニタ AIX(R) 編 (3000-9-130)
- 高信頼化システム監視機能 HA モニタ HP-UX 編 (3000-9-131)
- 高信頼化システム監視機能 HA モニタ Linux(R) 編 (3000-9-132)

なお、このマニュアルでは、次のマニュアルについて、対象 OS およびバージョン番号を省略して表記しています。マニュアルの正式名称とこのマニュアルでの表記を次の表に示します。

正式名称	このマニュアルでの表記
TPBroker Version 5 TPBroker ユーザーズガイド	TPBroker ユーザーズガイド
JP1 Version 7i JP1/Base	JP1/Base
JP1 Version 8 JP1/Automatic Job Management System 2 操作ガイド	JP1/Automatic Job Management System 2 操作ガイド
JP1 Version 7i JP1/Automatic Job Management System 2 操作ガイド	



正式名称	このマニュアルでの表記
スケーラブルデータベースサーバ HiRDB Version 8 UAP 開発ガイド	HiRDB UAP 開発ガイド
スケーラブルデータベースサーバ HiRDB Version 7 UAP 開発ガイド ( UNIX(R)/Windows(R) 用 )	
スケーラブルデータベースサーバ HiRDB Version 7 UAP 開発ガイド ( Windows(R) 用 )	
スケーラブルデータベースサーバ HiRDB Version 8 システム運用ガイド ( UNIX(R) 用 )	HiRDB システム運用ガイド
スケーラブルデータベースサーバ HiRDB Version 8 システム運用ガイド ( Windows(R) 用 )	
スケーラブルデータベースサーバ HiRDB Version 8 コマンドリファレンス ( UNIX(R) 用 )	HiRDB コマンドリファレンス
スケーラブルデータベースサーバ HiRDB Version 8 コマンドリファレンス ( Windows(R) 用 )	
スケーラブルデータベースサーバ HiRDB Version 7 コマンドリファレンス ( UNIX(R) 用 )	
スケーラブルデータベースサーバ HiRDB Version 7 コマンドリファレンス ( Windows(R) 用 )	
高信頼化システム監視機能 HA モニタ AIX(R) 編	高信頼化システム監視機能 HA モニタ
高信頼化システム監視機能 HA モニタ HP-UX 編	
高信頼化システム監視機能 HA モニタ Linux(R) 編	

## 読書手順

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

マニュアルを読む目的		記述箇所
システムの運用方法の概要について知りたい。	システムの運用方法の種類について知りたい。	第 1 編 1 章
システムの監視について知りたい。	システムで運用中に監視できる項目と監視方法について知りたい。	第 2 編 2 章
	性能解析トレースの収集方法および利用方法について知りたい。	第 2 編 3 章
システムの保守について知りたい。	J2EE アプリケーションとリソースの管理方法について知りたい。	第 3 編 4 章
	トラブルシューティングについて知りたい。	第 3 編 5 章
システムの監査について知りたい。	システムの監査で利用できる監査ログについて知りたい。	第 4 編 6 章
	データベースの監査証跡と連携した場合の運用について知りたい。	第 4 編 7 章

マニュアルを読む目的		記述箇所
システムの起動と停止について知りたい。	Windows を使用している場合の、クラスタソフトウェアを使用したシステムの起動と停止の手順について知りたい。	第 5 編 8 章
	UNIX を使用している場合の、クラスタソフトウェアを使用したシステムの起動と停止の手順について知りたい。	第 5 編 9 章
トレース取得ポイントと PRF トレース取得レベルについて知りたい。	システムの処理性能を解析する場合の性能解析トレース取得ポイントと PRF トレースの取得レベルについて知りたい。	付録 A
障害発生時の回復方法について知りたい。	snapshot ログの収集対象について知りたい。	付録 B
	CMR を含むアプリケーションの回復方法について知りたい。	付録 E
データベース関連の障害発生時の対処方法について知りたい。	データベース (HiRDB または Oracle) との接続中にトラブルが発生した接続の確認方法について知りたい。	付録 C
出力されるログ情報について知りたい。	特定の機能を使用している場合に出力されるログ情報について知りたい。	付録 D
互換用機能について知りたい。	ベーシックモードでのシステムの運用方法について知りたい。	付録 F
	サブレットエンジンモードでのシステムの運用方法について知りたい。	付録 G
運用作業の代替手段について知りたい。	運用作業で利用できる代替手段について知りたい。	付録 H
用語について知りたい。	このマニュアルで使用している用語について知りたい。	付録 I

## ご利用製品ごとの用語の読み替えについて

ご利用の製品によっては、マニュアルで使用している用語を、ご利用の製品名に読み替える必要があります。

次の表に従って、マニュアルで使用している用語をご利用の製品名に読み替えてください。

ご利用の製品名	マニュアルで使用している用語
uCosminexus Developer Professional <sup>1</sup>	Application Server および Application Server Enterprise
uCosminexus Developer Standard <sup>1 2</sup>	Application Server
uCosminexus Service Architect <sup>1</sup>	Application Server および Application Server Enterprise
uCosminexus Service Platform	

注 1 テスト環境で使用している場合にだけ読み替えが必要です。

注 2 uCosminexus Developer Standard と Application Server には一部機能差があります。

機能差については、マニュアル「Cosminexus アプリケーション開発ガイド」の Developer Standard 使用時の注意事項に関する説明を参照してください。

## このマニュアルでの表記

このマニュアルで使用している表記と、対応する製品名を次に示します。

表記		製品名
Application Server	Application Server Enterprise	uCosminexus Application Server Enterprise
	Application Server Standard	uCosminexus Application Server Standard
Developer	Developer Professional	uCosminexus Developer Professional
	Developer Standard	uCosminexus Developer Standard
Excel		Microsoft(R) Office Excel
HiRDB または HiRDB サーバ	HiRDB/Parallel Server	HiRDB/Parallel Server Version 7
		HiRDB/Parallel Server Version 8
	HiRDB/Single Server	HiRDB/Single Server Version 7
		HiRDB/Single Server Version 8
HiRDB Run Time または HiRDB クライアント		HiRDB/Run Time Version 7
		HiRDB/Run Time Version 8
Internet Explorer		Microsoft(R) Internet Explorer(R)
IPF		Itanium(R) Processor Family
JP1/AJS2	JP1/AJS2 - Agent	JP1/Automatic Job Management System 2 - Agent
	JP1/AJS2 - Light Edition	JP1/Automatic Job Management System 2 - Light Edition
	JP1/AJS2 - Manager	JP1/Automatic Job Management System 2 - Manager
	JP1/AJS2 - View	JP1/Automatic Job Management System 2 - View
Microsoft Cluster Service		Microsoft(R) Cluster Service
Microsoft IIS	Microsoft IIS 6.0	Microsoft(R) Internet Information Services 6.0
Oracle	Oracle9i	Oracle9i
		Oracle9i R2
	Oracle10g	Oracle 10g
		Oracle 10g R2
UNIX	AIX	AIX 5L V5.2
		AIX 5L V5.3
	HP-UX または HP-UX ( IPF )	HP-UX 11i V2 ( IPF )
		HP-UX 11i V3 ( IPF )
Linux	Linux ( IPF )	Red Hat Enterprise Linux AS 3 ( IPF )

表記		製品名	
		Red Hat Enterprise Linux AS 4 ( IPF )	
		Red Hat Enterprise Linux 5 Advanced Platform ( Intel Itanium )	
		Linux ( x86 / AMD64 & Intel EM64T )	Red Hat Enterprise Linux AS 3 ( x86 )
		Red Hat Enterprise Linux AS 4 ( x86 )	
		Red Hat Enterprise Linux 5 Advanced Platform ( x86 )	
		Red Hat Enterprise Linux ES 3 ( x86 )	
		Red Hat Enterprise Linux ES 4 ( x86 )	
		Red Hat Enterprise Linux 5 ( x86 )	
		Red Hat Enterprise Linux AS 3 ( AMD64 & Intel EM64T )	
		Red Hat Enterprise Linux AS 4 ( AMD64 & Intel EM64T )	
		Red Hat Enterprise Linux 5 Advanced Platform ( AMD/Intel 64 )	
		Red Hat Enterprise Linux ES 3 ( AMD64 & Intel EM64T )	
		Red Hat Enterprise Linux ES 4 ( AMD64 & Intel EM64T )	
		Red Hat Enterprise Linux 5 ( AMD/Intel 64 )	
		Solaris	Solaris 9
Solaris 10			
Web Redirector	uCosminexus Web Redirector		
Windows Server 2003	Windows Server 2003 Enterprise Edition	Microsoft(R) Windows Server(R) 2003 , Enterprise Edition Operating System ( x86 )	
	Windows Server 2003 Standard Edition	Microsoft(R) Windows Server(R) 2003 , Standard Edition Operating System ( x86 )	
Windows Server 2003 R2	Windows Server 2003 R2 Enterprise Edition	Microsoft(R) Windows Server(R) 2003 R2 , Enterprise Edition Operating System ( x86 )	
	Windows Server 2003 R2 Standard Edition	Microsoft(R) Windows Server(R) 2003 R2 , Standard Edition Operating System ( x86 )	
Windows Server 2003 ( x64 )	Windows Server 2003 Enterprise x64 Edition	Microsoft(R) Windows Server(R) 2003 , Enterprise x64 Edition Operating System	
	Windows Server 2003 Standard x64 Edition	Microsoft(R) Windows Server(R) 2003 , Standard x64 Edition Operating System	

表記		製品名
Windows Server 2003 R2 ( x64 )	Windows Server 2003 R2 Enterprise x64 Edition	Microsoft(R) Windows Server(R) 2003 R2 , Enterprise x64 Edition Operating System
	Windows Server 2003 R2 Standard x64 Edition	Microsoft(R) Windows Server(R) 2003 R2 , Standard x64 Edition Operating System
Windows Vista	Windows Vista Business	Microsoft(R) Windows Vista(R) Business
	Windows Vista Enterprise	Microsoft(R) Windows Vista(R) Enterprise
	Windows Vista Ultimate	Microsoft(R) Windows Vista(R) Ultimate
Windows XP		Microsoft(R) Windows(R) XP Professional Operating System
XDM/RD E2		VOS3 XDM/RD E2

なお，Windows Server 2003，Windows Server 2003 R2，Windows Server 2003 ( x64 )，Windows Server 2003 R2 ( x64 )，Windows Vista，および Windows XP を総称して Windows と表記することがあります。

このマニュアルで使用している表記と，対応する Cosminexus の機能名を次に示します。

表記	Cosminexus の機能名
Cosminexus Developer's Kit for Java	Cosminexus Developer's Kit for Java <sup>TM</sup>
Cosminexus RM	Cosminexus Reliable Messaging
CTM	Cosminexus Component Transaction Monitor
DB Connector for Cosminexus RM	DB Connector for Cosminexus Reliable Messaging
Management Server	Cosminexus Management Server
PRF	Cosminexus Performance Tracer
Server Plug-in	Cosminexus Server Plug-in
Smart Composer	Cosminexus Smart Composer
TPBroker	Cosminexus TPBroker

このマニュアルで使用している表記と，対応する Java 関連用語を次に示します。

表記	Java 関連用語
Connector 1.0	J2EE <sup>TM</sup> Connector Architecture 1.0
Connector 1.5	J2EE <sup>TM</sup> Connector Architecture 1.5
DI	Dependency Injection
EAR	Enterprise ARchive
EJB または Enterprise JavaBeans	Enterprise JavaBeans <sup>TM</sup>
J2EE または Java 2 Platform, Enterprise Edition	Java <sup>TM</sup> 2 Platform, Enterprise Edition

表記	Java 関連用語
J2SE	Java™ 2 Platform, Standard Edition
JAAS	Java™ Authentication and Authorization Service
JAR	Java™ Archive
Java	Java™
Java 2 Runtime Environment, Standard Edition	Java™ 2 Runtime Environment, Standard Edition
Java 2 SDK, Standard Edition	Java™ 2 Software Development Kit, Standard Edition
JavaBeans	JavaBeans™
JCA	J2EE™ Connector Architecture
JDBC	JDBC™ Java™ Database Connectivity
JDK	Java™ Development Kit
JMS	Java Message Service
JNDI	Java Naming and Directory Interface™
JSF	JavaServer™ Faces Reference Implementation (RI) Version: 1.1_01 FCS
JSP	JavaServer Pages™
JTA	Java™ Transaction API
Servlet またはサーブレット	Java™ Servlet
WAR	Web ARchive

### 適用 OS の違いによる機能相違点の表記

このマニュアルは、適用 OS が Windows , AIX , HP-UX , Linux , および Solaris の製品に対応します。OS によって記述を書き分ける場合、次に示す表記を使用して、それぞれの説明に OS 名を明記しています。

表記	意味
Windows の場合	Windows に該当する表記です。
AIX の場合	AIX に該当する表記です。
HP-UX の場合	HP-UX に該当する表記です。
Linux の場合	Linux に該当する表記です。
Solaris の場合	Solaris に該当する表記です。
UNIX の場合	UNIX ( AIX , HP-UX , Linux , Solaris ) に該当する表記です。

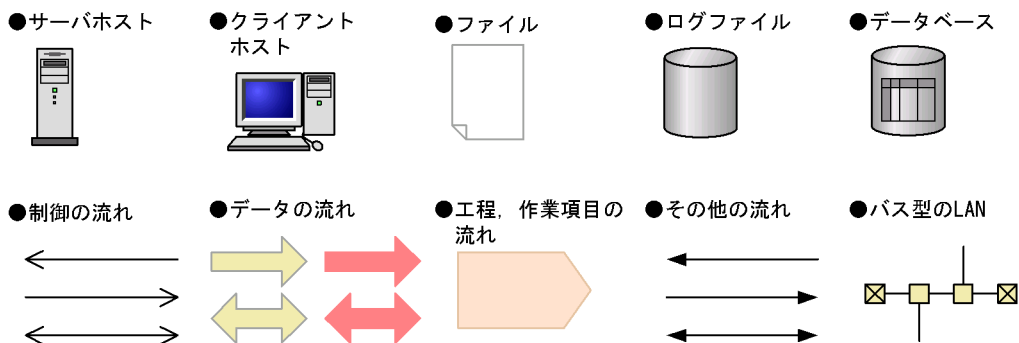
このマニュアルで使用している略語  
このマニュアルで使用している英略語を次に示します。

英略語	英字での表記
API	<u>A</u> pplication <u>P</u> rogramming <u>I</u> nterface
ASCII	<u>A</u> merican <u>S</u> tandard <u>C</u> ode for <u>I</u> nformation <u>I</u> nterchange
BMP	<u>B</u> ean- <u>M</u> anaged <u>P</u> ersistence
BMT	<u>B</u> ean- <u>M</u> anaged <u>T</u> ransaction
CMP	<u>C</u> ontainer- <u>M</u> anaged <u>P</u> ersistence
CMR	<u>C</u> ontainer- <u>M</u> anaged <u>R</u> elationship
CMT	<u>C</u> ontainer- <u>M</u> anaged <u>T</u> ransaction
CORBA	<u>C</u> ommon <u>O</u> bject <u>R</u> equest <u>B</u> roker <u>A</u> rchitecture
CSV	<u>C</u> omma <u>S</u> eparated <u>V</u> alue
CUI	<u>C</u> haracter <u>U</u> ser <u>I</u> nterface
DB	<u>D</u> atabase
DBMS	<u>D</u> atabase <u>M</u> anagement <u>S</u> ystem
DD	<u>D</u> eployment <u>D</u> escriptor
DMZ	<u>D</u> emilitarized <u>Z</u> one
DN	<u>D</u> istinguished <u>N</u> ame
DNS	<u>D</u> omain <u>N</u> ame <u>S</u> ystem
EIS	<u>E</u> nterprise <u>I</u> nformation <u>S</u> ystem
EL	<u>E</u> xpression <u>L</u> anguage
GUI	<u>G</u> raphical <u>U</u> ser <u>I</u> nterface
HA	<u>H</u> igh <u>A</u> vailability
HTML	<u>H</u> yper <u>T</u> ext <u>M</u> arkup <u>L</u> anguage
HTTP	<u>H</u> yper <u>T</u> ext <u>T</u> ransfer <u>P</u> rotocol
HTTPS	<u>H</u> yper <u>T</u> ext <u>T</u> ransfer <u>P</u> rotocol <u>S</u> ecurity
IIOP	<u>I</u> nternet <u>I</u> nter- <u>O</u> rb <u>P</u> rotocol
JIS	<u>J</u> apanese <u>I</u> ndustrial <u>S</u> tandards
JNI	<u>J</u> ava <u>N</u> ative <u>I</u> nterface
LAN	<u>L</u> ocal <u>A</u> rea <u>N</u> etwork
LDAP	<u>L</u> ightweight <u>D</u> irectory <u>A</u> ccess <u>P</u> rotocol
LDIF	<u>L</u> DAP <u>D</u> ata <u>I</u> nterchange <u>F</u> ormat
MDA	<u>M</u> odel <u>D</u> riven <u>A</u> rchitecture
MIB	<u>M</u> anagement <u>I</u> nformation <u>B</u> ase

英略語	英字での表記
OID	<u>O</u> bject <u>I</u> dentifier
OMG	<u>O</u> bject <u>M</u> anagement <u>G</u> roup
ORB	<u>O</u> bject <u>R</u> quest <u>B</u> roker
OS	<u>O</u> perating <u>S</u> ystem
OTM	<u>O</u> bject <u>T</u> ransaction <u>M</u> onitor
OTS	<u>O</u> bject <u>T</u> ransaction <u>S</u> ervice
PIM	<u>P</u> latform <u>I</u> ndependent <u>M</u> odel
PSM	<u>P</u> latform <u>S</u> pecific <u>M</u> odel
RDB	<u>R</u> elational <u>D</u> atabase
RMI	<u>R</u> emote <u>M</u> ethod <u>I</u> nvocation
RPC	<u>R</u> emote <u>P</u> rocedure <u>C</u> all
SFO	<u>S</u> ession <u>F</u> ail <u>O</u> ver
SNMP	<u>S</u> imple <u>N</u> etwork <u>M</u> anagement <u>P</u> rotocol
SOA	<u>S</u> ervice <u>O</u> riented <u>A</u> rchitecture
SOAP	<u>S</u> imple <u>O</u> bject <u>A</u> ccess <u>P</u> rotocol
UDDI	<u>U</u> niversal <u>D</u> escription, <u>D</u> iscovery and <u>I</u> ntegration
UML	<u>U</u> nified <u>M</u> odeling <u>L</u> anguage
URI	<u>U</u> niform <u>R</u> esource <u>I</u> dentifier
URL	<u>U</u> niform <u>R</u> esource <u>L</u> ocator
VM	<u>V</u> irtual <u>M</u> achine
XML	<u>E</u> xtensible <u>M</u> arkup <u>L</u> anguage

## このマニュアルの図中で使用している記号

このマニュアルの図中で使用している記号を、次のように定義します。





## このマニュアルで使用している記号

### ！ 注意事項

間違いやすい点、および間違うとトラブルが発生するような項目について説明しています。

### ポイント

その説明の要点について説明しています。

### 参考

補足的な情報を説明しています。

## このマニュアルの文法で使用している記号

このマニュアルで使用する記号について次に示します。

記号	意味
	横に並べられた複数の項目に対する項目間の区切りを示し、「または」を意味します。 (例) A   B A または B を指定することを示します。
{ }	この記号で囲まれている複数の項目のうちから一つを選択することを示します。項目が横に並べられ、記号   で区切られている場合は、そのうちの一つを選択します。 (例) {A   B   C} A, B または C のどれかを指定することを示します。
[ ]	この記号で囲まれている項目は省略してもよいことを示します。複数の項目が横に並べて記述されている場合には、すべてを省略するか、記号 { } と同じくどれか一つを選択します。 (例 1) [A] 「何も指定しない」か「A を指定する」ことを示します。 (例 2) [B   C] 「何も指定しない」か「B または C を指定する」ことを示します。
< >	この記号で囲まれている項目は、該当する要素やファイルなどを指定したり、該当する要素が表示されたりすることを示します。 (例 1) <プロパティ> プロパティを記述します。またはプロパティが表示されます。 (例 2) <ファイル名> ファイル名を指定します。

## このマニュアルの GUI の説明で使用している記号

記号	意味
[ ]	画面の名称および画面に表示されている項目を表します。

記号	意味
[ ] - [ ]	- の前に示したメニューから, - の後ろのメニューを選択することを表します。
< >	< > 内の名称がユーザの環境によって異なることを表します。
「 」	入力値, 可変値, またはメッセージなどを表します。

### 常用漢字以外の漢字の使用について

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

鍵(かぎ) 個所(かしょ) 全て(すべて) 汎用(はんよう) 必須(ひつす) 閉塞(へいそく)

### KB (キロバイト) などの単位表記について

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

# 目次

## 第 1 編 概要

<b>1</b>	<b>システム運用の概要</b>	<b>1</b>
1.1	Cosminexus のシステム運用	2
1.2	システム運用の特長	5
1.3	システムの運用で実行する作業	7
1.3.1	J2EE アプリケーションを実行するシステムの場合	7
1.3.2	バッチアプリケーションを実行するシステムの場合	14
1.4	システムの運用で使用する機能	19
1.5	システム運用の作業	21
1.5.1	システムを起動 / 停止する	21
1.5.2	システムの運用状況を監視する	21
1.5.3	システムの処理性能を監視する	23
1.5.4	J2EE アプリケーションを運用する	23
1.6	システム保守の作業	25
1.6.1	J2EE アプリケーションとリソースを保守する	25
1.6.2	トラブルに対処する	26
1.7	システム監査の作業	28
1.7.1	監査ログを使用してシステムを監査する	28
1.7.2	データベース監査証跡と連携して運用する	28
1.8	システム運用を始める前に	29
1.8.1	Component Container 管理者による運用に関する留意事項	29
1.8.2	トラブル発生時に資料を収集するための準備	30
1.8.3	Windows Vista 使用時の注意事項	31
1.8.4	運用管理コマンド (mngsvrutil) を利用する場合の留意事項	31
1.8.5	サーバ管理コマンドを利用する場合の留意事項	31
1.8.6	作業ディレクトリに関する注意事項	32
1.8.7	RMI レジストリに関する注意事項	32
1.8.8	コマンド実行時に指定する Web システム名, サービスユニット名, およびサーバ名に関する注意事項	33

## 第 2 編 システムの監視

<b>2</b>	<b>システムの運用監視</b>	<b>35</b>
2.1	運用監視作業の概要	36
2.2	論理サーバのステータスの監視	38
2.2.1	ステータスの監視方法	38
2.2.2	ステータス監視で確認できる項目	39
2.3	稼働情報の監視（稼働情報ファイル）	41
2.3.1	稼働情報ファイルで収集できる情報	41
2.3.2	稼働情報ファイルの出力	43
2.3.3	稼働情報ファイルの調査	44
2.4	稼働情報の監視（運用管理コマンド）	62
2.4.1	サーバの稼働情報の監視方法（J2EE サーバ，SFO サーバ，バッチサーバ）	62
2.4.2	稼働情報監視で確認できる項目	62
2.5	リソース枯渇の監視	73
2.5.1	出力される情報の種類	73
2.5.2	出力形式と出力内容	74
2.6	CTM の稼働統計情報の確認	81
2.6.1	CTM の稼働統計情報の収集方法	81
2.6.2	CTM の稼働統計情報の出力先と出力情報	82
2.6.3	CTM の稼働統計情報の出力例	85
2.7	トランザクション情報の確認	88
2.7.1	稼働中のトランザクションの確認	88
2.7.2	停止中のトランザクションの確認	88
<b>3</b>	<b>システムの処理性能の解析</b>	<b>91</b>
3.1	処理性能の解析作業の概要	92
3.2	性能解析トレースファイルの収集	95
3.2.1	性能解析トレースファイルの収集方法	95
3.2.2	性能解析トレースファイルの出力先と出力情報	96
3.3	性能解析トレースファイルの利用方法	100
3.3.1	Web サーバのレスポンスタイムの解析	100
3.3.2	アプリケーションサーバ内でのリクエストの処理状況の調査	101
3.3.3	セッションのライフサイクルの調査	103
3.3.4	タイムアウトが発生したトランザクションの特定	107

3.3.5	タイムアウトが発生したリクエストの特定	108
3.3.6	ルートアプリケーション情報を利用したログ調査	109
3.3.7	トラブルが発生した接続の特定	110
3.3.8	性能解析トレースファイルとスレッドダンプを対応づけた問題個所の調査	111

## 第3編 システムの保守

<b>4</b>	<b>J2EE アプリケーションとリソースの管理</b>	<b>113</b>
4.1	J2EE アプリケーションとリソースの管理の概要	114
4.2	Web アプリケーションの最大同時実行スレッド数の動的変更	117
4.2.1	Web アプリケーションの稼働状況の確認	118
4.2.2	Web アプリケーションの最大同時実行スレッド数の設定変更	120
4.2.3	最大同時実行スレッド数を動的に変更したときの Web アプリケーションの動作	121
4.3	CTM のスケジュールキューの稼働状況の確認と最大同時実行数の変更	123
4.3.1	CTM のスケジュールキューの稼働状況の確認	123
4.3.2	CTM のスケジュールキューの同時実行数の変更	123
4.4	J2EE アプリケーションの実行時間の監視とキャンセル	126
4.4.1	J2EE アプリケーションの実行状態の確認	127
4.4.2	タイムアウトが発生したリクエストのキャンセル	129
4.5	J2EE アプリケーションの閉塞と停止	130
4.5.1	J2EE アプリケーションの停止方法の種類	130
4.5.2	システムごとに実行できるサービスの閉塞方法および J2EE アプリケーションの停止方法	131
4.5.3	負荷分散機を利用したサービス閉塞	133
4.5.4	CTM を使用したサービス閉塞	135
4.5.5	J2EE アプリケーションの停止	137
4.6	J2EE アプリケーションの入れ替えと保守	144
4.6.1	J2EE アプリケーションの入れ替え	144
4.6.2	リデプロイによる J2EE アプリケーションの入れ替え	146
4.6.3	リロードによる J2EE アプリケーションの入れ替え	148
4.6.4	入れ替える J2EE アプリケーション内の JSP の事前コンパイル	149
4.6.5	J2EE アプリケーションの名称変更	151
4.6.6	CTM を使用したオンライン状態での J2EE アプリケーションの入れ替え	153
4.6.7	テストモードでの J2EE アプリケーションの運用	154
4.7	JavaBeans リソースの入れ替え	160

4.8	クラスタコネクションプールの運用	163
4.8.1	コネクションプールの状態の確認	163
4.8.2	コネクションプールの一時停止	163
4.8.3	コネクションプールの再開	164
4.8.4	コネクションプールの状態	165

## 5

### トラブルシューティング 169

5.1	対処の手順	170
5.2	取得が必要な資料の種類	172
5.2.1	トラブルの種別と取得が必要な資料	172
5.2.2	取得が必要な資料の一覧	174
5.2.3	取得方法および調査方法との対応	175
5.3	資料の取得	178
5.3.1	資料の取得方法	178
5.3.2	トラブル発生時に自動的に取得できる資料	178
5.3.3	snapshot ログの収集	186
5.4	資料の個別取得	188
5.4.1	Cosminexus のログの取得 (J2EE アプリケーションを実行するシステム)	188
5.4.2	Cosminexus のログの取得 (パッチアプリケーションを実行するシステム)	218
5.4.3	EJB クライアントアプリケーションのシステムログの取得	232
5.4.4	性能解析トレースの取得	240
5.4.5	JavaVM のスレッドダンプの取得	240
5.4.6	JavaVM のガーベジコレクションログの取得	245
5.4.7	メモリダンプの取得	246
5.4.8	日立固有の JavaVM ログの取得 (日立 JavaVM ログファイル)	249
5.4.9	JavaVM 出力メッセージログの取得 (標準出力またはエラーレポートファイル)	249
5.4.10	OS の状態情報と OS のログの取得	250
5.4.11	OS の統計情報の取得	254
5.4.12	Cosminexus の定義情報の取得	257
5.4.13	Cosminexus の作業ディレクトリの内容の取得	258
5.4.14	Cosminexus のリソース設定情報の取得	258
5.4.15	Web サーバログの取得	258
5.4.16	JavaVM のスタックトレース情報の取得	259
5.5	資料の調査	260
5.5.1	Cosminexus のログの調査	260
5.5.2	EJB クライアントアプリケーションのログの調査	272

5.5.3	Cosminexus DABroker Library のスプール情報の調査	273
5.5.4	性能解析トレースの調査	303
5.5.5	JavaVM のスレッドダンプの調査	303
5.5.6	JavaVM のガーベージコレクションログの調査	305
5.5.7	日立 JavaVM 固有のログの調査 (日立 JavaVM ログファイル)	306
5.5.8	JavaVM が出力するメッセージログの調査 (標準出力およびエラーレポートファイル)	308
5.5.9	OS の状態情報および OS のログの調査	320
5.5.10	スタックトレースの調査	321
5.6	トラブルへの対処と回復	330
5.6.1	構成ソフトウェアのプロセス (論理サーバ) が異常終了した場合	330
5.6.2	J2EE アプリケーションの強制停止に失敗した場合	333
5.6.3	セッションフェイルオーバー機能でトラブルが発生した場合	333
5.6.4	データベースと接続中にトラブルが発生した場合	335
5.6.5	JavaVM が異常終了した場合	336
5.6.6	JP1 と連携したシステムでトラブルが発生した場合	338
5.6.7	1:1 系切り替えシステムでトラブルが発生した場合	339
5.6.8	N:1 リカバリシステムでトラブルが発生した場合	340
5.6.9	EJB クライアントでトラブルが発生した場合	342
5.7	トラブルシューティングに関連する留意事項	344
5.7.1	EJB クライアントアプリケーションのシステムログに関する留意事項	344
5.7.2	Cosminexus DABroker Library のトレースファイルの運用時の留意事項	347
5.7.3	CTM 使用時の留意事項	349
5.7.4	PRF 使用時の留意事項	349

## 第 4 編 システムの監査

6	監査ログを使用したシステムの監査	351
6.1	監査ログを使用したシステム監査の概要	352
6.2	監査ログの収集と調査	353
6.2.1	監査ログの出力先	353
6.2.2	監査ログの出力形式	353
6.2.3	メッセージの出力項目	354
6.3	監査ログを出力するコマンド・操作一覧	361
6.3.1	J2EE サーバで使用するコマンド一覧	361

6.3.2	バッチサーバで使用するコマンド一覧	363
6.3.3	性能解析トレース・CTM で使用するコマンド一覧	364
6.3.4	Management Server で使用するコマンド一覧	365
6.3.5	EJB クライアントアプリケーションで使用するコマンド一覧	368
6.3.6	Hitachi Web Server で使用するコマンド・操作一覧 (Windows の場合)	369
6.3.7	Hitachi Web Server で使用するコマンド・操作一覧 (UNIX の場合)	369
6.4	監査ログの出力ポイント	370
6.4.1	J2EE サーバで使用するコマンドの監査ログの出力ポイント	370
6.4.2	バッチサーバで使用するコマンドの監査ログの出力ポイント	384
6.4.3	性能解析トレース・CTM で使用するコマンドの監査ログの出力ポイント	385
6.4.4	Management Server で使用するコマンドの監査ログの出力ポイント	389
6.4.5	EJB クライアントアプリケーションで使用するコマンドの監査ログの出力ポイント	401
6.4.6	Hitachi Web Server で使用するコマンド, および Hitachi Web Server に対する操作の監査ログの出力ポイント (Windows の場合)	402
6.4.7	Hitachi Web Server で使用するコマンド, および Hitachi Web Server に対する操作の監査ログの出力ポイント (UNIX の場合)	405

## 7

データベース監査証跡と連携した運用	411
7.1 データベース監査証跡と連携した運用の概要	412
7.2 データベース監査証跡と連携した運用例	413
7.2.1 アプリケーションのユーザログの取得	413
7.2.2 データベースの監査証跡情報の取得	414
7.2.3 性能解析トレースの取得	414
7.2.4 情報の特定	415
7.3 注意事項	416

## 第5編 システムの起動と停止

## 8

クラスタソフトウェアを使用したシステムの起動と停止 (Windows の場合)	417
8.1 アプリケーションサーバを対象にした 1:1 系切り替えシステムの起動と停止	418
8.1.1 アプリケーションサーバを対象にした 1:1 系切り替えシステムの起動	419
8.1.2 アプリケーションサーバを対象にした 1:1 系切り替えシステムの停止	419



8.1.3	アプリケーションサーバを対象にした 1:1 系切り替えシステムで計画的に系を切り替えるときの起動と停止	420
8.1.4	アプリケーションサーバを対象にした 1:1 系切り替えシステムをメンテナンスする場合の起動と停止	420
8.2	運用管理サーバを対象にした 1:1 系切り替えシステムの起動と停止	424
8.2.1	運用管理サーバを対象にした 1:1 系切り替えシステムの起動	424
8.2.2	運用管理サーバを対象にした 1:1 系切り替えシステムの停止	425
8.2.3	運用管理サーバを対象にした 1:1 系切り替えシステムで計画的に系を切り替えるときの起動と停止	425
8.3	相互系切り替えシステムの起動と停止	427
8.3.1	相互系切り替えシステムの起動	427
8.3.2	相互系切り替えシステムの停止	428
8.3.3	相互系切り替えシステムで計画的に系を切り替える場合の起動と停止	429
8.4	N:1 リカバリシステムの起動と停止	430
8.4.1	N:1 リカバリシステムの起動	430
8.4.2	N:1 リカバリシステムの停止	431

## 9

	クラスタソフトウェアを使用したシステムの起動と停止 (UNIX の場合)	433
9.1	1:1 系切り替えシステムの起動と停止	434
9.1.1	1:1 系切り替えシステムの起動	435
9.1.2	1:1 系切り替えシステムの停止	436
9.1.3	1:1 系切り替えシステムで計画的に系を切り替える場合の起動と停止	438
9.1.4	アプリケーションサーバの 1:1 系切り替えシステムをメンテナンスする場合の起動と停止	438
9.1.5	運用管理サーバの 1:1 系切り替えシステムをメンテナンスする場合の起動と停止	442
9.2	相互系切り替えシステムの起動と停止	444
9.2.1	相互系切り替えシステムの起動	446
9.2.2	相互系切り替えシステムの停止	448
9.2.3	相互系切り替えシステムで計画的に系を切り替える場合の起動と停止	451
9.2.4	相互系切り替えシステムをメンテナンスする場合の起動と停止	453
9.3	N:1 リカバリシステムの起動と停止	457
9.3.1	N:1 リカバリシステムの起動	457
9.3.2	N:1 リカバリシステムの停止	459

## 付録

		461
付録 A	トレース取得ポイントと PRF トレース取得レベル	462

付録 A.1	CTM のトレース取得ポイント	466
付録 A.2	リダイレクタのトレース取得ポイント	472
付録 A.3	Web コンテナのトレース取得ポイント (リクエスト処理のトレース)	476
付録 A.4	Web コンテナのトレース取得ポイント (セッショントレース)	483
付録 A.5	Web コンテナのトレース取得ポイント (フィルタのトレース)	489
付録 A.6	Web コンテナのトレース取得ポイント (セッションフェイルオーバ機能のトレース)	498
付録 A.7	EJB コンテナのトレース取得ポイント	504
付録 A.8	JNDI のトレース取得ポイント	517
付録 A.9	JTA のトレース取得ポイント	521
付録 A.10	DB Connector, JCA コンテナのトレース取得ポイント	526
付録 A.11	RMI のトレース取得ポイント	549
付録 A.12	OTS のトレース取得ポイント	551
付録 A.13	標準出力 / 標準エラー出力 / ユーザログのトレース取得ポイント	561
付録 A.14	DI のトレース取得ポイント	563
付録 A.15	バッチアプリケーション実行機能のトレース取得ポイント	565
付録 A.16	J2EE サーバの開始・終了時のトレース取得ポイント	566
<b>付録 B</b>	<b>snapshot ログの収集対象一覧</b>	<b>568</b>
付録 B.1	Cosminexus Component Container	569
付録 B.2	Cosminexus Component Transaction Monitor	586
付録 B.3	Cosminexus DABroker Library	587
付録 B.4	Cosminexus Developer's Kit for Java	588
付録 B.5	Cosminexus Performance Tracer	589
付録 B.6	Cosminexus TPBroker	589
付録 B.7	Cosminexus Web Services - Security	594
付録 B.8	Hitachi Web Server	597
付録 B.9	Cosminexus cFramework	598
付録 B.10	uCosminexus Service Adapter for TP1	600
付録 B.11	監査ログ	601
付録 B.12	そのほかの情報	602
<b>付録 C</b>	<b>データベースと接続中にトラブルが発生したコネクションの特定</b>	<b>603</b>
付録 C.1	Cosminexus Component Container	608
付録 C.2	Cosminexus DABroker Library	612
付録 C.3	HiRDB クライアント	615
付録 C.4	HiRDB サーバ	617
付録 C.5	Oracle クライアント	618
付録 C.6	Oracle サーバ	619

付録 D 機能ごとに出力されるログ情報	622
付録 D.1 J2EE アプリケーション実行時間監視で出力されるログ情報	622
付録 E 障害発生時の CMR 用の表の回復	628
付録 F ベーシックモードの利用 (互換機能)	630
付録 F.1 ベーシックモードの場合のプロセスの依存関係	630
付録 F.2 ベーシックモードの場合のプロセスの再起動方法	630
付録 G サーブレットエンジンモードでのシステム運用	632
付録 G.1 Component Container 管理者による運用に関する留意事項	632
付録 G.2 システムの起動	633
付録 G.3 システムの停止	635
付録 G.4 システムの処理性能の解析	636
付録 G.5 トラブルシューティング	638
付録 G.6 資料の調査	641
付録 G.7 トラブルへの対処と回復	641
付録 G.8 トラブルシューティングに関連する留意事項	643
付録 G.9 資料の取得方法	643
付録 H 代替手段で実行できる運用作業	656
付録 H.1 システムの起動と停止	656
付録 H.2 システムの運用監視	656
付録 H.3 システムの処理性能解析	658
付録 H.4 統合ユーザ管理のリソースの監視	658
付録 H.5 J2EE アプリケーションとリソースの管理	659
付録 H.6 システムの監査	662
付録 H.7 トラブルシューティング	662
付録 I 用語解説	664

## 索引



# 1

## システム運用の概要

この章では、Cosminexus で構築したシステムの運用方法の種類，およびシステムの運用サイクルについて説明します。

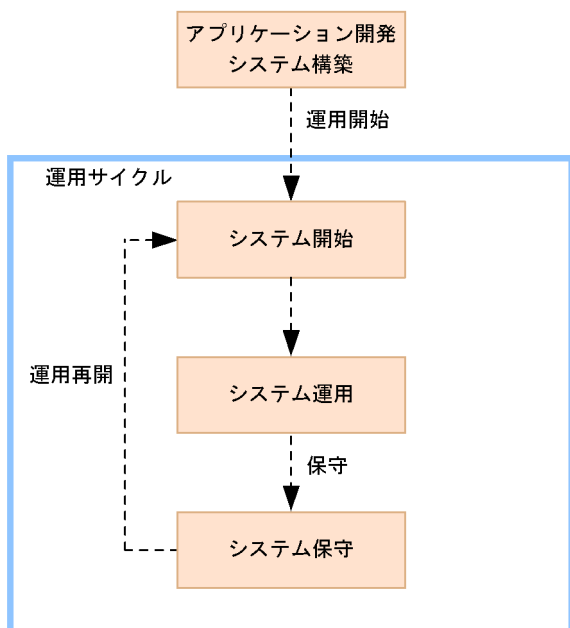
- 
- 1.1 Cosminexus のシステム運用
  - 1.2 システム運用の特長
  - 1.3 システムの運用で実行する作業
  - 1.4 システムの運用で使用する機能
  - 1.5 システム運用の作業
  - 1.6 システム保守の作業
  - 1.7 システム監査の作業
  - 1.8 システム運用を始める前に
-

## 1.1 Cosminexus のシステム運用

Cosminexus のシステムでは、アプリケーション開発とシステム構築の作業が終了したあと、システムの運用を開始します。システムの運用では、日常的なシステム運用と定期的なシステム保守を繰り返します。

Cosminexus のシステムの運用サイクルを次の図に示します。

図 1-1 システムの運用サイクル（概要）



（凡例）

---▶ : システムの状態遷移の流れ

システム運用は、日常的に実施する作業です。システム運用では、システムの起動や停止のほか、システムを安定稼働させるために、システムの稼働状況や性能などを監視します。

システム保守は、システムの動作環境や実行するアプリケーションを変更する場合や、システムにトラブルが発生した場合などに、必要に応じて実施する作業です。システムの保守では、動作環境の設定を変更したり、運用中に発生したトラブルに対処したりします。

システムの運用サイクルを構成する作業を次の表に示します。

表 1-1 システムの運用サイクルを構成する作業

分類	作業内容	参照先
システムの起動と停止	<ul style="list-style-type: none"> <li>• システムの起動 / 停止</li> <li>• アプリケーションの開始 / 停止</li> </ul>	マニュアル「Cosminexus 簡易構築・運用ガイド」, 第 5 編 システムの起動と停止 <sup>1</sup>
システムの監視	<ul style="list-style-type: none"> <li>• システムの運用状況の監視</li> <li>• システムの処理性能の監視</li> </ul>	第 2 編 システムの監視
システムの保守	<ul style="list-style-type: none"> <li>• J2EE アプリケーション<sup>2</sup>・リソースの保守</li> <li>• システムの動作環境の設定変更</li> <li>• トラブルへの対処</li> </ul>	第 3 編 システムの保守

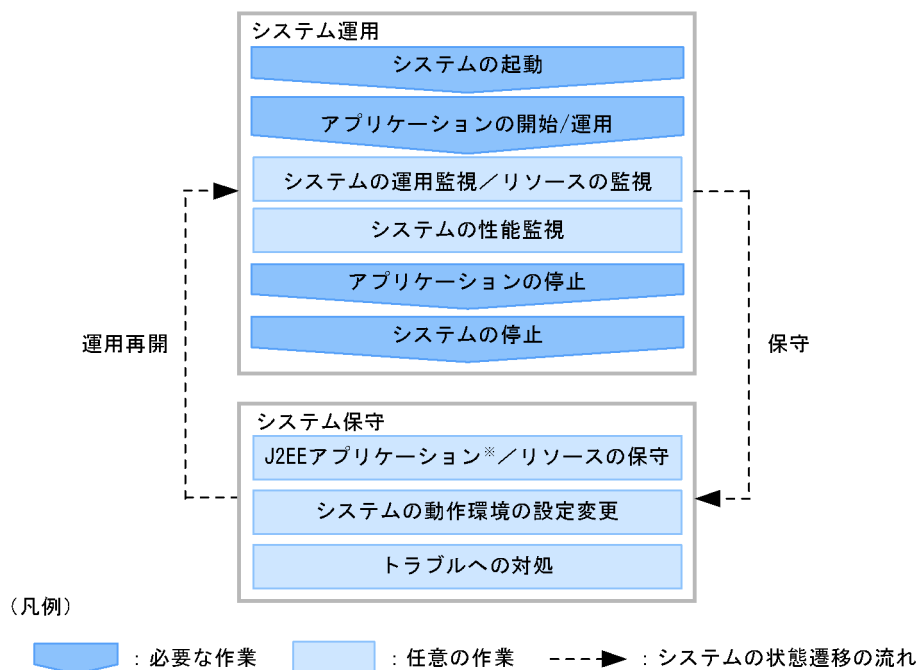
注 1 Smart Composer 機能を使用したシステムの起動と停止の方法については、マニュアル「Cosminexus 簡易構築・運用ガイド」のシステムの起動と停止の説明を参照してください。このマニュアルでは、クラスタソフトウェアを使用したシステムの起動と停止の方法について説明しています。

注 2 J2EE アプリケーションの保守は、バッチアプリケーションを実行するシステムの運用では実施しません。

運用サイクルを図で表すと、次のようになります。

## 1. システム運用の概要

図 1-2 システムの運用サイクル ( J2EE サーバモードの場合 )



注※ J2EEアプリケーションの保守は、バッチアプリケーション\*を実行するシステムの運用では実施しません。

なお、運用サイクルに含まれない運用作業として、システムの監査があります。システムの監査に関する運用作業には、監査ログを使用してシステムを監査する場合や、データベースが提供する監査証跡の機能と連携する場合の運用作業があります。

次の節以降では、システム運用、システム保守、およびシステム監査の各作業の概要について説明します。

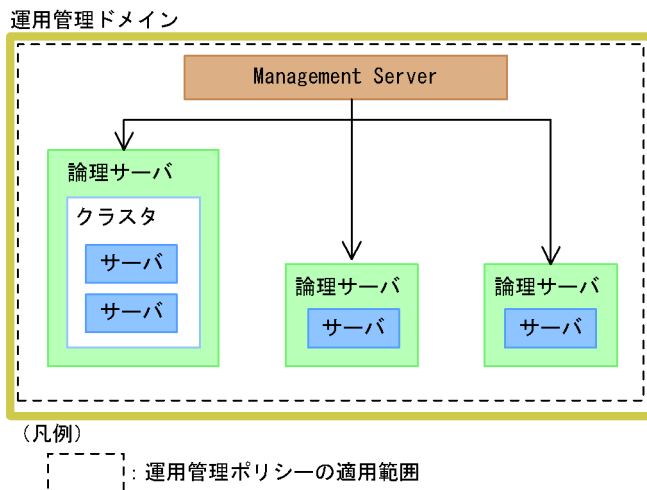


## 1.2 システム運用の特長

Cosminexus のシステムでは、Smart Composer 機能を使用して構築したシステムに対して、コマンドで一括して運用できます。ここでは、Cosminexus のシステムの一括運用について説明します。

Cosminexus のシステムは、一つのホストだけではなく複数のホストで構成されることがあります。このような複数のホストで構成されるシステムを Management Server を利用することで一元管理します。Management Server による運用管理の概要を次に示します。

図 1-3 Management Server による運用管理の概要



論理サーバとは、Management Server の運用管理の対象になる、サーバまたはクラスタです。サーバとは Cosminexus のサーバの実体で、Web サーバや J2EE サーバ、バッチサーバなどがあります。クラスタとは、ある共通の機能を持つ論理サーバの集合で、J2EE サーバクラスタや Web サーバクラスタなどがあります。なお、クラスタに対しては、一括して運用操作を実行できるため、単体の論理サーバと同様に管理できます。

Management Server では、複数の論理サーバをまとめて管理します。一つの Management Server が管理する範囲のことを、運用管理ドメインといいます。運用管理ドメインは、同一の運用管理ポリシーを適用する論理サーバの集合です。なお、運用管理ドメインは、互いに排他的関係になるため、一つの論理サーバが複数の運用管理ドメインに包含されることはありません。

なお、システムを一括運用するには、Smart Composer 機能、または運用管理コマンドを使用します。次に、それぞれの方法の概要について説明します。

## 1. システム運用の概要

### (1) Smart Composer 機能を使用する

Smart Composer 機能とは、システムを簡単に構築・運用するための機能です。

Smart Composer 機能を使用すると、次のシステム運用の作業を実行できます。

- サービスユニットを一括起動または一括停止する
- サービスユニットのステータスを確認する
- Web システム内のサービスユニットの構成を確認する

Smart Composer 機能を使用したシステム運用の設定および作業の詳細については、マニュアル「Cosminexus 簡易構築・運用ガイド」のシステム運用の説明を参照してください。

### (2) 運用管理コマンドを使用する

運用管理コマンド (mngsvrutil) とは、Management Server を操作するためのインタフェースです。

運用管理コマンドを使用すると、論理サーバ単位、ホスト単位、または運用管理ドメイン単位の操作を実行できるため、システムを多面的に運用・管理できます。運用管理コマンドの概要を次に示します。

運用管理コマンドを使用した運用では、次に示すような論理サーバの運用操作を実行できます。

- 論理サーバのアプリケーションを管理する
- 論理サーバの運用監視をする
- システム全体の処理性能を監視・解析するための性能情報を収集する

## 1.3 システムの運用で実行する作業

ここでは、Cosminexus のシステムの運用作業について説明します。

### 1.3.1 J2EE アプリケーションを実行するシステムの場合

ここでは、J2EE アプリケーションを実行するシステムの運用で実施する作業について説明します。J2EE アプリケーションを実行するシステムでは、次の運用作業を実施します。

- システムの起動と停止
- システムの運用監視
- システムの処理性能解析
- J2EE アプリケーションとリソースの管理
- システムの監査
- システムの変更
- システム環境の移行
- トラブルシューティング

これらの作業を実行するには、主に次の手段を使用します。

- Smart Composer 機能のコマンドを使用する
- 運用管理コマンド (mngsvrutil) を使用する
- Cosminexus のコマンドを使用する

なお、ここでは、Cosminexus システムで推奨する手段で、各運用作業を実行する場合について説明しています。これ以外の手段で実行する運用作業については、「付録 H 代替手段で実行できる運用作業」を参照してください。

#### (1) システムの起動と停止で実行できる作業

システムの起動と停止で実行できる作業について、次の表に示します。なお、Cosminexus のシステムで実行できる運用作業には、運用前に設定が必要なものもあります。

表 1-2 システムの起動と停止で実行できる作業

運用作業	運用前の設定の要否	運用前の設定方法の参照先	運用方法	
			参照先	手段
サービスユニットを一括起動します。	不要	-	簡易構築	cmx_start_target コマンド
サービスユニットを一括停止します。	不要	-	簡易構築	cmx_stop_target コマンド

## 1. システム運用の概要

運用作業	運用前の設定の要否	運用前の設定方法の参照先	運用方法	
			参照先	手段
クラスタソフトウェアを使用したシステムを起動・停止します。	必要	構築	8章, 9章	Cosminexusのコマンド, クラスタソフトウェアの機能

### (凡例)

必要：運用前の設定が必要です。

不要：運用前の設定は不要です。

-：該当しません。

簡易構築：マニュアル「Cosminexus 簡易構築・運用ガイド」

構築：マニュアル「Cosminexus システム構築ガイド」

### (2) システムの運用監視で実行できる作業

システムの運用監視で実行できる作業について、次の表に示します。なお、Cosminexusのシステムで実行できる運用作業には、運用前に設定が必要なものもあります。

表 1-3 システムの運用監視で実行できる作業

運用作業	運用前の設定の要否	運用前の設定方法の参照先	運用方法	
			参照先	手段
サービスユニットのステータスを確認します。	不要	-	簡易構築	cmx_list_status コマンド
論理サーバのステータスを確認します。	不要	構築	2.2	運用管理コマンド (mngsvrutil)
システム内のサービスユニットの構成を確認します。	不要	-	簡易構築	cmx_list_model コマンド
稼働情報ファイルを参照して、J2EE サーバおよびSFOサーバの稼働情報を監視します。	省略可	構築	2.3	-
運用管理コマンドを使用して、J2EE サーバおよびSFOサーバの稼働情報を監視します。	不要	-	2.4	運用管理コマンド (mngsvrutil)
リソースの使用率や使用数を監視して、しきい値を超えた場合に原因を調査します。	省略可	構築	2.5	-
稼働情報ファイルに出力される稼働情報、またはリソースの枯渇状況に応じてメッセージが出力された場合の処理を自動化します (Management アクション)。	必要	構築	-	-
CTMの稼働統計情報を出力します。	不要	-	2.6	運用管理コマンド (mngsvrutil)

運用作業	運用前の設定の要否	運用前の設定方法の参照先	運用方法	
			参照先	手段
トランザクションの状態を確認します。	不要	-	2.7	サーバ管理コマンド ( cjlisttrn )

## ( 凡例 )

必要：運用前の設定が必要です。

省略可：デフォルトの設定で使用できるため、運用前の設定は省略できます。この動作をデフォルトから変更する場合は設定が必要です。

不要：運用前の設定は不要です。

- : 該当しません。

簡易構築：マニュアル「Cosminexus 簡易構築・運用ガイド」

構築：マニュアル「Cosminexus システム構築ガイド」

注 CTM は、構成ソフトウェアに Cosminexus Component Transaction Monitor を含む製品だけで利用できます。利用できる製品については、マニュアル「Cosminexus 概説」を参照してください。

## ( 3 ) システムの処理性能解析で実行できる作業

システムの処理性能解析で実行できる作業について、次の表に示します。なお、Cosminexus のシステムで実行できる運用作業には、運用前に設定が必要なものもありません。

表 1-4 システムの処理性能解析で実行できる作業

運用作業	運用前の設定の要否	運用前の設定方法の参照先	運用方法	
			参照先	手段
性能解析トレースファイルを収集します。	不要	-	3.2	運用管理コマンド ( mngsvrutil )
収集した性能解析トレースファイルを利用してシステムの処理性能を解析します。	不要	-	3.3	-

## ( 凡例 )

不要：運用前の設定は不要です。

- : 該当しません。

## ( 4 ) J2EE アプリケーションとリソースの管理で実行できる作業

J2EE アプリケーションとリソースの管理で実行できる作業について、次の表に示します。なお、Cosminexus のシステムで実行できる運用作業には、運用前に設定が必要なものもあります。

## 1. システム運用の概要

表 1-5 J2EE アプリケーションとリソースの管理で実行できる作業

運用作業	運用前の設定の要否	運用前の設定方法の参照先	運用方法		
			参照先	手段	
Web アプリケーションの稼働状況を確認します。	不要	-	4.2.1	運用管理コマンド (mngsvrutil)	
アクセス状況に応じて、Web アプリケーションの同時実行スレッド数を動的に変更します。	不要	-	4.2.2	運用管理コマンド (mngsvrutil)	
CTM のスケジュールキューの稼働状況を確認して、同時実行数を動的に変更します。 <sup>1</sup>	不要	-	4.3	<ul style="list-style-type: none"> <li>確認：運用管理コマンド (mngsvrutil)</li> <li>確認と変更：CTM のコマンド (ctmlsque, ctmchpara)</li> </ul>	
J2EE アプリケーションの実行状態を確認して、J2EE アプリケーションでタイムアウトが発生したリクエストをキャンセルします (J2EE アプリケーションの実行時間監視機能)。	必要	構築	4.4	<ul style="list-style-type: none"> <li>確認：サーバ管理コマンド (cjlistthread)</li> <li>キャンセル：サーバ管理コマンド (cjstopthread)</li> </ul>	
J2EE アプリケーションを通常開始します。	不要	-	リファレンス	サーバ管理コマンド (cjstartapp)	
J2EE アプリケーションを通常停止します。	不要	-	4.5.5	サーバ管理コマンド (cjstopapp)	
J2EE アプリケーションを強制停止します。	不要	-	4.5	サーバ管理コマンド (cjstopapp)	
負荷分散機を使用してサービスを閉塞します。 <sup>2</sup>	-	-	4.5.3	負荷分散機の機能	
CTM を利用してサービスを閉塞します。 <sup>1</sup>	不要	-	4.5.4	運用管理コマンド (mngsvrutil)	
J2EE アプリケーションを入れ替えます。	リデプロイによる入れ替え	省略可	構築	4.6.2	サーバ管理コマンド (cjreplaceapp)
	通常の入替	不要	-	4.6.1	サーバ管理コマンド (cjstopapp, cjimportapp)
	リロードによる入れ替え	必要	構築	4.6.3	サーバ管理コマンド (cjreloadapp)
JSP を事前にコンパイルしてから、J2EE アプリケーションを入れ替えます。	省略可 / 不要	構築	4.6.4	cjjspc コマンド	

運用作業	運用前の設定の要否	運用前の設定方法の参照先	運用方法	
			参照先	手段
J2EE アプリケーションの名称を変更します。	不要	-	4.6.5	サーバ管理コマンド (cjrenameapp)
CTM を利用して、オンライン状態で J2EE アプリケーションを入れ替えます。 <sup>1</sup>	不要	-	4.6.6	運用管理コマンド (mngsvrutil)
J2EE アプリケーションを本番環境でテストしてから本番運用を開始します。	不要	-	4.6.7	サーバ管理コマンド (cjimportapp, cjstartapp)
コネクションプールの情報を表示します。	不要	-	解説	サーバ管理コマンド (cjlistpool)
コネクションプールをクリアします。	不要	-	解説	サーバ管理コマンド (cjclearpool)
JavaBeans リソースを入れ替えます。	不要	-	4.7	サーバ管理コマンド (cjdeletejb, cjimportjb)
クラスタ構成のデータベースをメンテナンスします。	不要	-	4.8	サーバ管理コマンド (cjlistrar, cjsuspendpool, cjresumepool)
リソースの接続テストをします。	必要	構築	解説	サーバ管理コマンド (cjtestres)

## ( 凡例 )

必要：運用前の設定が必要です。

省略可：デフォルトの設定で使用できるため、運用前の設定は省略できます。この動作をデフォルトから変更する場合は設定が必要です。

不要：運用前の設定は不要です。

-：該当しません。

解説：マニュアル「Cosminexus 機能解説」

構築：マニュアル「Cosminexus システム構築ガイド」

リファレンス：マニュアル「Cosminexus リファレンス コマンド編」

注 1 CTM は、構成ソフトウェアに Cosminexus Component Transaction Monitor を含む製品だけで利用できます。利用できる製品については、マニュアル「Cosminexus 概説」を参照してください。

注 2 負荷分散機を Smart Composer 機能で管理していない場合に実行します。負荷分散機を Smart Composer 機能で管理している場合は、cmx\_stop\_target コマンドを使用してサービスを閉塞してください。cmx\_stop\_target コマンドの詳細については、マニュアル「Cosminexus 簡易構築・運用ガイド」を参照してください。

## ( 5 ) システムの監査で実行できる運用作業

監査ログを使用したシステムの監査、およびデータベース監査証跡と連携した運用で実

## 1. システム運用の概要

行できる作業について、次の表に示します。なお、Cosminexus のシステムで実行できる運用作業には、運用前に設定が必要なものもあります。

表 1-6 システムの監査で実行できる作業

運用作業	運用前の設定の要否	運用前の設定方法の参照先	運用方法	
			参照先	手段
監査ログを出力して、管理します。	必要	構築	6.2	-
監査ログを収集して調査します。	必要	構築	6.2	-
データベース監査証跡に出力された情報を基に、データベースにアクセスしたリクエストの情報を特定します。	不要	-	7.2	-

(凡例)

必要：運用前の設定が必要です。

不要：運用前の設定は不要です。

-：該当しません。

構築：マニュアル「Cosminexus システム構築ガイド」

### (6) システムの変更でできる運用作業

システムの変更で実行できる作業について、次の表に示します。

表 1-7 システムの変更で実行できる作業

運用作業	運用前の設定の要否	運用前の設定方法の参照先	運用方法		
			参照先	手段	
設定変更	システム全体の設定を一括で変更します。	不要	-	簡易構築	-
構成変更	システムをスケールアウトします。	不要	-	簡易構築	-
	システムをスケールインします。	不要	-	簡易構築	-
削除	システムを削除します。	不要	-	簡易構築	-

(凡例)

不要：運用前の設定は不要です。

-：該当しません。

簡易構築：マニュアル「Cosminexus 簡易構築・運用ガイド」



## (7) システム環境の移行で実行できる運用作業

システム環境の移行で実行できる作業について、次の表に示します。

表 1-8 システム環境の移行で実行できる作業

運用作業	運用前の設定の要否	運用前の設定方法の参照先	運用方法	
			参照先	手段
構築済みのシステムの環境を移行します。	不要	-	簡易構築	-

(凡例)

不要：運用前の設定は不要です。

-：該当しません。

簡易構築：マニュアル「Cosminexus 簡易構築・運用ガイド」

## (8) トラブルシューティングで実行できる作業

トラブルシューティングで実行できる作業について、次の表に示します。なお、Cosminexus のシステムで実行できる運用作業には、運用前に設定が必要なものもあります。

表 1-9 トラブルシューティングで実行できる作業

運用作業	運用前の設定の要否	運用前の設定方法の参照先	運用方法	
			参照先	手段
トラブルシューティングに必要な資料を収集します。	必要 / 省略可	構築	5.3.1	運用管理コマンド (mngsvrutil)
論理サーバ (構成ソフトウェアのサーバプロセス) に異常が発生した場合の処理を自動化します。	省略可	構築	5.3.2	-
トラブルシューティングに必要な資料を調査します。	不要	-	5.5	-
負荷分散機を閉塞して、システムの部分メンテナンスをします。	不要	-	簡易構築	-

(凡例)

必要：運用前の設定が必要です。

省略可：デフォルトの設定で使用できるため、運用前の設定は省略できます。この動作をデフォルトから変更する場合は設定が必要です。

不要：運用前の設定は不要です。

-：該当しません。

構築：マニュアル「Cosminexus システム構築ガイド」

簡易構築：マニュアル「Cosminexus 簡易構築・運用ガイド」

## 1.3.2 バッチアプリケーションを実行するシステムの場合

ここでは、バッチアプリケーションを実行するシステムの運用で実施する作業について説明します。バッチアプリケーションを実行するシステムでは、次の運用作業を実施します。

- システムの起動と停止
- システムの運用監視
- システムの処理性能解析
- リソースの管理
- システムの監査
- システムの変更
- システム環境の移行
- トラブルシューティング

これらの作業を実行するには、主に次の手段を使用します。

- Smart Composer 機能のコマンドを使用する
- 運用管理コマンド (mngsvrutil) を使用する
- Cosminexus のコマンドを使用する

なお、ここでは、Cosminexus システムで推奨する手段で、各運用作業を実行する場合について説明しています。これ以外の手段で実行する運用作業については、「付録 H 代替手段で実行できる運用作業」を参照してください。

### (1) システムの起動と停止で実行できる作業

システムの起動と停止で実行できる作業について、次の表に示します。なお、Cosminexus のシステムで実行できる運用作業には、運用前に設定が必要なものもあります。

表 1-10 システムの起動と停止で実行できる作業

運用作業	運用前の設定の要否	運用前の設定方法の参照先	運用方法	
			参照先	手段
サービスユニットを一括起動します。	不要	-	簡易構築	cmx_start_target コマンド
サービスユニットを一括停止します。	不要	-	簡易構築	cmx_stop_target コマンド
クラスタソフトウェアを使用したシステムを起動・停止します。	必要	構築	8 章, 9 章	Cosminexus のコマンド, クラスタソフトウェアの機能

(凡例)

- 必要：運用前の設定が必要です。
- 不要：運用前の設定は不要です。

- : 該当しません。

簡易構築 : マニュアル「Cosminexus 簡易構築・運用ガイド」

構築 : マニュアル「Cosminexus システム構築ガイド」

## (2) システムの運用監視で実行できる作業

システムの運用監視で実行できる作業について、次の表に示します。なお、Cosminexus のシステムで実行できる運用作業には、運用前に設定が必要なものもあります。

表 1-11 システムの運用監視で実行できる作業

運用作業	運用前の設定の要否	運用前の設定方法の参照先	運用方法	
			参照先	手段
サービスユニットのステータスを確認します。	不要	-	簡易構築	cmx_list_status コマンド
論理サーバのステータスを確認します。	不要	構築	2.2	運用管理コマンド (mngsvrutil)
システム内のサービスユニットの構成を確認します。	不要	-	簡易構築	cmx_list_model コマンド
稼働情報ファイルを参照して、バッチサーバの稼働情報を監視します。	省略可	構築	2.3	-
運用管理コマンドを使用して、バッチサーバの稼働情報を監視します。	不要	-	2.4	運用管理コマンド (mngsvrutil)
リソースの使用率や使用数を監視して、しきい値を超えた場合に原因を調査します。	省略可	構築	2.5	-
稼働情報ファイルに出力される稼働情報、またはリソースの枯渇状況に応じてメッセージが出力された場合の処理を自動化します (Management アクション)。	必要	構築	-	-

### (凡例)

必要 : 運用前の設定が必要です。

省略可 : デフォルトの設定で使用できるため、運用前の設定は省略できます。この動作をデフォルトから変更する場合は設定が必要です。

不要 : 運用前の設定は不要です。

- : 該当しません。

簡易構築 : マニュアル「Cosminexus 簡易構築・運用ガイド」

構築 : マニュアル「Cosminexus システム構築ガイド」

## (3) システムの処理性能解析で実行できる作業

システムの処理性能解析で実行できる作業について、次の表に示します。

## 1. システム運用の概要

表 1-12 システムの処理性能解析で実行できる作業

運用作業	運用前の設定の要否	運用前の設定方法の参照先	運用方法	
			参照先	手段
性能解析トレースファイルを集めます。	不要	-	3.2	運用管理コマンド (mngsvrutil)
収集した性能解析トレースファイルを利用してシステムの処理性能を解析します。	不要	-	3.3	-

(凡例)

不要：運用前の設定は不要です。

-：該当しません。

### (4) リソースの管理で実行できる作業

リソースの管理で実行できる作業について、次の表に示します。なお、Cosminexus のシステムで実行できる運用作業には、運用前に設定が必要なものもあります。

表 1-13 リソースの管理で実行できる作業

運用作業	運用前の設定の要否	運用前の設定方法の参照先	運用方法	
			参照先	手段
コネクションプールの情報を表示します。	不要	-	解説	サーバ管理コマンド (cjlistpool)
コネクションプールをクリアします。	不要	-	解説	サーバ管理コマンド (cjclearpool)
クラスタ構成のデータベースをメンテナンスします。	不要	-	4.8	サーバ管理コマンド (cjlistrar, cjsuspendpool, cjresumepool)
リソースの接続テストをします。	必要	構築	解説	サーバ管理コマンド (cjtestres)

(凡例)

必要：運用前の設定が必要です。

不要：運用前の設定は不要です。

-：該当しません。

解説：マニュアル「Cosminexus 機能解説」

構築：マニュアル「Cosminexus システム構築ガイド」

リファレンス：マニュアル「Cosminexus リファレンス コマンド編」

### (5) システムの監査で実行できる運用作業

監査ログを使用したシステムの監査、およびデータベース監査証跡と連携した運用で実行できる作業について、次の表に示します。なお、Cosminexus のシステムで実行でき

る運用作業には、運用前に設定が必要なものもあります。

表 1-14 システムの監査で実行できる作業

運用作業	運用前の設定の要否	運用前の設定方法の参照先	運用方法	
			参照先	手段
監査ログを出力して、管理します。	必要	構築	6.2	-
監査ログを収集して調査します。	必要	構築	6.2	-
データベース監査証跡に出力された情報を基に、データベースにアクセスしたリクエストの情報を特定します。	不要	-	7.2	-

(凡例)

必要：運用前の設定が必要です。

不要：運用前の設定は不要です。

-：該当しません。

構築：マニュアル「Cosminexus システム構築ガイド」

## (6) システムの変更で実行できる運用作業

システムの変更で実行できる作業について、次の表に示します。

表 1-15 システムの変更で実行できる作業

運用作業	運用前の設定の要否	運用前の設定方法の参照先	運用方法		
			参照先	手段	
設定変更	システム全体の設定を一括で変更します。	不要	-	簡易構築	-
構成変更	システムをスケールアウトします。	不要	-	簡易構築	-
	システムをスケールインします。	不要	-	簡易構築	-
削除	システムを削除します。	不要	-	簡易構築	-

(凡例)

不要：運用前の設定は不要です。

-：該当しません。

簡易構築：マニュアル「Cosminexus 簡易構築・運用ガイド」

## (7) システム環境の移行で実行できる運用作業

システム環境の移行で実行できる作業について、次の表に示します。

## 1. システム運用の概要

表 1-16 システム環境の移行で実行できる作業

運用作業	運用前の設定の要否	運用前の設定方法の参照先	運用方法	
			参照先	手段
構築済みのシステムの環境を移行します。	不要	-	簡易構築	-

(凡例)

不要：運用前の設定は不要です。

-：該当しません。

簡易構築：マニュアル「Cosminexus 簡易構築・運用ガイド」

### (8) トラブルシューティングで実行できる作業

トラブルシューティングで実行できる作業について、次の表に示します。なお、Cosminexus のシステムで実行できる運用作業には、運用前に設定が必要なものもあります。

表 1-17 トラブルシューティングで実行できる作業

運用作業	運用前の設定の要否	運用前の設定方法の参照先	運用方法	
			参照先	手段
トラブルシューティングに必要な資料を収集します。	必要 / 省略可	構築	5.3.1	運用管理コマンド ( mngsvrutil )
論理サーバに異常が発生した場合の処理を自動化します。	省略可	構築	5.3.2	-
トラブルシューティングに必要な資料を調査します。	不要	-	5.5	-

(凡例)

必要：運用前の設定が必要です。

省略可：デフォルトの設定で使用できるため、運用前の設定は省略できます。この動作をデフォルトから変更する場合は設定が必要です。

不要：運用前の設定は不要です。

-：該当しません。

構築：マニュアル「Cosminexus システム構築ガイド」

## 1.4 システムの運用で使用する機能

---

ここでは、システムの運用で使用する機能の概要について説明します。システムの運用で使用する機能には、サーバプロセスを操作する機能とアプリケーションを操作する機能の二つがあります。それぞれの機能の概要について次に説明します。

### (1) サーバプロセスを操作する機能

サーバプロセスを操作する機能について説明します。

#### Smart Composer 機能

システムを簡単に構築・運用するための機能です。

Smart Composer 機能を使用して構築したシステムでは、J2EE サーバや Web サーバで構成される、業務サービスを提供する閉じた部分系を、サービスユニットとして扱います。Smart Composer 機能を使用することで、サービスユニット単位で一括したシステム運用ができます。

#### 運用管理コマンド (mngsvrutil)

Management Server の運用管理の対象となる、論理サーバの運用操作を実行するコマンドです。

#### 運用管理ポータル

Web ブラウザに表示されるポータル画面を通して、Management Server の運用管理の対象となる論理サーバの運用操作を実行する機能です。

#### J2EE サーバを操作するコマンド

J2EE サーバを起動、停止するためのコマンドです。cjstartsv コマンド、および cjstopsv コマンドがあります。

#### CTM のコマンド

リクエストのスケジューリングをするための機能を提供している、CTM を操作するためのコマンドです。CTM は、構成ソフトウェアに Cosminexus Component Transaction Monitor を含む製品だけで利用できます。利用できる製品については、マニュアル「Cosminexus 概説」を参照してください。

#### PRF のコマンド

システムの処理性能を解析するとき使用する、性能解析トレース情報を編集してファイルに出力したり、PRF トレースファイルの取得レベルを設定したりするコマンドです。

### (2) アプリケーションを操作する機能

#### サーバ管理コマンド

J2EE サーバで管理しているアプリケーションおよびリソースの設定をするためのコマンドです。

## 1. システム運用の概要

### Server Plug-in

リソースアダプタや J2EE アプリケーションを設定するための GUI です。Eclipse のプラグインとして使用できます。



## 1.5 システム運用の作業

---

ここでは、日常的なシステムの運用で実施する作業について説明します。

日常的なシステム運用の作業は、次のような作業で構成されます。

- システムの起動 / 停止
- システムの運用状況の監視
- システムの処理性能の監視
- J2EE アプリケーションの運用

なお、システムの起動以外の運用作業は、すべてシステムが起動した状態で実施する作業です。

注 J2EE アプリケーションの運用は、バッチアプリケーションを実行するシステムの運用では実施しません。

### 1.5.1 システムを起動 / 停止する

Cosminexus で構築したシステムを運用するためには、Cosminexus の構成ソフトウェアの複数のサーバプロセスを適切な順序で起動する必要があります。システムの構成によっては、同じ種類のサーバプロセスを複数起動することもあります。また、サービスを開始するためには、J2EE アプリケーションの開始も必要です。

Cosminexus のシステムでは、J2EE サーバや Web サーバで構成される業務サービスを提供する閉じた部分系を、サービスユニットとして管理します。Smart Composer 機能を使用して、サービスユニット単位でシステムを一括して起動したり、停止したりできます。Smart Composer 機能を使用したシステムの起動・停止方法については、マニュアル「Cosminexus 簡易構築・運用ガイド」を参照してください。

---

#### 参考

このマニュアルでは、クラスタソフトウェアを使用したシステムの起動・停止方法について説明しています。

---

### 1.5.2 システムの運用状況を監視する

ここでは、システムの運用状況を監視する作業の概要について説明します。

システムの運用状況を監視する方法の詳細については、「2. システムの運用監視」を参照してください。

#### (1) 論理サーバのステータスを監視する

システムを構成するホストおよびサーバのステータスを確認して、システムが正常に稼

## 1. システム運用の概要

働しているかどうかを監視できます。この情報を基に、必要に応じて起動、停止、再起動ができます。

### (2) 論理サーバの稼働情報を監視する

論理サーバや論理サーバで稼働しているアプリケーションの稼働情報を基に、システムの性能を監視できます。システムの性能を監視することで、システム内の問題個所を発見したり、パフォーマンスチューニングに役立てたりすることができます。

---

#### 参考

論理サーバの稼働情報を監視するには、稼働情報ファイルを使用できます。稼働情報ファイルに出力されるフルガーベージコレクション回数、または URL グループ単位の実行待ちリクエスト数がしきい値を超えたときに、メッセージが出力されます。稼働情報ファイルに出力される情報を基にしてメッセージを出力する機能を、しきい値イベントといいます。また、しきい値イベントによってメッセージが出力されたときに、適切な処置を自動実行するように設定できます。しきい値イベントによって出力されたメッセージを基にして自動実行する処理を、Management アクションといいます。

Management アクションの設定方法の詳細は、マニュアル「Cosminexus システム構築ガイド」の Management イベントによる処理の自動実行の設定に関する説明を参照してください。

---

### (3) リソースの使用率または使用数を基にリソースの枯渇を監視する

リソースの使用率または使用数の推移を監視して、使用率または使用数がしきい値を超えた場合、メッセージを出力できます。リソースの使用率または使用数の推移は、リソース枯渇監視情報として定期的に出力できます。しきい値を超えた場合、出力されているリソース枯渇監視情報を基に原因を分析して適切に対処することで、トラブルの発生を未然に防げます。また、リソース枯渇監視情報は、トラブル発生後の要因の解析にも役立ちます。

---

#### 参考

しきい値を超えたときに出力されるメッセージを使用して、システムにトラブルが発生する前に適切な処置を自動実行するように設定できます。例えば、JavaVM のヒープメモリを監視してフルガーベージコレクションの発生を防ぐ設定などができます。このように、リソース枯渇監視情報を基にして自動実行する処理を、Management アクションといいます。しきい値を超えたときに出力されるメッセージによって発生するイベントのことを、Management イベントといいます。Management イベントによる処理の自動実行の設定については、マニュアル「Cosminexus システム構築ガイド」の Management イベントによる処理の自動実行の設定に関する説明を参照してください。

---

### (4) CTM の稼働情報を監視する

CTM を利用しているシステムでは、CTM の稼働情報を出力できます。また、運用管理ドメイン内の CTM の稼働情報をまとめて収集、出力することもできます。

なお、CTM は、構成ソフトウェアに Cosminexus Component Transaction Monitor を含む製品だけで利用できます。利用できる製品については、マニュアル「Cosminexus 概説」を参照してください。

### (5) トランザクションの状態を確認する

起動中および停止中の J2EE サーバのトランザクションの状態を確認することができます。稼働中のトランザクションの状態、未決着のトランザクションがあるかどうかなどを把握できます。

## 1.5.3 システムの処理性能を監視する

システムを構成する各種ソフトウェアの稼働状況やリクエストの処理過程で各サーバが出力したトレース情報などを監視して、システム全体の処理性能を確認できます。処理性能を監視すると、システムのボトルネックの調査や、パフォーマンスチューニングを実施する必要があるかどうかの判断ができるようになります。

Cosminexus で構築したシステムでは、リクエストの処理過程で各サーバが出力するトレース情報を、パフォーマンストレーサというプロセスを使用してトレースファイルに出力します。このトレースファイルには、クライアントから送信されたリクエストが、データベースなどの EIS で処理されて、その処理結果がクライアントに返却されるまでのトレース情報が出力されます。また、セッションのライフサイクルを判断するための情報もこのファイルに出力されます。

システムの運用作業では、このトレース情報を基に、システムの処理性能やボトルネックを解析できます。なお、性能解析時には、性能解析トレースファイルを使用します。性能解析トレースファイルとは、パフォーマンストレーサが出力したバイナリ形式のファイル（PRF トレースファイル）をテキスト形式に編集したファイルです。

トレース情報の収集時には、運用管理ドメイン内のホスト上で動作している、任意のパフォーマンストレーサが出力する性能解析トレースファイルを収集します。運用管理ドメイン内のすべてのホストの性能解析トレースファイルをまとめて収集することもできます。

システムの処理性能を監視する方法については、「3. システムの処理性能の解析」を参照してください。

## 1.5.4 J2EE アプリケーションを運用する

ここでは、J2EE アプリケーションを運用する作業の概要を説明します。

J2EE アプリケーションの運用方法の詳細については、「4. J2EE アプリケーションとリソースの管理」を参照してください。

## 1. システム運用の概要

### (1) J2EE アプリケーションの稼働状態に応じたパフォーマンスチューニングを実行する

J2EE アプリケーションに含まれる Web アプリケーションの稼働状況を監視して、必要に応じて、同時実行スレッド数を変更できます。このとき、J2EE アプリケーションを停止する必要はありません。これによって、クライアントにサービスを提供しながら、J2EE アプリケーションのパフォーマンスをチューニングできます。

### (2) J2EE アプリケーションを停止する

一日のうち特定の時間だけクライアントにサービスを提供する J2EE アプリケーションでは、毎日決まった時間に J2EE アプリケーションを停止して、次の日に再度開始する作業が必要です。また、24 時間連続稼働させているサービスをメンテナンスする場合は、一度にすべての J2EE アプリケーションを停止するのではなく、一部ずつ停止させて、サービスを提供し続けられるような運用を検討する必要があります。J2EE アプリケーションを停止するとき、まず、リクエストの受け付けを停止して、すでに受け付けたリクエストについては処理を実行してから J2EE アプリケーションを停止させる方法を、サービス閉塞といいます。日常運用の中で J2EE アプリケーションを停止する場合、サービス閉塞を適切に実行することで、安全にサービスを停止できます。実行できるサービス閉塞の方法は、J2EE アプリケーションの構成やシステムの構成によって異なります。

## 1.6 システム保守の作業

---

Cosminexus のシステム上で動作する J2EE アプリケーションに対して実行できる保守作業を次に示します。

- J2EE アプリケーションおよびリソースの保守
- システムの動作環境の設定変更
- トラブルへの対処

なお、システムの運用形態や業務の変化に応じて、システムの動作環境の設定を変更する場合、システム構築時と同じ手順で設定します。システム構築の方法については、マニュアル「Cosminexus システム構築ガイド」を参照してください。

### 1.6.1 J2EE アプリケーションとリソースを保守する

ここでは、J2EE アプリケーションとリソースを保守する作業の概要を説明します。

J2EE アプリケーションとリソースの保守の詳細については、「4. J2EE アプリケーションとリソースの管理」を参照してください。

#### (1) タイムアウトしたリクエストをキャンセルする

J2EE アプリケーションで実行しているリクエストが無限ループなどの原因によって制御できなくなった場合、リクエストを強制的にキャンセルして、リソースを解放する必要があります。あらかじめ J2EE アプリケーション内のメソッドの実行状態を監視しておくことで、予定した時間内に完了しないリクエストに対して、キャンセルを実行できます。

#### (2) J2EE アプリケーションを強制停止する

日常運用で J2EE アプリケーションを停止する場合や、トラブルが発生して即座に J2EE アプリケーションを停止したい場合などに、J2EE アプリケーション内のリクエストが完了していないことが原因で停止処理が完了しないことがあります。この場合、J2EE アプリケーションを強制的に停止する必要があります。

J2EE アプリケーションの強制停止では、完了しないリクエストを強制的に終了させます。これによって、J2EE アプリケーションを停止できる状態にできます。

#### (3) J2EE アプリケーションを入れ替える

運用中の J2EE アプリケーションをバージョンアップしたりメンテナンスしたりする場合、J2EE アプリケーションの入れ替え作業を実施します。

J2EE アプリケーションを入れ替える場合にサービスを停止したくないときは、適切なサービス閉塞をする必要があります。CTM を利用して J2EE サーバへのクライアントが

## 1. システム運用の概要

らのリクエストを制御して、システム全体を停止させないで J2EE アプリケーションを入れ替えられます。これによって、クライアントからのリクエストを受け付けながら、オンライン状態で J2EE アプリケーションを入れ替えられます。なお、CTM で制御できるのは、RMI-IIOP 通信を使用する、Stateless Session Bean に対するリモートインタフェース呼び出しのリクエストだけです。

また、J2EE アプリケーションを入れ替える場合、新しい J2EE アプリケーションが本番環境で問題なく動作することを確認してから入れ替えたい場合があります。Cosminexus のシステムでは、テストモードというモードを利用して、J2EE アプリケーションを本番環境でテストしてから本番運用に切り替えられます。

なお、CTM は、構成ソフトウェアに Cosminexus Component Transaction Monitor を含む製品だけで利用できます。利用できる製品については、マニュアル「Cosminexus 概説」を参照してください。

### (4) リソースをメンテナンスする

運用中に、システムで使用しているリソースをメンテナンスする場合、システムの運用状況に応じてリソースをメンテナンスします。Cosminexus のシステムでは、JavaBeans リソースやクラスタ構成のデータベースをメンテナンスする場合、リソースを入れ替えるために一時停止・再開したり、状態を確認したりできます。

## 1.6.2 トラブルに対処する

ここでは、トラブルに対処する作業の概要について説明します。

トラブルに対処する方法の詳細については、「5. トラブルシューティング」を参照してください。

トラブル発生時、構成ソフトウェアでは、トラブルシュート情報としてその時点の状態をログに出力します。このログを収集・分析して、トラブルの発生要因を調査します。また、性能解析トレースを使用して、リクエストの処理状況からトラブルの発生個所を特定することもできます。

Cosminexus で構築したシステムでは、構成ソフトウェアのトラブル発生時のログと定義ファイルを一括して収集できます。この情報を snapshot ログといいます。運用管理ドメイン内の論理サーバが停止される直前、または J2EE サーバを手動で一括再起動する直前に、snapshot ログの一括収集が自動で実行されます。

また、CTM を利用すると、J2EE サーバが異常終了した場合でも、すぐにクライアントにエラーが返却されないように設定できます。J2EE サーバが再起動されるまでの間、CTM によって J2EE アプリケーションのスケジュールキューが閉塞制御され、すでに登録されているリクエストを保持して、さらにクライアントからのリクエストを受け付け続けます。これによって、すぐに再起動すれば、クライアントにトラブルを気づかせないで運用を続けられます。

なお、CTM は、構成ソフトウェアに Cosminexus Component Transaction Monitor を含む製品だけで利用できます。利用できる製品については、マニュアル「Cosminexus 概説」を参照してください。

## 1.7 システム監査の作業

---

Cosminexus のシステムで実行できるシステム監査の作業を次に示します。

- 監査ログを使用したシステムの監査
- データベース監査証跡と連携した運用

### 1.7.1 監査ログを使用してシステムを監査する

ここでは、監査ログを使用してシステムを監査する作業の概要について説明します。

監査ログを使用してシステムを監査する作業の詳細については、「6. 監査ログを使用したシステムの監査」を参照してください。

監査ログとは、システム構築者や運用者が Cosminexus のプログラムに対して実行した操作、およびその操作に伴うプログラムの動作の履歴が出力されるファイルです。監査者が監査ログを調査することで、「いつ」「だれが」「何をしたか」を知ることができ、システムの運用が法規制やセキュリティ評価基準、または業界ごとの各種の基準に準拠していることを証明できます。

### 1.7.2 データベース監査証跡と連携して運用する

ここでは、データベース監査証跡と連携した運用の作業の概要について説明します。

データベース監査証跡と連携した運用の詳細については、「7. データベース監査証跡と連携した運用」を参照してください。

データベース監査証跡とは、データベースの監査に利用するためのデータベースの操作記録です。データベース監査証跡に関する機能はデータベースが提供しています。Cosminexus システムでは、データベースの監査証跡機能と連携し、J2EE サーバやバッチサーバがデータベースにアクセスした際に記録されるデータベース監査証跡の情報の一部に、Cosminexus システムが持つ情報を出力させることができます。この機能をデータベース監査証跡連携機能といいます。データベース監査証跡連携機能については、マニュアル「Cosminexus 機能解説」のデータベース監査証跡との連携に関する説明を参照してください。

データベース監査証跡と連携した運用では、データベース監査証跡、性能解析トレースおよびアプリケーションのユーザログを利用して、データベースの監査対象のテーブルにアクセスしたリクエストを特定できます。



## 1.8 システム運用を始める前に

ここでは、システム運用を始める前に確認、準備しておくことについて説明します。

### 1.8.1 Component Container 管理者による運用に関する留意事項

UNIX で Component Container 管理者を設定している場合の留意事項について説明します。Component Container 管理者とは、UNIX を使用している場合に、J2EE サーバ、またはバッチサーバの構築・運用に関する操作を実行できる権限を持つユーザです。Component Container 管理者を設定することで、root 権限を持たないユーザでも、J2EE サーバ、またはバッチサーバの起動・停止やサーバ管理コマンドの実行など、システム運用時の操作を実行できるようになります。

#### ！ 注意事項

Component Container 管理者を設定している場合、Component Container 管理者が実行できる操作は、一部の操作を除き、スーパーユーザでは操作できなくなるので注意してください。詳細については、マニュアル「Cosminexus 機能解説」の、J2EE サーバの運用とシステムの運用管理者に関する説明を参照してください。

なお、Component Container 管理者の設定については、マニュアル「Cosminexus システム構築ガイド」の Component Container 管理者の設定に関する説明を参照してください。

Component Container 管理者が実施できるシステム運用時の操作と、Component Container 管理者を設定している場合にパーミッションが変更されるディレクトリについて説明します。

#### (1) Component Container 管理者が実施できるシステム運用時の操作

Component Container 管理者を設定した場合に、スーパーユーザおよび Component Container 管理者が実施できるシステム運用時の操作を次の表に示します。

表 1-18 Component Container 管理者が実施できるシステム運用時の操作

操作	スーパーユーザ	Component Container 管理者
J2EE サーバ、バッチサーバ、および Web コンテナサーバの起動 / 停止 ( cjstartsv コマンド , cjstopsv コマンド )	×	
サーバ管理コマンドの実行		
スレッドダンプの取得 ( ejdumpsv コマンド )		

## 1. システム運用の概要

操作	スーパーユーザ	Component Container 管理者
OS 状態情報の取得 (cjetagetsysinfo コマンド)		
バッチアプリケーションの実行 (cjexecjob コマンド)		
バッチアプリケーションの強制停止 (cjkilljob コマンド)		

(凡例) : 実行できます。 × : 実行できません。

なお、Component Container 管理者を設定しない場合は、スーパーユーザが運用管理者となるため、Component Container 管理者の操作もスーパーユーザが実施します。

### 参考

Component Container 管理者を設定していても、運用管理コマンドの実行には影響ありません。

Management Server のセットアップ時に設定した管理ユーザ ID で、システム運用時の操作およびコマンドを実行できます。

## (2) Component Container 管理者を設定している場合にパーミッションが変更されるディレクトリ

Component Container 管理者を設定している場合、次に示すディレクトリ下のサブディレクトリおよびファイルの属性 (所有者 / グループ) が「Component Container 管理者 / Component Container 管理者グループ」に変更されます。

- /opt/Cosminexus/CC
- <作業ディレクトリ>/ejb/
- <作業ディレクトリ>/web/

ただし、/opt/Cosminexus/CC 下の一部のファイル (/opt/Cosminexus/CC/server/bin/cjenvsetup、/opt/Cosminexus/CC/server/bin/cjetagetsysinfo や、インストールログなど) の属性は変更されません。

## 1.8.2 トラブル発生時に資料を収集するための準備

トラブルが発生した場合に備えて、資料を収集するための準備を必要に応じてしてください。

次に示す資料を収集したい場合は、運用開始前の準備が必要です。

日立固有の JavaVM のログ

JavaVM のガーベージコレクションのログなどは、このログに出力されます。

ユーザダンプ (Windows の場合, J2EE サーバモードの場合)

core ダンプ (UNIX の場合, J2EE サーバモードの場合)

OS の統計情報 (Windows の場合)

また, 必要であれば, 取得資料のサイズ, 出力レベルなどを調整しておきます。

トラブルシューティングの資料取得の設定については, マニュアル「Cosminexus システム構築ガイド」のトラブルシューティングの資料取得の設定に関する説明を参照してください。

### 1.8.3 Windows Vista 使用時の注意事項

Windows Vista を使用する場合には, 次の点に留意してください。

Cosminexus が提供するコマンドは, 管理者特権で実行する必要があります。

Cosminexus が提供するコマンドは, 「管理者: コマンドプロンプト」で実行してください。「管理者: コマンドプロンプト」は, Windows Vista で提供されている機能を使用して起動してください。起動方法の例を次に示します。

1. [スタート] ボタンをクリックします。
2. [プログラム] - [アクセサリ] を選択します。
3. [コマンド プロンプト] を右クリックして, [管理者として実行] をクリックします。

管理者のパスワードまたは確認を求められた場合は, 画面の指示に従って, パスワードを入力するか, または確認情報を提供してください。

### 1.8.4 運用管理コマンド (mngsvrutil) を利用する場合の留意事項

運用管理コマンド (mngsvrutil) は, 運用操作を実行するコマンドです。

mngsvrutil コマンドは, 運用管理エージェントおよび Management Server が起動している状態で実行できるコマンドです。運用管理エージェントおよび Management Server を起動してから実行してください。

なお, このマニュアルで説明する実行形式および実行例では, その操作に主に関連する引数だけを示して説明します。指定できる引数の詳細など, コマンドの詳細については, マニュアル「Cosminexus リファレンス コマンド編」を参照してください。

### 1.8.5 サーバ管理コマンドを利用する場合の留意事項

サーバ管理コマンドは, J2EE アプリケーションやリソースの設定およびカスタマイズを実行するコマンドです。

サーバ管理コマンドは, J2EE サーバが起動している状態で実行できるコマンドです。

## 1. システム運用の概要

J2EE サーバおよびその前提プロセスを起動してから実行してください。

サーバ管理コマンドには、参照系コマンド、更新系コマンド、および特権系コマンドがあります。これらは、組み合わせによって、同時に実行できるものと、排他処理されるものがあります。実行時には注意してください。サーバ管理コマンドの同時実行についての制約に関しては、マニュアル「Cosminexus アプリケーション設定操作ガイド」のサーバ管理コマンドの処理と制御に関する説明を参照してください。また、参照系コマンド、更新系コマンドおよび特権系コマンドの分類については、マニュアル「Cosminexus リファレンス コマンド編」を参照してください。

なお、このマニュアルで説明する実行形式および実行例では、その操作に主に関連する引数だけを示して説明します。指定できる引数の詳細など、コマンドの詳細については、マニュアル「Cosminexus リファレンス コマンド編」を参照してください。

### 1.8.6 作業ディレクトリに関する注意事項

作業ディレクトリとは、usrconf.cfg (J2EE サーバ用オプション定義ファイル) の `ejb.public.directory` キーに指定したディレクトリです。usrconf.cfg ファイルの詳細については、マニュアル「Cosminexus リファレンス 定義編」を参照してください。

作業ディレクトリの運用では、次の点に注意してください。

Cosminexus Component Container 以外のアプリケーションで、作業ディレクトリ以下のディレクトリおよびファイルを使用しないでください。J2EE サーバを起動した場合に、例外が発生することがあります。

作業ディレクトリ以下のディレクトリ構成を変更しないでください。また、ファイルを追加、または削除しないでください。

### 1.8.7 RMI レジストリに関する注意事項

Management Server は、J2EE サーバの運用監視に、J2EE サーバ上で動作する RMI レジストリを使用します。そのため、Management Server が管理している RMI レジストリの情報と実際に動作している J2EE サーバの RMI レジストリの情報が一致していないと、運用監視に失敗する場合があります。Management Server の管理している RMI レジストリの情報は、次のタイミングで更新されます。J2EE サーバ上で動作する RMI レジストリと異ならないように注意してください。

- 運用管理ポータル「Cosminexus Management Server の設定」で構成情報の回復を実行した場合
- 運用管理ポータル「論理サーバの環境設定」で「適用」ボタンを押した場合
- 運用管理ポータル「論理サーバの環境設定」からサーバの設定の読み込みを実行した場合
- `cmx_build_model.exe` コマンドを実行した場合
- `cmx_change_model.exe` コマンドを実行した場合

運用管理ポータルでの操作については、マニュアル「Cosminexus 運用管理操作ガイド」を参照してください。cmx\_build\_model.exe コマンド、および cmx\_change\_model.exe コマンドの詳細については、マニュアル「Cosminexus 簡易構築・運用ガイド」を参照してください。

## 1.8.8 コマンド実行時に指定する Web システム名，サービスユニット名，およびサーバ名に関する注意事項

運用作業でコマンドを実行する際、コマンドによって次の四つの異なる名称を指定します。各名称の対応を確認するには、コマンドや簡易構築定義ファイルを使用する必要があります。

- Web システム名  
Smart Composer 機能のコマンドに指定します。  
例：cmx\_start\_target -s <Web システム名 >
- サービスユニット名  
Smart Composer 機能のコマンドに指定します。  
例：cmx\_start\_target -s <Web システム名 > -unit < サービスユニット名 >
- 論理サーバ名  
運用管理コマンド (mngsvrutil) に指定します。  
例：mngsvrutil start -t < 論理サーバ名 >
- 実サーバ名  
サーバ管理コマンドに指定します。  
例：cjstartsv < 実サーバ名 >

各名称の対応は、次の方法で確認してください。

1. cmx\_list\_model コマンドを実行することで、次の名称の対応を確認します。

- ホスト名
- Web システム名
- サービスユニット名
- 論理サーバ名

実行例

```
cmx_list_model -m localhost:28080 -u admin -p admin -modeltype SERVEREX
cmx_sample-system_unit1_PRF_01,PerformanceTracer,host1,combined-tier,unit1,sam
ple-system
cmx_sample-system_unit1_J2EE_01,J2EEServer,host1,combined-tier,unit1,sampl
e-system
cmx_sample-system_unit1_HTTP_01,WebServer,host1,combined-tier,unit1,sampl
e-system
cmx_sample-system_unit2_PRF_01,PerformanceTracer,host2,combined-tier,unit2,sam
ple-system
cmx_sample-system_unit2_J2EE_01,J2EEServer,host2,combined-tier,unit2,sampl
e-system
cmx_sample-system_unit2_HTTP_01,WebServer,host2,combined-tier,unit2,sampl
e-system
```

## 1. システム運用の概要

2. 簡易構築定義ファイルで、次の名称の対応を確認します。

- 論理サーバ名 (logical-server-name)  
論理サーバ名が定義されていない場合は、自動生成されます。
- 実サーバ名 (realservername)  
実サーバ名が定義されていない場合は、論理サーバ名と同じになります。

# 2

## システムの運用監視

この章では、システムの運用監視について説明します。運用開始後のシステムで、システムの稼働状況やトランザクション情報などを適宜確認して、安定した稼働状態を保っていくための運用作業のことを、システムの運用監視といいます。運用監視は、運用管理コマンド (`mngsvrutil`)、およびサーバ管理コマンドで実行できます。

このほか、Cosminexus で構築したシステムでは、J2EE サーバまたは SFO サーバでのリソースの使用率または使用数を基にリソース枯渇を監視して推移をログに出力したり、CTM の稼働統計情報を収集したりできます。これらの情報を監視することで、システムの安定稼働を図ります。

- 
- 2.1 運用監視作業の概要
  - 2.2 論理サーバのステータスの監視
  - 2.3 稼働情報の監視（稼働情報ファイル）
  - 2.4 稼働情報の監視（運用管理コマンド）
  - 2.5 リソース枯渇の監視
  - 2.6 CTM の稼働統計情報の確認
  - 2.7 トランザクション情報の確認
-

## 2.1 運用監視作業の概要

システムの運用を開始したら、論理サーバのステータスやシステムの稼働情報を適宜確認して、安定した稼働状態を保っていくことが必要です。稼働情報に出力されたシステム性能を基に、チューニングが必要かどうか判断する必要があります。

システムの運用監視作業について、次の表に示します。

表 2-1 システムの運用監視

作業内容	手段	作業概要	参照先
論理サーバのステータス監視	運用管理コマンド (mngsvrutil)	論理サーバのステータスを確認します。論理サーバのステータスは、次の単位で確認できます。 <ul style="list-style-type: none"> <li>運用管理ドメイン単位</li> <li>ホスト単位</li> <li>論理サーバ単位</li> </ul>	2.2
システムの稼働監視	稼働情報収集機能 <sup>1</sup> <sup>2</sup>	J2EE サーバおよび SFO サーバの稼働情報を監視します。	2.3
	運用管理コマンド (mngsvrutil)	J2EE サーバ、SFO サーバ、またはバッチサーバの稼働情報を監視します。	2.4
	リソース枯渇監視機能 <sup>1</sup> <sup>3</sup>	リソース枯渇監視ログファイルを調査して、リソースの使用率や使用数を監視します。また、しきい値を超えた場合に原因を調査します。	2.5
	運用管理コマンド (mngsvrutil)	運用管理ドメイン内の CTM の稼働統計情報を確認します。次の単位で確認できます。 <ul style="list-style-type: none"> <li>運用管理ドメイン内のすべての CTM</li> <li>運用管理ドメイン内の指定した CTM、または指定したホスト内の CTM</li> </ul>	2.6
トランザクション情報の確認	サーバ管理コマンド (cjlisttrn)	稼働中の J2EE サーバでのトランザクションの状態や、停止中の J2EE サーバでの未決着のトランザクションの有無などの情報を確認します。	2.7

注 1 システム構築時に設定した一定の間隔ごとにファイルが出力されるため、サーバ管理コマンドや運用管理コマンドなどのツールを使用して出力する必要はありません。

注 2 稼働情報ファイルに出力されるフルガーベージコレクション回数、または URL グループ単位の実行待ちリクエスト数がしきい値を超えたときに、メッセージが出力されます（しきい値イベント）。このメッセージを利用して Management イベントを発行し、対処を Management アクションとして自動化できます。しきい値イベントの設定については、マニュアル「Cosminexus システム構築ガイド」の、稼働情報ファイルの取得とイベント発行の設定に関する説明を参照してください。また、Management イベントによる処理の自動実行の設定については、マニュアル「Cosminexus システム構築ガイド」の、Management イベントによる処理の自動実行の設定に関する説明を参照してください。

注 3 リソースの使用状況がしきい値を超えた場合、Management イベントを発行できます。Management イベントを利用すると、対処を Management アクションとして自動化できます。Management イベントによる処理の自動実行の設定については、マニュアル「Cosminexus シス



「システム構築ガイド」の「Management イベントによる処理の自動実行の設定に関する説明を参照してください。

## 2.2 論理サーバのステータスの監視

この節では、論理サーバの稼働状態（ステータス）の監視方法と、監視できる項目について説明します。

論理サーバとは、Management Server の運用管理の対象になる、サーバまたはクラスタです。

論理サーバのステータスの監視には、運用管理コマンド（mngsvrutil）を使用します。

なお、cmx\_list\_status コマンドを使用して、サービスユニットのステータスを監視できます。cmx\_list\_status コマンドの使用方法については、マニュアル「Cosminexus 簡易構築・運用ガイド」を参照してください。

### 参考

Management Server および運用管理エージェントのステータスを確認するには、次の方法があります。

- 運用管理コマンド（mngsvrutil）のサブコマンド「check」
- adminagentcheck コマンド

運用管理コマンド、および adminagentcheck コマンドの詳細については、マニュアル「Cosminexus リファレンス コマンド編」を参照してください。

### 2.2.1 ステータスの監視方法

論理サーバのステータスは、運用管理コマンド（mngsvrutil）で監視できます。

論理サーバのステータスを監視するには、運用管理コマンド（mngsvrutil）にサブコマンド「list」を指定して実行します。これによって、論理サーバのステータス情報を標準出力に出力したり、CSV 形式または SNMP 連携用形式にファイル出力したりできます。

「list」の引数に指定する値によって、ステータスを出力する対象を指定できます。

運用管理コマンドおよびそのサブコマンドの詳細については、マニュアル「Cosminexus リファレンス コマンド編」を参照してください。

次に、実行形式と実行例を示します。

運用管理ドメイン内の論理サーバ名とステータス情報を出力する場合

実行形式

```
mngsvrutil -m <Management Serverのホスト名> -u <管理ユーザID> -p <管理パスワード>  
list status
```

## 実行例

```
mngsvrutil -m mnghost -u user01 -p pw1 list status
```

指定した論理サーバにインポートされている J2EE アプリケーション名とステータス情報を出力する場合

## 実行形式

```
mngsvrutil -m <Management Serverのホスト名> -u <管理ユーザID> -p <管理パスワード> -t <論理サーバ名> list appStatus
```

## 実行例

```
mngsvrutil -m mnghost -u user01 -p pw1 -t J2EEserver1 list appStatus
```

## 2.2.2 ステータス監視で確認できる項目

ステータス監視では、論理サーバの起動 / 停止状態が確認できます。ステータスは、すべての論理サーバで確認できます。

ステータスは、次の単位で確認できます。

### 運用管理ドメイン単位

運用管理ドメインに含まれるすべての論理サーバのステータスをまとめて確認できます。

### ホスト単位

選択したホスト内のすべての論理サーバのステータスをまとめて確認できます。

### 論理サーバ単位

選択した論理サーバのステータスを確認できます。種類ごとに確認することも、個々の論理サーバだけを確認することもできます。

確認できるステータスの種類と意味を次に示します。

表 2-2 稼働状況のステータスの種類と意味

稼働状況のステータス	説明
停止	初期状態および停止要求を受け付け停止処理が完了したあとの、論理サーバが停止した状態です。または、通信障害の回復後に運用管理エージェントが停止状態であることを確認できた状態です。
起動中	起動要求を受け付けてから論理サーバが稼働状態になるまでの起動処理中の状態です。

## 2. システムの運用監視

稼働状況のステータス	説明
稼働中	起動要求を受け付け起動処理が完了したあとの、論理サーバが稼働している状態です。または、通信障害の回復後に運用管理エージェントが稼働状態であることを確認できた状態です。
停止中	停止要求を受け付けてから論理サーバが停止状態になるまでの停止処理中の状態です。
強制停止中	強制停止要求を受け付けてから論理サーバが停止状態になるまでの強制停止処理中の状態です。
異常停止	停止要求を受け付けていない状態で、論理サーバの停止を検出した状態です。
回復中	異常停止状態で起動要求を受け付けてから稼働状態になるまでの、論理サーバの起動処理中の状態です。
通信障害	運用管理エージェントとの通信で障害が発生し、ステータスの表示ができない状態です。
自動停止中	運用管理エージェントから論理サーバの異常通知（プロセスはあるが動作していない状態の通知）を受けて論理サーバの強制停止中の状態です。
自動再起動中	自動再起動指定ありの論理サーバで、稼働中状態に運用管理エージェントから論理サーバの停止通知を受けて自動再起動処理中の状態です。
計画停止中	計画停止要求を受け付けてから、論理 Web サーバが停止状態になるまでの計画停止処理中の状態です。このステータスに遷移するのは、論理 Web サーバの場合だけです。

## 2.3 稼働情報の監視（稼働情報ファイル）

---

この節では、稼働情報ファイルを使用した、J2EE サーバ、および SFO サーバの稼働情報の監視について説明します。

稼働情報ファイルとは、J2EE サーバと SFO サーバ内の稼働情報が、定期的に出力されるファイルです。稼働情報ファイルは、J2EE サーバと SFO サーバの機能ごとに出力されます。稼働情報ファイルを参照すると、次のことができます。

- J2EE サーバ・SFO サーバの稼働状態を確認する
- J2EE サーバ・SFO サーバの稼働実績を統計情報として参照する
- J2EE サーバ・SFO サーバの設定パラメタをチューニングするときの参考にする

なお、稼働情報ファイルに出力されるフルガーページコレクション回数、または URL グループ単位の実行待ちリクエスト数を監視して、しきい値を超えた場合にメッセージを出力できます（しきい値イベント）。また、出力されたメッセージを利用して Management イベントを発行し、このアラートに対応して必要な処理を自動実行するように設定できます。Management イベントによる処理の自動実行の設定については、マニュアル「Cosminexus システム構築ガイド」の Management イベントによる処理の自動実行の設定に関する説明を参照してください。

---

### ポイント

稼働情報の監視を行う場合、通常の運用では、ここで説明している稼働情報ファイルを使用した稼働監視を行ってください。稼働情報ファイルで取得できる情報より詳細な情報を取得したい場合だけ、運用管理コマンドを使用した稼働情報の監視を行ってください。運用管理コマンドを使用した稼働情報の監視は、「2.4 稼働情報の監視（運用管理コマンド）」を参照してください。

---

### 2.3.1 稼働情報ファイルで収集できる情報

稼働情報ファイルを使用すると、J2EE サーバ、および SFO サーバの機能ごとの稼働情報を収集できます。稼働情報ファイルには、稼働情報として、サーバ性能、およびサーバリソースに関する情報が出力されます。稼働情報ファイルの対象となる J2EE サーバと SFO サーバの機能、およびそれぞれの機能で取得できる稼働情報の種類を次に示します。

## 2. システムの運用監視

表 2-3 機能ごとに取得できる稼働情報の種類

機能	機能の説明	稼働情報の種類
JavaVM	J2EE サーバ・SFO サーバが使用する JavaVM	<ul style="list-style-type: none"> <li>• JavaVM ヒープサイズ</li> <li>• コピーガーベージコレクション回数</li> <li>• フルガーベージコレクション回数</li> <li>• ロードされているクラス数</li> <li>• 稼働中のスレッド数</li> <li>• モニタロックのためにブロック状態であるスレッド数</li> </ul>
プロセスリソース	J2EE サーバ・SFO サーバが使用するプロ セスリソース	<ul style="list-style-type: none"> <li>• スレッド数</li> <li>• ファイルディスクリプタ数</li> </ul>
Stateful Session Bean	J2EE サーバ上で動作 する Stateful Session Bean	<ul style="list-style-type: none"> <li>• 接続セッション数</li> </ul>
Stateless Session Bean	J2EE サーバ上で動作 する Stateless Session Bean	<ul style="list-style-type: none"> <li>• プールされたインスタンス数</li> <li>• プール内の使用中インスタンス数</li> <li>• 実行待ちリクエスト数</li> </ul>
Message-driven Bean	J2EE サーバ上で動作 する Message-driven Bean	<ul style="list-style-type: none"> <li>• プールされたインスタンス数</li> <li>• プール内の使用中インスタンス数</li> <li>• 受け付けメッセージ数</li> </ul>
DB Connector	J2EE サーバ上で動作 する DB Connector	<ul style="list-style-type: none"> <li>• プールされた PreparedStatement 数</li> <li>• プールされた CallableStatement 数</li> <li>• PreparedStatement メソッドが呼び出された回数</li> <li>• PrepareCall メソッドが呼び出された回数</li> <li>• プール内 PreparedStatement ヒット回数</li> <li>• プール内 CallableStatement ヒット回数</li> </ul>
JCA リソース	リソースアダプタが使用 する JCA リソース	<ul style="list-style-type: none"> <li>• プールされたコネクション数</li> <li>• プール内の使用中コネクション数</li> <li>• コネクション取得待ちスレッド数</li> <li>• コネクション取得失敗数</li> </ul>
トランザクションサー ビス	J2EE サーバで使用さ れるトランザクション サービス	<ul style="list-style-type: none"> <li>• 決着済みトランザクション数</li> <li>• トランザクションロールバック数</li> </ul>
Web アプリケーション	J2EE サーバ上で動作 する Web アプリケー ション	<ul style="list-style-type: none"> <li>• 同時実行スレッド数</li> <li>• 実行待ちリクエスト数</li> <li>• 実行待ちリクエスト数の上限からあふれたリク エスト数</li> <li>• 受け付けリクエスト数</li> <li>• 応答済みリクエスト数</li> <li>• セッション数</li> </ul>
Web コンテナ	J2EE サーバ上で動作 する Web コンテナ	<ul style="list-style-type: none"> <li>• 同時実行スレッド数</li> <li>• 実行待ちリクエスト数</li> <li>• 実行待ちリクエスト数の上限からあふれたリク エスト数</li> </ul>

機能	機能の説明	稼働情報の種類
URL グループ	Web アプリケーションに定義された URL グループ単位の同時実行数制御	<ul style="list-style-type: none"> <li>同時実行スレッド数</li> <li>実行待ちリクエスト数</li> <li>実行待ちリクエスト数の上限からあふれたリクエスト数</li> <li>受け付けリクエスト数</li> <li>応答済みリクエスト数</li> </ul>

注 稼働情報監視の対象が SFO サーバの場合、JavaVM とプロセスリソースの稼働情報が取得できません。

## 2.3.2 稼働情報ファイルの出力

ここでは、稼働統計情報ファイルの出力先、ファイル面数、面の切り替え間隔、およびファイル名について説明します。

### (1) 稼働情報ファイルの出力先

デフォルトの設定の場合、稼働統計情報ファイルは、J2EE サーバの起動時に次のフォルダに作成されます。

Windows の場合

```
<作業ディレクトリ> %ejb%<J2EE サーバ名>%stats
```

UNIX の場合

```
<作業ディレクトリ> /ejb/<J2EE サーバ名>/stats
```

注 <作業ディレクトリ> は、J2EE サーバのユーザ定義（usrconf.cfg ファイル中の ejb.public.directory）で指定されたディレクトリを指します。デフォルト値は、Windows の場合、「<Cosminexus のインストールディレクトリ>%CC%server%public」、UNIX の場合、「/opt/Cosminexus/CC/server/public」です。

また、稼働情報ファイルは、出力先を変更することもできます。

稼働情報ファイルの出力先の設定については、マニュアル「Cosminexus システム構築ガイド」の稼働情報ファイル取得の設定に関する説明を参照してください。

### (2) 稼働情報ファイルのファイル面数と面の切り替え間隔

稼働情報ファイルは、出力先に保存するファイルの面数とファイルの面を切り替える時間間隔を設定できます。例えば、保存するファイル面数を「7」、ファイルの面を切り替える時間間隔を「24（時間）」と設定した場合、稼働情報ファイルは 1 日に 1 回面が切り替わり、最新の 1 週間分（7 日分）のファイルを保存できることとなります。

稼働情報ファイルの面数と面を切り替える時間間隔の設定については、マニュアル「Cosminexus システム構築ガイド」の稼働情報ファイル取得の設定に関する説明を参照してください。

### (3) 稼働情報ファイルのファイル名

稼働情報ファイルのファイル名は、稼働情報を収集する機能ごとに異なります。機能ごとの稼働情報ファイルのファイル名を次の表に示します。

表 2-4 稼働情報ファイルのファイル名

機能	ファイル名
JavaVM	HJVMStats_<YYYYMMDDhhmm><TZ>.csv
プロセスリソース	HOSStats_<YYYYMMDDhhmm><TZ>.csv
Stateful Session Bean	HStatefulSessionBeanStats_<YYYYMMDDhhmm><TZ>.csv
Stateless Session Bean	HStatelessSessionBeanStats_<YYYYMMDDhhmm><TZ>.csv
Message-driven Bean	HMessageDrivenBeanStats_<YYYYMMDDhhmm><TZ>.csv
DB Connector	HDBConnectorStats_<YYYYMMDDhhmm><TZ>.csv
JCA リソース	HJCAConnectionPoolStats_<YYYYMMDDhhmm><TZ>.csv
トランザクションサービス	HJTASStats_<YYYYMMDDhhmm><TZ>.csv
Web アプリケーション	HWebModuleStats_<YYYYMMDDhhmm><TZ>.csv
Web コンテナ	HWebContainerStats_<YYYYMMDDhhmm><TZ>.csv
URL グループ	HWebURLGroupStats_<YYYYMMDDhhmm><TZ>.csv

#### 注

- <YYYYMMDDhhmm> には、ファイルが出力された時刻を表す次の値が表示されます。  
YYYY：西暦年，MM：月，DD：日，hh：時，mm：分
- <TZ> には、タイムゾーンが、GMT（グリニッジ標準時）からの時差で表示されます。日本の場合、「+0900」が表示されます。

### 2.3.3 稼働情報ファイルの調査

稼働情報ファイルを調査するためには、ヘッダファイルと組み合わせて参照する必要があります。ヘッダファイルは、稼働情報ファイルの対象となる J2EE サーバや SFO サーバの機能と 1 対 1 の関係で出力されて、稼働情報ファイルに出力される情報に対応する項目名が出力されます。そのため、ヘッダファイルと稼働情報ファイルを対応させながら、稼働情報を調査する必要があります。

ヘッダファイルと稼働情報ファイルの対応の例を次に示します。



図 2-1 ヘッダファイルと稼働情報ファイルの対応

ヘッダファイル（稼働情報ファイルに出力される項目が表示される）

Date (+0900)	ObjectName	StatsPath	HeapSize	StartTime (+0900)	...
2006/1/14	com.cosmi	HJVMStats	2006/1/14	5.19E	
2006/1/14	com.cosmi	HJVMStats	2006/1/14	5.19E	
2006/1/14	com.cosmi	HJVMStats	2006/1/14	5.19E	
2006/1/14	com.cosmi	HJVMStats	2006/1/14	5.19E	
2006/1/14	com.cosmi	HJVMStats	2006/1/14	5.19E	

ヘッダファイルと稼働情報ファイルが対応している

稼働情報ファイル（ヘッダファイルに表示された項目の値が表示される）

稼働情報ファイルを調査する場合、まずファイル名に表示される日付などを参考にして、調査したい稼働情報ファイルを選択してください。次に、選択した稼働情報ファイルと、それに対応するヘッダファイルを一つのファイルに編集して、Excel などのアプリケーションプログラムで参照してください。その場合、グラフや表などの形式に変換して参照することをお勧めします。また、時刻を表す項目を Excel などのアプリケーションプログラムで参照する場合、セルの表示設定を、時刻を表示する形式に設定することをお勧めします。例えば Excel を使用している場合は、セルの表示形式を「時刻」に設定してください。

ヘッダファイルは、J2EE サーバの起動時に次のフォルダに作成されます。

Windows の場合

<作業ディレクトリ> ¥ejb¥<J2EE サーバ名>¥stats

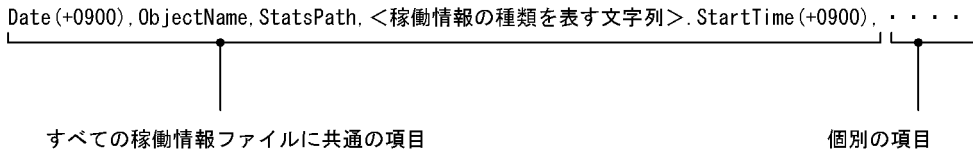
UNIX の場合

<作業ディレクトリ> /ejb/<J2EE サーバ名>/stats

注 <作業ディレクトリ> は、J2EE サーバのユーザ定義（usrconf.cfg ファイル中の ejb.public.directory）で指定されたディレクトリを指します。デフォルト値は、Windows の場合、「<Cosminexus のインストールディレクトリ>¥CC¥server¥public」、UNIX の場合、「/opt/Cosminexus/CC/server/public」です。

稼働情報ファイルとヘッダファイルは、すべての稼働情報ファイルに共通の項目と、個別の項目で構成されます。個別の項目は、対象となる機能ごとに異なります。ヘッダファイルの構成は、次のようになります。

## 2. システムの運用監視



すべての稼働情報ファイルに共通の項目の見方を次の表に示します。

表 2-5 すべての稼働情報ファイルに共通の項目の見方

項目 (ヘッダファイルに出力される文字列)	説明
Date	稼働情報ファイルを収集した時刻が、「YYYY/MM/DD hh:mm:ss.nnn」で表示されます。 YYYY：西暦年，MM：月，DD：日，hh：時，mm：分，ss：秒，nnn：ミリ秒
ObjectName	稼働情報ファイルごとに同じ形式の情報が表示されます。
StatsPath	稼働情報ファイルごとに固有の情報が出力されます。
<稼働情報の種類を表す文字列> .StartTime	稼働情報の種類ごとに，稼働情報の収集対象が稼働中になった時刻が表示されます。時刻は，1970年1月1日午前0時から経過した時間がミリ秒単位で表示されます。

注 ヘッダファイルにはこの情報に加えて，タイムゾーンが GMT (グリニッジ標準時) からの時差で表示されます。

例 (日本の場合): 「Date(+0900)」, 「HeapSize.StartTime(+0900)」

ただし，リロードによる J2EE アプリケーションの入れ替えを実行した場合，リロードを実行した時刻が表示されます。

なお，「ObjectName」には，稼働情報ファイルごとに次の形式で情報が出力されます。

表 2-6 ObjectName の形式

稼働情報ファイル (ファイル名)	ObjectName の形式
JavaVM (HJVMStats_<YYYYMMDDhhmm><TZ>.csv)	com.cosminexus.management.j2ee:J2EEServer=<J2EE サーバ名>,j2eeType=JVM,name=jvm
プロセスリソース (HOSStats_<YYYYMMDDhhmm><TZ>.csv)	com.cosminexus.management.j2ee:J2EEServer=<J2EE サーバ名>,j2eeType=OSResource,name=os
Stateful Session Bean (HStatefulSessionBeanStats_<YYYYMMD Dhhmm><TZ>.csv)	com.cosminexus.management.j2ee:EJBModule=<EJB アプリケーション表示名>,J2EEApplication=<J2EE アプリケーション表示名>,J2EEServer=<J2EE サーバ名>,j2eeType=StatefulSessionBean,mode=<アプリケーションの動作モード> ^1,name=<Stateful Session Bean 名>

稼働情報ファイル (ファイル名)	ObjectName の形式
Stateless Session Bean ( HStatelessSessionBeanStats_<YYYYMMDDhhmm><TZ>.csv )	com.cosminexus.management.j2ee:EJBModule=<EJB アプリケーション表示名 >,J2EEApplication=<J2EE アプリケーション表示名 >,J2EEServer=<J2EE サーバ名 >,j2eeType=StatelessSessionBean,mode=< アプリケーションの動作モード > <sup>1</sup> ,name=<Stateless Session Bean 名 >
Message-driven Bean ( HMessageDrivenBeanStats_<YYYYMMDhhmm><TZ>.csv )	com.cosminexus.management.j2ee:EJBModule=<EJB アプリケーション表示名 >,J2EEApplication=<J2EE アプリケーション表示名 >,J2EEServer=<J2EE サーバ名 >,j2eeType=MessageDrivenBean,mode=< アプリケーションの動作モード > <sup>1</sup> ,name=<Message-driven Bean 名 >
DB Connector ( HDBConnectorStats_<YYYYMMDDhhmm><TZ>.csv )	com.cosminexus.management.j2ee:J2EEApplication =<J2EE アプリケーション表示名 > <sup>2</sup> ,J2EEServer=<J2EE サーバ名 >,ResourceAdapterModule=< リソースアダプタ表示 名 >,j2eeType=ResourceAdapter,mode=< アプリケーションの動作モード > <sup>1</sup> ,raType=< リソースアダプタの種別 > <sup>3</sup> ,name=< リソースアダプタ表示名 >
JCA リソース ( HJCAConnectionPoolStats_<YYYYMMDDhhmm><TZ>.csv )	com.cosminexus.management.j2ee:J2EEServer=<J2EE サーバ名 >,ResourceAdapter=< リソースアダプタ表示名 >,j2eeType=JCAResource,mode=< アプリケーションの動作モード > <sup>1</sup> ,raType=< リソースアダプタの種別 > <sup>3</sup> ,app=<J2EE アプリケーションの表示名 > <sup>4</sup> ,name=< リソースアダプタ表示名 >
トランザクションサービス ( HJTASStats_<YYYYMMDDhhmm><TZ>.csv )	com.cosminexus.management.j2ee:J2EEServer=<J2EE サーバ名 >,j2eeType=JTAResource,name=JTAResource
Web アプリケーション ( HWebModuleStats_<YYYYMMDDhhmm><TZ>.csv )	com.cosminexus.management.j2ee:J2EEApplication =<J2EE アプリケーション表示名 >,J2EEServer=<J2EE サーバ名 >,j2eeType=WebModule,mode=< アプリケーションの動作モード > <sup>1</sup> ,name=<Web アプリケーション表示名 >
Web コンテナ ( HWebContainerStats_<YYYYMMDDhhmm><TZ>.csv )	com.cosminexus.management.j2ee:J2EEServer=<J2EE サーバ名 >,j2eeType=WebContainer,name=WebContainer

## 2. システムの運用監視

稼働情報ファイル (ファイル名)	ObjectName の形式
URL グループ ( HWebURLGroupStats_<YYYYMMDDhhmm><TZ>.csv )	com.cosminexus.management.j2ee:J2EEApplication=<J2EE アプリケーション表示名>,J2EESEServer=<J2EE サーバ名>,WebModule=<Web アプリケーション名>,j2eeType=WebURLGroup,mode=< アプリケーションの動作モード > <sup>1</sup> ,name=<URL グループ単位の同時実行スレッド数制御の定義名 >

注 1 <アプリケーションの動作モード>には、次の値が出力されます

- test : アプリケーションの動作モードがテストモードの場合
- normal : アプリケーションの動作モードが通常モードの場合

注 2 <J2EE アプリケーション表示名>は、リソースアダプタが J2EE アプリケーションに含まれてデプロイされている場合にだけ出力されます。直接 J2EE サーバにデプロイされたリソースアダプタの場合は、「null」が出力されます。

注 3 「raType=<リソースアダプタの種類>」は、ルートリソースアダプタ、またはメンバリソースアダプタを使用している場合にだけ、次の形式で出力されます。

- RootResourceAdapter : ルートリソースアダプタの場合
- MemberResourceAdapter : メンバリソースアダプタの場合

注 4 「app=<J2EE アプリケーションの表示名>」は、リソースアダプタが J2EE アプリケーションに含まれてデプロイされている場合にだけ出力されます。

次に、それぞれの稼働情報ファイル個別の項目の見方について説明します。個別の項目は、ヘッダファイルに次の形式で表示されます。

<稼働情報の種類を表す文字列> . <項目名>

各項目は稼働情報の種類を表す文字列のあとに、「.」で区切られて出力されます。稼働情報の種類を表す文字列、および項目名について次の表に示します。

表 2-7 稼働情報の種類を表す文字列

稼働情報の種類	稼働情報の種類を表す文字列
JavaVM ヒープサイズ	HeapSize
コピーガーベージコレクション回数	CopyGCCCount
フルガーベージコレクション回数	FullGCCCount
ロードされているクラス数	LoadedClassCount
モントロックのためにブロック状態であるスレッド数	ThreadBlockedCount
スレッド数	ThreadCount
ファイルディスクリプタ数	FileDescriptorCount
接続セッション数	ActiveSessionCount
プールされたインスタンス数	PooledInstanceCount

稼働情報の種類	稼働情報の種類を表す文字列
プール内の使用中インスタンス数	ActivePooledInstanceCount
受け付けメッセージ数	MessageCount
実行待ちリクエスト数	WaitingRequestCount
プールされた PreparedStatement 数	PooledPreparedStatementCount
プールされた CallableStatement 数	PooledCallableStatementCount
PrepareStatement メソッドが呼び出された回数	InvokedPrepareStatementMethodCount
prepareCall メソッドが呼び出された回数	InvokedPrepareCallMethodCount
プール内 PreparedStatement ヒット回数	PooledPreparedStatementHitCount
プール内 CallableStatement ヒット回数	PooledCallableStatementHitCount
プールされたコネクション数	PoolSize
プール内の使用中コネクション数	ActivePoolSize
コネクション取得待ちスレッド数	WaitingThreadCount
コネクション取得失敗数	FailedRequestCount
決着済みトランザクション数	CompletionCount
トランザクションロールバック数	RolledbackCount
応答済みリクエスト数	ResponseCount
同時実行スレッド数	ActiveThreadCount
実行待ちリクエスト数の上限からあふれたリクエスト数	OverflowRequestCount
受け付けリクエスト数	RequestCount
セッション数	SessionCount

表 2-8 稼働統計情報ファイルの項目名

項目名 (ヘッダファイルに出力される文字列)	説明
Count	前回の稼働情報ファイルの出力時からの累積回数, または累積個数
HighWaterMark	前回の稼働情報ファイルの出力時からの最大値
LowWaterMark	前回の稼働情報ファイルの出力時からの最小値
Current	現在値
UpperBound	上限値
LowerBound	下限値

次に, 個別に出力される項目について, 稼働情報ファイルごとに説明します。

### (1) JavaVM の稼働情報ファイルに出力される情報

JavaVM の稼働情報ファイルに出力される情報について説明します。JavaVM の稼働情

## 2. システムの運用監視

報ファイルでは、J2EE サーバが使用する JavaVM の稼働情報を調査することができます。

稼働情報ファイル名

HJVMStats\_<YYYYMMDDhhmm><TZ>.csv

対応するヘッダファイル名

HJVMStats.txt

出力される稼働情報

JavaVM の稼働情報ファイルに出力される内容を次に示します。

表 2-9 JavaVM の稼働情報ファイルの出力内容

稼働情報の種類	項目（ヘッダファイルに出力される文字列）
JavaVM ヒープサイズ	HeapSize.UpperBound
	HeapSize.LowerBound
	HeapSize.HighWaterMark
	HeapSize.LowWaterMark
	HeapSize.Current
コピーガーベージコレクション回数	CopyGCCCount.Count
フルガーベージコレクション回数	FullGCCCount.Count
ロードされているクラス数	LoadedClassCount.HighWaterMark
	LoadedClassCount.LowWaterMark
	LoadedClassCount.Current
稼働中のスレッド数	ThreadCount.HighWaterMark
	ThreadCount.LowWaterMark
	ThreadCount.Current
モナタロックのためにブロック状態であるスレッド数	ThreadBlockedCount.HighWaterMark
	ThreadBlockedCount.LowWaterMark
	ThreadBlockedCount.Current

### （2）プロセスリソースの稼働情報ファイルに出力される情報

プロセスリソースの稼働情報ファイルに出力される情報について説明します。プロセスリソースの稼働情報ファイルでは、J2EE サーバプロセスが使用する OS リソースの稼働情報を調査することができます。

稼働情報ファイル名

HOSStats\_<YYYYMMDDhhmm><TZ>.csv

対応するヘッダファイル名

HOSStats.txt

出力される稼働情報

プロセスリソースの稼働情報ファイルに出力される内容を次に示します。

表 2-10 プロセスリソースの稼働情報ファイルの出力内容

稼働情報の種類	項目（ヘッダファイルに出力される文字列）
J2EE サーバプロセスが生成したスレッド数	ThreadCount.UpperBound <sup>1</sup>
	ThreadCount.LowerBound <sup>2</sup>
	ThreadCount.HighWaterMark <sup>2</sup>
	ThreadCount.LowWaterMark <sup>2</sup>
	ThreadCount.Current <sup>2</sup>
J2EE サーバプロセスが使用するファイルディスクリプタ数	FileDescriptorCount.UpperBound <sup>3</sup>
	FileDescriptorCount.LowerBound <sup>3</sup>
	FileDescriptorCount.HighWaterMark <sup>4</sup>
	FileDescriptorCount.LowWaterMark <sup>4</sup>
	FileDescriptorCount.Current <sup>4</sup>

注 1 Windows, Linux, または Solaris の場合は無効になり, 「-1」が出力されます。

注 2 Linux の場合は無効になり, 「-1」が出力されます。

注 3 Windows の場合は無効になり, 「-1」が出力されます。

注 4 Windows, または AIX の場合は無効になり, 「-1」が出力されます。

### (3) Stateful Session Bean の稼働情報ファイルに出力される情報

Stateful Session Bean の稼働情報ファイルに出力される情報について説明します。

Stateful Session Bean の稼働情報ファイルでは, Stateful Session Bean の稼働情報を調査することができます。

稼働情報ファイル名

HStatefulSessionBeanStats\_<YYYYMMDDhhmm><TZ>.csv

対応するヘッダファイル名

HStatefulSessionBeanStats.txt

出力される稼働情報

Stateful Session Bean の稼働情報ファイルに出力される内容を次に示します。

## 2. システムの運用監視

表 2-11 Stateful Session Bean の稼働情報ファイルの出力内容

稼働情報の種類	項目（ヘッダファイルに出力される文字列）	備考
接続中のセッション数	ActiveSessionCount.UpperBound	Stateful Session Bean のアクティブ・セッションの最大数が無制限のとき、「0」が出力されます。
	ActiveSessionCount.LowerBound	下限値を設定できないため、無効になり「-1」が出力されます。
	ActiveSessionCount.HighWaterMark	-
	ActiveSessionCount.LowWaterMark	-
	ActiveSessionCount.Current	-

（凡例） - : 該当なし

### （4）Stateless Session Bean の稼働情報ファイルに出力される情報

Stateless Session Bean の稼働情報ファイルに出力される情報について説明します。  
Stateless Session Bean の稼働情報ファイルでは、Stateless Session Bean の稼働情報を調査することができます。

稼働情報ファイル名

HStatelessSessionBeanStats\_<YYYYMMDDhhmm><TZ>.csv

対応するヘッダファイル名

HStatelessSessionBeanStats.txt

出力される稼働情報

Stateless Session Bean の稼働情報ファイルに出力される内容を次に示します。

表 2-12 Stateless Session Bean の稼働情報ファイルの出力内容

稼働情報の種類	項目（ヘッダファイルに出力される文字列）	備考
プールされたインスタンス数	PooledInstanceCount.UpperBound	Stateless Session Bean のプール内インスタンスの最大数が無制限のとき、「0」が出力されます。
	PooledInstanceCount.LowerBound	-
	PooledInstanceCount.HighWaterMark	-
	PooledInstanceCount.LowWaterMark	-
	PooledInstanceCount.Current	-
プール内の使用中インスタンス数	ActivePooledInstanceCount.UpperBound	Stateless Session Bean のプール内インスタンスの最大数が無制限のとき、「0」が出力されます。



稼働情報の種類	項目（ヘッダファイルに出力される文字列）	備考
	ActivePooledInstanceCount.LowerBound	下限値を設定できないため、無効になり「-1」が出力されます。
	ActivePooledInstanceCount.HighWaterMark	-
	ActivePooledInstanceCount.LowWaterMark	-
	ActivePooledInstanceCount.Current	-
実行待ちリクエスト数	WaitingRequestCount.HighWaterMark	-
	WaitingRequestCount.LowWaterMark	-
	WaitingRequestCount.Current	-

（凡例） - : 該当なし

### （5）Message-driven Bean の稼働情報ファイルに出力される情報

Message-driven Bean の稼働情報ファイルに出力される情報について説明します。

Message-driven Bean の稼働情報ファイルでは、Message-driven Bean の稼働情報を調査することができます。

稼働情報ファイル名

HMessageDrivenBeanStats\_<YYYYMMDDhhmm><TZ>.csv

対応するヘッダファイル名

HMessageDrivenBeanStats.txt

出力される稼働情報

Message-driven Bean の稼働情報ファイルに出力される内容を次に示します。

表 2-13 Message-driven Bean の稼働情報ファイルの出力内容

稼働情報の種類	項目（ヘッダファイルに出力される文字列）	備考
プールされたインスタンス数	PooledInstanceCount.UpperBound	Message-driven Bean のプール内インスタンスの最大数が無制限のとき、「0」が出力されます。
	PooledInstanceCount.LowerBound	-
	PooledInstanceCount.HighWaterMark	-
	PooledInstanceCount.LowWaterMark	-
	PooledInstanceCount.Current	-
プール内の使用中インスタンス数	ActivePooledInstanceCount.UpperBound	Message-driven Bean のプール内インスタンスの最大数が無制限のとき、「0」が出力されます。
	ActivePooledInstanceCount.LowerBound	下限値を設定できないため、無効になり「-1」が出力されます。

## 2. システムの運用監視

稼働情報の種類	項目（ヘッダファイルに出力される文字列）	備考
	ActivePooledInstanceCount.HighWaterMark	-
	ActivePooledInstanceCount.LowWaterMark	-
	ActivePooledInstanceCount.Current	-
受け付けメッセージ数	MessageCount.Count	-

（凡例） - : 該当なし

### （6）DB Connector の稼働情報ファイルに出力される情報

DB Connector の稼働情報ファイルに出力される情報について説明します。DB Connector の稼働情報ファイルでは、DB Connector の稼働情報を調査することができます。

ただし、コネクションプーリング機能を使用していない場合、prepareStatement メソッドが呼び出された回数（InvokedPrepareStatementMethodCount.Count）、および PrepareCall メソッドが呼び出された回数（InvokedPrepareCallMethodCount.Count）以外の値は無効になり、「-1」が出力されます。

稼働情報ファイル名

HDBConnectorStats\_<YYYYMMDDhhmm><TZ>.csv

対応するヘッダファイル名

HDBConnectorStats.txt

出力される稼働情報

DB Connector の稼働情報ファイルに出力される内容を次に示します。

表 2-14 DB Connector の稼働情報ファイルの出力内容

稼働情報の種類	項目（ヘッダファイルに出力される文字列）	備考
ステートメントプーリング機能によってキャッシュされた PreparedStatement 数	PooledPreparedStatementCount.UpperBound <sup>1</sup>	次の値が出力されます。 「出力値 = DB Connector に設定したコネクションごとの PreparedStatement サイズ × コネクションプールの上限値」 コネクションプールの上限値を無制限にしたときは、無効になり「-1」が出力されます。
	PooledPreparedStatementCount.LowerBound	下限値を設定できないため、無効になり「-1」が出力されます。

稼働情報の種類	項目（ヘッダファイルに出力される文字列）	備考
	PooledPreparedStatementCount.HighWaterMark <sup>1 2</sup>	-
	PooledPreparedStatementCount.LowWaterMark <sup>1 2</sup>	-
	PooledPreparedStatementCount.Current <sup>1 2</sup>	-
ステートメントプーリング機能によってキャッシュされたCallableStatement数	PooledCallableStatementCount.UpperBound <sup>3</sup>	次の値が出力されます。 「出力値 = DB Connector に設定したコネクションごとのPooledCallableStatement サイズ×コネクションプールの上限值」 コネクションプールの上限値を無制限にしたときは、無効になり「-1」が出力されます。
	PooledCallableStatementCount.LowerBound	下限値を設定できないため、無効になり「-1」が出力されます。
	PooledCallableStatementCount.HighWaterMark <sup>2 3</sup>	-
	PooledCallableStatementCount.LowWaterMark <sup>2 3</sup>	-
	PooledCallableStatementCount.Current <sup>2 3</sup>	-
PrepareStatement メソッドが呼び出された回数	InvokedPrepareStatementMethodCount.Count <sup>2</sup>	-
PrepareCall メソッドが呼び出された回数	InvokedPrepareCallMethodCount.Count <sup>2</sup>	-
ステートメントプーリング機能によってキャッシュされたPreparedStatement がヒットした回数	PooledPreparedStatementHitCount.Count <sup>1 2</sup>	-
ステートメントプーリング機能によってキャッシュされたCallableStatement がヒットした回数	PooledCallableStatementHitCount.Count <sup>2 3</sup>	-

（凡例） - : 該当なし

注 1 DB Connector の設定であるコネクションごとの PreparedStatement のプールサイズに「0」を設定した場合、PreparedStatement がプールされなくなるため、無効になり「-1」が出力されます。

注 2 接続先データベースに Oracle を使用している場合、または接続先データベースに HiRDB 以外を使用してコネクションの障害検知を行った場合、ステートメントオブジェクトを作成していても、1 以上の値が出力されます。

注 3 DB Connector の設定であるコネクションごとの CallableStatement のプールサイズに 0

## 2. システムの運用監視

を設定した場合、CallableStatement がプールされなくなるため、無効になり「-1」が出力されま  
す。

### (7) JCA リソースの稼働情報ファイルに出力される情報

JCA リソースの稼働情報ファイルに出力される情報について説明します。JCA リソース  
の稼働情報ファイルでは、JCA リソースの稼働情報を調査することができます。ただし、  
コネクションプールが無効の場合、またはルートリソースアダプタの場合、コネクショ  
ン取得失敗数だけが出力されます。メンバリソースアダプタの場合、単体のリソースア  
ダプタと同じ稼働情報が出力されます。DB Connector for Cosminexus RM を使用して  
いる場合、稼働情報は Cosminexus RM 側に出力されます。DB Connector for  
Cosminexus RM 側には出力されません。

Connector 1.5 仕様に準拠したリソースアダプタを使用した場合は、一つ目のコネクショ  
ン定義 (ra.xml 内でいちばん先頭の定義) に対応する稼働情報が出力されます。

Outbound のコネクション定義がない場合は、HJCAConnectionPoolStats の稼働情報は  
出力されません。

稼働情報ファイル名

HJCAConnectionPoolStats\_<YYYYMMDDhhmm><TZ>.csv

対応するヘッダファイル名

HJCAConnectionPoolStats.txt

出力される稼働情報

JCA リソースの稼働情報ファイルに出力される内容を次に示します。

表 2-15 JCA リソースの稼働情報ファイルの出力内容

稼働情報の種 類	項目 (ヘッダファイルに出力される文字列)	備考
プールされた コネクション 数	PoolSize.UpperBound	-
	PoolSize.LowerBound	-
	PoolSize.HighWaterMark	-
	PoolSize.LowWaterMark	-
	PoolSize.Current	-
プール内の使 用中コネク ション数	ActivePoolSize.UpperBound	-
	ActivePoolSize.LowerBound	下限値を設定できない ため、無効になり 「-1」が出力されます。
	ActivePoolSize.HighWaterMark	-
	ActivePoolSize.LowWaterMark	-

稼働情報の種類	項目（ヘッダファイルに出力される文字列）	備考
	ActivePoolSize.Current	-
コネクション取得待ちスレッド数	WaitingThreadCount.UpperBound	上限値を設定できないため、無効になり「-1」が出力されます。
	WaitingThreadCount.LowerBound	下限値を設定できないため、無効になり「-1」が出力されます。
	WaitingThreadCount.HighWaterMark	-
	WaitingThreadCount.LowWaterMark	-
	WaitingThreadCount.Current	-
コネクション取得失敗数	FailedRequestCount.Count	DB Connector for Cosminexus RM を使用している場合、Cosminexus RM 側に出力されます。DB Connector for Cosminexus RM 側には常に「0」が出力されます。

（凡例） - : 該当なし

### （8）トランザクションサービスの稼働情報ファイルに出力される情報

トランザクションサービスの稼働情報ファイルに出力される情報について説明します。トランザクションサービスの稼働情報ファイルでは、トランザクションサービスの稼働情報を調査することができます。

稼働情報ファイル名

HJTASStats\_<YYYYMMDDhhmm><TZ>.csv

対応するヘッダファイル名

HJTASStats.txt

出力される稼働情報

トランザクションサービスの稼働情報ファイルに出力される内容を次に示します。

表 2-16 トランザクションサービスの稼働情報ファイルの出力内容

稼働情報の種類	項目（ヘッダファイルに出力される文字列）
決着したトランザクション数	CompletionCount.Count
ロールバックしたトランザクション数	RolledbackCount.Count

(9) Web アプリケーションの稼働情報ファイルに出力される情報

Web アプリケーションの稼働情報ファイルに出力される情報について説明します。Web アプリケーションの稼働情報ファイルでは、対象の Web アプリケーション全体の稼働情報を調査することができます。

稼働情報ファイル名

HWebModuleStats\_<YYYYMMDDhhmm><TZ>.csv

対応するヘッダファイル名

HWebModuleStats.txt

出力される稼働情報

Web アプリケーションの稼働情報ファイルに出力される内容を次に示します。

表 2-17 Web アプリケーションの稼働情報ファイルの出力内容

稼働情報の種類	項目（ヘッダファイルに出力される文字列）	備考
最大同時実行スレッド数	ActiveThreadCount.UpperBound <sup>1 2</sup>	-
	ActiveThreadCount.LowerBound <sup>3</sup>	-
	ActiveThreadCount.HighWaterMark <sup>1</sup>	-
	ActiveThreadCount.LowWaterMark <sup>1</sup>	-
	ActiveThreadCount.Current <sup>1</sup>	-
実行待ちリクエスト数	WaitingRequestCount.UpperBound <sup>1 2 4</sup>	-
	WaitingRequestCount.LowerBound <sup>3 4</sup>	-
	WaitingRequestCount.HighWaterMark <sup>1 4</sup>	-
	WaitingRequestCount.LowWaterMark <sup>1 4</sup>	-
	WaitingRequestCount.Current <sup>1 4</sup>	-
実行待ちリクエスト数の上限からあふれたリクエスト数	OverflowRequestCount.Count <sup>1</sup>	-

稼働情報の種類	項目（ヘッダファイルに出力される文字列）	備考
受け付けリクエスト数	RequestCount.Count <sup>4</sup>	Web コンテナが受け付けた Web アプリケーション単位のリクエスト数が出力されます。実行待ちリクエスト数もカウントされます。Web サーバでは受信されていても、Web コンテナに転送されていないリクエスト数はカウントされません。
応答済みリクエスト数	ResponseCount.Count	-
セッション数	SessionCount.UpperBound	セッション数の上限を設定していない場合、無効になり「-1」が出力されます。
	SessionCount.LowerBound <sup>3</sup>	-
	SessionCount.HighWaterMark	-
	SessionCount.LowWaterMark	-
	SessionCount.Current	-

（凡例） - : 該当なし

注 1 Web アプリケーション単位の同時実行数制御機能を使用していない場合、または対象となる Web アプリケーションに、同時実行数制御機能を使用していない場合、無効になり、「-1」が出力されます。

注 2 対象となる Web アプリケーションに最大同時実行スレッド数制御が設定されている場合、Web アプリケーション単位の最大同時実行スレッド数が出力されます。最大同時実行スレッド数を動的に変更した場合、変更後の値が出力されます。

注 3 無効になり「-1」が出力されます。

注 4 リロード機能による入れ替え処理中の場合、Web アプリケーションが受け付けた実行待ちリクエスト数はカウントされません。リロード処理の完了後にカウントされます。

## （10）Web コンテナの稼働情報ファイルに出力される情報

Web コンテナの稼働情報ファイルに出力される情報について説明します。Web コンテナの稼働情報ファイルでは、開始している Web アプリケーションの稼働情報を調査することができます。

稼働情報ファイル名

HWebContainerStats\_<YYYYMMDDhhmm><TZ>.csv

対応するヘッダファイル名

HWebContainerStats.txt

出力される稼働情報

Web コンテナの稼働情報ファイルに出力される内容を次に示します。

## 2. システムの運用監視

表 2-18 Web コンテナの稼働情報ファイルの出力内容

稼働情報の種類	項目（ヘッダファイルに出力される文字列）
最大同時実行スレッド数	ActiveThreadCount.UpperBound <sup>1</sup>
	ActiveThreadCount.LowerBound <sup>1 2</sup>
	ActiveThreadCount.HighWaterMark <sup>1</sup>
	ActiveThreadCount.LowWaterMark <sup>1</sup>
	ActiveThreadCount.Current <sup>1</sup>
実行待ちリクエスト数	WaitingRequestCount.UpperBound <sup>3</sup>
	WaitingRequestCount.LowerBound <sup>2 3</sup>
	WaitingRequestCount.HighWaterMark <sup>4</sup>
	WaitingRequestCount.LowWaterMark <sup>4</sup>
	WaitingRequestCount.Current <sup>4</sup>
実行待ちリクエスト数の上限からあふれたリクエスト数	OverflowRequestCount.Count <sup>3</sup>

注 1 Web アプリケーション単位の最大同時実行スレッド数制御を無効に設定している場合、J2EE アプリケーションのリロード中に実行待ちになった新しいリクエストは、稼働中のスレッドとしてカウントされます。

注 2 無効になり「-1」が出力されます。

注 3 Web アプリケーション単位の最大同時実行スレッド数制御を設定していない場合、無効になり「-1」が出力されます。Web アプリケーション単位の最大同時実行スレッド数制御を設定している場合、デフォルトの実行待ちキューの情報が出力されます。

注 4 Web アプリケーション単位の最大同時実行スレッド数制御を設定していない場合、Web コンテナ単位の実行待ちキューの情報が出力されます。Web アプリケーション単位の最大同時実行スレッド数制御を設定している場合、デフォルトの実行待ちキューの情報が出力されます。

### (11) URL グループの稼働情報ファイルに出力される情報

URL グループの稼働情報ファイルに出力される情報について説明します。URL グループの稼働情報ファイルでは、URL グループの稼働情報を調査することができます。

稼働情報ファイル名

HWebURLGroupStats\_<YYYYMMDDhhmm><TZ>.csv

対応するヘッダファイル名

HWebURLGroupStats.txt

出力される稼働情報

URL グループの稼働情報ファイルに出力される内容を次に示します。



表 2-19 URL グループの稼働情報ファイルの出力内容

稼働情報の種類	項目 (ヘッダファイルに出力される文字列)
最大同時実行スレッド数	ActiveThreadCount.UpperBound <sup>1</sup>
	ActiveThreadCount.LowerBound <sup>2</sup>
	ActiveThreadCount.HighWaterMark
	ActiveThreadCount.LowWaterMark
	ActiveThreadCount.Current
実行待ちリクエスト数	WaitingRequestCount.UpperBound <sup>3</sup>
	WaitingRequestCount.LowerBound <sup>2 3</sup>
	WaitingRequestCount.HighWaterMark <sup>3</sup>
	WaitingRequestCount.LowWaterMark <sup>3</sup>
	WaitingRequestCount.Current <sup>3</sup>
実行待ちリクエスト数の上限からあふれたリクエスト数	OverflowRequestCount.Count
受け付けリクエスト数	RequestCount.Count <sup>4</sup>
応答済みリクエスト数	ResponseCount.Count

注 1 Web アプリケーション単位の最大同時実行スレッド数を動的に変更した場合、設定値ではなく変更後の情報が出力されます。

注 2 無効になり「-1」が出力されます。

注 3 リロード機能による入れ替え処理中の場合、Web アプリケーションが受け付けた実行待ちリクエスト数はカウントされません。リロード処理の完了後にカウントされます。

注 4 Web コンテナが受け付けた URL 単位のリクエスト数が出力されます。実行待ちリクエスト数もカウントされます。Web サーバでは受信されていても、Web コンテナに転送されていないリクエスト数はカウントされません。

## 2.4 稼働情報の監視（運用管理コマンド）

この節では、J2EE サーバ、SFO サーバ、またはバッチサーバの稼働情報の監視方法と、監視できる項目について説明します。

### ポイント

稼働情報の監視を行う場合、通常の運用では、稼働情報ファイルを使用した稼働監視を行ってください。稼働情報ファイルで取得できる情報より詳細な情報を取得したい場合だけ、運用管理コマンドを使用した稼働情報の監視を行ってください。稼働情報ファイルを使用した稼働情報の監視は、「2.3 稼働情報の監視（稼働情報ファイル）」を参照してください。

### 2.4.1 サーバの稼働情報の監視方法（J2EE サーバ、SFO サーバ、バッチサーバ）

J2EE サーバ、SFO サーバ、またはバッチサーバの稼働情報は、運用管理コマンド（mngsvrutil）で監視できます。

稼働情報を監視するには、運用管理コマンド（mngsvrutil）にサブコマンド「get」を指定して実行します。これによって、J2EE サーバまたは SFO サーバの稼働情報を標準出力に出力したり、CSV 形式または SNMP 連携用形式のファイルに出力したりできます。

「get」の引数に稼働情報を出力する対象を指定できます。

J2EE サーバの稼働情報を取得する場合の実行形式と実行例を次に示します。

#### 実行形式

```
mngsvrutil -m <Management Serverのホスト名> -u <管理ユーザID> -p <管理パスワード> -t <J2EEサーバ名> get <ドメイン名または取得するカテゴリ>
```

#### 実行例

```
mngsvrutil -m mnghost -u user01 -p pwl -t J2EEServer1 get j2eeContainer
```

mngsvrutil コマンド、そのサブコマンド、および取得できる情報の詳細については、マニュアル「Cosminexus リファレンス コマンド編」を参照してください。

### 2.4.2 稼働情報監視で確認できる項目

稼働情報監視では、J2EE サーバ、バッチサーバおよび SFO サーバの稼働状況が確認できます。

稼働情報として監視できる項目を次の表に示します。

表 2-20 稼働情報監視で表示できる項目 (J2EE サーバの場合)

監視の対象		監視できる項目
J2EE サーバ	J2EE コンテナ	<ul style="list-style-type: none"> <li>• J2EE サーバ名</li> <li>• Naming Service ホスト</li> <li>• Naming Service ポート番号</li> <li>• コンテナ起動時刻</li> </ul>
	EJB コンテナ	<ul style="list-style-type: none"> <li>• コンテナ名</li> </ul>
	Web コンテナ	<p>基本情報</p> <ul style="list-style-type: none"> <li>• コンテナ名</li> <li>• Web コンテナ起動時刻</li> <li>• 管理用サーバのポート番号</li> <li>• Web サーバとの通信に使用するポート番号</li> <li>• Web サーバとの通信ソケットのバックログ設定上限値</li> <li>• 稼働中 (リクエスト処理の実行中) スレッド数設定上限値</li> <li>• インプロセス HTTP サーバのポート番号</li> <li>• インプロセス HTTP サーバとの通信ソケットのバックログ設定上限値</li> <li>• インプロセス HTTP サーバのリクエスト処理用スレッド数の上限値</li> <li>• インプロセス HTTP サーバのリクエスト処理用スレッド数の下限値</li> <li>• インプロセス HTTP サーバの稼働スレッド数の上限値</li> <li>• インプロセス HTTP サーバのプーリングスレッド数</li> <li>• インプロセス HTTP サーバのコネクション数の上限値</li> <li>• インプロセス HTTP サーバの Persistent コネクション数の上限値</li> <li>• インプロセス HTTP サーバの Persistent コネクション切断時のリクエスト処理総数の上限値</li> <li>• インプロセス HTTP サーバの Persistent コネクションリクエスト待ち時間の上限値</li> <li>• 最大同時実行スレッド数</li> <li>• 実行待ちキューサイズ</li> </ul>

## 2. システムの運用監視

監視の対象	監視できる項目
	<p>稼働情報</p> <ul style="list-style-type: none"> <li>• Web サーバと Web コンテナとのコネクション数</li> <li>• 稼働中スレッド数 (HTTP サーバコネクタ)</li> <li>• 実行待ちしているリクエスト数</li> <li>• インプロセス HTTP サーバのリクエスト処理用スレッド数</li> <li>• インプロセス HTTP サーバの稼働中スレッド数</li> <li>• インプロセス HTTP サーバのプーリングスレッド数</li> <li>• インプロセス HTTP サーバのコネクション数</li> <li>• インプロセス HTTP サーバの Persistent コネクション数</li> <li>• インプロセス HTTP サーバの Persistent コネクション切断時のリクエスト処理総数</li> <li>• インプロセス HTTP サーバの Persistent コネクションリクエスト待ち時間</li> <li>• インプロセス HTTP サーバの実行待ちしているリクエスト数</li> <li>• インプロセス HTTP サーバのアクセスを拒否されたリクエスト数</li> <li>• 同時実行可能スレッド数上限値</li> <li>• 稼働スレッド数</li> <li>• 実行待ちリクエスト数</li> <li>• 実行待ちキューからあふれたリクエスト数</li> <li>• 稼働中スレッド数 (HTTP サーバ)</li> </ul>
JavaVM	<p>基本情報</p> <ul style="list-style-type: none"> <li>• JavaVM 名</li> <li>• JavaVM バージョン</li> <li>• JavaVM の最大メモリ使用量</li> </ul> <p>稼働情報</p> <ul style="list-style-type: none"> <li>• JavaVM の空きメモリ量</li> <li>• JavaVM の総メモリ容量</li> </ul>
アプリケーション	<p>J2EE アプリケーション</p> <ul style="list-style-type: none"> <li>• アプリケーション名 ( display name )</li> <li>• 説明</li> <li>• デプロイされた時刻</li> </ul>
	<p>EJB アプリケーション</p> <ul style="list-style-type: none"> <li>• EJB-JAR 名</li> <li>• 説明</li> </ul>

監視の対象	監視できる項目
Enterprise Bean ( Stateful Session Bean )	<p><b>基本情報</b></p> <ul style="list-style-type: none"> <li>• Enterprise Bean 名</li> <li>• 説明</li> <li>• Bean 名 ( 内部識別用 )</li> <li>• Home インタフェース名</li> <li>• Local Home インタフェース名</li> <li>• Component インタフェース名</li> <li>• Local Component インタフェース名</li> <li>• EJB クラス名</li> <li>• トランザクションタイプ</li> <li>• 同時接続最大値</li> <li>• 同時実行最大値</li> <li>• 非活性セッションのタイムアウト</li> <li>• 実行中セッションのタイムアウト</li> </ul> <p><b>稼働情報</b></p> <ul style="list-style-type: none"> <li>• 現在の接続セッション数</li> <li>• 実行中のセッション数</li> <li>• 非活性セッション数</li> <li>• 接続待ちセッション数</li> </ul> <p><b>Home インタフェース</b></p> <ul style="list-style-type: none"> <li>• Home インタフェース名</li> <li>• レスポンス</li> <li>• EJB メソッド実行時間</li> </ul> <p><b>Local Home インタフェース</b></p> <ul style="list-style-type: none"> <li>• Home インタフェース名</li> <li>• レスポンス</li> <li>• EJB メソッド実行時間</li> </ul> <p><b>Component インタフェース</b></p> <ul style="list-style-type: none"> <li>• Component インタフェース名</li> <li>• レスポンス</li> <li>• EJB メソッド実行時間</li> </ul> <p><b>Local Component インタフェース</b></p> <ul style="list-style-type: none"> <li>• Component インタフェース名</li> <li>• レスポンス</li> <li>• EJB メソッド実行時間</li> </ul>
Enterprise Bean ( Stateless Session Bean )	<p><b>基本情報</b></p> <ul style="list-style-type: none"> <li>• Enterprise Bean 名</li> <li>• 説明</li> <li>• Bean 名 ( 内部識別用 )</li> <li>• Home インタフェース名</li> <li>• Local Home インタフェース名</li> <li>• Component インタフェース名</li> <li>• Local Component インタフェース名</li> <li>• EJB クラス名</li> <li>• トランザクションタイプ</li> <li>• Bean インスタンスプール</li> </ul>

## 2. システムの運用監視

監視の対象	監視できる項目
	<p>稼働情報</p> <ul style="list-style-type: none"> <li>• 接続待ちセッション数</li> <li>• Bean インスタンスプール現在値</li> <li>• 使用中の Session Bean 数</li> <li>• 未使用の Session Bean 数</li> </ul> <p>Home インタフェース Stateful Session Bean と同じ</p> <p>Local Home インタフェース Stateful Session Bean と同じ</p> <p>Component インタフェース Stateful Session Bean と同じ</p> <p>Local Component インタフェース Stateful Session Bean と同じ</p>
Enterprise Bean ( Entity Bean )	<p>基本情報</p> <ul style="list-style-type: none"> <li>• Enterprise Bean 名</li> <li>• 説明</li> <li>• Bean 名 ( 内部識別用 )</li> <li>• Home インタフェース名</li> <li>• Local Home インタフェース名</li> <li>• Component インタフェース名</li> <li>• Local Component インタフェース名</li> <li>• EJB クラス名</li> <li>• EntityBean の永続化タイプ</li> <li>• EntityBean のキャッシュモデル</li> <li>• 同時接続最大値</li> <li>• Bean インスタンスプール</li> <li>• 接続タイムアウト</li> </ul> <p>稼働情報</p> <ul style="list-style-type: none"> <li>• 現在の接続セッション数</li> <li>• 接続待ちセッション数</li> <li>• Bean インスタンスプール現在値</li> <li>• 使用中の Entity Bean 数</li> <li>• 未使用の Entity Bean 数</li> </ul> <p>Home インタフェース Stateful Session Bean と同じ</p> <p>Local Home インタフェース Stateful Session Bean と同じ</p> <p>Component インタフェース Stateful Session Bean と同じ</p> <p>Local Component インタフェース Stateful Session Bean と同じ</p>

監視の対象	監視できる項目
Enterprise Bean ( Message-driven Bean )	<p>基本情報</p> <ul style="list-style-type: none"> <li>Enterprise Bean 名</li> <li>説明</li> <li>Bean 名 ( 内部識別用 )</li> <li>EJB クラス名</li> <li>トランザクションタイプ</li> <li>デスティネーションタイプ <sup>1</sup></li> <li>Bean インスタンスプール</li> </ul> <p>稼働情報</p> <ul style="list-style-type: none"> <li>現在の接続セッション数</li> <li>Bean インスタンスプール現在値</li> <li>EJB メソッド実行時間</li> </ul>
Web アプリケーション	<p>基本情報</p> <ul style="list-style-type: none"> <li>コンテキストルート</li> <li>占有スレッド数</li> <li>最大同時実行スレッド数</li> <li>実行待ちキューサイズ</li> </ul> <p>稼働情報</p> <ul style="list-style-type: none"> <li>有効なセッション数</li> <li>同時実行可能スレッド数上限値</li> <li>稼働スレッド数</li> <li>実行待ちリクエスト数</li> <li>実行待ちキューからあふれたリクエスト数</li> </ul>
サーブレット	<ul style="list-style-type: none"> <li>サーブレット名</li> <li>サーブレットの実装クラス名</li> <li>サーブレット実行回数</li> <li>サーブレット失敗回数</li> <li>サーブレット実行時間</li> <li>出力データサイズ</li> </ul>
URL	<ul style="list-style-type: none"> <li>URL</li> <li>URL 呼び出し回数</li> <li>URL 呼び出し失敗回数</li> <li>URL 実行時間</li> <li>出力データサイズ</li> </ul>
リソース	<p>データソース ( SimpleJTA )</p> <p>基本情報</p> <ul style="list-style-type: none"> <li>リソース名</li> <li>リソースタイプ</li> <li>説明</li> <li>認証タイプ</li> <li>ログインタイムアウト</li> <li>ユーザ ID</li> <li>コネクションプール</li> </ul> <p>稼働情報</p> <ul style="list-style-type: none"> <li>リソース名</li> <li>プール現在値 ( 総数 )</li> <li>使用中のコネクション数</li> <li>未使用のコネクション数</li> </ul>

## 2. システムの運用監視

監視の対象	監視できる項目
データソース ( FullJTA )	<p>基本情報</p> <ul style="list-style-type: none"> <li>• リソース名</li> <li>• リソースタイプ</li> <li>• 説明</li> <li>• ログインタイムアウト</li> <li>• ユーザ ID</li> <li>• コネクションプール</li> </ul> <p>稼働情報</p> <ul style="list-style-type: none"> <li>• リソース名</li> <li>• プール現在値</li> <li>• 使用中のコネクション数</li> <li>• 未使用のコネクション数</li> <li>• getConnection() メソッドの実行時間</li> <li>• getXACConnection() メソッドの実行時間</li> <li>• getConnection() メソッドの失敗回数</li> <li>• Connection で FATAL エラーが発生した回数</li> </ul>
リソースアダプタ	<p>基本情報</p> <ul style="list-style-type: none"> <li>• リソース名</li> <li>• リソースタイプ</li> <li>• 説明</li> <li>• リソースアダプタ提供ベンダ名</li> <li>• 準拠する JCA 仕様のバージョン</li> <li>• リソースアダプタのバージョン</li> <li>• 接続先 EIS のタイプ</li> <li>• ConnectionFactory のインタフェース名</li> <li>• ConnectionFactory の実装クラス名</li> <li>• ManagedConnectionFactory の実装クラス名</li> <li>• Connection のインタフェース名</li> <li>• Connection の実装クラス名</li> <li>• トランザクションサポートモデル</li> <li>• 設定プロパティ情報</li> <li>• ユーザ ID</li> <li>• コネクションプール <sup>2</sup> <sup>3</sup></li> </ul>



監視の対象	監視できる項目
	<p>稼働情報</p> <ul style="list-style-type: none"> <li>• リソース名</li> <li>• プール現在値 (総数) <sup>2</sup> <sup>3</sup></li> <li>• 使用中のコネクション数 <sup>2</sup> <sup>3</sup></li> <li>• 未使用のコネクション数 <sup>2</sup> <sup>3</sup></li> <li>• ManagedConnectionFactory の createManagedConnection() メソッドの実行回数 <sup>2</sup> <sup>3</sup></li> <li>• ManagedConnection の getConnection() メソッドの実行回数 <sup>2</sup> <sup>3</sup></li> <li>• ManagedConnection の cleanup() メソッドの実行回数 <sup>2</sup> <sup>3</sup></li> <li>• ManagedConnection の destroy() メソッドの実行回数 <sup>2</sup> <sup>3</sup></li> <li>• ConnectionManager の allocateConnection() メソッドの実行時間 <sup>2</sup> <sup>3</sup> <sup>4</sup></li> <li>• ManagedConnectionFactory の createManagedConnection() メソッドの実行時間 <sup>2</sup> <sup>3</sup></li> <li>• ConnectionManager の allocateConnection() メソッドの失敗回数 <sup>2</sup> <sup>3</sup> <sup>4</sup></li> <li>• ManagedConnection で FATAL エラーが発生した回数 <sup>2</sup> <sup>3</sup></li> </ul>
サービス	<p>トランザクション</p> <p>基本情報</p> <ul style="list-style-type: none"> <li>• サービス名</li> <li>• サービスタイプ</li> <li>• トランザクションタイムアウトデフォルト値</li> </ul> <p>稼働情報</p> <ul style="list-style-type: none"> <li>• アクティブトランザクション数</li> <li>• 平均トランザクション時間</li> </ul>

注 使用する機能 (Web サーバ連携機能, インプロセス HTTP サーバ, 最大同時実行スレッド数制御機能などを使用するかどうか), オプションの設定, 使用するリソースの違いなどによっては表示されない項目があります。

注 1 Connector 1.5 仕様に準拠したリソースアダプタを使用した場合は, 「Other」が出力されません。

注 2 ルートリソースアダプタを使用している場合, 「0」または「Not Available」が出力されません。

注 3 Cosminexus RM を使用している場合, これらの項目は Cosminexus RM 側の稼働情報として出力されます。Management Server の稼働情報監視では確認できません。

注 4 メンバリソースアダプタを使用している場合, 「0」または「Not Available」が出力されません。

2. システムの運用監視

表 2-21 稼働情報監視で表示できる項目 ( バッチサーバの場合 )

監視の対象		監視できる項目
バッチサーバ	J2EE コンテナ	<ul style="list-style-type: none"> <li>• バッチサーバ名</li> <li>• Naming Service ホスト</li> <li>• Naming Service ポート番号</li> <li>• コンテナ起動時刻</li> </ul>
	JavaVM	基本情報 <ul style="list-style-type: none"> <li>• JavaVM 名</li> <li>• JavaVM バージョン</li> <li>• JavaVM の最大メモリ使用量</li> </ul> 稼働情報 <ul style="list-style-type: none"> <li>• JavaVM の空きメモリ量</li> <li>• JavaVM の総メモリ容量</li> </ul>
リソース	データソース ( SimpleJTA )	基本情報 <ul style="list-style-type: none"> <li>• リソース名</li> <li>• リソースタイプ</li> <li>• 説明</li> <li>• 認証タイプ</li> <li>• ログインタイムアウト</li> <li>• ユーザ ID</li> <li>• コネクションプール</li> </ul> 稼働情報 <ul style="list-style-type: none"> <li>• リソース名</li> <li>• プール現在値 ( 総数 )</li> <li>• 使用中のコネクション数</li> <li>• 未使用のコネクション数</li> </ul>
	データソース ( FullJTA )	基本情報 <ul style="list-style-type: none"> <li>• リソース名</li> <li>• リソースタイプ</li> <li>• 説明</li> <li>• ログインタイムアウト</li> <li>• ユーザ ID</li> <li>• コネクションプール</li> </ul> 稼働情報 <ul style="list-style-type: none"> <li>• リソース名</li> <li>• プール現在値</li> <li>• 使用中のコネクション数</li> <li>• 未使用のコネクション数</li> <li>• getConnection() メソッドの実行時間</li> <li>• getXACONNECTION() メソッドの実行時間</li> <li>• getConnection() メソッドの失敗回数</li> <li>• Connection で FATAL エラーが発生した回数</li> </ul>

監視の対象	監視できる項目
リソースアダプタ	<p>基本情報</p> <ul style="list-style-type: none"> <li>• リソース名</li> <li>• リソースタイプ</li> <li>• 説明</li> <li>• リソースアダプタ提供ベンダ名</li> <li>• 準拠する JCA 仕様のバージョン</li> <li>• リソースアダプタのバージョン</li> <li>• 接続先 EIS のタイプ</li> <li>• ConnectionFactory のインタフェース名</li> <li>• ConnectionFactory の実装クラス名</li> <li>• ManagedConnectionFactory の実装クラス名</li> <li>• Connection のインタフェース名</li> <li>• Connection の実装クラス名</li> <li>• トランザクションサポートモデル</li> <li>• 設定プロパティ情報</li> <li>• ユーザ ID</li> <li>• コネクションプール <sup>1</sup> <sup>2</sup></li> </ul> <p>稼働情報</p> <ul style="list-style-type: none"> <li>• リソース名</li> <li>• プール現在値 (総数) <sup>1</sup> <sup>2</sup></li> <li>• 使用中のコネクション数 <sup>1</sup> <sup>2</sup></li> <li>• 未使用のコネクション数 <sup>1</sup> <sup>2</sup></li> <li>• ManagedConnectionFactory の createManagedConnection() メソッドの実行回数 <sup>1</sup> <sup>2</sup></li> <li>• ManagedConnection の getConnection() メソッドの実行回数 <sup>2</sup> <sup>3</sup></li> <li>• ManagedConnection の cleanup() メソッドの実行回数 <sup>2</sup> <sup>3</sup></li> <li>• ManagedConnection の destroy() メソッドの実行回数 <sup>1</sup> <sup>2</sup></li> <li>• ConnectionManager の allocateConnection() メソッドの実行時間 <sup>1</sup> <sup>2</sup> <sup>3</sup></li> <li>• ManagedConnectionFactory の createManagedConnection() メソッドの実行時間 <sup>1</sup> <sup>2</sup></li> <li>• ConnectionManager の allocateConnection() メソッドの失敗回数 <sup>1</sup> <sup>2</sup> <sup>3</sup></li> <li>• ManagedConnection で FATAL エラーが発生した回数 <sup>1</sup> <sup>2</sup></li> </ul>
サービス	<p>トランザクション</p> <p>基本情報</p> <ul style="list-style-type: none"> <li>• サービス名</li> <li>• サービスタイプ</li> <li>• トランザクションタイムアウトデフォルト値</li> </ul> <p>稼働情報</p> <ul style="list-style-type: none"> <li>• アクティブトランザクション数</li> <li>• 平均トランザクション時間</li> </ul>

注 使用する機能 (Web サーバ連携機能, インプロセス HTTP サーバ, 最大同時実行スレッド数制

## 2. システムの運用監視

御機能などを使用するかどうか)、オプションの設定、使用するリソースの違いなどによっては表示されない項目があります。

注 1 ルートリソースアダプタを使用している場合、「0」または「Not Available」が出力されません。

注 2 Cosminexus RM を使用している場合、これらの項目は Cosminexus RM 側の稼働情報として出力されます。Management Server の稼働情報監視では確認できません。

注 3 メンバリソースアダプタを使用している場合、「0」または「Not Available」が出力されません。

表 2-22 稼働情報監視で表示できる項目 (SFO サーバの場合)

監視の対象		監視できる項目
SFO サーバ	J2EE コンテナ	<ul style="list-style-type: none"><li>• J2EE サーバ名</li><li>• Naming Service ホスト</li><li>• Naming Service ポート番号</li><li>• コンテナ起動時刻</li></ul>
	JavaVM	<p>基本情報</p> <ul style="list-style-type: none"><li>• JavaVM 名</li><li>• JavaVM バージョン</li><li>• JavaVM の最大メモリ使用量</li></ul> <p>稼働情報</p> <ul style="list-style-type: none"><li>• JavaVM の空きメモリ量</li><li>• JavaVM の総メモリ容量</li></ul>
アプリケーション (SFO 用)	J2EE アプリケーション	<ul style="list-style-type: none"><li>• アプリケーション名 (display name)</li><li>• 説明</li><li>• デプロイされた時刻</li></ul>

### ポイント

#### 統計情報のサンプリング時間の設定

稼働情報を表示する場合、統計情報のサンプリング時間を指定してください。

サンプリング時間を指定すると、画面を表示した時刻または更新した時刻からさかのぼって、サンプリング時間 (これを N 秒といいます) に指定した時間内のデータを統計情報として表示できます。「N 秒ピーク」や「N 秒平均値」とはこの統計情報です。また、最大値、最小値などは、サンプリングを開始した以降のデータの中から値が抽出されます。

運用管理コマンド (mngsvrutil) にサブコマンド「set」を指定してサンプリング時間を設定できます。詳細は、マニュアル「Cosminexus リファレンス コマンド編」を参照してください。

## 2.5 リソース枯渇の監視

この節では、J2EE サーバまたは SFO サーバのリソースの使用率または使用数の推移を基に、リソース枯渇を監視する方法について説明します。

J2EE サーバまたは SFO サーバのリソース枯渇を監視する設定にしている場合、監視対象のリソースについての情報が、一定間隔でファイルに出力されます。この情報を、リソース枯渇監視情報といいます。リソース枯渇監視情報は、監視するリソースの種類ごとに出力するかどうかを選択できます。なお、リソース枯渇監視情報を出力したファイルを、リソース枯渇監視ログファイルといいます。

リソース枯渇監視情報を利用すると、リソースの使用率または使用数の変化のしかたを確認できます。また、この情報は、リソースの使用率または使用数がしきい値を超えた場合に、その原因を調査するためにも利用できます。

リソース枯渇監視ログファイルの出力先については、「5.4.1 Cosminexus のログの取得 (J2EE アプリケーションを実行するシステム)」を参照してください。

リソース枯渇監視の設定については、マニュアル「Cosminexus システム構築ガイド」のリソース枯渇監視の設定に関する説明を参照してください。

なお、リソースの使用率または使用数にしきい値を設定して監視すると、しきい値を超えた場合にアラートを出力できます。このアラートに対応して必要な処理を自動実行するように設定できます。Management イベントによる処理の自動実行の設定については、マニュアル「Cosminexus システム構築ガイド」の Management イベントによる処理の自動実行の設定に関する説明を参照してください。

### 2.5.1 出力される情報の種類

リソース枯渇監視情報には、次の表に示す情報が出力されます。なお、Windows、または AIX の場合、ファイルディスクリプタの数は監視できません。また、Linux の場合、スレッド監視は使用できません。

表 2-23 リソース枯渇監視情報に出力される情報

情報の種類	ファイル名	情報の概要
メモリ枯渇監視情報	cjmemorywatch	JavaVM のヒープおよびメモリ領域の使用状況の監視結果が出力されます。 この情報は、usrconf.properties の ejbserver.watch.memory.writefile.enabled キーに true が設定されている場合に出力されます。
ファイルディスクリプタ枯渇監視情報	cjfiledescriptorwatch	J2EE サーバプロセスが使用しているファイルディスクリプタの数の監視結果が出力されます。 この情報は、usrconf.properties の ejbserver.watch.fileDescriptor.writefile.enabled キーに true が設定されている場合に出力されます。

## 2. システムの運用監視

情報の種類	ファイル名	情報の概要
スレッド枯渇監視情報	cjthreadwatch	J2EE サーバプロセスが生成したスレッド数の監視結果が出力されます。 この情報は、usrconf.properties の ejbserver.watch.thread.writefile.enabled キーに true が設定されている場合に出力されます。
スレッドダンプ枯渇監視情報	cjthreaddumpwatch	出力されたスレッドダンプのファイル数の監視結果が出力されます。 この情報は、usrconf.properties の ejbserver.watch.threaddump.writefile.enabled キーに true が設定されている場合に出力されます。
HTTP リクエスト実行待ちキュー枯渇監視情報	cjrequestqueuewatch	HTTP リクエスト実行待ちキューの監視結果として、次の 2 種類の実行待ちキューが監視され、結果が出力されます。 <ul style="list-style-type: none"> <li>Web アプリケーション単位の最大同時実行スレッド数が設定されていない Web アプリケーションの実行待ちキューが監視されて、リクエスト格納数の監視結果が出力されます。 この情報は、usrconf.properties の ejbserver.watch.defaultRequestQueue.writefile.enabled キーに true が設定されている場合に出力されます。</li> <li>Web アプリケーション単位の最大同時実行スレッド数が設定されている Web アプリケーションの実行待ちキューが監視されて、リクエスト格納数の監視結果が出力されません。 この情報は、アプリケーションの属性として、<code>&lt;war&gt;&lt;hitachi-war-property&gt;&lt;thread-control&gt;&lt;resource-watcher&gt;</code> タグ下の <code>&lt;watcher-writefile-enabled&gt;</code> タグに true が設定されている場合に出力されます。</li> </ul> <p>なお、Web アプリケーション単位の最大同時実行スレッド数の制御が有効な場合（usrconf.properties の <code>webserver.container.thread_control.enabled</code> キーに true が設定されている場合）に、これらの監視が実行されます。</p>
セッション数枯渇監視情報	cjhttpsessionwatch	Web アプリケーション単位のセッション数の監視結果が出力されます。 この情報は、アプリケーションの属性として、 <code>&lt;war&gt;&lt;hitachi-war-property&gt;&lt;http-session&gt;&lt;resource-watcher&gt;</code> タグ下の <code>&lt;watcher-writefile-enabled&gt;</code> タグに true が設定されている場合に出力されます。
コネクションプール枯渇監視情報	cjconnectionpoolwatch	リソースアダプタ単位のコネクションプール数の監視結果が出力されます。 この情報は、リソースアダプタのプロパティとして、 <code>WatchWriteFileEnabled</code> プロパティに true が設定されている場合に出力されます。

### 2.5.2 出力形式と出力内容

リソース枯渇監視情報は、出力レコードが CSV 形式ファイルで出力されます。

出力形式を次の図に示します。

図 2-2 リソース枯渇監視情報の出力形式

yyyy/mm/dd hh:mm:ss.sss	pid	tid	message-id	message (LANG=ja)
0000 2004/99/99 99:99:99.999	HEJB	00000000	00000000	
				,"Title1", "Title2", "Title3", ...
0000 2004/99/99 99:99:99.999	HEJB	00000000	00000000	,"Data1", "Data2", "Data3", ...
0000 2004/99/99 99:99:99.999	HEJB	00000000	00000000	,"Data1", "Data2", "Data3", ...

出力形式について説明します。

- それぞれのカラムの出力内容がわかるように、タイトル行が、100 行ごとに出力されます。
- タイトルには、リソース共通の情報として、次の項目が出力されます。これらの項目は、図中の、"Title1", "Title2", "Title3"...の個所に出力されます。
  - Rate：割合（使用率）
  - Current：現在値
  - Max：最大値
  - Threshold：しきい値

このほか、リソースによって、必要な情報が出力されます。詳細は、それぞれのリソースの出力内容を参照してください。

- しきい値判定の計算結果、現在値情報、設定したしきい値の順で出力されます。しきい値判定の結果が複数ある場合は、複数出力されます。
- しきい値が % で設定されている場合、しきい値の計算結果も % で出力されます。例えば、計算結果が 0.752 であった場合、しきい値が % で指定されているときには、75.2 と出力されます。
- しきい値が絶対値の場合は、しきい値の計算結果は出力されません。
- しきい値として上限が設定されていない場合、しきい値の計算結果は、「-」と出力されます。
- 値が小数になる場合は、小数点以下第 2 位が四捨五入され、小数点以下第 1 位までが出力されます。なお、しきい値判定にも、小数点以下第 2 位が四捨五入され、小数点以下第 1 位までにした値が利用されます。

なお、ファイルに出力される内容については、usrconf.properties、アプリケーションの属性、またはリソースアダプタの属性の設定内容に従います。

それぞれの出力情報について説明します。

### (1) メモリ枯渇監視情報

メモリ枯渇監視情報には、次の表に示す情報が出力されます。

2. システムの運用監視

表 2-24 メモリ枯渇監視情報の出力内容

出力タイトル文字列	出力内容
Rate1	Tenured 領域のメモリ使用率が出力されます。 単位は % です。 Tenured 領域のメモリ使用率は、次の計算式で算出されます。 • Tenured 領域のメモリ使用率 = Tenured 領域消費サイズ / Tenured 領域合計サイズ × 100
Rate2	Tenured 領域最大空きメモリに対する New 領域の合計メモリ率が出力されます。 単位は % です。 Tenured 領域最大空きメモリに対する New 領域の合計メモリ率は、次の計算式で算出されます。 • Tenured 領域最大空きメモリに対する New 領域の合計メモリ率 = New 領域合計サイズ / Tenured 領域最大空きサイズ × 100
Rate3	Permanent 領域のメモリ使用率が出力されます。 単位は % です。 Permanent 領域のメモリ使用率は、次の計算式で算出されます。 • Permanent 領域のメモリ使用率 = Permanent 領域消費サイズ / Permanent 領域合計サイズ × 100
Free [New]	New 領域の空きメモリサイズが出力されます。 単位はバイトです。
Total [New]	New 領域の合計メモリサイズが出力されます。 単位はバイトです。
Max [New]	New 領域の最大メモリサイズが出力されます。 単位はバイトです。
Free [Old]	Tenured 領域の空きメモリサイズが出力されます。 単位はバイトです。
Total [Old]	Tenured 領域の合計メモリサイズが出力されます。 単位はバイトです。
Max [Old]	Tenured 領域の最大メモリサイズが出力されます。 単位はバイトです。
Free [Permanent]	Permanent 領域の空きメモリサイズが出力されます。 単位はバイトです。
Total [Permanent]	Permanent 領域の合計メモリサイズが出力されます。 単位はバイトです。
Max [Permanent]	Permanent 領域の最大メモリサイズが出力されます。 単位はバイトです。
Threshold	しきい値が出力されます。 単位は % です。



**!** 注意事項

メモリ枯渇監視機能は、十分にメモリ設計をして、フルガーベージコレクションが起こりにくいようにチューニングされたシステムを前提としています。メモリ枯渇監視機能は、このようにチューニングされたシステムであっても、フルガーベージコレクションが発生してアプリケーションが一時的に停止する予兆を検知することを目的としています。したがって、メモリチューニングが十分でない場合は、不必要にアラートが出力されることがありますので、ご注意ください。

メモリ枯渇監視をする場合は、Permanent 領域の最大サイズと初期サイズに同じ値を設定することをお勧めします。具体的には `usrconf.cfg` の `-XX:MaxPermSize` と `-XX:PermSize` を同じ値にします。異なる値を設定した場合、Permanent 領域の領域拡張でもアラートが出力される場合があります。そのほかのチューニングパラメタについては、Sun Microsystems, Inc. の Web サイトなどの情報を参照してください。

また、この機能で監視しているメモリ使用率が 100% に近づいた場合に必ずフルガーベージコレクションが発生するわけではありません。監視した値は、フルガーベージコレクションが発生する可能性が高いという予兆を検知するために使用されます。

## 参考

JavaVM でのヒープおよびメモリには、次の 3 種類の領域があります。

- New 領域  
Eden と Survivor に該当する領域です。新しいオブジェクトが格納されます。
- Tenured 領域  
Tenured に該当する領域です。長時間存在するオブジェクトが格納されます。
- Permanent 領域  
JavaVM にロードされたクラスが格納される領域です。

メモリ監視結果情報には、これらの領域ごとに、その時点での空き領域、消費されている領域、および最大領域のサイズが出力されます。

メモリ監視を設定した場合、次の場合にメッセージが出力されます。

- Tenured 領域の合計サイズに対する Tenured 領域の消費サイズがしきい値以上の場合  
「Tenured 領域消費サイズ / Tenured 領域合計サイズ × 100 しきい値 (%)」の場合に出力されます。
- Tenured 領域の最大空きサイズに対する New 領域の合計サイズがしきい値以上の場合  
「New 領域合計サイズ / Tenured 領域最大空きサイズ × 100 しきい値 (%)」の場合に出力されます。
- Permanent 領域の合計サイズに対する Permanent 領域の消費サイズがしきい値以上の場合  
「Permanent 領域消費サイズ / Permanent 領域合計サイズ × 100 しきい値 (%)」の場合に出力されます。

## (2) ファイルディスクリプタ枯渇監視情報

ファイルディスクリプタ監視結果情報には、次の表に示す情報が出力されます。なお、Windows、または AIX の場合、ファイルディスクリプタの数は監視できません。

## 2. システムの運用監視

表 2-25 ファイルディスクリプタ枯渇監視情報の出力内容

出力タイトル文字列	出力内容
Current	J2EE サーバプロセスが使用しているファイルディスクリプタ数が出力されます。
Max	プロセスに割り当て可能なファイルディスクリプタの数が出力されます。 Solaris の場合、常に "-" が出力されます。
Threshold	しきい値が出力されます。

### (3) スレッド枯渇監視情報

スレッド枯渇監視情報には、次の表に示す情報が出力されます。

表 2-26 スレッド枯渇監視情報の出力内容

出力タイトル文字列	出力内容
Current	自プロセスが使用しているスレッド数が出力されます。
Max	<ul style="list-style-type: none"><li>• Windows の場合 常に "-" が出力されます。</li><li>• UNIX の場合 プロセスに生成できるスレッド数が出力されます。</li></ul>
Threshold	しきい値が出力されます。

### (4) スレッドダンプ枯渇監視情報

スレッドダンプ枯渇監視情報には、次の表に示す情報が出力されます。

表 2-27 スレッドダンプ枯渇監視情報の出力内容

出力タイトル文字列	出力内容
Rate	現在のスレッドダンプのファイル数が、最大値に占める割合が出力されます。 単位は % です。
Current	スレッドダンプのファイル数の現在値が出力されます。
Max	スレッドダンプのファイル数の上限値が出力されます。
Threshold	しきい値が出力されます。 単位は % です。

### (5) HTTP リクエスト実行待ちキュー枯渇監視情報

HTTP リクエスト実行待ちキューとは、Web アプリケーションの同時実行スレッドに対する、実行待ちキューのことです。

HTTP リクエスト実行待ちキュー枯渇監視情報には、次の表に示す情報が出力されます。  
なお、この枯渇監視情報には、次の 2 種類の情報が含まれます。

J2EE サーバ単位のリクエスト実行待ちキュー格納数の監視結果

## Web アプリケーション単位のリクエスト実行待ちキュー格納数の監視結果

表 2-28 HTTP リクエスト実行待ちキュー枯渇監視情報の出力内容

出力タイトル文字列	出力内容
J2eeApplicationName	J2EE アプリケーション名が出力されます。 デフォルトのリクエスト実行待ちキューの監視結果の場合は, "-" が出力されます。
ContextRootName	コンテキストルート名が出力されます。 デフォルトのリクエスト実行待ちキューの監視結果の場合は, "-" が出力されます。
Rate	現在のキュー格納数の最大値に占める割合が出力されます 単位は % です。
Current	キュー格納数の現在値が出力されます。
Max	実行待ちキューのリクエスト格納数の最大値が出力されます。
Threshold	しきい値が出力されます。 単位は % です。

## (6) セッション数枯渇監視情報

セッション数枯渇監視情報には、次の表に示す情報が出力されます。

表 2-29 セッション数枯渇監視情報の出力内容

出力タイトル文字列	出力内容
J2eeApplicationName	J2EE アプリケーション名が出力されます。
ContextRootName	コンテキストルート名が出力されます。
Rate	現在のセッション作成数の最大値に占める割合が出力されます。 単位は % です。 最大値が設定されていない場合は, "-" が出力されます。
Current	セッション数の現在値が出力されます。
Max	セッション数の最大値が出力されます。 設定されていない場合は "-" が出力されます。
Threshold	しきい値が出力されます。 単位は % です。

## (7) コネクションプール枯渇監視情報

コネクションプール枯渇監視情報には、次の表に示す情報が出力されます。

2. システムの運用監視

表 2-30 コネクションプール枯渇監視情報の出力内容

出力タイトル文字列	出力内容
ResourceName	<p>ユーザが付けたリソースアダプタ名が次の形式で出力されます。                      リソースアダプタを直接 J2EE サーバにデプロイした場合                      &lt;リソースアダプタ名&gt;                      リソースアダプタを J2EE アプリケーションに含めてデプロイした場合                      &lt;J2EE アプリケーション名&gt; : &lt;リソースアダプタ名&gt;                      リソースアダプタを J2EE アプリケーションに含めてデプロイして、テスト                      モードで開始した場合                      TEST#&lt;J2EE アプリケーション名&gt; : &lt;リソースアダプタ名&gt;</p>
Rate	<p>コネクションプールの使用率が出力されます。                      単位は % です。                      コネクション数の最大値が無制限の場合, "-" が出力されます。</p>
Active	<p>使用中のコネクション数が出力されます。</p>
Free	<p>未使用のコネクション数が出力されます。</p>
Current	<p>コネクション数の現在値が出力されます。</p>
Min	<p>コネクション数の最小値が出力されます。</p>
Max	<p>コネクション数の最大値が出力されます。                      コネクションプールが無制限の場合, "-1" が出力されます。</p>
Threshold	<p>しきい値が出力されます。                      単位は % です。</p>
All	<p>コネクションの総数 ( コネクションプール管理内のコネクションとコネク                      ションプール管理外のコネクションの総数 ) が出力されます。</p>

## 2.6 CTM の稼働統計情報の確認

この節では、運用管理コマンド (mngsvrutil) を使用して、CTM の稼働統計情報を収集する方法について説明します。また、収集した情報を確認する方法についても説明します。

次の単位で CTM の稼働統計情報を収集できます。

運用管理ドメイン内のすべての CTM

運用管理ドメイン内の指定した CTM

運用管理ドメイン内の指定したホスト内の CTM

### 2.6.1 CTM の稼働統計情報の収集方法

CTM の稼働統計情報の出力と収集は、mngsvrutil コマンドにサブコマンド「collect」を指定して実行します。

CTM の稼働統計情報は、次のどれかの単位で収集できます。

運用管理ドメイン内のすべての CTM

運用管理ドメイン内の指定した CTM

運用管理ドメイン内の指定したホスト内の CTM

なお、収集対象になるのは稼働中の CTM だけです。

CTM の稼働統計情報を出力、収集する場合の mngsvrutil コマンドの実行形式と実行例を次に示します。

運用管理ドメイン内のすべての CTM を対象にする場合

実行形式

```
mngsvrutil -m <Management Serverのホスト名> -u <管理ユーザID> -p <管理パスワード>
collect allCtmStatistics
```

実行例

```
mngsvrutil -m mnghost -u user01 -p pwl collect
allCtmStatistics
```

運用管理ドメイン内の指定した CTM、または指定したホスト内の CTM を対象にする場合

実行形式

```
mngsvrutil -m <Management Serverのホスト名> -u <管理ユーザID> -p <管理パスワード> -t <論理CTM名またはホスト名> [-k host] collect ctmStatistics
```

実行例

- 指定した CTM の稼働統計情報を収集する場合  
`mngsvrutil -m mnghost -u user01 -p pw1 -t ctm01 collect ctmStatistics`
- 指定したホスト内の CTM の稼働統計情報を収集する場合  
`mngsvrutil -m mnghost -u user01 -p pw1 -t host01 -k host collect ctmStatistics`

コマンドの実行結果は、標準出力またはファイルに出力されます。

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

## 2.6.2 CTM の稼働統計情報の出力先と出力情報

収集した稼働統計情報は CSV 形式で出力されて、ZIP 形式で圧縮されます。稼働統計情報が出力される CSV ファイルを、稼働統計出力ファイルといいます。

収集した稼働統計出力ファイルの出力先と出力情報について説明します。

収集した稼働統計出力ファイルの出力先

- Windows の場合  
`<Manager のログ出力ディレクトリ >%ctm`
- UNIX の場合  
`<Manager のログ出力ディレクトリ >/ctm`

なお、稼働統計出力ファイルは、稼働統計情報を収集する対象ごとに次の表に示すファイル名で出力されます。

表 2-31 収集対象ごとの稼働統計出力ファイル名

稼働統計情報の収集対象	ファイル名
運用管理ドメイン内のすべての CTM	<ドメイン名><日時>.zip
運用管理ドメイン内の指定した CTM	<論理サーバ名><日時>.zip
指定したホスト内の CTM	<ホスト名><日時>.zip

注 出力された稼働統計出力ファイル（CSV 形式）は、ZIP 形式で圧縮されます。

また、CSV 形式に出力される稼働統計情報は、CTM ノード単位、キュー単位、およびメソッド単位でそれぞれ異なるファイルで出力されます。出力単位ごとの稼働統計情報

のファイル名を次に示します。

表 2-32 出力単位ごとの稼働統計出力ファイル名

出力単位	ファイル名
CTM ノード単位	noddata.csv
キュー単位	quedata.csv
メソッド単位	mtddata.csv

#### 稼働統計情報の出力情報

稼働統計出力ファイルに出力される情報を次の表に示します。

表 2-33 稼働統計出力ファイルに出力される情報

統計情報種別	情報	説明
リクエスト情報	CTM レギュレータレスポンス時間	TPBroker クライアント、および EJB クライアントからの要求を受信した CTM レギュレータが、J2EE サーバに対してリクエストを送信してから応答を受信するまでの時間の差です。CTM レギュレータで出力されます。発生件数、および最大値・最小値・平均値がマイクロ秒単位で出力されます。
	CTM レギュレータリクエストタイムアウト	CTM レギュレータがリクエストを送信してから応答が返されるまでの間に発生したタイムアウト障害の発生件数です。CTM レギュレータで出力されます。
	CTM レギュレータリクエスト障害	CTM レギュレータがリクエストを送信してからその応答が返されるまでの処理で発生した障害の発生件数です。CTM レギュレータで出力されます。
	OTM ゲートウェイレスポンス時間	TPBroker OTM クライアントからの要求を受信した OTM ゲートウェイが、J2EE サーバに対してリクエストを送信してから、応答を受信するまでの時間の差です。OTM ゲートウェイで出力されます。発生件数、および最大値・最小値・平均値がマイクロ秒単位で出力されます。
	OTM ゲートウェイリクエストタイムアウト	OTM ゲートウェイでのリクエストの応答待ちで発生したタイムアウト障害の件数です。OTM ゲートウェイで出力されます。
	OTM ゲートウェイリクエスト障害	OTM ゲートウェイがリクエストを送信してからその応答が返されるまでの処理で発生した障害の件数です。OTM ゲートウェイで取得します。

## 2. システムの運用監視

統計情報種別	情報	説明
スケジュール情報	スケジュール待ち件数	スケジュールキューに滞留したリクエストの数です。 発生件数、および最大値・最小値・平均値が出力されます。
	スケジュール待ち時間	リクエストがスケジュールキューに登録されてから取り出されるまでの滞留時間です。 発生件数、および最大値・最小値・平均値がマイクロ秒単位で出力されます。
	スケジュールできなかったリクエスト数	スケジュールキューに同時に登録できる数を超えたため、スケジュールできなかったリクエストの数です。
プロセス情報	CTM レギュレータ異常終了件数	CTM レギュレータの異常終了が発生した件数です。 異常終了が発生するたびに、この情報が出力されます。
	OTM ゲートウェイ異常終了件数	OTM ゲートウェイの異常終了が発生した件数です。 異常終了が発生するたびに、この情報を取得します。



統計情報種別	情報	説明
コネクション情報	CTM レギュレータとクライアント間	CTM レギュレータとクライアントとのコネクション情報です。ctmstsstart コマンドの -CTMCheckInterval オプションに指定した間隔で出力されます。 最大値・最小値・平均値がマイクロ秒単位で出力されます。
	OTM ゲートウェイと TPBroker OTM クライアント間	OTM ゲートウェイと TPBroker OTM クライアントとのコネクション情報です。ctmstsstart コマンドの -CTMCheckInterval オプションに指定した間隔でこの情報を取得します。 最大値・最小値・平均値がマイクロ秒単位で出力されます。
	CTM デーモンと CTM レギュレータ間	CTM デーモンと CTM レギュレータとのコネクション情報です。ctmstsstart コマンドの -CTMCheckInterval オプションに指定した間隔でこの情報を取得します。 最大値・最小値・平均値がマイクロ秒単位で出力されます。
	CTM デーモンと OTM ゲートウェイ間	CTM デーモンと OTM ゲートウェイとのコネクション情報です。ctmstsstart コマンドの -CTMCheckInterval オプションに指定した間隔でこの情報を取得します。 最大値・最小値・平均値がマイクロ秒単位で出力されます。
	CTM デーモンと J2EE サーバ間 (制御用)	CTM デーモンと J2EE サーバ間の制御用コネクション情報です。ctmstsstart コマンドの -CTMCheckInterval オプションに指定した時間間隔でこの情報を取得します。
	CTM デーモンと J2EE サーバ間 (リクエスト用)	CTM デーモンと J2EE サーバ間のリクエスト要求処理コネクション情報です。ctmstsstart コマンドの -CTMCheckInterval オプションに指定した時間間隔でこの情報を取得します。

### 2.6.3 CTM の稼働統計情報の出力例

運用管理コマンド (mngsvrutil) を利用して収集した稼働統計出力ファイルの出力例を次に示します。

CTM ノード単位で出力した稼働統計情報の出力例

## 2. システムの運用監視

```

*** Statistics Information by CTMNode ***
CTMDomain, CTMID, start, end, Event, Count, Maximum, Minimum, Average, unit
CTMDOMAIN,10.209.15.55,2004/11/05 09:57,2004/11/05 10:07,Request Response
Time(Reg), 26525, 94000, 0, 9099,(usec)
CTMDOMAIN,10.209.15.55,2004/11/05 09:57,2004/11/05 10:07,Request Time Out(Reg),
0,,,,(count)
CTMDOMAIN,10.209.15.55,2004/11/05 09:57,2004/11/05 10:07,Request Error(Reg),
2,,,,(count)
CTMDOMAIN,10.209.15.55,2004/11/05 09:57,2004/11/05 10:07,Request Response
Time(OTM), 0, 0, 0, 0,(usec)
CTMDOMAIN,10.209.15.55,2004/11/05 09:57,2004/11/05 10:07,Request Time Out(OTM),
0,,,,(count)
CTMDOMAIN,10.209.15.55,2004/11/05 09:57,2004/11/05 10:07,Request Error(OTM),
0,,,,(count)
CTMDOMAIN,10.209.15.55,2004/11/05 09:57,2004/11/05 10:07,Schedule Wait Count,
26526, 1, 1, 1,(count)
CTMDOMAIN,10.209.15.55,2004/11/05 09:57,2004/11/05 10:07,Schedule Wait Time,
26526, 5688000, 0, 560,(usec)
CTMDOMAIN,10.209.15.55,2004/11/05 09:57,2004/11/05 10:07,Request Overflow,
0,,,,(count)
CTMDOMAIN,10.209.15.55,2004/11/05 09:57,2004/11/05 10:07,Regulator down,
0,,,,(count)
CTMDOMAIN,10.209.15.55,2004/11/05 09:57,2004/11/05 10:07,OTMGateway down,
0,,,,(count)
CTMDOMAIN,10.209.15.55,2004/11/05 09:57,2004/11/05 10:07,Regulator-Client,,
2, 0, 1,(count)
CTMDOMAIN,10.209.15.55,2004/11/05 09:57,2004/11/05 10:07,OTMGateway-Client,,
0, 0, 0,(count)
CTMDOMAIN,10.209.15.55,2004/11/05 09:57,2004/11/05
10:07,CTMDaemon-J2EEServer(Ctr),, 2, 0, 1,(count)
CTMDOMAIN,10.209.15.55,2004/11/05 09:57,2004/11/05
10:07,CTMDaemon-J2EEServer(Req),, 1, 0, 1,(count)
CTMDOMAIN,10.209.15.55,2004/11/05 09:57,2004/11/05 10:07,CTMDaemon-Regulator,,
1, 1, 1,(count)
CTMDOMAIN,10.209.15.55,2004/11/05 09:57,2004/11/05
10:07,CTMDaemon-OTMGateway,, 5, 5, 5,(count)

```

### キュー単位で出力した稼働統計情報の出力例

```

*** Statistics Information by Queue ***
CTMDomain, CTMID, QueueName, start, end, Event, Count, Maximum, Minimum,
Average, unit
CTMDOMAIN,10.209.15.55,converter,2004/11/05 09:57,2004/11/05 10:07,Schedule
Wait Count, 26526, 1, 1, 1,(count)
CTMDOMAIN,10.209.15.55,converter,2004/11/05 09:57,2004/11/05 10:07,Schedule
Wait Time, 26526, 5688000, 0, 560,(usec)
CTMDOMAIN,10.209.15.55,converter,2004/11/05 09:57,2004/11/05 10:07,Request
Overflow, 0,,,,(count)
CTMDOMAIN,10.209.15.55,converter,2004/11/05 09:57,2004/11/05
10:07,CTMDaemon-J2EEServer(Req),, 1, 1, 1,(count)

```

### メソッド単位で出力した稼働統計情報の出力例

```

*** Statistics Information by Method ***
CTMDomain, CTMID, Interface, Method, start, end, Event, Count, Maximum, Minimum,
Average, unit
CTMDOMAIN,10.209.15.55,Converter,dollarToYen,2004/11/05 09:57,2004/11/05
10:07,Request Response Time(Reg), 10393, 78000, 0,
9143,(usec)
CTMDOMAIN,10.209.15.55,Converter,dollarToYen,2004/11/05 09:57,2004/11/05
10:07,Request Time Out(Reg), 0,,,,(count)
CTMDOMAIN,10.209.15.55,Converter,dollarToYen,2004/11/05 09:57,2004/11/05
10:07,Request Error(Reg), 0,,,,(count)
CTMDOMAIN,10.209.15.55,Converter,dollarToYen,2004/11/05 09:57,2004/11/05
10:07,Request Response Time(OTM), 0, 0, 0, 0,(usec)
CTMDOMAIN,10.209.15.55,Converter,dollarToYen,2004/11/05 09:57,2004/11/05
10:07,Request Time Out(OTM), 0,,,,(count)
CTMDOMAIN,10.209.15.55,Converter,dollarToYen,2004/11/05 09:57,2004/11/05
10:07,Request Error(OTM), 0,,,,(count)
CTMDOMAIN,10.209.15.55,Converter,yenToEuro,2004/11/05 09:57,2004/11/05
10:07,Request Response Time(Reg), 10393, 94000, 0,
9119,(usec)
CTMDOMAIN,10.209.15.55,Converter,yenToEuro,2004/11/05 09:57,2004/11/05
10:07,Request Time Out(Reg), 0,,,,(count)
CTMDOMAIN,10.209.15.55,Converter,yenToEuro,2004/11/05 09:57,2004/11/05
10:07,Request Error(Reg), 2,,,,(count)
CTMDOMAIN,10.209.15.55,Converter,yenToEuro,2004/11/05 09:57,2004/11/05
10:07,Request Response Time(OTM), 0, 0, 0, 0,(usec)
CTMDOMAIN,10.209.15.55,Converter,yenToEuro,2004/11/05 09:57,2004/11/05
10:07,Request Time Out(OTM), 0,,,,(count)
CTMDOMAIN,10.209.15.55,Converter,yenToEuro,2004/11/05 09:57,2004/11/05
10:07,Request Error(OTM), 0,,,,(count)

```

## 2.7 トランザクション情報の確認

この節では、稼働中および停止中の J2EE サーバのトランザクションの情報を確認する方法について説明します。稼働中の J2EE サーバでのトランザクションの状態や、停止中の J2EE サーバでの未決着のトランザクションの有無などの情報を確認できます。

### 2.7.1 稼働中のトランザクションの確認

J2EE サーバで稼働中のトランザクションの情報を確認できます。トランザクションの状態、グローバルトランザクション ID、経過時間、ブランチの種類などの情報を確認できます。

稼働中のトランザクションの確認には、`cjlisttrn` コマンドを使用します。実行形式と実行例を次に示します。

実行形式

```
cjlisttrn [<サーバ名称>] -bqual
```

実行例

```
cjlisttrn MyServer -bqual
```

また、未決着のトランザクションの状態も確認できます。未決着のトランザクションについての情報を確認するときは、引数に「`-pending`」を指定します。未決着のトランザクションについての情報を確認する場合の実行形式と実行例を次に示します。

実行形式

```
cjlisttrn [<サーバ名称>] -pending -bqual
```

実行例

```
cjlisttrn MyServer -pending -bqual
```

`cjlisttrn` コマンド、および取得できる情報の詳細については、マニュアル「[Cosminexus リファレンス コマンド編](#)」を参照してください。

### 2.7.2 停止中のトランザクションの確認

停止中の J2EE サーバのトランザクションの情報を確認できます。トランザクションの状態、グローバルトランザクション ID、経過時間、ブランチの種類等の情報を確認できます。また、未決着のトランザクションが残っているかどうかを確認できます。

停止中のトランザクションの確認には、`cjlisttrnfile` コマンドを使用します。実行形式と

実行例を次に示します。

実行形式

```
cjlisttrnfile [<サーバ名称>] -bqual
```

実行例

```
cjlisttrnfile MyServer -bqual
```

J2EE サーバが停止中の状態で未決着のトランザクションが存在する場合は、必要に応じて次の処理を実行して、トランザクションを決着させてください。

- J2EE サーバを再起動する (cjstartsv コマンド)
- J2EE サーバをリカバリモードで起動する (cjstartrecover コマンド)

cjlisttrnfile コマンド、および取得できる情報の詳細については、マニュアル「Cosminexus リファレンス コマンド編」を参照してください。



# 3

## システムの処理性能の解析

この章では、システムでの処理性能の解析方法について説明します。

Cosminexus では、システムの処理性能の解析に、パフォーマンストレーサという論理サーバを使用します。

---

3.1 処理性能の解析作業の概要

---

3.2 性能解析トレースファイルの収集

---

3.3 性能解析トレースファイルの利用方法

---

## 3.1 処理性能の解析作業の概要

---

Cosminexus で構築したシステムの処理性能は、クライアントからデータベースなどの EIS までの一連の処理、およびその処理結果がクライアントに返却されるまでのリクエストの一連の処理で、リダイレクタや Web コンテナなどから出力されるトレース情報を基に解析できます。トレース情報を出力するリダイレクタや Web コンテナなどを、機能レイヤといいます。リクエストの処理過程では、次に示す機能レイヤからトレース情報が出力されます。

- CTM
- リダイレクタ
- Web コンテナ
- EJB コンテナ
- JNDI
- JTA
- JCA コンテナ
- DB Connector
- RMI (通信処理)
- OTS
- 標準出力 / 標準エラー出力 / ユーザログ
- DI
- バッチアプリケーション実行機能

各機能レイヤでは決まった処理のポイント (トレース取得ポイント) でトレース情報を出力します。各機能レイヤが出力するトレース情報は、パフォーマンストレーサ (Cosminexus Performance Tracer) の PRF デーモンによって、トレースファイル (PRF トレースファイル) に出力されます。PRF デーモンにはトレース取得レベル (標準または詳細) を設定できます。PRF デーモンに設定するトレース取得レベルを PRF トレース取得レベルといいます。PRF トレース取得レベルの設定方法については、「付録 H.3 システムの処理性能解析」を参照してください。各機能レイヤのトレース取得ポイントおよび PRF トレース取得レベルの詳細については、「付録 A トレース取得ポイントと PRF トレース取得レベル」を参照してください。

なお、RMI (通信処理) の機能レイヤについては、トレース情報の取得を抑止することができます。その場合、トレース取得レベルに「抑止」を設定します。設定方法については、マニュアル「Cosminexus リファレンス コマンド編」の `cprfstart` コマンド (PRF デーモンの開始) の説明、または `cprflevel` コマンド (PRF トレース取得レベルの表示と変更) の説明を参照してください。



## 参考

このほか、次に示す Application Server のプロセス、構成ソフトウェアおよび関連プログラムでも、PRF トレースが取得できます。

- Cosminexus Web Services - Base
- uCosminexus TP1 Connector
- Cosminexus RM
- TP1/Client/J
- TP1/MQ Access
- HCSC サーバ
- uCosminexus Service Coordinator Interactive Workflow
- uCosminexus Service Adapter for Object Access
- uCosminexus Service Adapter for Flat Files

なお、各機能レイヤのトレース情報はバイナリ形式で PRF トレースファイルに出力されます。システムの処理性能を解析する際は、PRF トレースファイルをテキスト形式 (CSV 形式) に変換したファイル (性能解析トレースファイル) を使用します。性能解析トレースファイルの出力先および出力情報については、「3.2.2 性能解析トレースファイルの出力先と出力情報」を参照してください。

システムの処理性能を解析する作業について、次の表に示します。なお、これらの作業は、運用管理コマンド (mngsvrutil) を使用して実行します。

## 参考

保守レベルのトレース情報について

PRF トレース取得レベルには、標準レベルと詳細レベルのほかに、保守レベルがあります。保守レベルとは、障害発生時などの保守情報を取得するためのレベルを指します。通常は指定しないでください。

PRF トレース取得レベルに保守レベルを設定する方法については、マニュアル「Cosminexus リファレンス コマンド編」の cprflevel コマンドについての説明を参照してください。

## (1) 性能解析トレースファイルの収集

性能解析トレースファイルを収集する作業の概要について次の表に示します。

### 3. システムの処理性能の解析

表 3-1 性能解析トレースファイルの収集

作業内容	手段	作業概要	参照先
性能解析トレースファイルの出力/収集	運用管理コマンド (mngsvrutil)	PRF トレースファイルを CSV 形式に変換した性能解析トレースファイルを出力、収集します。 性能解析トレースファイルの対象範囲は次の中から選択できます。 <ul style="list-style-type: none"> <li>運用管理ドメイン内のすべてのパフォーマンスストレサ</li> <li>特定のホスト内のすべてのパフォーマンスストレサ</li> <li>特定のパフォーマンスストレサ</li> </ul>	3.2.1

#### (2) 性能解析トレースファイルを利用したシステムの処理性能の解析

収集した性能解析トレースファイルを利用してシステムの処理性能を解析する作業の概要について次の表に示します。なお、出力された性能解析トレースファイルは、CSV 形式のファイルを編集するアプリケーションプログラムなどを使用して解析します。

表 3-2 性能解析トレースファイルを利用したシステムの処理性能の解析

作業内容	作業概要	参照先
Web サーバのレスポンスタイムの解析	Web サーバがクライアントからリクエストを受けてから返却するまでに掛かった時間を解析します。	3.3.1
アプリケーションサーバ内でのリクエストの処理状況の調査	Web サーバがクライアントから受けたリクエストがアプリケーションサーバで処理されているかどうか調査します。	3.3.2
セッションのライフサイクルの調査	PRF トレース取得レベルを「詳細」に設定している場合に出力されるトレース（セッショントレース）を利用して、セッションのライフサイクルを調査します。	3.3.3
タイムアウトしたトランザクションの特定	J2EE アプリケーションまたはバッチアプリケーションのトランザクションがタイムアウトした場合に、メッセージまたは性能解析トレースファイルを使用してタイムアウトしたトランザクションを特定します。	3.3.4
タイムアウトしたリクエストの特定	リダイレクタでのレスポンス受信時にタイムアウトが発生した場合に、性能解析トレースを利用してタイムアウトが発生したリクエストを特定します。	3.3.5
ルートアプリケーション情報を利用したログの調査	J2EE アプリケーションまたはバッチアプリケーション内で Cosminexus の API を使用して、性能解析情報のルートアプリケーション情報の文字列表現を任意のタイミングでログファイルに出力します。	3.3.6
トラブルが発生した接続の特定	性能解析トレースファイルに出力された情報から、トラブルが発生した接続を特定します。	3.3.7
性能解析トレースとスレッドダンプの対応づけ	J2EE アプリケーションまたはバッチアプリケーションでスローダウンやハングアップが発生したときに、性能解析トレースとスレッドダンプを対応づけることで、問題が発生した個所を調査できます。	3.3.8

## 3.2 性能解析トレースファイルの収集

性能解析トレースファイルの収集方法、および収集したファイルの出力先について説明します。性能解析トレースファイルは、運用管理コマンドを使用して収集します。

### 3.2.1 性能解析トレースファイルの収集方法

運用管理コマンド (mngsvrutil) を使用して性能解析トレースファイルを収集する場合、mngsvrutil コマンドに、サブコマンド「collect」を指定して収集します。性能解析トレースファイルの収集時、トレース情報の収集対象を選択します。運用管理コマンドを使用した場合のトレース情報の収集対象を次に示します。

- 運用管理ドメイン内のすべてのパフォーマンストレーサ
- 特定のホスト内のすべてのパフォーマンストレーサ
- 特定のパフォーマンストレーサ

それぞれの場合の実行形式と実行例を次に示します。

運用管理ドメイン内のすべてのパフォーマンストレーサを対象にする場合

実行形式

```
mngsvrutil -m <Management Serverのホスト名> -u <管理ユーザID> -p <管理パスワード>
collect allPrfTrace
```

実行例

```
mngsvrutil -m mngghost -u user01 -p pwl collect allPrfTrace
```

特定のホスト内のすべてのパフォーマンストレーサを対象にする場合

実行形式

```
mngsvrutil -m <Management Serverのホスト名> -u <管理ユーザID> -p <管理パスワード> -t
<ホスト名> -k host collect prfTrace
```

実行例

```
mngsvrutil -m mngghost -u user01 -p pwl -t host01 -k host
collect prfTrace
```

特定のパフォーマンストレーサを対象にする場合

実行形式

```
mngsvrutil -m <Management Serverのホスト名> -u <管理ユーザID> -p <管理パスワード> -t
<論理パフォーマンストレーサ名> -k logicalServer collect prfTrace
```

実行例

### 3. システムの処理性能の解析

```
mngsvrutil -m mnghost -u user01 -p pw1 -t ID01 -k  
logicalServer collect prfTrace
```

## 3.2.2 性能解析トレースファイルの出力先と出力情報

収集した性能解析トレースファイルの出力先と出力情報について説明します。

### (1) 性能解析トレースファイルの出力先

- Windows の場合  
    <Manager のログ出力ディレクトリ >%prf
- UNIX の場合  
    <Manager のログ出力ディレクトリ >/prf

なお、性能解析トレースファイルは、トレース情報の収集対象別に次の表に示すファイル名で出力されます。

表 3-3 性能解析トレースファイルのファイル名

トレース情報の収集対象	ファイル名
運用管理ドメイン内のすべてのパフォーマンスストレサ	<運用管理ドメイン名><日時 <sup>1</sup> >.zip
特定のホスト内のすべてのパフォーマンスストレサ	<ホスト名><日時 <sup>1</sup> >.zip
特定のパフォーマンスストレサ	<論理サーバ名 <sup>2</sup> ><日時 <sup>1</sup> >.zip

注 1

性能解析トレースファイルを収集した日時が表示されます。

注 2

指定したパフォーマンスストレサの名称が表示されます。

### (2) 性能解析トレースファイルの出力情報

パフォーマンスストレサから性能解析トレースファイル (CSV 形式) に出力される情報を次の表に示します。なお、CTM とそれ以外の機能レイヤでは、取得項目が異なります。また、付加情報の有無など、取得ポイントごとに、出力される項目は異なります。各取得ポイントで出力される項目の詳細については、「付録 A トレース取得ポイントと PRF トレース取得レベル」を参照してください。

表 3-4 性能解析トレースファイルに出力する情報

トレース情報のヘッダ	説明	値の範囲
PRF	そのプロセスのレコードの状態 (正常または異常)。	次のどちらかが出力されます。 正常: Rec 異常: ErrRec

トレース情報のヘッダ	説明	値の範囲
Process	トレース情報を取得したプロセスのプロセス ID。	10 けたの 10 進数が出力されます。
Thread	トレース情報を取得したプロセス内スレッドのスレッド ID, およびスレッドのハッシュ値 <sup>1</sup> 。	スレッド ID: 20 けた以内の 10 進数が出力されます。 ハッシュ値: 10 けた以内の 10 進数が出力されます。
Trace	トレース情報を取得したプロセス内スレッドでのトレース通番。	10 けたの 10 進数が出力されます。
ProcessName	プロセス名。	32 文字以内のプロセスを表す文字列 <sup>2</sup> が出力されます。
Event	トレース取得ポイントを示すイベント ID。	6 けたの 16 進数 (6 けたには, "0x" も含まれます) が出力されます <sup>3</sup> 。
Date	トレース情報を取得した年月日。	年月日が「yyyy/mm/dd」のフォーマットで出力されます。 yyyy: 年 mm: 月 dd: 日
Time	トレース情報を取得した時刻 (時:分:秒)。	時刻が「hh:mm:ss」のフォーマットで出力されます。 hh: 時 mm: 分 ss: 秒
Time(msec/usec/nsec)	トレース情報を取得した時刻 (ミリ秒 / マイクロ秒 / ナノ秒)。	時刻が「ms/us/ns」のフォーマットで出力されます。 ms: ミリ秒 us: マイクロ秒 ns: ナノ秒
Rc	リターンコード。	16 けたの 16 進数 (16 けたには, "0x" も含む) が出力されます。 正常: 0 異常: 1 (または 0 以外)
ClientAP IP <sup>4</sup>	トレース情報を取得したクライアントアプリケーションの IP アドレス。	IP アドレスが, 「aaa.bbb.ccc.ddd」のフォーマットで出力されます。
ClientAP PID <sup>4</sup>	トレース情報を取得したクライアントアプリケーションのプロセス ID。	10 けたの 10 進数が出力されます。
ClientAP CommNo. <sup>4</sup>	トレース情報を取得したクライアントアプリケーションの通信番号。	18 けたの 16 進数 (18 けたには, "0x" も含まれます) が出力されます。
RootAP IP <sup>5</sup>	トレース情報を取得したルートアプリケーションの IP アドレス。	IP アドレスが「aaa.bbb.ccc.ddd」のフォーマットで出力されます。
RootAP PID <sup>5</sup>	トレース情報を取得したルートアプリケーションのプロセス ID。	10 けたの 10 進数が出力されます。

### 3. システムの処理性能の解析

トレース情報のヘッダ	説明	値の範囲
RootAP CommNo. <sup>5</sup>	トレース情報を取得したルートアプリケーションの通信番号。	18 けたの 16 進数 (18 けたには, "0x" も含む) が出力されます
SendSCD IP <sup>5</sup>	リクエスト要求元 CTM の IP アドレス。	IP アドレスが「aaa.bbb.ccc.ddd」のフォーマットで出力されます。
SendSCD PID <sup>6</sup>	リクエスト要求元 CTM のプロセス ID。	10 けたの 10 進数が出力されま す。
ReceiveSCD IP <sup>6</sup>	リクエスト要求先 CTM の IP アドレス。	IP アドレスが「aaa.bbb.ccc.ddd」 のフォーマットで出力されます。
ReceiveSCD PID <sup>6</sup>	リクエスト要求先 CTM のプロセス ID。	10 けたの 10 進数
INT	取得ポイントごとのインタフェース名。	33 文字以内の文字列 <sup>7</sup> が出力 されます。
OPR	取得ポイントに関連するオペレーション 情報。	33 文字以内の文字列 <sup>7</sup> が出力 されます。
LookupName <sup>6</sup>	ルックアップ名。	33 文字以内の文字列 <sup>7</sup> が出力 されます。
OPT <sup>8</sup>	取得ポイントごとの付加情報。	514 文字以内の 16 進数値文字列 が出力されます。
ASCII	取得ポイントごとの付加情報を ASCII 文字出力。	OPT の内容が 258 文字以内の ASCII 文字列として出力されま す。

#### 注 1

CTM およびリダイレクタで取得したトレース情報には、スレッドのハッシュ値は出力されない場合があります。

#### 注 2

プロセス名称は次のように決定されます。

- EJB クライアントアプリケーションの場合

EJB クライアントアプリケーションのシステムプロパティ

`ejbserver.server.prf.processName` に指定された名称。

このシステムプロパティを指定していない場合、またはこのプロパティに null 文字を指定した場合は、"EJBClient" になります。

- J2EE サーバ、バッチサーバ、および Web コンテナサーバの場合

サーバ名がプロセス名称となります。

- リダイレクタの場合

Web サーバとして Hitachi Web Server を使用している場合は、"RD-`{Web サーバ待ち受けポート番号}`" がプロセス名称となります。Microsoft IIS を使用している場合は、"Redirector" がプロセス名称となります。

- CTM の場合

CTM の各プロセス名称となります。

## 注 3

イベント ID は、機能レイヤのトレース取得ポイントごとに割り当てられています。詳細は、「付録 A トレース取得ポイントと PRF トレース取得レベル」を参照してください。

## 注 4

クライアントアプリケーション情報の構成要素です。

## 注 5

ルートアプリケーション情報の構成要素です。Web コンテナで出力されるトレースポイントでは、トレース情報を取得したルートアプリケーションの IP アドレス、プロセス ID、および通信番号が「0.0.0. 0/0/0x0000000000000000」と出力される場合があります。詳細は「3.3.6 ルートアプリケーション情報を利用したログ調査」を参照してください。

## 注 6

CTM の場合だけ出力される情報です。CTM 以外のレイヤでは、「\*\*\*\*」が表示されません。

## 注 7

インタフェース名、およびオペレーション名が 33 文字を超える場合、次のどれかの方法で、33 文字になって出力されます。詳細は、「付録 A トレース取得ポイントと PRF トレース取得レベル」を参照してください。

先頭 32 文字 +\*

先頭 16 文字 ++ 末尾 16 文字

\*+ 末尾 32 文字

## 注 8

機能レイヤによっては、「OPT」に入リ口時刻が出力されるトレース取得ポイントがあります。入リ口時刻とは、そのトレース取得ポイントのトレースに対応する入リ口トレースが出力された時間を示します。例えば、リダイレクタのトレース取得ポイントの場合、Web サーバへの HTTP レスポンスボディ情報の送信完了時 (0x8104) のトレース取得ポイントで出力される入リ口時刻は、Web コンテナからの HTTP レスポンスボディ情報の送信開始時 (0x8004) の時刻になります。なお、入リ口時刻は、「1970 年 1 月 1 日 00:00:00」からの通算時間が 16 バイトの値で出力されます。前半の 8 バイトの値は秒単位を、後半の 8 バイトの値はマイクロ秒単位を示します。ただし、リダイレクタの性能解析トレースの場合は、後半の 8 バイトはミリ秒単位を示します。

## 3.3 性能解析トレースファイルの利用方法

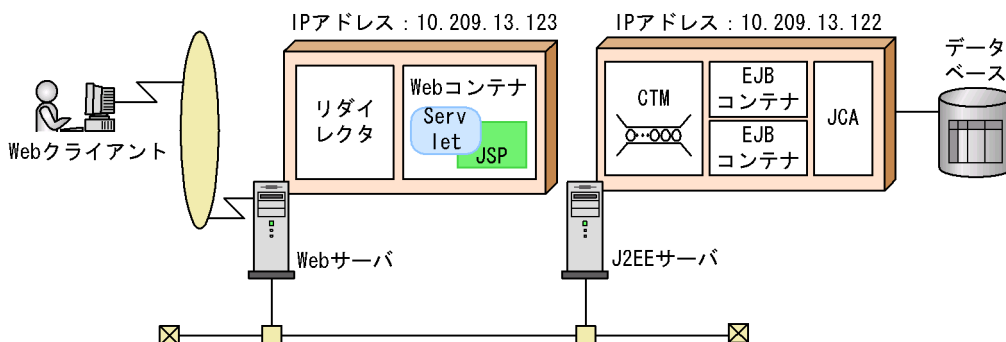
性能解析トレースファイルを使用して性能を解析する場合、Microsoft Excel など、CSV 形式のファイルを編集できるアプリケーションプログラムで表示して、目的に合わせてフィルタリングや並べ替えの機能を利用します。

性能解析トレースファイルの利用方法について、例を使用して説明します。ここでは、次の利用方法について説明します。

- Web サーバのレスポンスタイムの解析
- アプリケーションサーバ内でのリクエストの処理状況の調査
- セッションのライフサイクルの調査
- タイムアウトが発生したトランザクションの特定
- タイムアウトが発生したリクエストの特定
- ルートアプリケーション情報を利用したログ調査
- トラブルが発生したコネクションの特定

なお、ここでは、次のような Web クライアント構成の環境で性能解析トレースファイルを収集したことを前提とします。

図 3-1 Web クライアント構成の例



### 3.3.1 Web サーバのレスポンスタイムの解析

ここでは、Web サーバがクライアントからリクエストを受けてから返却するまでに掛かった時間を解析する方法を、例を使用して説明します。

リダイレクタに割り当てられたイベント ID 「0x8001」 および 「0x8101」 をキーにして、Web サーバで収集した性能解析トレースファイルをフィルタリングします。フィルタリングに使用するイベント ID が示すトレース取得ポイントは次のとおりです。



表 3-5 フィルタリングに使用するイベント ID が示すトレース取得ポイント

イベント ID	トレース取得ポイント
0x8001	Web コンテナへのリクエストヘッダ情報の送信
0x8101	Web コンテナからのレスポンス完了通知の受信

イベント ID 「0x8001」および「0x8101」をキーにして、性能解析トレースをフィルタリングした例を次に示します。

図 3-2 性能解析トレースファイルをフィルタリングした例

1	Event	Date	Time	Time(msec)	Client AP CommNo.	F
686	0x8101	2004/2/5	16:31:27	668/000/000	0x00000000000000c4	1
690	0x8001	2004/2/5	16:32:31	489/000/000	0x00000000000000c7	1
708	0x8101	2004/2/5	16:32:38	379/000/000	0x00000000000000c7	1
712	0x8001	2004/2/5	16:32:45	550/000/000	0x00000000000000cc	1
718	0x8101	2004/2/5	16:32:46	561/000/000	0x00000000000000cc	1
722	0x8001	2004/2/5	16:32:49	015/000/000	0x00000000000000cd	1

「0x8001」および「0x8101」のトレース取得時刻から、リクエストごとのレスポンスタイムを解析できます。図 3-2 の例では、クライアントアプリケーション情報の通信番号が「0x00000000000000c7」と「0x00000000000000cc」のリクエストのレスポンスタイムを比較すると、「0x00000000000000c7」のリクエストの方が処理に時間が掛かっていることがわかります。

### 3.3.2 アプリケーションサーバ内でのリクエストの処理状況の調査

ここでは、Web サーバがクライアントから受けたリクエストがアプリケーションサーバで処理されているかどうか調査する方法を、例を使用して説明します。

次に示す Web サーバで収集した性能解析トレースファイルから、2004/2/5 16:32:31 にリクエストを受け付けたことがわかります。

### 3. システムの処理性能の解析

図 3-3 Web サーバで収集した性能解析トレースファイルの例

	6	7	8	14	15	16
1	Event	Date	Time	RootAP IP	RootAF	RootAP CommNo.
686	0x8101	2004/2/5	16:31:27	10.209.13.123	2200	0x00000000000000c4
690	0x8001	2004/2/5	16:32:31	10.209.13.123	2200	0x00000000000000c7
708	0x8101	2004/2/5	16:32:38	10.209.13.123	2200	0x00000000000000c7
712	0x8001	2004/2/5	16:32:45	10.209.13.123	2200	0x00000000000000cc
718	0x8101	2004/2/5	16:32:46	10.209.13.123	2200	0x00000000000000cc
722	0x8001	2004/2/5	16:32:49	10.209.13.123	2200	0x00000000000000cd

このリクエストのルートアプリケーション情報をキーにして、アプリケーションサーバで収集した性能解析トレースファイルを調査します。

「RootAP IP : 10.209.13.123」, 「RootAP ID : 2200」, および「RootAP CommNo. : 0x00000000000000c7」をキーにして、アプリケーションサーバで収集した性能解析トレースファイルをフィルタリングした例を次に示します。

図 3-4 アプリケーションサーバで収集した性能解析トレースファイルをフィルタリングした例

	6	7	8	14	15	16
1	Even	Date	Time	RootAP IP	RootAP	RootAP CommNo.
538	0x1401	2004/2/5	16:32:33	10.209.13.123	2200	0x00000000000000c7 *
539	0x1402	2004/2/5	16:32:34	10.209.13.123	2200	0x00000000000000c7 *
540	0x2101	2004/2/5	16:32:34	10.209.13.123	2200	0x00000000000000c7 1 *
541	0x1301	2004/2/5	16:32:34	10.209.13.123	2200	0x00000000000000c7 *
542	0x1403	2004/2/5	16:32:34	10.209.13.123	2200	0x00000000000000c7 *
543	0x3000	2004/2/5	16:32:34	10.209.13.123	2200	0x00000000000000c7 *
544	0x3001	2004/2/5	16:32:34	10.209.13.123	2200	0x00000000000000c7 *
545	0x1404	2004/2/5	16:32:34	10.209.13.123	2200	0x00000000000000c7 1 *
546	0x8e05	2004/2/5	16:32:34	10.209.13.123	2200	0x00000000000000c7 *
547	0x8405	2004/2/5	16:32:34	10.209.13.123	2200	0x00000000000000c7 *
548	0x8c00	2004/2/5	16:32:34	10.209.13.123	2200	0x00000000000000c7 *
549	0x8c01	2004/2/5	16:32:36	10.209.13.123	2200	0x00000000000000c7 *
550	0x8c20	2004/2/5	16:32:37	10.209.13.123	2200	0x00000000000000c7 *
551	0x8c21	2004/2/5	16:32:37	10.209.13.123	2200	0x00000000000000c7 *
552	0x8406	2004/2/5	16:32:37	10.209.13.123	2200	0x00000000000000c7 *
553	0x8e06	2004/2/5	16:32:37	10.209.13.123	2200	0x00000000000000c7 *
554	0x1405	2004/2/5	16:32:37	10.209.13.123	2200	0x00000000000000c7 *
555	0x1406	2004/2/5	16:32:37	10.209.13.123	2200	0x00000000000000c7 1 *
556	0x1302	2004/2/5	16:32:37	10.209.13.123	2200	0x00000000000000c7 1 *
557	0x2102	2004/2/5	16:32:37	10.209.13.123	2200	0x00000000000000c7 *
558	0x2103	2004/2/5	16:32:37	10.209.13.123	2200	0x00000000000000c7 1 *
559	0x2104	2004/2/5	16:32:37	10.209.13.123	2200	0x00000000000000c7 *
3538						

このようにアプリケーションサーバで収集した性能解析トレースファイルを、ルートアプリケーション情報をキーにフィルタリングすると、アプリケーションサーバ内でのリクエストの一連の処理を調査できます。

### 3.3.3 セッションのライフサイクルの調査

ここでは、性能解析トレースを利用したセッションのライフサイクルの調査方法について説明します。

セッションのライフサイクルは、PRFトレース取得レベルを「詳細」に設定している場合に出力されるトレース（セッショントレース）を利用して調査できます。

セッショントレース情報は、出力された性能解析トレースファイルのセッション ID またはグローバルセッション ID（セッションフェイルオーバー機能を使用している場合）をキーにして調査できます。

セッション ID は、セッショントレース情報として出力されたイベントのオプション領域に、次の形式で出力されます。

```
セッション文字数:セッションID
```

例えば、「abc123」というセッション ID の場合、オプション領域には「6:abc123」と出力されます。セッション ID が空文字の場合は、セッション文字数として「0:」と出力されます。セッション ID を取得できなかった場合は、何も出力されません。

グローバルセッション ID の場合も同様の形式で出力されます。

また、特定のイベントのセッション ID は、リクエストヘッダの Cookie または URL 書き換えをした場合の URL から取得されます。どちらから取得したのかは、セッション ID の末尾に出力された文字から確認できます。Cookie から取得した場合は、これらのイベントで出力されるセッション ID に続いて「C」が出力されます。URL から取得した場合は、セッション ID に続いて「U」が出力されます。セッション ID の取得元が確認できるイベントを、次に示します。

- 0x8000（リクエスト処理要求取得後）
- 0x8002（Web コンテナから HTTP リクエストボディ要求受信）
- 0x8102（Web コンテナへの HTTP リクエストボディ送信受信）
- 0x8200（Web サーバ経由のリクエスト取得およびリクエストヘッダ解析完了）

セッショントレース情報が出力されている性能解析トレースファイルの例を次の図に示します。この例は、一つのセッションの有効期間内での、セッションを生成するリクエスト、セッションを利用するリクエスト、およびセッションを破棄するリクエストのトレース情報を出力したものです。また、ここでは、セッションを生成するリクエスト、セッションを利用するリクエスト、およびセッションを破棄するリクエストを分けて示します。実際はこれら三つの出力例は、続けて出力されます。また、これらの画面では、セッショントレースに関係ない出力項目は表示していません。

### 3. システムの処理性能の解析

図 3-5 セッショントレース情報が出力されている性能解析トレースファイルの例（セッションを生成するリクエスト部分）

	G	W	X	Z	AA
1	Even	INT	OFF	OPT	ASCII
82	0x8000	127.0.0.	/PrfSe:		
83	0x8001	127.0.0.			
84	0x8204	4			
85	0x8304	4		0000000	...Af\$.....
86	0x8204	426			
87	0x8304	426		0000000	...Af\$.....
88	0x8200	GET	/PrfSe:		
89	0x8203	com.hit	/PrfSe:3a00000		
90	0x8203	com.hit	/PrfSe:3a00000		
91	0x8202	Session	/PrfSe:3a00000		
92	0x8208	/PrfSe:	60	33373a3	37:01065ADADD60E0F41AB3E9E2642EAAD7hos
101	0x8302	Session	/PrfSe:0000000	...Af\$.....	37:01065ADADD60E0F41AB3E9E2642E
102	0x8303	com.hit	/PrfSe:0000000	...Af\$.....	37:01065ADADD60E0F41AB3E9E2642E
111	0x8303	com.hit	/PrfSe:0000000	...Af\$.....	37:01065ADADD60E0F41AB3E9E2642E
112	0x8205	197			
113	0x8305	197		0000000	...Af\$.....8
114	0x8205	332			
115	0x8305	332		0000000	...Af\$.....8
116	0x8205	6			
117	0x8003				
118	0x8305	6		0000000	...Af\$.....8
119	0x8300	GET	/PrfSe:0000000		...Af\$.....37:01065ADADD60E0F41AB3E9E2642EAAD7host1:53:000000FF76FF12B501065ADA DD60E0F41AB3E9E2642EAAD7host1...
120	0x8204	4			
121	0x8103			0000000	...Af\$.....837:01065ADADD60E0F41AB3E9E2642E
122	0x8004	324		33373a3	37:01065ADADD60E0F41AB3E9E2642EAAD7hos
123	0x8006	324			
124	0x8106	324		0000000	...Af\$.....8
125	0x8007				
126	0x8107			0000000	...Af\$.....8
127	0x8104	324		0000000	...Af\$.....837:01065ADADD60E0F41AB3E9E2642E
128	0x8101	127.0.0.		0000000	...Af\$.....37:01065ADADD60E0F41AB3E9E2642E
129	0x8100	127.0.0.	/PrfSe:0000000	...Af\$.....	37:01065ADADD60E0F41AB3E9E2642E

1. イベント「0x8208」の ASCII の列に、生成されたセッション ID が出力されます。オペレーション列の 60 は、セッションの有効期間を示します。
2. イベント「0x8300」の ASCII の列に、生成されたグローバルセッション ID が出力されます。

図 3-6 セッショントレース情報が出力されている性能解析トレースファイルの例（セッションを利用するリクエスト部分）

	G	W	X	Z	AA
1	Even	INT	OPR	OF	ASCII
138	0x8000	127.0.0.	/PrfSessionTrace/se	33373	37:01065ADADD60E0F41 AB3E9E26 42EAAD7host1:C:
139	0x8001	127.0.0.		33373	37:01065ADADD60E0F41 AB3E9E26 42EAAD7host1:C:
140	0x8304	4		00000	...A\$.....8
141	0x8204	538			
142	0x8304	538		00000	...A\$.....
143	0x8200	GET	/PrfSessionTrace/se	33373	37:01065ADADD60E0F41 AB3E9E264
144	0x8203	com.hit	/PrfSessionTrace	33373	37:01065ADADD60E0F41 AB3E9E264
153	0x8203	com.hit	/PrfSessionTrace	33373	37:01065ADADD60E0F41 AB3E9E264
154	0x8202	Normal	/PrfSessionTrace	33373	37:01065ADADD60E0F41 AB3E9E264
155	0x8302	Normal	/PrfSessionTrace	00000	...A\$.....37:01065ADADD60E0F41 AE
156	0x8303	com.hit	/PrfSessionTrace	00000	...A\$.....37:01065ADADD60E0F41 AE
166	0x8205	29			
167	0x8305	29		00000	...A\$.....y.
168	0x8205	419			
169	0x8305	419		00000	...A\$.....y.
170	0x8205	6			
171	0x8305	6		00000	...A\$.....y.
172	0x8003				...A\$.....37:01065ADADD60E0F41
173	0x8300	GET	/PrfSessionTrace*et	00000	AB3E9E2642EAAD7host1:...
174	0x8204	4			
175	0x8103			00000	...A\$.....y.
176	0x8004	411			
177	0x8006	411			
178	0x8106	411		00000	...A\$.....y.
179	0x8007				
180	0x8107			00000	...A\$.....y.
181	0x8104	411		00000	...A\$.....y.
182	0x8101	127.0.0.		00000	...A\$.....
183	0x8100	127.0.0.	/PrfSessionTrace*et	00000	...A\$.....

3. リクエストヘッダの Cookie から取得したセッション ID であることを示しています。

### 3. システムの処理性能の解析

図 3-7 セッショントレース情報が出力されている性能解析トレースファイルの例（セッションを破棄するリクエスト部分）

	G	W	X	Z	AA
1	Even	INT	OPR	OF	ASCII
184	0x8000	127.0.0.	/PrfSessionTrace/st	33373	37:01065ADADD60E0F41 AB3E9E264
185	0x8001	127.0.0.		33373	37:01065ADADD60E0F41 AB3E9E264
186	0x8304	4		0000C	...Af\$.....y.
187	0x8204	549			
188	0x8304	549		0000C	...Af\$.....
189	0x8200	GET	/PrfSessionTrace/st	33373	37:01065ADADD60E0F41 AB3E9E264
190	0x8203	com.hit	/PrfSessionTrace	33373	37:01065ADADD60E0F41 AB3E9E264
199	0x8203	com.hit	/PrfSessionTrace	33373	37:01065ADADD60E0F41 AB3E9E264
200	0x8202	org.apa	/PrfSessionTrace	33373	37:01065ADADD60E0F41 AB3E9E264
201	0x8209	/PrfSes	1097210000000.00	33373	37:01065ADADD60E0F41 AB3E9E264
202	0x8302	org.apa	/PrfSessionTrace	0000C	...Af\$.....
203	0x8303	com.hit	/PrfSessionTrace	0000C	...Af\$.....
212	0x8303	com.hit	/PrfSessionTrace	0000C	...Af\$.....
213	0x8205	107			
214	0x8305	107		0000C	...Af\$.....
215	0x8205	581			
216	0x8305	581		0000C	...Af\$.....
217	0x8205	6			
218	0x8305	6		0000C	...Af\$.....
219	0x8003				
220	0x8300	GET	/PrfSessionTrace*m	0000C	37:01065ADADD60E0F41 AB3E9E264
221	0x8204	4			
222	0x8103			0000C	...Af\$.....
225	0x8004	573			
226	0x8006	573			
227	0x8106	573		0000C	...Af\$.....
228	0x8007				
229	0x8107			0000C	...Af\$.....
230	0x8104	573		0000C	...Af\$.....
231	0x8101	127.0.0.		0000C	...Af\$.....
232	0x8100	127.0.0.	/PrfSessionTrace*m	0000C	...Af\$.....

4. イベント「0x8209」の ASCII 列に、破棄したセッションのセッション ID が出力されます。オペレーション列は、破棄したセッションの生成時刻を示します。
5. グローバルセッションが破棄されていることを示します。破棄されたグローバルセッション ID は出力されません。

このトレースファイルで確認できることについて説明します。

#### セッションの有効期間の確認

セッション生成時には、イベント「0x8208」が出力されます。セッション破棄時には、イベント「0x8209」が出力されます。これらの取得時刻から、セッションの有効範囲を確認できます。また、イベント「0x8208」では、オペレーション名から生成されたセッションの有効期間（秒）も確認できます。

セッション ID は、セッションを破棄するリクエスト内のイベント「0x8209」でセッションが破棄されるまで、同じ ID が出力されます。なお、イベント「0x8209」のオペレーション名から破棄されたセッションの生成時刻も確認できます。

#### グローバルセッションの有効期間の確認

グローバルセッションの有効期間は、出力されたグローバルセッション ID を基に確認できます。グローバルセッション ID とは、セッションフェイルオーバー機能を使用している場合に、冗長化された J2EE サーバ間で引き継がれたセッションが一意であることを示すための ID です。詳細は、マニュアル「Cosminexus 機能解説」のセッションフェイルオーバーによるセッション情報の引き継ぎに関する説明を参照してください。

グローバルセッション ID が生成されている場合、特定のトレース取得ポイントでセッション ID に続けてグローバルセッション ID が出力されます。

図 3-5 の場合、セッションを生成するリクエスト内のイベント「0x8300」のタイミングでグローバルセッション ID が生成されて、その ID が出力されています。グローバルセッション ID は、セッションを破棄するリクエスト内のイベント「0x8209」でリクエスト処理が完了するまで、同じ ID が出力されます。

### 3.3.4 タイムアウトが発生したトランザクションの特定

ここでは、J2EE アプリケーションまたはバッチアプリケーションのトランザクションがタイムアウトした場合に、性能解析トレースの情報を利用してタイムアウトしたトランザクションを特定する方法について説明します。

ここでは、次の 2 種類の方法について説明します。

メッセージを利用した特定方法

性能解析トレースファイルの出力内容による特定方法

#### (1) メッセージを利用した特定方法

トランザクションでタイムアウトが発生した場合、メッセージ KDJE31002-W、および KDJE50080-W が出力されます。これらのメッセージには、次の情報が出力されます。

KDJE31002-W

- トランザクションを開始した J2EE アプリケーション名、またはバッチアプリケーションのクラス名
- トランザクションを開始した J2EE コンポーネント（Enterprise Bean、サーレットまたは JSP）のクラス名とインスタンスのハッシュコード

### 3. システムの処理性能の解析

- 保守情報
- 性能解析トレースのルートアプリケーション情報

メッセージに出力された性能解析トレースのルートアプリケーション情報を、性能解析トレースファイルに出力されたルートアプリケーション情報と突き合わせて確認することで、性能解析トレース中のどこでトランザクションタイムアウトが発生したかを確認できます。

また、その前後のトレース情報を確認して、タイムアウトが発生したトランザクションでの処理内容を確認できます。

KDJE50080-W

- IP アドレス、プロセス ID、ルートアプリケーション情報
- SQL を実行したコネクションのコネクション ID
- SQL を実行したメソッド名
- SQL を実行したメソッドが実行中かどうか
- 最後に実行された SQL

メッセージに出力された SQL の情報を参照することで、タイムアウトが発生した原因となった個所を調査できます。

なお、メッセージの詳細については、マニュアル「Cosminexus メッセージ 1 KAWS / KDAL / KDJE 編」を参照してください。

#### (2) 性能解析トレースファイルの出力内容による特定方法

性能解析トレースファイルには、トランザクションタイムアウトが発生する直前と直後のトレース取得ポイントで、トレース情報が出力されています。

次のイベント ID の処理内容を確認してください。

表 3-6 トランザクションタイムアウト時に出力される性能解析トレース

イベント ID	説明
0x8819	トランザクションタイムアウトの直前の処理で出力される情報です。インタフェース名に、タイムアウトするトランザクションのルートアプリケーション情報が出力されます。
0x8820	トランザクションタイムアウトの直後の処理で出力される情報です。
0x8C41	障害調査用 SQL が出力されたときに出力される情報です。

### 3.3.5 タイムアウトが発生したリクエストの特定

ここでは、リダイレクタでのレスポンス受信時にタイムアウトが発生した場合に、性能解析トレースを利用してタイムアウトが発生したリクエストを特定する方法について説明します。

リダイレクタでのレスポンス受信時にタイムアウトが発生した場合、メッセージ KDJE41019-E が出力されます。このメッセージには、次の情報が出力されます。



- リクエストの URI
- Web サーバとの通信に使用する Web コンテナが動作するホストの IP アドレス
- Web サーバとの通信に使用する Web コンテナのポート番号
- Web サーバとの通信に使用する Web コンテナのファイルディスクリプタ
- 性能解析トレースのルートアプリケーション情報

メッセージに出力された性能解析トレースのルートアプリケーション情報を、性能解析トレースファイルに出力されたルートアプリケーション情報と突き合わせて確認することで、性能解析トレース中のどこでリクエストのタイムアウトが発生したかを確認できます。

タイムアウトが発生したリクエストは、該当するルートアプリケーション情報を出力しているリクエスト処理内の、次の表に示すイベント ID のトレース情報で確認できます。オペレーション名に出力されている URI で、該当するリクエストを特定してください。

表 3-7 タイムアウトしたリクエストを確認するために使用できる性能解析トレース

イベント ID	説明
0x8000	リクエスト処理要求の取得後に出力された情報です。 オペレーション名として、リクエストの URI が出力されます。
0x8100	リダイレクタ処理の完了後に出力された情報です。 オペレーション名として、リクエストの URI が出力されます。

### 3.3.6 ルートアプリケーション情報を利用したログ調査

J2EE アプリケーションまたはバッチアプリケーション内で Cosminexus の API を使用すると、性能解析情報のルートアプリケーション情報の文字列表現を任意のタイミングでログファイルに出力できます。

ルートアプリケーション情報の文字列表現は、次の形式で出力されます（最大 48 文字）。

IPアドレス/プロセスID/通信番号  
 (例: 10.209.15.130/1234/0x0000000000000001)

API を使用してログファイルにルートアプリケーション情報を出力すると、ログファイルと性能解析トレースファイルを突き合わせた調査ができるようになります。

Web コンテナでは、新しいリクエストを受け付けた時点で、リクエスト単位で新しいルートアプリケーション情報が割り当てられます。新しいルートアプリケーション情報が割り当てられるトレースポイントを次に示します。

表 3-8 新しいルートアプリケーション情報が割り当てられるトレースポイント

条件	イベント ID (処理内容)	処理内容
インプロセス HTTP サーバを使用している場合	0x8211	リクエスト取得時, リクエストヘッダ解析完了時
Web サーバと連携している場合	0x8200	リクエスト取得時, リクエストヘッダ解析完了時

注 Web サーバと連携している場合, リダイレクタで取得したルートアプリケーション情報が割り当てられます。リダイレクタまたはリダイレクタが使用する PRF デーモンに異常が発生して, ルートアプリケーション情報が発行されなかった場合, Web コンテナで新しいルートアプリケーション情報が割り当てられます。

また, 次のトレース取得ポイントでは, 「IP アドレス / プロセス ID / 通信番号」が「0.0.0. 0/0/0x0000000000000000」と出力されることがあります。

0x8212

インプロセス HTTP サーバ使用時の Web クライアントからのデータ読み込み開始時

0x8312

インプロセス HTTP サーバ使用時の Web クライアントへのデータ書き込み開始時

「IP アドレス / プロセス ID / 通信番号」が「0.0.0. 0/0/0x0000000000000000」と出力されるのは, 次の場合です。

HTTP リクエストヘッダを受信した場合

HTTP リクエストではない, 不正なデータを受信した場合

リクエスト処理中に例外が発生した場合

なお, ルートアプリケーション情報の文字列表現での出力に使用する API は, com.hitachi.software.ejb.application.prf パッケージで提供されている, CprfTrace クラスの getRootApInfo メソッドです。

J2EE アプリケーションまたはバッチアプリケーション開発時のルートアプリケーション情報の取得の実装については, マニュアル「Cosminexus 機能解説」の, 性能解析トレースのルートアプリケーション情報の取得のための実装に関する説明を参照してください。API の詳細については, マニュアル「Cosminexus リファレンス API 編」を参照してください。

### 3.3.7 トラブルが発生したコネクションの特定

データベースとして HiRDB または Oracle を使用している場合, 性能解析トレースファイルに出力されたコネクション ID と, Cosminexus DABroker Library や HiRDB クライアントおよび Oracle クライアントのログファイルやトレースファイルに出力されたコ

ネクション ID を突き合わせて確認することで、トラブルが発生したコネクションを確認できます。確認方法については、「付録 C データベースと接続中にトラブルが発生したコネクションの特定」を参照してください。

### 3.3.8 性能解析トレースファイルとスレッドダンプを対応づけた問題個所の調査

J2EE アプリケーションまたはバッチアプリケーションでスローダウンやハングアップが発生したときに、性能解析トレースファイルとスレッドダンプを対応づけることで、問題が発生した個所を調査できます。

性能解析トレースファイルとスレッドダンプの対応づけには、性能解析トレースファイルに出力されたスレッド ID と、スレッドダンプに出力されたスレッド情報の nativeID (OS レベルのスレッド ID) を使用します。ここでは、性能解析トレースファイルから対応するスレッドダンプを特定する方法について説明します。

性能解析トレースファイルから対応するスレッドダンプを特定する手順を次に示します。

1. 性能解析トレースファイル、およびスレッドダンプを収集します。  
性能解析トレースファイルの収集方法については、「3.2.1 性能解析トレースファイルの収集方法」を参照してください。スレッドダンプの収集方法については、「5.4.5 JavaVM のスレッドダンプの取得」を参照してください。
2. 使用する性能解析トレースファイル、およびスレッドダンプを選びます。  
性能解析トレースファイルとスレッドダンプが出力された時刻を基に、調査に使用する性能解析トレースファイル、およびスレッドダンプを選びます。出力された時刻は、次の情報を参考にしてください。

性能解析トレースファイル

「Time」および「Time(msec/usec/nsec)」

性能解析トレースファイルの「Time」および「Time(msec/usec/nsec)」を次の図に示します。

	C	D	E	F	G	H	I
1	Thread(hashcode)	Trace	ProcessName	Event	Date	Time	Time(msec/usec/nsec)
2999	4736(7719486)	132	MyServer	0x8e04	2006/7/21	19:22:37	168/000/000
3000	4388(348051)	27	MyServer	0x8e05	2006/7/21	19:22:37	184/000/000
3001	4388(348051)	28	MyServer	0x8e06	2006/7/21	19:22:37	184/000/000
3002	4388(348051)	29	MyServer	0x8e05	2006/7/21	19:22:37	184/000/000

「Time」および「Time(msec/usec/nsec)」

スレッドダンプ

ファイル名、およびファイルの末尾に出力される時刻  
ファイルの末尾に出力される時刻の例を次に示します。

### 3. システムの処理性能の解析

```
⋮  
⋮  
Full thread dump completed.   Fri Jul 21 19:22:47 2006
```

3. 性能解析トレースファイルの調査したいイベント ID に対応する「Thread (hashcode)」のスレッド ID (10 進数) を 16 進数に変換します。

	C	D	E	F	G	H	I
1	Thread(hashcode)	Trace	ProcessName	Event	Date	Time	Time(msec/usec/nsec)
2999	4736(7719486)	132	MyServer	0x8e04	2006/7/21	19:22:37	168/000/000
3000	4388(348051)	27	MyServer	0x8e05	2006/7/21	19:22:37	184/000/000
3001	4388(348051)	28	MyServer	0x8e06	2006/7/21	19:22:37	184/000/000
3002	4388(348051)	29	MyServer	0x8e05	2006/7/21	19:22:37	184/000/000



4388 (10進数) =1124 (16進数)

4. スレッドダンプの「nid」が手順 .3 で 16 進数に変換したスレッド ID と一致するスレッド情報を探して、対応するスレッド情報を特定します。

```
⋮  
⋮  
"VBJ ThreadPool Worker" daemon prio=5 jid=0x00054f93 tid=0x04cef380 nid=0x1124  
in Object.wait() [0x0632f000..0x0632fd18]  
  stack=[0x06330000..0x062f5000..0x062f1000..0x062f0000]  
  [user cpu time=0ms, kernel cpu time=15ms] [blocked count=1, waited count=29]  
at java.lang.Object.wait(Native Method)  
⋮  
⋮
```

## 4

## J2EE アプリケーションとリソースの管理

この章では、J2EE サーバ上で動作する J2EE アプリケーション、および JavaBeans リソースの管理について説明します。また、クラスタ構成のデータベースを管理する方法について説明します。

J2EE アプリケーションの管理では、運用状況に合わせて J2EE アプリケーションの設定を変更したり、サービスの提供方法やシステムのメンテナンスに応じて J2EE アプリケーションを開始・停止したり、入れ替えたりします。リクエスト処理が終了しない場合には、必要に応じてキャンセルもできます。

- 
- 4.1 J2EE アプリケーションとリソースの管理の概要

---

  - 4.2 Web アプリケーションの最大同時実行スレッド数の動的変更

---

  - 4.3 CTM のスケジュールキューの稼働状況の確認と最大同時実行数の変更

---

  - 4.4 J2EE アプリケーションの実行時間の監視とキャンセル

---

  - 4.5 J2EE アプリケーションの閉塞と停止

---

  - 4.6 J2EE アプリケーションの入れ替えと保守

---

  - 4.7 JavaBeans リソースの入れ替え

---

  - 4.8 クラスタコネクションプールの運用
-

## 4.1 J2EE アプリケーションとリソースの管理の概要

J2EE アプリケーションの管理では、運用状況に合わせて J2EE アプリケーションの設定を変更したり、サービスの提供方法やシステムのメンテナンスに応じて J2EE アプリケーションを開始、停止したり、入れ替えたりします。

J2EE アプリケーションの管理の作業について、次の表に示します。

表 4-1 アクセス状況に応じた Web アプリケーションの最大同時実行数の動的変更

作業内容	手段	作業概要	参照先
Web アプリケーションの稼働状況の確認	運用管理コマンド (mngsvrutil)	その時点の Web アプリケーションの稼働状況を確認します。	4.2.1
Web アプリケーションの最大同時実行スレッド数の設定変更	運用管理コマンド (mngsvrutil)	Web アプリケーションの同時実行スレッド数を動的に変更します。	4.2.2

表 4-2 CTM のスケジュールキューの稼働状況の確認と同時実行数の変更

作業内容	手段	作業概要	参照先
CTM のスケジュールキューの稼働状況の確認	運用管理コマンド (mngsvrutil)	CTM のキューに接続された J2EE アプリケーションに関する情報を確認します。	4.3.1
CTM のスケジュールキューの同時実行数の変更	CTM のコマンド (ctmchpara)	J2EE アプリケーション (Enterprise Bean) の同時実行数をスケジュールキュー単位で動的に変更します。	4.3.2

表 4-3 J2EE アプリケーションの実行時間の監視とキャンセル

作業内容	手段	作業概要	参照先
J2EE アプリケーションの実行状態の確認	サーバ管理コマンド (cjlistthread)	実行中の J2EE アプリケーションの実行状態を確認します。	4.4.1
タイムアウトが発生したリクエストのキャンセル	サーバ管理コマンド (cjstopthread)	J2EE アプリケーション内で実行されているリクエストのうち、終了しないリクエストをキャンセルします。	4.4.2

表 4-4 J2EE アプリケーションの閉塞

作業内容	手段	作業概要	参照先
負荷分散機を使用したサービス閉塞	負荷分散機の機能	Web フロントに負荷分散機を使用している場合に、次の方法でサービスを閉塞します。 <ul style="list-style-type: none"> <li>負荷分散機のリクエストの振り分け先を変更する</li> <li>負荷分散機を停止する</li> <li>サービスを構成する一部の Web アプリケーションだけを停止する</li> </ul>	4.5.3
CTM を利用したサービス閉塞	運用管理コマンド (mngsvrutil)	CTM を使用しているバックシステムで、ホスト内の J2EE アプリケーションを一度に停止したり、キューを共有する J2EE アプリケーションを一度に停止したりする場合、J2EE アプリケーションのスケジュールキューに対して直接閉塞を実行します。	4.5.4

表 4-5 J2EE アプリケーションの停止

作業内容	手段	作業概要	参照先
J2EE アプリケーションの停止	サーバ管理コマンド (cjstopapp)	J2EE アプリケーションを通常停止、または強制停止します。通常停止にタイムアウトを設定して、強制停止を自動で実行することもできます。	4.5.5

表 4-6 J2EE アプリケーションの入れ替えと保守

作業内容	手段	作業概要	参照先
J2EE アプリケーションの通常入れ替え	サーバ管理コマンド (cjstopapp, cjimportapp)	J2EE アプリケーションを停止してから新しいアプリケーションに入れ替えます。入れ替え後、J2EE アプリケーションを再開します。	4.6.1
リデプロイによる J2EE アプリケーションの入れ替え	サーバ管理コマンド (cjreplaceapp)	リデプロイによって J2EE アプリケーションを入れ替えます。アーカイブ形式の J2EE アプリケーションを入れ替える場合に、少ない手順で入れ替えることができます。	4.6.2
リロードによる J2EE アプリケーションの入れ替え	サーバ管理コマンド (cjreloadapp)	リロードによって J2EE アプリケーションを入れ替えます。展開ディレクトリ形式の J2EE アプリケーションを入れ替える場合に、少ない手順で入れ替えることができます。	4.6.3

#### 4. J2EE アプリケーションとリソースの管理

作業内容	手段	作業概要	参照先
JSP を事前にコンパイルしてから実行する J2EE アプリケーションの入れ替え	サーバ管理コマンド (cjstartapp) / cjspec コマンド	JSP 事前コンパイル機能を使用して、JSP を事前にコンパイルしてから J2EE アプリケーションを入れ替えます。	4.6.4
J2EE アプリケーションの名称変更	サーバ管理コマンド (cjrenameapp)	名称を変更して、入れ替え前や入れ替え後の J2EE アプリケーションを管理しやすくします。	4.6.5
CTM を利用したオンライン状態での J2EE アプリケーションの入れ替え	運用管理コマンド (mngsvrutil)	スケジュールキューの出口を閉じたあと、J2EE アプリケーションを入れ替えます。	4.6.6
テストモードでの J2EE アプリケーションの運用	サーバ管理コマンド (cjimportapp, cjstartapp)	本番環境で問題なく J2EE アプリケーションが動作するかを確認するためのテストモードで、J2EE アプリケーションを運用します。	4.6.7

表 4-7 JavaBeans リソースの入れ替え

作業内容	手段	作業概要	参照先
JavaBeans リソースの入れ替え	サーバ管理コマンド (cjdeletejb, cjimportjb)	JavaBeans リソースを入れ替えます。	4.7

表 4-8 クラスタ構成のデータベースのメンテナンス

作業内容	手段	作業概要	参照先
コネクシオンプールの状態の確認	サーバ管理コマンド (cjlistrar)	クラスタ構成のデータベースをメンテナンスする前、およびメンテナンスしたあとに、コネクシオンプールの状態を確認します。	4.8.1
コネクシオンプールの一時停止	サーバ管理コマンド (cjsuspendpool)	コネクシオンプールを一時停止します。	4.8.2
コネクシオンプールの再開	サーバ管理コマンド (cjresumepool)	コネクシオンプールを再開します。	4.8.3

#### ! 注意事項

SFO サーバ上で動作する SFO サーバアプリケーションは、サーバ管理コマンドを利用して直接操作しないでください。



## 4.2 Web アプリケーションの最大同時実行スレッド数の動的変更

---

Cosminexus を使用して構築したシステムでは、Web アプリケーション単位の最大同時実行スレッド数、占有スレッド数、および実行待ちキューサイズを、サービスを停止しないで動的に変更できます。Web アプリケーション単位の最大同時実行スレッド数、占有スレッド数、および実行待ちキューサイズは、運用管理コマンド (mngsvrutil) を使用して変更します。

Web アプリケーション単位の最大同時実行スレッド数を変更すると、次のような局面に対応できます。

Web アプリケーション単位の稼働状態でのパフォーマンスチューニング  
クライアントにサービスを提供しながらパフォーマンスをチューニングできます。

アクセス状況に応じた一時的な Web アプリケーション単位の最大同時実行スレッド数の変更

アクセス状況に応じて、特定の Web アプリケーションの最大同時実行スレッド数を一時的に増加または減少させたい場合に、対処できます。

時間帯に応じた計画的な Web アプリケーション単位の最大同時実行スレッド数の変更  
時間帯に応じて、Web アプリケーションの最大同時実行スレッド数を計画的に増加または減少させたい場合に、対処できます。

なお、ここで設定した項目は、サービスを停止すると無効になり、設定内容は J2EE サーバに保存されません。また、この方法で動的に変更できるのは、Web アプリケーションについての情報だけです。Web コンテナについての設定を変更する場合は、J2EE サーバを再起動しないと有効になりません。

---

### 参考

Web コンテナ単位の最大同時実行スレッド数を変更したい場合、また、Web アプリケーションの最大同時実行スレッド数を一時的ではなく恒常的に変更したい場合は、システムの構築時と同じ手順で設定してください。Web コンテナでの最大同時実行スレッド数の制御の設定については、マニュアル「Cosminexus システム構築ガイド」の同時実行スレッド数の設定に関する説明を参照してください。

---

##### ! 注意事項

Web アプリケーションの最大同時実行スレッド数を変更した場合、URL グループ単位の最大同時実行スレッド数や占有スレッド数との関係によっては、URL グループ単位の同時実行スレッド数制御の動作へ影響が出ることがあります。URL グループ単位の同時実行スレッド数制御への影響については、マニュアル「Cosminexus 機能解説」を参照してください。

Web アプリケーションの最大同時実行スレッド数を動的に変更するための準備と手順を次に示します。

##### 準備

Web アプリケーションの最大同時実行スレッド数の動的変更は、J2EE サーバおよび Web アプリケーションを含む J2EE アプリケーションが起動、開始されている状態で実行します。

J2EE サーバの起動、および J2EE アプリケーションの開始を含むシステムの起動方法については、マニュアル「Cosminexus 簡易構築・運用ガイド」を参照してください。

##### 手順

次の手順で実行します。

1. Web アプリケーションの稼働状況を監視して、最大同時実行スレッド数を変更する必要があるかどうかを確認する（4.2.1 参照）  
運用管理コマンドを使用して実行します。
2. 必要と判断した場合、Web アプリケーションの最大同時実行スレッド数を変更する（4.2.2 参照）  
運用管理コマンドを使用して実行します。
3. Web アプリケーションの稼働状況を確認して、改善されたことを確認する（4.2.1 参照）  
運用管理コマンドを使用して実行します。

### 4.2.1 Web アプリケーションの稼働状況の確認

稼働中の Web アプリケーションの稼働状況を確認します。Web アプリケーションの稼働状況は、運用管理コマンド（mngsvrutil）で確認できます。確認した結果、例えば、次のような場合には、最大同時実行スレッド数を変更することを検討してください。

稼働スレッド数に比べて実行待ちスレッド数が極端に多い状況を想定していない場合に、稼働スレッド数に比べて実行待ちスレッド数が極端に多いとき

実行待ちリクエスト数の現在値が実行待ちリクエスト数の最大値に近づいている状況を想定していない場合に、実行待ちリクエスト数の現在値が、実行待ちリクエスト数の最大値に近づいているとき

実行待ちキューからリクエストがあふれる状況を想定していない場合に、実行待ち

### キューからリクエストがあふれているとき

なお、監視した結果、一時的にはではなく恒常的に最大同時実行スレッド数を変更する必要があると判断した場合は、動的変更をするのではなく、Web アプリケーションを停止して、最大同時実行スレッド数の設定をし直してください。Web アプリケーションを停止した状態での Web コンテナでの最大同時実行スレッド数の制御の設定については、マニュアル「Cosminexus システム構築ガイド」の同時実行スレッド数の設定に関する説明を参照してください。

Web アプリケーションの稼働状況を確認するには、mngsvrutil コマンドに、サブコマンド「get」を指定して実行します。

実行形式および実行例を次に示します。なお、mngsvrutil コマンドの詳細については、マニュアル「Cosminexus リファレンス コマンド編」を参照してください。

#### 実行形式

```
mngsvrutil -m <Management Serverのホスト名> -u <管理ユーザID> -p <管理パスワード> -t <ホスト名> -k host get webApps
```

#### 実行例

```
mngsvrutil -m mnghost -u user01 -p pwl -t host01 -k host get webApps
```

コマンドの実行結果は、標準出力またはファイルに出力されます。

Web アプリケーションの稼働情報のうち、Web アプリケーションの最大同時実行スレッド数を変更する場合に参考になる情報は、次に示すヘッダ情報の項目で確認できます。なお、N 秒とは、運用監視で設定しているサンプリング時間です。

表 4-9 Web アプリケーションの最大同時実行スレッド数を変更する場合に参考になる情報

ヘッダ情報	内容
contextRoot	Web アプリケーションのコンテキストルート。
exclusiveThreadCountUpperBound	Web アプリケーションの占有スレッド数。
activeThreadCountUpperBound	Web アプリケーションの最大同時実行スレッド数。
waitingRequestCountUpperBound	Web アプリケーションの実行待ちキューサイズ。
currentThreadCountUpperBound	Web アプリケーションの同時実行可能スレッド数の上限値。
activeThreadCount	稼働スレッド数の現在値。
activeThreadCountPeak	稼働スレッド数の N 秒ピーク。
activeThreadCountAverage	稼働スレッド数の N 秒平均値。

#### 4. J2EE アプリケーションとリソースの管理

ヘッダ情報	内容
activeThreadCountHighWaterMark	稼働スレッド数の最大値。
activeThreadCountLowWaterMark	稼働スレッド数の最小値。
waitingRequestCount	実行待ちリクエスト数の現在値。
waitingRequestCountPeak	実行待ちリクエスト数の N 秒ピーク。
waitingRequestCountAverage	実行待ちリクエスト数の N 秒平均値。
waitingRequestCountHighWaterMark	実行待ちリクエスト数の最大値。
waitingRequestCountLowWaterMark	実行待ちリクエスト数の最小値。
overflowRequestCount	実行待ちキューからあふれたリクエスト数。

### 4.2.2 Web アプリケーションの最大同時実行スレッド数の設定変更

稼働状況を確認した Web アプリケーションの次の項目を、必要に応じて変更します。

最大同時実行スレッド数

占有スレッド数

実行待ちキューサイズ

これらの項目は、運用管理コマンド (mngsvrutil) を使用して変更できます。ここで設定した値は、Web アプリケーションを停止するまで有効です。

#### ! 注意事項

Web アプリケーションの最大同時実行スレッド数の動的変更実行中 (mngsvrutil コマンドにサブコマンド「change」を指定して実行している間) は、J2EE アプリケーションのデプロイおよびアンデプロイは実行しないでください。

Web アプリケーションの最大同時実行スレッド数を動的に変更するには、mngsvrutil コマンドに、サブコマンド「change」を指定して実行します。

実行形式を次に示します。なお、mngsvrutil コマンドの詳細については、マニュアル「Cosminexus リファレンス コマンド編」を参照してください。

実行形式

```
mngsvrutil -m <Management Serverのホスト名> -u <管理ユーザID> -p <管理パスワード> -t
<ホスト名> -k host change webAppThreadCtrl <Webアプリケーションのコンテキストルート> <
最大同時実行スレッド数>, <占有スレッド数>, <実行待ちキューサイズ>
```

次に、実行例を示します。この例では、次の表に示すように設定を変更します。なお、Web アプリケーションの名称は、「WebAP1」とします。

表 4-10 Web アプリケーション (WebAP1) の最大同時実行スレッド数の動的変更の設定例

設定対象	設定項目	変更前の設定値	変更後の設定値
Web コンテナ	最大同時実行スレッド数	10	10 (変更できません)
Web アプリケーション (WebAP1)	最大同時実行スレッド数	7	8
	占有スレッド数	4	5
	実行待ちキューサイズ	8	10

#### 実行例

```
mngsvrutil -m mnghost -u user01 -p pw1 -t host01 -k host change
webAppThreadCtrl "WebAP1" 8,5,10
```

設定内容は、コマンド実行後、すぐに反映されます。

### 4.2.3 最大同時実行スレッド数を動的に変更したときの Web アプリケーションの動作

ここでは、Web アプリケーションの最大同時実行スレッド数を動的に変更したときの Web アプリケーションの動作について説明します。

#### (1) 最大同時実行スレッドを変更したことによる動作

最大同時実行スレッド数を変更した場合、Web アプリケーションは次のように動作します。

##### 最大同時実行スレッド数を増やした場合

実行待ち状態だったリクエストのうち、実行可能になったリクエストが直ちに実行されます。

例えば、設定変更によって最大同時実行スレッド数を 7 から 8 に変更した場合に、実行待ちキューにリクエストがあったときには、実行待ちキューのリクエストの一つがすぐに実行されることとなります。

##### 最大同時実行スレッド数を減らした場合

実行可能な最大同時実行スレッド数が減少されます。

ただし、最大同時実行スレッド数のスレッドをすべて使用している状態で設定を変更しようとした場合、実行中のスレッド数は減らないため、一時的に最大同時実行スレッド数を超える数のスレッドが実行されます。

例えば、設定変更によって最大同時実行スレッド数を 8 から 7 に変更した場合に、その時点で使用されているスレッド数が 8 個あったときは、一時的に設定変更後の最大同時実行スレッド数を超える 8 個のスレッドが実行されます。

稼働中のスレッドの一つが終了したときにスレッド数が減られ、以降は設定値どおり、最大で 7 個のスレッドが同時実行されるようになります。

### (2) 占有スレッド数を変更したことによる動作

占有スレッド数を変更した場合、Web アプリケーションは次のように動作します。

#### 占有スレッド数を増やした場合

実行待ち状態だったリクエストのうち、該当する Web アプリケーションの占有スレッド数が増えたことで実行可能になったリクエストが、直ちに実行されます。また、アクセスのピーク時など、Web コンテナ単位の最大同時実行スレッド数に設定されているスレッド数をすべて使用している状態で特定の Web アプリケーションの占有スレッド数を増やした場合、その Web アプリケーションの実行待ちキューにリクエストがあるときには、実行待ちキューのリクエストがすぐに実行されます。これによって、一時的に、Web コンテナ単位の最大同時実行スレッド数を超える数のスレッドが実行されることがあります。

例えば、Web コンテナ単位の最大同時実行スレッド数が 10 で、Web アプリケーションである WebAP1 と WebAP2 でそれぞれ 7 個と 3 個のスレッドを使用している状態で WebAP1 の占有スレッド数を 8 に変更した場合、一時的に Web コンテナ単位で 11 個のスレッドが実行されます。

#### 占有スレッド数を減らした場合

特定の Web アプリケーションの占有スレッド数を減らすことで、Web コンテナ単位で共有されるスレッド数が増加します。実行待ちキューのリクエストのうち、共有スレッド数が増えることで実行可能になったスレッドがある場合は、直ちに実行されます。

### (3) 実行待ちキューサイズを変更したことによる動作

実行待ちキューサイズを変更した場合、Web アプリケーションは次のように動作します。

#### 実行待ちキューサイズを増やした場合

実行待ちキューサイズが直ちに増やされます。

#### 実行待ちキューサイズを減らした場合

実行待ちキューで待っているリクエストの数よりも少ない値に実行待ちキューサイズを変更した場合、実行待ちキューサイズを超えるリクエストは HTTP 503 エラーとして返却されます。

## 4.3 CTM のスケジュールキューの稼働状況の確認と最大同時実行数の変更

ここでは、CTM のスケジュールキューの稼働状況を確認する方法、および J2EE アプリケーションの最大同時実行数をスケジュールキュー単位で動的に変更する方法について説明します。J2EE アプリケーションの最大同時実行数の変更では、J2EE アプリケーションの内容によって、同時実行数を一時的に増加させたり減少させたりできます。

CTM のスケジュールキューの稼働状況の確認をするには、「4.3.1 CTM のスケジュールキューの稼働状況の確認」を参照してください。また、J2EE アプリケーションの最大同時実行数を CTM スケジュールキュー単位で動的に変更する場合は、「4.3.2 CTM のスケジュールキューの同時実行数の変更」を参照してください。

### 4.3.1 CTM のスケジュールキューの稼働状況の確認

ここでは、CTM のスケジュールキューの稼働状況を確認する方法について説明します。CTM のスケジュールキューの稼働状況は運用管理コマンド (mngsvrutil) を使用して確認できます。

CTM のスケジュールキューの稼働状況を確認するには、運用管理コマンドのサブコマンド「get」の引数に、「queueApps」を指定して実行します。コマンドを実行すると、J2EE アプリケーション開始時の同時実行数、J2EE アプリケーションに対する現在の常駐スレッド数などの情報が取得できます。

次に、実行形式と実行例を示します。

実行形式

```
mngsvrutil -m <Management Serverのホスト名> -u <管理ユーザID> -p <管理パスワード> -t <論理サーバ名> get queueApps
```

実行例

```
mngsvrutil -m mnghost -u user01 -p pw1 -t myServer get queueApps
```

mngsvrutil コマンド、そのサブコマンド、および取得できる情報の詳細については、マニュアル「Cosminexus リファレンス コマンド編」を参照してください。

### 4.3.2 CTM のスケジュールキューの同時実行数の変更

J2EE アプリケーションの同時実行数をスケジュールキュー単位で動的に変更する方法について説明します。

#### 4. J2EE アプリケーションとリソースの管理

J2EE アプリケーションの最大同時実行数を CTM スケジュールキュー単位で動的に変更する作業は次の流れで行います。

1. 現在の CTM のスケジュールキューの同時実行数を確認する  
CTM のコマンド (ctmlsqe) を使用して実行します (「(1) CTM のスケジュールキューの稼働状況の確認」参照)。
2. CTM のスケジュールキューの同時実行数を変更する  
CTM のコマンド (ctmchpara) を使用して実行します (「(2) CTM のスケジュールキューの同時実行数の変更」参照)。
3. 変更後の CTM のスケジュールキューの同時実行数を確認する  
CTM のコマンド (ctmlsqe) を使用して実行します (「(1) CTM のスケジュールキューの稼働状況の確認」参照)。

なお、CTM のスケジュールキューの同時実行数の変更は、スケジュールキューの状態が次の場合に実行できます。

- A : スケジューリング可能状態
- H : スケジューリング閉塞中
- C : スケジューリング可能閉塞

#### (1) CTM のスケジュールキューの稼働状況の確認

CTM のスケジュールキューの稼働状況を確認するには、ctmlsqe コマンドの引数に「-CTMAppInfo」を指定して実行します。このコマンドを実行すると、スケジュールキューを共有している J2EE アプリケーションの情報を確認できます。実行形式と実行例を次に示します。

実行形式

```
ctmlsqe -CTMDomain <CTMドメイン名称> -CTMID <CTM識別子> -CTMAppInfo
```

実行例

```
ctmlsqe -CTMDomain domain01 -CTMID CTM01 -CTMAppInfo
```

ctmlsqe コマンドの詳細、および出力される情報の詳細については、マニュアル「Cosminexus リファレンス コマンド編」を参照してください。

#### (2) CTM のスケジュールキューの同時実行数の変更

CTM のスケジュールキューの同時実行数を変更するには、ctmchpara コマンドを実行します。実行形式と実行例を次に示します。

実行形式



```
ctmchpara -CTMDomain <CTMドメイン名称> -CTMID <CTM識別子> -CTMQueue <スケジュール  
キュー登録名称> -CTMChangeCount <同時実行数>
```

#### 実行例

```
ctmchpara -CTMDomain domain01 -CTMID CTM01-CTMQueue que01  
-CTMChangeCount 10
```

実行後、変更が反映されていることを確認してください。スケジュールキューの状態を確認する方法については、「(1) CTM のスケジュールキューの稼働状況の確認」を参照してください。

ctmchpara コマンドの詳細、および出力される情報の詳細については、マニュアル「Cosminexus リファレンス コマンド編」を参照してください。

## 4.4 J2EE アプリケーションの実行時間の監視とキャンセル

---

J2EE アプリケーションの内部で無限ループなどが発生すると、J2EE アプリケーションの動作を制御できなくなります。このようなトラブルへの対策として、J2EE アプリケーションの実行時間にタイムアウト値を設定しておき、一定時間を過ぎたら処理が完了しなくても制御が戻るように設定する方法があります。

Cosminexus で構築したシステムでは、次の 2 種類の機能が使用できます。

メソッドタイムアウト機能

メソッドキャンセル機能

これらの機能では、J2EE アプリケーション内のメソッド単位にタイムアウト時間を設定して監視し、一定時間内に終了しなかったリクエストのタイムアウトを検知できます（メソッドタイムアウト機能）。また、タイムアウトが発生した場合、強制的にメソッドをキャンセルできます（メソッドキャンセル機能）。このとき、トランザクションも強制決着されます。メソッドキャンセルは、自動的に実行されるように設定できます。

J2EE アプリケーションの実行時間を監視してタイムアウトが発生したリクエストを自動的にキャンセルするための設定は、サーバ管理コマンドを使用して実行します。また、J2EE アプリケーションの実行時間を監視するかどうか、および監視時間間隔は、J2EE サーバの動作設定のカスタマイズで設定します。J2EE アプリケーション実行時間の監視の設定については、マニュアル「Cosminexus システム構築ガイド」を参照してください。

また、J2EE アプリケーションでメソッドタイムアウトが発生した場合や、メソッドキャンセルに失敗したスレッドが存在する場合は、サーバ管理コマンドを使用して、手動でメソッドキャンセルを実行（再実行）する必要があります。

サーバ管理コマンドによってメソッドキャンセルを実行する手順を次に示します。

手順

タイムアウトが発生したことを示すメッセージ、またはスレッドの状態が不正なことを示すメッセージを受け取ったあとで、次の手順で実行します。

1. J2EE アプリケーションの実行状態を確認する（4.4.1 参照）  
サーバ管理コマンドを使用して実行します。
2. タイムアウトが発生している場合、J2EE アプリケーションで実行しているリクエストをキャンセルする（4.4.2 参照）  
サーバ管理コマンドを使用して実行します。

なお、J2EE アプリケーションを強制的に停止した場合も、メソッドはキャンセルされません。J2EE アプリケーションの強制停止についての詳細は、「4.5.5 J2EE アプリケー

ションの停止」を参照してください。

#### 4.4.1 J2EE アプリケーションの実行状態の確認

J2EE アプリケーション内のメソッドでタイムアウトが発生したことを示すメッセージが出力された場合、またはメソッドキャンセルに失敗したことによってスレッドの状態が不正になったことを示すメッセージが出力された場合に、実行中の J2EE アプリケーションの実行状態を確認します。確認結果によって、メソッドキャンセルを実行できるかどうかを判断します。

実行状態の確認には、サーバ管理コマンド (cjlistthread) を使用します。

なお、このコマンドは、スレッドの状態が遷移した時に取得したスタックトレースを確認する場合にも使用できます。ただし、スレッドの状態が「running」の場合は、スタックトレースは出力されません。また、スレッドの状態が「stopping」の場合は、複数回スタックトレースが出力されるため、遷移した時点のスタックトレースではなく、最新のスタックトレースが出力されます。

実行形式と実行例を次に示します。なお、cjlistthread コマンドの詳細については、マニュアル「Cosminexus リファレンス コマンド編」を参照してください。

##### 実行形式

```
cjlistthread <J2EEサーバ名>
```

-detail オプションを指定すると、動作中のスレッド情報の詳細として、スタックトレースも出力できます。

```
cjlistthread <J2EEサーバ名> -detail
```

##### 実行例

```
cjlistthread MyServer -detail
```

-detail オプションを指定した場合の実行結果の形式を次に示します。

##### 実行結果

#### 4. J2EE アプリケーションとリソースの管理

```
Current Time=HH:MM:SS
ThreadID=11111,RootApInfo=RootAP1,Status=timeout,AppName=AP1,StartTime=HH:MM:SS,TimeOut=60
  com.hitachi.XXXX
    at com.hitachi.YYYY
    at user.code.UserClass1
    at com.hitachi.ZZZZ
    .
    .
ThreadID=22222,RootApInfo=RootAP2,Status=stopping,AppName=AP2,StartTime=HH:MM:SS,TimeOut=60
  com.hitachi.xxxx
    at com.hitachi.yyyy
    at user.code.UserClass2
    at com.hitachi.zzzz
```

実行結果の、「Status=」の後ろに出力されているのが、スレッドの状態です。

スレッドの状態とメソッドキャンセルの実行可否を、次の表に示します。

表 4-11 スレッドの状態とメソッドキャンセルの実行可否

スレッドの状態	意味	メソッドキャンセルの実行可否
running	実行時間が監視されている状態です。正常に動作しています。	実行できます。
timeout	メソッドタイムアウトによって、タイムアウトが検知された状態です。	実行できます。
stopping	メソッドキャンセル処理が実行されている状態です。	実行できません。
stopped	メソッドキャンセル処理の完了待ちの状態です。	実行できません。
failed	メソッドキャンセル処理が失敗した状態です。	実行できます。

ただし、スレッドの状態がメソッドキャンセルを実行できる状態の場合でも、そのメソッドが保護区で実行されている場合は、メソッドはキャンセルできません。保護区については、マニュアル「Cosminexus 機能解説」のメソッドキャンセルに関する説明を参照してください。

#### ポイント

このコマンドで出力されるスレッドの情報は、cjisthread コマンドを実行した時点でのスナップショットです。このため、刻々と変化します。正確なスレッドの状態を取得するためには、複数回コマンドを実行して確認することをお勧めします。

また、-detail オプションを指定すると、実行結果の出力量が多くなります。-detail オプションを指定する場合は、出力先をファイルにすることをお勧めします。

## 4.4.2 タイムアウトが発生したリクエストのキャンセル

J2EE アプリケーションでタイムアウトが発生したリクエストをキャンセルします。この処理をメソッドキャンセルといいます。メソッドキャンセルは、「4.4.1 J2EE アプリケーションの実行状態の確認」の内容に従って `cjlistthread` コマンドを実行してスレッドの実行状態を確認した結果、メソッドキャンセルが可能な状態（`running`、`timeout` または `failed`）だった場合に実行してください。

メソッドキャンセルには、サーバ管理コマンド（`cjstopthread`）を使用します。一度のコマンド実行で、複数のスレッドのメソッドをキャンセルできます。

実行形式と実行例を次に示します。なお、`cjstopthread` コマンドの詳細については、マニュアル「Cosminexus リファレンス コマンド編」を参照してください。

### 実行形式

```
cjstopthread <J2EEサーバ名> -tid <スレッドID> [-tid <スレッドID>...]
```

### 実行例

```
cjstopthread MyServer -tid 11111
```

`-tid` オプションのオプション引数の `<スレッド ID>` には、タイムアウトが発生したことを示すメッセージ、またはスレッドの状態が不正なことを示すメッセージに出力されていたスレッド ID を指定してください。

なお、このコマンドを実行したときに、すでにメソッドキャンセルが実行中の場合、またはメソッドキャンセルの対象に指定したスレッドが存在しない場合は、メソッドキャンセルは実行されず、コマンドは正常終了します。コマンド実行後のスレッドの状態は、`cjlistthread` コマンドで確認できます。「4.4.1 J2EE アプリケーションの実行状態の確認」に示した手順で確認してください。

## 4.5 J2EE アプリケーションの閉塞と停止

---

この節では、J2EE アプリケーションの閉塞方法と停止方法について説明します。なお、J2EE アプリケーションだけでなく、システム全体を停止させる場合の実手順については、マニュアル「Cosminexus 簡易構築・運用ガイド」の日常運用の操作に関する説明を参照してください。

### 4.5.1 J2EE アプリケーションの停止方法の種類

J2EE アプリケーションの停止とは、J2EE アプリケーションを構成する Web アプリケーションや Enterprise Bean の終了処理を実行して、J2EE アプリケーションをクライアントからのリクエストを受け付けられない状態にすることです。この状態を、停止状態といいます。そのあとで、J2EE アプリケーション自体を停止します。

J2EE アプリケーションの停止方法には、次の 2 種類があります。

通常停止

強制停止

それぞれについて説明します。

#### (1) 通常停止

日常運用の中で、J2EE アプリケーションを停止するときの停止方法です。J2EE アプリケーション内の Web アプリケーションや Enterprise Bean を順番に停止して、サービスを安全に停止させます。なお、J2EE アプリケーション内でリクエストが実行されている場合、リクエストが終了するまで J2EE アプリケーションは停止しません。ただし、通常停止を実行するときには、タイムアウトを設定できます。タイムアウトを設定しておくと、J2EE アプリケーションが停止していない場合でも、一定時間が経過したらクライアントに制御が戻ります。

通常停止では、まず、サービスを閉塞してリクエストの受け付けを中止してから、J2EE アプリケーションを停止します。サービス閉塞とは、新しいリクエストを受け付けられない状態で、すでに受け付けたリクエストを実行することです。

例えば、次のような場合に計画的に J2EE アプリケーションを停止するときには、まず、サービス閉塞を実行してから、J2EE アプリケーションを停止します。

一日のうち特定の時間内にだけサービスを提供する場合

システムをメンテナンスする場合

#### (2) 強制停止

J2EE アプリケーションで処理中のリクエストが完了するのを待たないで、リクエストを強制的に停止する方法です。強制停止によって、使用されていたリソースは解放されま

す。

J2EE アプリケーションでトラブルが発生した場合など、クライアントに制御が戻らないときには、強制停止によってサービスが迅速に停止できるようになります。

#### ポイント

---

次の場合には J2EE アプリケーションの強制停止を実行できません。

- ステートメントキャンセルが実行できない場合  
データベースを操作している場合に実行中の SQL の処理が戻らなくなったとき、ステートメントキャンセルが実行されます。ステートメントキャンセルは、実行中の SQL の処理を取り消す処理のことです。  
ただし、リソースアダプタの設定、システム構成、使用しているトランザクションの種類などによっては、ステートメントキャンセルができない場合があります。ステートメントキャンセルができない場合、SQL を実行中のトランザクションのタイムアウト、および SQL の処理が戻らない J2EE アプリケーションの強制停止は実行できません。  
ステートメントキャンセルについては、マニュアル「Cosminexus 機能解説」のトランザクションタイムアウトに関する説明を参照してください。
  - メソッドキャンセルが実行できない場合  
強制停止を実行すると、実行中のスレッドは、メソッドキャンセルによって予期しない場所で中断されます。このため、メソッドキャンセルの対象スレッドとほかのスレッドで更新または削除を実行する共有データがある J2EE アプリケーションの場合、メソッドキャンセルを使用できません。  
メソッドキャンセルが実行できない J2EE アプリケーションの構成については、マニュアル「Cosminexus 機能解説」の J2EE アプリケーション実行時間の監視機能に関する説明を参照してください。
  - CTM を使用する環境で、J2EE アプリケーションのアンデプロイ時のスケジュールキューを非活性にするための設定に、タイムアウト時間として 0 (ゼロ) を設定した場合  
CTM のスケジュールキューを非活性にするための処理のタイムアウト時間として 0 を設定した場合、リクエストが終了するまで CTM による閉塞が完了しません。このため、J2EE アプリケーションを停止できなくなります。  
詳細は、マニュアル「Cosminexus 機能解説」のスケジュールキューの閉塞制御に関する説明を参照してください。
- 

## 4.5.2 システムごとに実行できるサービスの閉塞方法および J2EE アプリケーションの停止方法

サービス閉塞は、クライアントから見てリクエストの受け口になる要素（フロント）に対して実行します。フロントを閉塞すると、クライアントから新しいリクエストを受け付けることなく、すでに受け付けた J2EE アプリケーション内のリクエストを継続実行できます。

Web フロントシステムの場合や CTM を使用したバックシステムの場合、J2EE アプリケーションのフロントに、負荷分散機、Web サーバまたは CTM などがあります。J2EE アプリケーションを直接停止する前にこれのフロントに当たる要素を閉塞すると、确实

#### 4. J2EE アプリケーションとリソースの管理

なサービス閉塞が実行できます。

なお、サービス閉塞後に実行する J2EE アプリケーションは、運用管理コマンド (mngsvrutil) で停止してください。

Web フロントシステム、バックシステムなどのシステムの種類の意味については、マニュアル「Cosminexus システム設計ガイド」の Cosminexus のシステム構成の分類に関する説明を参照してください。

##### (1) Web フロントシステムでのサービス閉塞

Web フロントシステムとは、フロントエンドである Web ブラウザから送信されるリクエストを受け付けて、そのリクエストを処理するシステムです。

サービス閉塞の対象は、J2EE アプリケーション内の Web アプリケーションです。

このシステムでは、次の表に示す方法でサービス閉塞を実行できます。J2EE アプリケーションの停止を実行する前にほかの方法でサービス閉塞を実行することで、サービスを確実に閉塞できます。

表 4-12 Web フロントシステムで実行できるサービス閉塞

サービスの閉塞方法	説明	手段	参照先
負荷分散機のリクエストの振り分け先を変更する	負荷分散機の設定を変えて、J2EE アプリケーションを停止する J2EE サーバにリクエストを振り分けないようにします。これによって、J2EE サーバはリクエストを受け付けなくなります。	負荷分散機の機能	4.5.3
負荷分散機を停止する	負荷分散機を停止して、J2EE サーバへのリクエストの転送を停止します。これによって、J2EE サーバはリクエストを受け付けなくなります。	負荷分散機の機能	4.5.3
J2EE アプリケーションを通常停止する	J2EE アプリケーションを通常停止すると、その J2EE アプリケーションは新たなリクエストを受け付けなくなります。ただし、実行中の処理が終了するまで、停止は完了しません。	サーバ管理コマンド	4.5.5

注

負荷分散機を使用したシステムの場合に実行できます。

##### (2) バックシステムでのサービス閉塞 (CTM を使用しているシステムの場合)

バックシステムとは、Web フロントシステムの背後などで動作する、複数の業務システムに共通な業務サービスを実行するためのシステムです。

CTM を使用しているシステムの場合、CTM のスケジュールキューを対象にしたサービス閉塞を実行できます。



このシステムでは、次の表に示す方法でサービス閉塞を実行できます。J2EE アプリケーションの停止を実行する前に、ほかの方法でサービス閉塞を実行することで、サービスを確実に閉塞できます。

表 4-13 バックシステム (CTM を使用している場合) で実行できるサービス閉塞

サービスの閉塞方法	説明	手段	参照先
CTM の機能を使用してサービス閉塞を実行する	CTM の機能を使用してスケジュールキューを閉塞して、J2EE サーバへのリクエストの転送を停止します。これによって、J2EE サーバはリクエストを受け付けなくなります。	運用管理コマンド	4.5.4
J2EE アプリケーションを通常停止する	J2EE アプリケーションを通常停止すると、その J2EE アプリケーションは新たなリクエストを受け付けなくなります。ただし、実行中の処理が終了するまで、停止は完了しません。	サーバ管理コマンド	4.5.5

### (3) バックシステムでのサービス閉塞 (CTM を使用していないシステムの場合)

CTM を使用していないバックシステムの場合、サービス閉塞の対象は、J2EE アプリケーション内のフロントの Enterprise Bean (フロント EJB) です。

このシステムでは、次の方法でサービス閉塞を実行できます。

表 4-14 バックシステム (CTM を使用していない場合) で実行できるサービス閉塞

サービスの閉塞方法	説明	手段	参照先
J2EE アプリケーションを通常停止する	J2EE アプリケーションを通常停止すると、その J2EE アプリケーションは新たなリクエストを受け付けなくなります。ただし、実行中の処理が終了するまで、停止は完了しません。	サーバ管理コマンド	4.5.5

なお、システム構築時にあらかじめフロント EJB を定義しておく必要があります。

## 4.5.3 負荷分散機を利用したサービス閉塞

ここでは、負荷分散機を利用したサービス閉塞の実行方法について説明します。負荷分散機を利用したサービス閉塞は、Web フロントシステムの場合に、負荷分散機を利用しているときに実行できます。負荷分散機を利用したサービスの閉塞には、次の二つの方法があります。

- 通常のサービス閉塞
- サービス部分閉塞

それぞれの実行方法について説明します。

##### ポイント

---

負荷分散機を利用したサービス閉塞は、Smart Composer 機能で負荷分散機を管理していない場合に使用します。Smart Composer 機能で負荷分散機を管理している場合のサービス閉塞は、`cmx_stop_target` コマンドを使用します。`cmx_stop_target` コマンドについては、マニュアル「Cosminexus 簡易構築・運用ガイド」を参照してください。

---

### (1) 通常のサービス閉塞

負荷分散機を利用した通常のサービス閉塞は、次のどちらかの方法で実行します。

負荷分散機のリクエストの振り分け先を変更する

通常使用する Web サーバのほかにサービス閉塞用の Web サーバを用意しておいて、サービス閉塞するタイミングでリクエストの振り分け先をサービス閉塞用の Web サーバに切り替える方法です。サービス閉塞用の Web サーバには、エラーページ表示の設定などをおきます。

負荷分散機を停止する

負荷分散機を停止して、リクエストの受け付けを停止する方法です。すでに Web サーバに送られたリクエストはそのまま実行されます。

なお、サービス閉塞を解除する場合は、リクエストの振り分け先を再度変更するか、または負荷分散機を起動します。

手順については、ご使用の負荷分散機の使用方法に従ってください。

### (2) 負荷分散機を利用したサービス部分閉塞

リクエストを受け付ける Web サーバ、およびリクエストを実行する J2EE サーバを負荷分散機を使用して冗長化している構成の場合、サービス全体を停止させないで、サービスを構成する一部の Web アプリケーションだけを停止できます。この閉塞方法を、サービス部分閉塞といいます。

なお、Web アプリケーションのサービス部分閉塞ができるのは、次のような構成のシステムです。

負荷分散機を使用したシステム

Web サーバと J2EE サーバが冗長化されていて、Web サーバと J2EE サーバが 1:1 または 1:N の構成であること (N は 1 以上の整数)

Web アプリケーションのサービス部分閉塞は、負荷分散機によるリクエストの振り分け先を変更して、サービスを閉塞する系にリクエストを振り分けないことで実現できます。実行方法については、負荷分散機の使用方法を確認ください。

---

## ポイント

負荷分散機によってサービス部分閉塞を実行する場合、停止する Web サーバに対して作成されたセッションを使用したリクエストについては、セッションが切断されます。この場合、稼働中の Web サーバに対してセッションを接続し直す必要があります。ただし、セッションフェイルオーバー機能を使用している場合は、セッションの一部の情報を引き継ぎます。セッションフェイルオーバー機能については、マニュアル「Cosminexus 機能解説」の J2EE サーバ間のセッション情報の引き継ぎに関する説明を参照してください。

---

### 4.5.4 CTM を使用したサービス閉塞

ここでは、CTM を使用したサービス閉塞を、運用管理コマンド (mngsvrutil) で実行する方法について説明します。

この方法は、CTM を使用しているときに実行できます。

CTM を使用しているバックシステムで、ホスト内の J2EE アプリケーションを一度に停止したり、キューを共有する J2EE アプリケーションを一度に停止したりする場合、J2EE アプリケーションのスケジュールキューに対して直接閉塞を実行します。そのあと、J2EE アプリケーションを停止します。

運用管理コマンドを使用して J2EE アプリケーションのスケジュールキューを直接閉塞する場合、スケジュールキューを共有する J2EE アプリケーション単位、ホスト単位、または運用管理ドメイン単位で J2EE アプリケーションを一度に停止できます。また、スケジュールキューに登録済みのリクエストを破棄するか、一定時間処理を続けるかどうかを選択します。スケジュールキューに登録済みのリクエストを破棄する場合、登録済みのリクエストはすべてエラーとしてクライアントに返却されます。一定時間処理を続ける場合、時間内に終了しなかった処理はエラーとしてクライアントに返却されます。

運用管理コマンドを使用した場合の CTM を使用したサービス閉塞の実行形式および実行例について説明します。ここでは、通常の手順で閉塞する方法と、強制的に閉塞する方法について説明します。強制閉塞は、CTM デーモンの負荷が高い場合などに、すぐにキューを閉塞したいときに実行する方法です。

#### (a) 通常の閉塞

運用管理コマンド (mngsvrutil) を使用して通常の手順で閉塞する方法について説明します。

CTM の閉塞処理では、次の作業が実行されます。

新規リクエストの受け付けが終了されます。

スケジュールキューに格納されているリクエストのうち、すでに J2EE サーバに振り分けられて処理中のリクエストは引き続き処理されます。

スケジュールキューに格納されているリクエストのうち、J2EE サーバへの振り分け

#### 4. J2EE アプリケーションとリソースの管理

がされていないリクエストに対しては、java.rmi.RemoteException エラーが返却されます。

特定のホスト内の J2EE アプリケーションを一括停止する場合の運用管理コマンドの実行形式と実行例を次に示します。なお、運用管理コマンドの詳細については、マニュアル「Cosminexus リファレンス コマンド編」を参照してください。

##### 実行形式

```
mngsvrutil -m <Management Serverのホスト名> -u <管理ユーザID> -p <管理パスワード> -t <ホスト名> -k host hold queues in:<リクエスト終了待ち時間(秒)>
```

##### 実行例

- サービス閉塞をして、すべてのリクエスト処理の完了を待つ場合  
mngsvrutil -m mnghost -u user01 -p pw1 -t host01 -k host hold queues in:0
- サービス閉塞をして、5 分間リクエストの処理を続けて、終了しないリクエストは破棄する場合  
mngsvrutil -m mnghost -u user01 -p pw1 -t host01 -k host hold queues in:300
- サービス閉塞をして、リクエストはすぐに破棄する場合  
mngsvrutil -m mnghost -u user01 -p pw1 -t host01 -k host hold queues in:-1

スケジュールキューの閉塞を解除するときは、mngsvrutil コマンドにサブコマンド「release」を指定して実行します。mngsvrutil コマンドの実行形式と実行例を次に示します。

##### 実行形式

```
mngsvrutil -m <Management Serverのホスト名> -u <管理ユーザID> -p <管理パスワード> -t <ホスト名> -k host release queues
```

##### 実行例

```
mngsvrutil -m mnghost01 -u user01 -p pw1 -t host01 -k host release queues
```

##### (b) 強制閉塞

強制閉塞は、CTM デーモンの負荷が高いときにすぐにキューを閉塞するための方法です。通常の閉塞方法では、キューを閉塞するときに、CTM デーモンと通信して、その処理の延長で滞留しているリクエストを破棄しています。しかし、この方法では、CTM デーモンの負荷が高い場合、通信処理に時間が掛かるため、閉塞処理にも時間が掛かってしまいます。

強制閉塞を使用すると、CTM デーモンとの通信処理をしないで、即座にキューを閉塞できます。なお、強制閉塞を使用した場合、滞留しているリクエストの破棄は、CTM デーモン間で負荷情報を監視するタイミングにあわせて実行されます。

なお、閉塞の解除方法については、通常の閉塞をした場合と同じです。コマンドの詳細については、マニュアル「Cosminexus リファレンス コマンド編」を参照してください。

強制閉塞を実行する場合の `mngsvrutil` コマンドの実行例を次に示します。

#### 実行形式

```
mngsvrutil -m <Management Serverのホスト名> -u <管理ユーザID> -p <管理パスワード> -t
<ホスト名> -k host hold queues force
```

#### 実行例

```
mngsvrutil -m mngghost -u user01 -p pw1 -t host01 -k host hold
queues force
```

### 4.5.5 J2EE アプリケーションの停止

ここでは、サーバ管理コマンドを使用して J2EE アプリケーションを停止する方法について説明します。

サーバ管理コマンドを使用した J2EE アプリケーションの停止には、次の五つの方法があります。

1. デフォルトのタイムアウト時間で通常停止を実行する
2. 任意のタイムアウト時間を設定して通常停止を実行する
3. デフォルトのタイムアウト時間で通常停止を実行して、停止しなかった場合に強制停止を実行する
4. 任意のタイムアウト時間を設定して通常停止を実行して、停止しなかった場合に強制停止を実行する
5. 任意のタイムアウト時間を設定して通常停止を実行して、停止しなかった場合に自動的に強制停止を実行する

これらの方法以外の順序でコマンドを実行した場合、コマンドが異常終了します。例えば、通常停止をしない状態では、強制停止の形式でコマンドを実行できません。

また、1.、2.、または 5. の処理は、一つの J2EE アプリケーションに対してどれか 1 回だけ実行できます。

ただし、強制停止は、J2EE サーバの動作設定をカスタマイズして、強制停止が有効になるように設定されているシステムだけで実行できます。J2EE サーバの動作設定のカスタマイズの方法については、マニュアル「Cosminexus システム構築ガイド」を参照してください。

#### 4. J2EE アプリケーションとリソースの管理

J2EE アプリケーションの停止処理では、次の処理が実行されます。

J2EE アプリケーションのフロントが Web アプリケーションの場合

- 新規リクエストを受け付けなくなります。
- 処理中のリクエストは引き続き処理されます。
- Web アプリケーションの同時実行数制御のキューに格納されているリクエストのうち、Web コンテナでの処理が開始されていないものに対して、HTTP503 エラーが返却されます。

J2EE アプリケーションのフロントが Enterprise Bean の場合

- 新規リクエストを受け付けなくなります。
- 処理中のリクエストは引き続き処理されます。

停止後のアプリケーションに含まれている Enterprise Bean のメソッドの呼び出しはできません。停止後のアプリケーションに含まれている Enterprise Bean のメソッドを呼び出そうとすると、「stop could not lock <文字列>」というメッセージが J2EE サーバの標準エラー出力に出力されることがありますが、問題ありません。

次に、それぞれのパターンについて、サーバ管理コマンドの実行手順を説明します。

##### (1) デフォルトのタイムアウト時間で通常停止を実行する

コマンドの実行手順、実行形式および実行例を次に示します。

###### 1. J2EE アプリケーションを通常停止します。

cjstopapp コマンドを実行します。

実行形式

```
cjstopapp <J2EEサーバ名> -name <J2EEアプリケーション名>
```

実行例

```
cjstopapp Myserver -name Appl
```

なお、デフォルトのタイムアウト時間でコマンドを実行した場合、停止処理が完了しなくても 60 秒後にコマンドの制御が戻ります。

###### 2. コマンドの制御がタイムアウトによって戻った場合は、J2EE アプリケーションの停止状態を確認します。

タイムアウトによってコマンドの制御が戻った場合、J2EE アプリケーションの停止処理は完了していないことがあります。このため、タイムアウトが発生した場合は、J2EE アプリケーションの状態を確認してください。

J2EE アプリケーションの状態は、サーバ管理コマンドの cjlistapp コマンド実行して確認します。

実行形式

```
cjlistapp <サーバ名称>
```

#### 実行例

```
cjlistapp MyServer
```

J2EE アプリケーションの状態として、次のどれかの状態が出力されます。

表 4-15 J2EE アプリケーションの状態

出力される文字列	状態の意味
running	開始状態
stopped	停止状態
stopFailure	通常停止失敗状態
forceStopFailure	強制停止失敗状態
blockadeFailure	閉塞失敗状態
blockading	閉塞中
blockaded	閉塞状態
stopping	通常停止中
forceStopping	強制停止中

なお、手順の実行後、J2EE サーバの状態を確認してください。次の状態の場合は、J2EE サーバを再起動する必要があります。

閉塞失敗

通常停止失敗

#### (2) 任意のタイムアウト時間を設定して通常停止を実行する

コマンドの実行手順、実行形式および実行例を次に示します。

1. J2EE アプリケーションを通常停止します。このとき、コマンドのオプションに、コマンドの制御を戻すためのタイムアウト時間を設定します。  
cjstopapp コマンドに、-t オプションを指定して実行します。

#### 実行形式

```
cjstopapp <J2EEサーバ名> -name <J2EEアプリケーション名> -t <タイムアウト時間>
```

#### 実行例

```
cjstopapp MyServer -name App1 -t 120
```

#### 4. J2EE アプリケーションとリソースの管理

この実行例を実行した場合、停止処理が完了しなくても 120 秒後にコマンドの制御が戻ります。

2. タイムアウトによってコマンドの制御が戻った場合は、J2EE アプリケーションの停止状態を確認します。

コマンドの制御がタイムアウトによって戻った場合、J2EE アプリケーションの停止処理は完了していないことがあります。このため、タイムアウトが発生した場合は、J2EE アプリケーションの状態を確認してください。

J2EE アプリケーションの状態は、`cjlistapp` コマンドを実行して確認します。

実行形式

```
cjlistapp <J2EEサーバ名>
```

実行例

```
cjlistapp MyServer
```

J2EE アプリケーションの状態として出力される内容については、「(1) デフォルトのタイムアウト時間で通常停止を実行する」の手順 2. を参照してください。

なお、手順の実行後、J2EE サーバの状態を確認してください。次の状態の場合は、J2EE サーバを再起動する必要があります。

閉塞失敗

通常停止失敗

### (3) デフォルトのタイムアウト時間で通常停止を実行して、停止しなかった場合に強制停止を実行する

コマンドの実行手順、実行形式および実行例を次に示します。

1. J2EE アプリケーションを通常停止します。

`cjstopapp` コマンドを実行します。

実行形式

```
cjstopapp <J2EEサーバ名> -name <J2EEアプリケーション名>
```

実行例

```
cjstopapp MyServer -name App1
```

なお、デフォルトのタイムアウト時間でコマンドを実行した場合、停止処理が完了しなくても 60 秒後にコマンドの制御が戻ります。

2. タイムアウトによってコマンドの制御が戻った場合は、J2EE アプリケーションの停



止状態を確認します。

コマンドの制御がタイムアウトによって戻った場合、J2EE アプリケーションの停止処理は完了していないことがあります。このため、タイムアウトが発生した場合は、J2EE アプリケーションの状態を確認してください。

J2EE アプリケーションの状態は、cjlistapp コマンドを実行して確認します。

実行形式

```
cjlistapp <J2EEサーバ名>
```

実行例

```
cjlistapp MyServer
```

J2EE アプリケーションの状態として出力される内容については、「(1) デフォルトのタイムアウト時間で通常停止を実行する」の手順 2. を参照してください。

### 3. J2EE アプリケーションの強制停止を実行します。

J2EE アプリケーションが通常停止で停止しなかった場合、強制停止を実行します。

cjstopapp コマンドに、-cancel オプションを指定して実行します。

実行形式

```
cjstopapp <J2EEサーバ名> -name <J2EEアプリケーション名> -cancel
```

実行例

```
cjstopapp MyServer -name App1 -cancel
```

強制停止が成功すると、J2EE アプリケーションで処理中のリクエストが中止されて、先に実行していた通常停止が終了します。

なお、手順の実行後、J2EE サーバの状態を確認してください。次の状態の場合は、J2EE サーバを再起動する必要があります。

強制停止失敗

閉塞失敗

通常停止失敗

### (4) 任意のタイムアウト時間を設定して通常停止を実行して、停止しなかった場合に強制停止を実行する

コマンドの実行手順、実行形式および実行例を次に示します。

1. J2EE アプリケーションを通常停止します。このとき、コマンドのオプションに、コマンドの制御を戻すためのタイムアウト時間を設定します。

#### 4. J2EE アプリケーションとリソースの管理

cjstopapp コマンドに、-t オプションを指定して実行します。

実行形式

```
cjstopapp <J2EEサーバ名> -name <J2EEアプリケーション名> -t <タイムアウト時間>
```

実行例

```
cjstopapp MyServer -name Appl -t 120
```

この実行例を実行した場合、停止処理が完了しなくても 120 秒後にコマンドの制御が戻ります。

- タイムアウトによってコマンドの制御が戻った場合は、J2EE アプリケーションの停止状態を確認します。  
コマンドの制御がタイムアウトによって戻った場合、J2EE アプリケーションの停止処理は完了していないことがあります。このため、タイムアウトが発生した場合は、J2EE アプリケーションの状態を確認してください。  
J2EE アプリケーションの状態は、cjlistapp コマンドを実行して、確認します。

実行形式

```
cjlistapp <J2EEサーバ名>
```

実行例

```
cjlistapp MyServer
```

J2EE アプリケーションの状態として出力される内容については、「(1) デフォルトのタイムアウト時間で通常停止を実行する」の手順 2. を参照してください。

- J2EE アプリケーションの強制停止を実行します。  
J2EE アプリケーションが通常停止で停止しなかった場合、強制停止を実行します。  
cjstopapp コマンドに、-cancel オプションを指定して実行します。

実行形式

```
cjstopapp <J2EEサーバ名> -name <J2EEアプリケーション名> -cancel
```

実行例

```
cjstopapp Myserver -name Appl -cancel
```

強制停止が成功すると、J2EE アプリケーションで処理中のリクエストが中止されて、先に実行していた通常停止が終了します。

なお、手順の実行後、J2EE サーバの状態を確認してください。次の状態の場合は、

J2EE サーバを再起動する必要があります。

強制停止失敗

閉塞失敗

通常停止失敗

(5) 任意のタイムアウト時間を設定して通常停止を実行して、停止しなかった場合に自動的に強制停止を実行する

コマンドの実行手順、実行形式および実行例を次に示します。

1. タイムアウト時間後に強制停止を実行する形式で、J2EE アプリケーションを停止します。

この形式では、J2EE アプリケーションをまず通常停止で停止して、指定したタイムアウト時間内に停止しなかった場合に、強制停止を実行します。

cjstopapp コマンドに、`-t` オプションと `-force` オプションを指定して実行します。

実行形式

```
cjstopapp <J2EEサーバ名> -name <J2EEアプリケーション名> -t <タイムアウト時間> -force
```

実行例

```
cjstopapp MyServer -name Appl -t 120 -force
```

なお、手順の実行後、J2EE サーバの状態を確認してください。次の状態の場合は、J2EE サーバを再起動する必要があります。

強制停止失敗

閉塞失敗

通常停止失敗

## 4.6 J2EE アプリケーションの入れ替えと保守

この節では、J2EE アプリケーションの入れ替えと保守について説明します。また、テストモードでの J2EE アプリケーションの運用についても説明します。

J2EE アプリケーションのバージョンアップやメンテナンスを実施する場合、J2EE アプリケーションを入れ替えることがあります。

また、24 時間サービス提供が必要な J2EE アプリケーションを入れ替える場合、CTM を使用したオンライン状態での J2EE アプリケーションの入れ替えの方法を使用すると、サービスを停止しないで J2EE アプリケーションを入れ替えられます。

### 4.6.1 J2EE アプリケーションの入れ替え

ここでは、サーバ管理コマンドを使用して J2EE アプリケーションを入れ替える手順を説明します。J2EE アプリケーションは、停止して、削除してから新しいアプリケーションに入れ替えます。入れ替え後、J2EE アプリケーションを再開します。

入れ替えは、次の手順で実行します。

1. 入れ替える J2EE アプリケーションを停止します。

cjstopapp コマンドを実行します。実行形式と実行例を次に示します。

実行形式

```
cjstopapp <J2EEサーバ名> -name <J2EEアプリケーション名>
```

実行例

```
cjstopapp MyServer -name App1
```

2. J2EE アプリケーションを削除します。

cjdeleteapp コマンドを実行します。実行形式と実行例を次に示します。

実行形式

```
cjdeleteapp <J2EEサーバ名> -name <J2EEアプリケーション名>
```

実行例

```
cjdeleteapp MyServer -name App1
```

3. 入れ替え後の J2EE アプリケーションをインポートします。

サーバ管理コマンドを使用する場合、cjimportapp コマンドを実行します。実行形式

と実行例を次に示します。

#### 実行形式

```
cjimportapp <J2EEサーバ名> -f <EARファイルのパス>
```

#### 実行例

```
cjimportapp MyServer -f Appl.ear
```

4. J2EE アプリケーションに必要な情報を設定，およびカスタマイズします。
- サーバ管理コマンドを使用する場合は，cjgetappprop コマンドで属性ファイルを取得して，属性ファイルを編集してから，cjsetappprop コマンドを実行して設定します。cjgetappprop コマンドの実行形式と実行例を次に示します。

#### 実行形式

```
cjgetappprop <J2EEサーバ名> -name <J2EEアプリケーション名> -type all -encoding Shift_JIS -c <アプリケーション統合属性ファイル>
```

#### 実行例

```
cjgetappprop MyServer -name Appl -type all -encoding Shift_JIS  
-c Appl.xml
```

属性ファイルはテキストエディタなどで編集します。

cjsetappprop コマンドの実行形式と実行例を次に示します。

#### 実行形式

```
cjsetappprop <J2EEサーバ名> -name <J2EEアプリケーション名> -type all -c <アプリケーション統合属性ファイル>
```

#### 実行例

```
cjsetappprop MyServer -name Appl -type all -c Appl.xml
```

5. J2EE アプリケーションを開始します。
- サーバ管理コマンドを使用する場合，cjstartapp コマンドを実行します。実行形式と実行例を次に示します。

#### 実行形式

```
cjstartapp <J2EEサーバ名> -name <J2EEアプリケーション名>
```

##### 実行例

```
cjstartapp MyServer -name Appl
```

### 4.6.2 リデプロイによる J2EE アプリケーションの入れ替え

ここでは、リデプロイによる J2EE アプリケーションの入れ替えについて説明します。

リデプロイとは、アーカイブ形式の J2EE アプリケーションを入れ替える場合に、少ない手順で高速に入れ替えられるデプロイ方法です。ロジックだけを変更した J2EE アプリケーションを入れ替えたい場合などに利用できます。リデプロイは、サーバ管理コマンドを使用して実行できます。

リデプロイを実行できる条件を次に示します。

#### リデプロイを実行できる条件

- 入れ替えられるのは、実行時情報を含まない J2EE アプリケーション（EAR ファイル）だけです。実行時情報を含む J2EE アプリケーション（ZIP ファイル）はリデプロイできません。
- 入れ替え前と入れ替え後の J2EE アプリケーションの構成が同じである必要があります。J2EE アプリケーションに含まれる EJB-JAR、リソースアダプタおよび WAR の数が異なったり、それらのファイル名称が異なったりする場合は、リデプロイはできません。また、J2EE アプリケーションの名称も同じである必要があります。
- 入れ替え前と入れ替え後の J2EE アプリケーションに含まれる EJB-JAR 内のホームインタフェース（ローカル、リモート）、コンポーネントインタフェース（ローカル、リモート）、ビジネスインタフェース（ローカル、リモート）のメソッド定義、およびアノテーションの値が同じである必要があります。
- ランタイム属性だけを引き継ぐ設定をしている場合に、アプリケーション開発環境で設定済みの DD ファイル（application.xml、ejb-jar.xml、ra.xml および web.xml）の定義内容が同じである必要があります。

また、J2EE アプリケーションを入れ替える時に、入れ替え前の J2EE アプリケーションを別な名称に変更して退避しておく、名称による J2EE アプリケーションの世代管理が実現できます。ここでは、J2EE アプリケーションの名称変更についてもあわせて説明します。

## 参考

---

通常の入れ替えまたはリデプロイによる入れ替えによって J2EE アプリケーションを入れ替える場合、入れ替え後の J2EE アプリケーションをテストモードで運用できます。テストモードを使用すると、実運用に入る前に、実際の運用環境での J2EE アプリケーションの動作を確認できます。テストモードによる運用については、「4.6.7 テストモードでの J2EE アプリケーションの運用」を参照してください。

---

リデプロイでは、入れ替え前の J2EE アプリケーションの情報を、入れ替え後の J2EE アプリケーションに引き継げます。次の情報が引き継げます。

## Application Server のバージョンが 06-70 以降の場合

デフォルトの設定の場合、入れ替え後の J2EE アプリケーションには、入れ替え前の J2EE アプリケーションのすべての属性が引き継がれます。06-70 よりも前のバージョンと同じように、ランタイム属性だけを引き継ぎたい場合は、オプションを指定して `cjreplaceapp` コマンドを実行する必要があります。コマンドの詳細は、マニュアル「Cosminexus リファレンス コマンド編」を参照してください。

## Application Server のバージョンが 06-70 より前の場合

入れ替え前の J2EE アプリケーションの属性ファイルに設定したランタイム属性が引き継がれます。

注 属性ファイルには、DD(application.xml, ejb-jar.xml, ra.xml, web.xml) の定義と属性ファイル独自の定義が設定できます。属性ファイル独自の定義のことを、ランタイム属性といいます。

リデプロイを実行するとき、J2EE アプリケーションは開始、停止どちらの状態でもかまいません。開始状態の J2EE アプリケーションを入れ替えた場合、J2EE アプリケーションは入れ替え後に自動的に開始されます。ただし、プールやキャッシュに格納されていた J2EE アプリケーション関連のオブジェクトは破棄されます。停止状態の J2EE アプリケーションを入れ替えた場合は、入れ替え後の J2EE アプリケーションも停止した状態になります。

##### ポイント

J2EE アプリケーションが開始されている状態でリデプロイを実行した場合、リデプロイ処理の中で J2EE アプリケーションは停止され、入れ替え後に再開されます。このとき、停止処理の実行時間がサーバ管理コマンド (cjreplaceapp) で設定したタイムアウト時間を超過した場合、J2EE アプリケーションの強制停止が実行されます。タイムアウト時間を指定しなかった場合は、デフォルトのタイムアウト時間である 60 秒を超過すると、強制停止が実行されます。強制停止実行後にさらにタイムアウト時間を超過した場合は、コマンドが異常終了します。

また、入れ替え後に J2EE アプリケーションを再開するとき、開始処理の実行時間がサーバ管理コマンド用の usrcnf.properties の ejbserver.rmi.request.timeout キーに指定したタイムアウト時間を超過した場合も、コマンドが異常終了します。

入れ替え作業の実行形式と実行例を次に示します。

##### 実行形式

```
cjreplaceapp <J2EEサーバ名> -name <J2EEアプリケーション名> -f <入れ替えるEARファイルのパス>
```

##### 実行例

```
cjreplaceapp MyServer -name Appl1 -f Appl1.ear
```

### 4.6.3 リロードによる J2EE アプリケーションの入れ替え

ここでは、リロードによる J2EE アプリケーションの入れ替え方法について説明します。

リロードとは、少ない手順で展開ディレクトリ形式の J2EE アプリケーションの入れ替えを実行できる機能です。リロードによる J2EE アプリケーションの入れ替えでは、既存の J2EE アプリケーションの停止、削除、入れ替え後の J2EE アプリケーションのアーカイブ、インポート、再開などの作業が不要です。クラスファイルを更新してリロードを実行するだけで J2EE アプリケーションを更新できるため、メンテナンスが頻繁に発生するシステムの運用などで特に有効な機能です。

なお、リロードによる J2EE アプリケーションの入れ替えを実行するには、事前に設定が必要です。設定方法の詳細については、マニュアル「Cosminexus システム構築ガイド」の J2EE アプリケーションの更新検知とリロードの設定に関する説明を参照してください。

リロードによる J2EE アプリケーションの入れ替えは、サーバ管理コマンド (cjreloadapp コマンド) を使用して実行できます。cjreloadapp コマンドの詳細についてはマニュアル「Cosminexus アプリケーション設定操作ガイド」を参照してください。

リロードは、次の手順で実行します。

1. メンテナンスの内容に従って Java ソースファイルを編集、または作成して、クラス



ファイルにコンパイルします。

## 2. J2EE アプリケーションのリロードを実行します。

cjreloadapp コマンドを実行します。実行形式と実行例を次に示します。

実行形式

```
cjreloadapp <J2EEサーバ名> -name <J2EEアプリケーション名>
```

実行例

```
cjreloadapp MyServer -name Appl
```

### ! 注意事項

リロードに失敗したアプリケーションを削除するには、リロードの成功後、アプリケーションを停止、削除するか、J2EE サーバを再起動したあとにアプリケーションを削除してください。

## 4.6.4 入れ替える J2EE アプリケーション内の JSP の事前コンパイル

J2EE アプリケーションのメンテナンスで JSP を編集した場合、通常、入れ替え後の J2EE アプリケーションに最初のリクエストがあったときに、JSP のコンパイルが実行されます。JSP 事前コンパイル機能を使用すると、デプロイ前の J2EE アプリケーションに対して JSP のコンパイルを実行しておけるため、最初のリクエストに対するレスポンスの時間を短縮することができます。

JSP 事前コンパイル機能を実行するには、サーバ管理コマンド、または cjjspc コマンドを使用します。ここでは、JSP 事前コンパイル機能の実行タイミングと実行方法について説明します。

システム運用時に JSP 事前コンパイル機能を実行するタイミングを次に示します。

- J2EE アプリケーション開始時 (サーバ管理コマンド)
- リロードによる J2EE アプリケーションの入れ替え時 (cjjspc コマンド)
- リデプロイによる J2EE アプリケーションの入れ替え時 (cjjspc コマンド)

それぞれのタイミングでの、JSP 事前コンパイル機能の実行方法について説明します。

### (1) J2EE アプリケーションを開始するときの JSP の事前コンパイル

J2EE アプリケーションを開始するとき JSP 事前コンパイル機能を実行します。この場合、cjstartapp コマンドを、-jspc オプションを指定して実行します。

#### 4. J2EE アプリケーションとリソースの管理

実行形式と実行例を次に示します。

実行形式

```
cjstartapp <サーバ名称> -name <J2EEアプリケーション名> -jspc
```

実行例

```
cjstartapp MyServer -name account -jspc
```

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

#### (2) リロードによって J2EE アプリケーションを入れ替えるときの JSP の事前コンパイル

リロードによって J2EE アプリケーションを入れ替える場合、cjreloadapp コマンドを実行する前に JSP 事前コンパイル機能を実行します。この場合、cjjspc コマンドを実行します。

実行形式と実行例を次に示します。

実行形式

```
cjjspc -root <Webアプリケーションのルートディレクトリ>
```

実行例 (Windows の場合)

```
cjjspc -root d:\app\webappl
```

実行例 (UNIX の場合)

```
cjjspc -root /tmp/app/webappl
```

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

#### (3) リデプロイによって J2EE アプリケーションを入れ替えるときの JSP の事前コンパイル

リデプロイによって J2EE アプリケーションを入れ替える場合、cjreplaceapp コマンドを実行する前に JSP 事前コンパイル機能を実行します。この場合、cjjspc コマンドを実行します。

実行形式と実行例を次に示します。

実行形式

```
cjjspc -root <Webアプリケーションのルートディレクトリ>
```

実行例 (Windows の場合)

```
cjjspc -root d:¥app¥webapp1
```

実行例 (UNIX の場合)

```
cjjspc -root /tmp/app/webapp1
```

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

#### ! 注意事項

- リデプロイによって入れ替える J2EE アプリケーションに事前コンパイル機能を使用する場合、入れ替える J2EE アプリケーションに JSP コンパイル結果が含まれていないと JSP 事前コンパイル機能は実行されません。入れ替え後も JSP 事前コンパイル機能を使用する場合は、Web アプリケーション単位の JSP 事前コンパイルを実行して、入れ替える J2EE アプリケーションに JSP コンパイル結果を含めるようにしてください。
- タグファイル、静的インクルードされたファイル、または TLD ファイルを更新した場合、更新したファイルを参照するすべての JSP ファイルをコンパイルしてください。
- JSP 事前コンパイル機能を使用した展開ディレクトリ形式のアプリケーションに、JSP ファイル、またはタグファイルを追加した場合、JSP 事前コンパイルを再度実施して、JSP ファイル、またはタグファイルを参照するすべての JSP ファイルをコンパイルしてください。
- 展開ディレクトリ形式のアプリケーションの JSP ワークディレクトリに含まれるクラスファイルを、開発環境からコピーして更新する場合は、開発環境で JSP 事前コンパイルを実施したすべてのクラスファイルをコピーしてください。
- cjjspc コマンドを使用して JSP の事前コンパイルを実施した場合、JSP ファイルまたはタグファイルのトランスレーション時にエラーが発生すると、エラーメッセージがコンソールに出力されます。また、ejstartapp コマンドを使用して JSP の事前コンパイルを実施した場合、JSP ファイルまたはタグファイルのトランスレーション時にエラーが発生すると、エラーメッセージがサーブレットログに出力されます。

### 4.6.5 J2EE アプリケーションの名称変更

J2EE アプリケーションを入れ替える時、既存の J2EE アプリケーションの名称を変更して退避しておくと、J2EE アプリケーションの世代管理ができます。また、入れ替え前の J2EE アプリケーションに戻りたい場合などにもスムーズに対処できます。

J2EE アプリケーションの名称変更は、サーバ管理コマンドで実行します。すでに変更後の名称と同じ名称の J2EE アプリケーションがある場合、その名称には変更できません。大文字、小文字の違いによる区別はできません。ただし、通常モードまたはテストモードなどの運用時のモードが異なる場合、それぞれのモードで一つずつ同じ名称の J2EE

#### 4. J2EE アプリケーションとリソースの管理

アプリケーションを登録できます。

なお、名称を変更すると、ルックアップ名称の変更も必要になります。J2EE アプリケーションの構成要素に、リモート呼び出しをするホームインタフェースまたはコンポーネントインタフェースを使用した Enterprise Bean が含まれている場合、RMI-IIOP スタブおよびインタフェースを取得し直してください。

RMI-IIOP スタブおよびインタフェースの取得は、サーバ管理コマンド (cjgetstubsjar) を使用して実行します。J2EE アプリケーションを一度も実行していない状態では取得できません。また、次の場合、RMI-IIOP スタブおよびインタフェースの取得はエラーになります。

J2EE アプリケーションに WAR しか含まれない場合

Enterprise Bean で使用されているホームインタフェースおよびコンポーネントインタフェースがすべてローカル呼び出しの場合

Enterprise Bean が Message-driven Bean だけの場合

名称変更は、次の手順で実行します。

1. 名称を変更する J2EE アプリケーションを停止します。  
サーバ管理コマンド (cjstopapp) を実行します。
2. J2EE アプリケーションの名称を変更します。  
サーバ管理コマンド (cjrenameapp) を実行します。  
実行形式と実行例を次に示します。

実行形式

```
cjrenameapp <J2EEサーバ名> -name <変更前のJ2EEアプリケーションの名称> -newname <変更後のJ2EEアプリケーションの名称>
```

実行例

```
cjrenameapp MyServer -name App1 -newname App1bak
```

3. J2EE アプリケーションを開始します。  
サーバ管理コマンド (cjstartapp) を実行します。
4. J2EE アプリケーションに含まれる Enterprise Bean 内のホームインタフェースまたはコンポーネントインタフェースがリモート呼び出しを実行するように定義されている場合、RMI-IIOP スタブおよびインタフェースを取得します。  
サーバ管理コマンド (cjgetstubsjar) を実行します。  
実行形式と実行例を次に示します。

実行形式

```
cjgetstubsjar <J2EEサーバ名> -name <変更後のJ2EEアプリケーションの名称> -d <RMI-IIOP
スタブおよびインタフェースを格納するディレクトリのパス>
```

### 実行例

```
cjgetstubsjar MyServer -name Applbak -d temp
```

### ! 注意事項

Application Server のバージョンが 06-50 よりも前の場合、Web ブラウザで index.html を表示して RMI-IIOP スタブおよびインタフェースを取得できました。06-50 以降の場合、次の条件を満たすときだけ、この機能を使用できます。

- J2EE サーバに旧バージョンで作成した J2EE アプリケーションがある状態で、06-50 以降の Application Server をアップグレードインストールした場合
- cjrenameapp コマンドで名称を変更していない場合

次の場合は、index.html による RMI-IIOP スタブおよびインタフェースの取得はできませんので、ご注意ください。

- 06-50 以降の環境で、J2EE アプリケーションを新規にインポートした場合  
実行時情報を含む J2EE アプリケーションまたは実行時情報を含まない J2EE アプリケーションのどちらかをインポートした場合も、実行できません。
- 06-50 以降の環境で新規に J2EE アプリケーションを作成した場合
- cjrenameapp コマンドで名称を変更した場合

## 4.6.6 CTM を使用したオンライン状態での J2EE アプリケーションの入れ替え

CTM を利用したシステムでは、オンライン状態で J2EE アプリケーションを入れ替える場合、J2EE アプリケーションのスケジュールキューの出口を閉じたあと、入れ替えを実行します。この操作は、運用管理コマンド (mngsvrutil) で実行できます。

J2EE アプリケーションの入れ替えは、J2EE アプリケーション単位、ホスト単位、または運用管理ドメイン単位で実行できます。

スケジュールキューの出口を閉じるには、mngsvrutil コマンドにサブコマンド「hold」を指定して実行します。mngsvrutil コマンドを実行してスケジュールキューの出口を閉塞している間も、クライアントからのリクエストはスケジュールキューに登録し続けられます。ただし、スケジュールキューの最大リクエスト登録数を超えた場合、クライアントにエラーが返却されます。

J2EE アプリケーションの入れ替えが終了したら、スケジュールキューの閉塞を解除します。スケジュールキューの閉塞解除は、mngsvrutil コマンドにサブコマンド「release」を指定して実行します。スケジュールキューの閉塞を解除すると、J2EE アプリケーション

#### 4. J2EE アプリケーションとリソースの管理

ンで、スケジュールキューに保持されていたリクエストの処理を再開します。

CTM を使用したオンライン状態での J2EE アプリケーションの入れ替え手順を次に示します。

1. 入れ替える J2EE アプリケーションに対応する CTM のスケジュールキューの出口を閉じます。

J2EE アプリケーションを入れ替える場合の mngsvrutil コマンドの実行形式と実行例を次に示します。

実行形式

```
mngsvrutil -m <Management Serverのホスト名> -u <管理ユーザID> -p <管理パスワード> -t <CTMの名称> hold queue <キューの名称> out
```

実行例

```
mngsvrutil -m mnghost -u user01 -p pw1 -t ctm01 hold queue App1  
out
```

2. J2EE アプリケーションを入れ替えます。

J2EE アプリケーションを停止して、新しいアプリケーションに入れ替えます。そのあとで、J2EE アプリケーションを再開します。

J2EE アプリケーションの入れ替え方法については、「4.6.1 J2EE アプリケーションの入れ替え」を参照してください。

3. CTM のスケジュールキューの閉塞を解除するときは、mngsvrutil コマンドにサブコマンド「release」を指定して実行します。

この場合の mngsvrutil コマンドの実行形式と実行例を次に示します。

実行形式

```
mngsvrutil -m <Management Serverのホスト名> -u <管理ユーザID> -p <管理パスワード> -t <CTMの名称> release queue <キューの名称>
```

実行例

```
mngsvrutil -m mnghost -u user01 -p pw1 -t ctm01 release queue  
App1
```

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

### 4.6.7 テストモードでの J2EE アプリケーションの運用

ここでは、テストモードでの J2EE アプリケーションの運用について説明します。

J2EE アプリケーションを新しく登録する場合や、すでに動作中の J2EE アプリケー

ションを入れ替えるとき、本番運用を開始する前に、本番環境で問題なく動作することを確認したいことがあります。このような場合にテストモードを使用します。

テストモードでの運用は、J2EE アプリケーション単位で実行できます。テストモードから本番運用時のモード（通常モード）に変更する場合、J2EE アプリケーションの設定やカスタマイズをし直す必要はありません。また、一つの本番環境で表示名が同じ J2EE アプリケーションを、通常モードとテストモードで同時に開始できます。これによって、既存の J2EE アプリケーションでクライアントにサービスを提供しながら、入れ替える予定の J2EE アプリケーションの動作を同じ環境で確認できます。

テストモードで運用できるのは、J2EE アプリケーションだけです。J2EE アプリケーション内で DB Connector などのリソースアダプタを使用する場合は、本番時に使用するものを使用してください。また、Web サーバ上の静的コンテンツについても、本番用に構築した Web サーバ上に配置して使用してください。

### (1) 運用管理コマンド使用時の注意

運用管理コマンド (mngsvrutil) では、テストモードで動作している J2EE アプリケーションを操作できません。テストモードで J2EE アプリケーションを操作したい場合は、サーバ管理コマンドを実行してください。ただし、テストモードの場合も、運用管理コマンドで稼働情報を出力できます。

なお、J2EE アプリケーションのモードが、サーバ管理コマンドでテストモードに変更された場合、運用管理コマンドにサブコマンド「get」を指定して取得した J2EE アプリケーションの稼働情報の fullyQualifiedName に、「\_TEST」という文字列が付いて出力されます。

### (2) J2EE アプリケーションを入れ替える場合にテストモードを使用する手順

J2EE アプリケーションを入れ替えるときに、J2EE アプリケーションをテストモードで開始する手順の例を示します。なお、この手順では、入れ替え元の J2EE アプリケーションは、J2EE サーバ上で開始されていることを前提としています。

#### 準備

Web アプリケーションを含む J2EE アプリケーションをテストモードで運用する場合、あらかじめ次の作業が必要です。

- リダイレクタの設定 (URL のマッピング)

通常モードのコンテキストルートを含む URL のマッピングに加えて、テストモードのコンテキストルートを含む URL のマッピングを設定してください。Web アプリケーションを含む J2EE アプリケーションをテストモードで運用する場合、Web アプリケーションのコンテキストルートにプリフィックス「\_test\_」を付けたものが、コンテキストルートとして使用されます。例えば、通常モードのコンテキストルートが「/jsptobean」の場合、テストモードのコンテキストルートは

#### 4. J2EE アプリケーションとリソースの管理

「/\_test\_/jsptobean」になります。

URL パターンでのリクエスト振り分けの設定については、マニュアル

「Cosminexus システム構築ガイド」を参照してください。

- Web サーバの再起動

Web サーバを再起動して、リダイレクタの設定を有効にしてください。

#### 手順

##### 1. J2EE アプリケーションをテストモードでインポートします。

J2EE アプリケーションのインポートには、サーバ管理コマンド (cjimportapp) を使用します。

実行形式と実行例を次に示します。

#### 実行形式

```
cjimportapp <J2EEサーバ名> -test -f <EARファイルのパス>
```

#### 実行例

```
cjimportapp MyServer -test -f Appl.ear
```

##### 2. インポートした J2EE アプリケーションのプロパティ設定とカスタマイズを実行します。

サーバ管理コマンド (cjgetappprop および cjsetappprop) を使用して実行します。それぞれのコマンドには -test オプションを指定して、テストモードで実行します。

cjgetappprop コマンドの実行形式と実行例を次に示します。

#### 実行形式

```
cjgetappprop <J2EEサーバ名> -test -name <J2EEアプリケーション名> -type all -c <アプリケーション統合属性ファイルの出力先パス>
```

#### 実行例

```
cjgetappprop MyServer -test -name Appl -type all -c  
ApplProp.xml
```

属性ファイルはテキストエディタなどで編集します。

cjsetappprop コマンドの実行形式と実行例を次に示します。

#### 実行形式

```
cjsetappprop <J2EEサーバ名> -test -name <J2EEアプリケーション名> -type all -c <編集したアプリケーション統合属性ファイルのパス>
```



## 実行例

```
cjsetappprop MyServer -test -name Appl -type all -c
App1Prop.xml
```

- J2EE アプリケーションをテストモードで開始します。  
cjstartapp コマンドの実行形式と実行例を次に示します。

## 実行形式

```
cjstartapp <J2EEサーバ名> -test -name <J2EEアプリケーション名>
```

## 実行例

```
cjstartapp MyServer -test -name Appl
```

- テストを実施します。  
テスト用のクライアントから適宜操作を実行して、テストします。問題ないことが確認できたら、手順 5. に進みます。  
テスト結果に問題があった場合は、J2EE アプリケーションを修正して、手順 1. からもう一度実行してください。なお、同じ名称の J2EE アプリケーションは同じモードで登録できません。このため、テストが失敗したテストモードの J2EE アプリケーションは削除してください。テストモードの J2EE アプリケーションの削除は、サーバ管理コマンド (cjdeleteapp) に -test オプションを付けた形式で実行してください。

## ポイント

- テストモードで動作している J2EE アプリケーションは、テストモード用に設定した特定のクライアントからだけ利用できます。Web アプリケーションを含む J2EE アプリケーションをテストモードで運用する場合、Web アプリケーションのコンテキストルートにプリフィックス「\_test\_」を付けたものが、コンテキストルートとして使用されます。例えば、コンテキストルートが「/jsptobean」で、本番運用での URL が「http://www.hitachi.co.jp/jsptobean/account.jsp」の場合、テストモードでの URL は、「http://www.hitachi.co.jp/\_test\_/jsptobean/account.jsp」になります。テスト用のクライアントからはこの URL を使用して J2EE アプリケーションに対して操作を実行してください。
- EJB を含む J2EE アプリケーションをテストモードで運用する場合、EJB はテスト用の名前空間に登録されます。EJB クライアントアプリケーションから EJB を呼び出す場合には、通常モード用の名前空間のルートコンテキスト名で始まる検索文字列「HITACHI\_EJB/...」ではなく、テストモード用の名前空間のルートコンテキスト名で始まる検索文字列「\$HITACHI\_TEST/...」を指定して lookup を実行してください。

- テストモードで運用している J2EE アプリケーションを停止します。

## 実行形式

```
cjstopapp <J2EEサーバ名> -test -name <J2EEアプリケーション名>
```

#### 4. J2EE アプリケーションとリソースの管理

##### 実行例

```
cjstopapp MyServer -test -name App1
```

6. 通常モードで運用している J2EE アプリケーションを停止します。  
なお、J2EE アプリケーションを入れ替えではなく新規にインポートした場合、この操作は不要です。

##### 実行形式

```
cjstopapp <J2EEサーバ名> -name <J2EEアプリケーション名>
```

##### 実行例

```
cjstopapp MyServer -name App1
```

7. 通常モードで運用していた J2EE アプリケーションの名称を変更します。  
なお、J2EE アプリケーションを入れ替えではなく新規にインポートした場合、この操作は不要です。  
また、この操作は、インポートしたファイルの世代を管理したい場合などに、必要に応じて実行してください。例えば、「App1」という J2EE アプリケーションの場合、「App1\_bak」などに変更します。

##### 実行形式

```
cjrenameapp <J2EEサーバ名> -name <変更前のJ2EEアプリケーション名> -newname <変更後のJ2EEアプリケーション名>
```

##### 実行例

```
cjrenameapp MyServer -name App1 -newname App1_bak
```

8. テストモードで運用していた J2EE アプリケーションのモードを通常モードに変更します。

##### 実行形式

```
cjchmodapp <J2EEサーバ名> -test -name <テストモードで動作を確認したJ2EEアプリケーション名> -mode normal
```

##### 実行例

```
cjchmodapp MyServer -test -name App1 -mode normal
```

9. 手順 8. でモードを変更した J2EE アプリケーションを開始します。

##### 実行形式

```
cjstartapp <J2EEサーバ名> -name <J2EEアプリケーション名>
```

## 実行例

```
cjstartapp MyServer -name Appl
```

## (3) テストモードで実行できるサーバ管理コマンド

テストモードで実行できるサーバ管理コマンドを示します。

それぞれのコマンドの詳細については、マニュアル「Cosminexus リファレンス コマンド編」を参照してください。

表 4-16 テストモードで実行できるサーバ管理コマンド

コマンド	テストモードでの動作
cjchmodapp	テストモードでインポートした J2EE アプリケーションのモードを通常モードに切り替えられます。また、通常モードでインポートした J2EE アプリケーションのモードをテストモードに切り替えられます。
cjdeleteapp	テストモードの J2EE アプリケーションを削除できます。
cjgetappprop	テストモードの J2EE アプリケーションのアプリケーション属性、WAR 属性および EJB-JAR 属性を取得できます。
cjgetstubsjar	テストモードの J2EE アプリケーションから RMI-IIOP スタブおよびインタフェースを取得できます。
cjimportapp	J2EE アプリケーションをテストモードでインポートできます。インポートは、実行時情報を含む形式 (ZIP ファイル) または実行時情報を含まない形式 (EAR ファイル) のどちらも実行できます。
cjlistapp	J2EE サーバに含まれるすべての J2EE アプリケーションのうち、テストモードの J2EE アプリケーションの一覧を表示できます。また、テストモードの J2EE アプリケーションに含まれる WAR ファイルおよび EJB-JAR ファイルの一覧を表示できます。
cjlistthread	テストモードの J2EE アプリケーションのスレッド一覧を表示できます。
cjreplaceapp	テストモードの J2EE アプリケーションをリデプロイできます。
cjreloadapp	テストモードの J2EE アプリケーションをリロードできます。
cjsetappprop	テストモードの J2EE アプリケーションのアプリケーション属性、WAR 属性および EJB-JAR 属性を設定できます。
cjstartapp	J2EE アプリケーションをテストモードで開始できます。
cjstopapp	テストモードで開始した J2EE アプリケーションを停止できます。
cjstopthread	テストモードの J2EE アプリケーションに対してメソッドキャンセルを実行できます。

## 4.7 JavaBeans リソースの入れ替え

この節では、JavaBeans リソースを入れ替える手順について説明します。JavaBeans リソースは、J2EE アプリケーションおよび JavaBeans リソースを停止して、J2EE サーバを再起動してから入れ替えます。JavaBeans リソースの入れ替えには、サーバ管理コマンドを使用します。

手順を次に示します。

1. 入れ替える JavaBeans リソースを使用している J2EE アプリケーションを停止します。

J2EE アプリケーションを停止するには、`cjstopapp` コマンドを実行します。

`cjstopapp` コマンドの実行形式と実行例を次に示します。

実行形式

```
cjstopapp <J2EEサーバ名> -name <J2EEアプリケーション名>
```

実行例

```
cjstopapp MyServer -name Appl
```

2. 入れ替える JavaBeans リソースを停止します。

JavaBeans リソースを停止するには、`cjstopjb` コマンドを実行します。`cjstopjb` コマンドの実行形式と実行例を次に示します。

実行形式

```
cjstopjb <サーバ名称> -resname <JavaBeansリソース表示名>
```

実行例

```
cjstopjb MyServer -resname javabeansname
```

3. J2EE サーバを再起動します。

いったん開始した JavaBeans リソースを削除するには、J2EE サーバを再起動する必要があります。J2EE サーバを停止するには、`cjstopsv` コマンドを実行します。

`cjstopsv` コマンドの実行形式と実行例を次に示します。

実行形式

```
cjstopsv <J2EEサーバ名>
```

実行例

```
cjstopsv MyServer
```

J2EE サーバを停止したあと、もう一度開始します。J2EE サーバを開始するには、cjstartsv コマンドを実行します。cjstartsv コマンドの実行形式と実行例を次に示します。

#### 実行形式

```
cjstartsv <J2EEサーバ名>
```

#### 実行例

```
cjstartsv MyServer
```

4. 入れ替える JavaBeans リソースを削除します。

JavaBeans リソースを削除するには、cjdeletejb コマンドを実行します。cjdeletejb コマンドの実行形式と実行例を次に示します。

#### 実行形式

```
cjdeletejb <サーバ名称> -resname <JavaBeansリソース表示名>
```

#### 実行例

```
cjdeletejb MyServer -resname MyJavaBeans
```

5. 入れ替え後の JavaBeans リソースをインポートします。

JavaBeans リソースをインポートするには、cjimportjb コマンドを実行します。cjimportjb コマンドの実行形式と実行例を次に示します。

#### 実行形式

```
cjimportjb <サーバ名称> -f <JARファイルパス> -c <JavaBeansリソース属性ファイルパス>
```

#### 実行例

```
cjimportjb MyServer -f Myjavabeans.jar -c Myjavabeansprop.xml
```

6. JavaBeans リソースを開始します。

JavaBeans リソースを開始するには、cjstartjb コマンドを実行します。cjstartjb コマンドの実行形式と実行例を次に示します。

#### 実行形式

```
cjstartjb <サーバ名称> -resname <JavaBeansリソース表示名>
```

#### 実行例

```
cjstartjb MyServer -resname javabeansname
```

#### 4. J2EE アプリケーションとリソースの管理

##### 7. J2EE アプリケーションを開始します。

J2EE アプリケーションを開始するには、cjstartapp コマンドを実行します。

cjstartapp コマンドの実行形式と実行例を次に示します。

実行形式

```
cjstartapp <J2EEサーバ名> -name <J2EEアプリケーション名>
```

実行例

```
cjstartapp MyServer -name Appl
```

## 4.8 クラスタコネクションプールの運用

この節では、クラスタコネクションプールの運用について説明します。データベースに発生した障害に対応する場合や、データベースをメンテナンスする場合、一部のコネクションプールを手動で停止、再開することで、システム全体を止めることなくデータベースをメンテナンスできます。一部のコネクションプールを手動で停止、再開して、データベースをメンテナンスする作業は次の流れで行います。

1. コネクションプールの状態を確認する（4.8.1 参照）  
サーバ管理コマンドを使用して実行します。
2. コネクションプールを一時停止する（4.8.2 参照）  
サーバ管理コマンドを使用して実行します。
3. コネクションプールを再開する（4.8.3 参照）  
サーバ管理コマンドを使用して実行します。
4. コネクションプールの状態を確認する（4.8.1 参照）  
サーバ管理コマンドを使用して実行します。

### 4.8.1 コネクションプールの状態の確認

クラスタ構成のデータベースをメンテナンスする前、およびメンテナンスしたあとに、コネクションプールの状態を確認します。

コネクションプールの状態の詳細については「4.8.4 コネクションプールの状態」を参照してください。

コネクションプールの状態の確認は、`cjlistrar` コマンドで実行できます。

`cjlistrar` コマンドの実行形式と実行例を次に示します。

実行形式

```
cjlistrar <サーバ名> -clusterpool
```

実行例

```
cjlistrar MyServer -clusterpool
```

### 4.8.2 コネクションプールの一時停止

ここでは、コネクションプールを一時停止する手順について説明します。

コネクションプールの一時停止は、`cjsuspendpool` コマンドで実行できます。

`cjsuspendpool` コマンドを実行すると、コネクションプールが閉塞されて一時停止状態になり、コネクション取得要求を受け付けなくなります。

## 4. J2EE アプリケーションとリソースの管理

cjsuspendpool コマンドは、コネクションプールが次の状態の場合に実行できます。

- 開始
- 開始予約
- 自動一時停止
- 自動一時停止予約

コネクションプールの状態を確認する方法については、「4.8.1 コネクションプールの状態の確認」を参照してください。

cjsuspendpool コマンドの実行形式と実行例を次に示します。

実行形式

```
cjsuspendpool <サーバ名> -resname <一時停止対象となるメンバリソースアダプタの表示名>
```

実行例

```
cjsuspendpool MyServer -resname  
DB_Connector_for_Oracle_ClusterPool_Member
```

### 4.8.3 コネクションプールの再開

ここでは、コネクションプールを再開する手順について説明します。

コネクションプールの再開は、cjresumepool コマンドで実行できます。cjresumepool コマンドを実行すると、コネクションプールは手動再開中状態になり、再開処理が実行されます。再開処理が完了すると、コネクションプールは開始状態になり、コネクション取得要求を受け付けるようになります。

cjresumepool コマンドは、コネクションプールが次の場合に実行できます。

- 使用されていないコネクション管理スレッドが、コネクションプールのコネクションの最大値以上ある
- コネクションプールが、手動一時停止、手動一時停止予約、自動一時停止、または自動一時停止予約の状態である

コネクションプールの状態を確認する方法については、「4.8.1 コネクションプールの状態の確認」を参照してください。

cjresumepool コマンドの実行形式と実行例を次に示します。

実行形式

```
cjresumepool <サーバ名> -resname <再開対象となるメンバリソースアダプタの表示名>
```

実行例



```

cjresumepool MyServer -resname
DB_Connector_for_Oracle_ClusterPool_Member

```

なお、cjresumepool コマンドを実行したあと、コネクションプールの状態を確認して、再開処理が正しく実行されたかどうかを確認してください。コネクションプールの状態を確認する方法については、「4.8.1 コネクションプールの状態の確認」を参照してください。

## 4.8.4 コネクションプールの状態

ここでは、メンバコネクションプールの状態について説明します。

メンバコネクションプールの状態は、J2EE サーバやメンバリソースアダプタを再起動しても維持されます。コネクションプールの状態を確認する方法については、「4.8.1 コネクションプールの状態の確認」を参照してください。

なお、メンバリソースアダプタ以外のコネクションプールには状態はありません。

### (1) コネクションプールの状態遷移

メンバコネクションプールの状態遷移を次の図に示します。

図 4-1 メンバコネクションプールの状態遷移（手で一時停止を実行する場合）

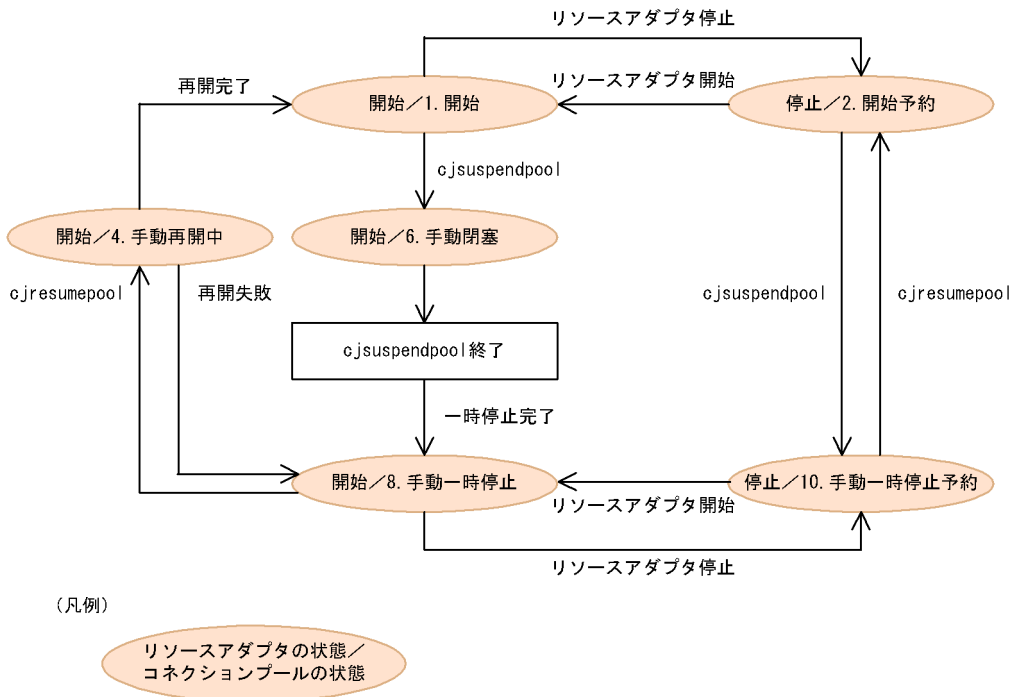
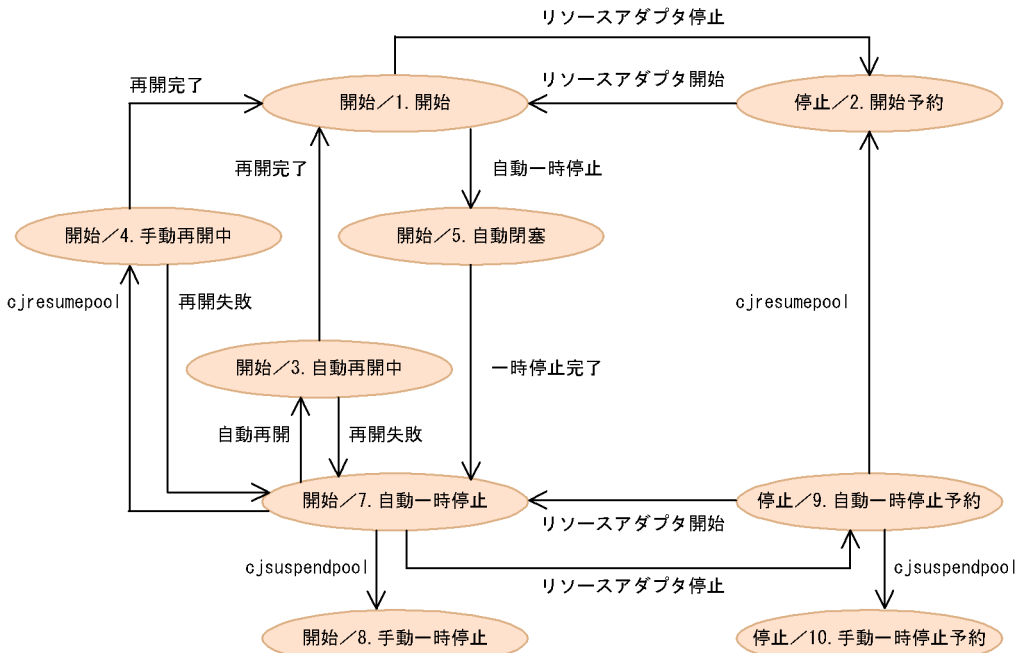


図 4-2 メンバコネクションプールの状態遷移（自動で一時停止が実行される場合）



(凡例)

リソースアダプタの状態 / コネクションプールの状態

なお、J2EE サーバの開始時にコネクションプールが再開中状態、または閉塞状態の場合、コネクションプールは一時停止状態に遷移します。

それぞれの状態の詳細について次の表に示します。

表 4-17 メンバコネクションプールの状態

番号	状態	説明
1	開始状態	コネクションプールが処理を受け付けている状態です。ルートリソースアダプタへのコネクション取得要求時には、開始状態のコネクションプールだけに処理が行われます。
2	開始予約状態	コネクションプールが開始状態のときにリソースアダプタが停止された状態です。開始予約状態のコネクションプールは、リソースアダプタの開始時に開始状態になります。リソースアダプタをデプロイした直後は、この状態になります。
3	自動再開中状態	自動再開機能によって、コネクションプールが再開処理を行っている状態です。再開処理が成功すると開始状態になります。再開処理が失敗すると、自動一時停止状態に戻ります。自動再開中状態のコネクションプールには、ルートリソースアダプタへのコネクション取得要求時に処理が出されません。

番号	状態	説明
4	手動再開中状態	自動一時停止状態、または手動一時停止状態のときに <code>cjresume</code> コマンドを実行して、コネクションプールが再開処理を行っている状態です。再開処理が成功すると、開始状態になります。再開処理が失敗すると、コマンド実行前の状態に戻ります。手動再開中状態のコネクションプールには、ルートリソースアダプタへのコネクション取得要求時に処理が出されません。
5	自動閉塞状態	自動一時停止機能によって、コネクションプールが閉塞されて、一時停止処理を行っている状態です。一時停止処理が完了すると、自動一時停止状態になります。自動閉塞状態のコネクションプールには、ルートリソースアダプタへのコネクション取得要求時に処理が出されません。
6	手動閉塞状態	<code>cjsuspend</code> コマンドによって、コネクションプールが閉塞され、一時停止処理を行っている状態です。一時停止処理が完了すると、手動一時停止状態になります。手動閉塞状態のコネクションプールには、ルートリソースアダプタへのコネクション取得要求時に処理が出されません。
7	自動一時停止状態	自動一時停止機能によって、コネクションプールが一時停止された状態です。コネクションプール内にコネクションは存在しません。自動一時停止状態のコネクションプールには、ルートリソースアダプタへのコネクション取得要求時に処理が出されません。
8	手動一時停止状態	<code>cjsuspend</code> コマンドによって、コネクションプールが一時停止された状態です。コネクションプール内にコネクションは存在しません。手動一時停止状態のコネクションプールには、ルートリソースアダプタへのコネクション取得要求時に処理が出されません。
9	自動一時停止予約状態	コネクションプールが自動一時停止状態のときにリソースアダプタを停止した状態です。自動一時停止予約状態のコネクションプールは、リソースアダプタの開始時に自動一時停止状態になります。なお、このとき、コネクションプールのウォーミングアップ機能は無効になります。
10	手動一時停止予約状態	コネクションプールが手動一時停止状態のときにリソースアダプタを停止した状態です。手動一時停止予約状態のコネクションプールは、リソースアダプタの開始時に手動一時停止状態になります。なお、このとき、コネクションプールのウォーミングアップ機能は無効になります。

注 番号は、図 4-1 および図 4-2 中の番号を示します。

## (2) コネクションプールの状態によるコマンド実行の可否

コネクションプールの状態によって、実行できるコマンドと実行できないコマンドがあります。コネクションプールの状態ごとに、それぞれのコマンドの実行の可否を次に示します。

4. J2EE アプリケーションとリソースの管理

表 4-18 コネクションプールの状態によるコマンド実行の可否

コマンド	コネクションプールの状態							
	開始	開始予約	自動再開中 / 手動再開中	自動閉塞 / 手動閉塞	自動一時停止	手動一時停止	自動一時停止予約	手動一時停止予約
cjstartrar	×		×	×	×	×		
cjstoprar		×	1	2			×	×
cjclearpool		×	×	×	×	×	×	×
cjlistrar								
cjsuspendpool			×	×		×		×
cjresumepool	×	×	×	×				
cjstopsv(通常停止時)			1	2				
cjstopsv(強制停止時)								

(凡例) : 実行できる : 制限あり × : 実行できない

注 1 コマンドは受け付けられませんが、開始または一時停止状態になってから処理が実行されません。

注 2 コマンドは受け付けられませんが、一時停止状態になってから処理が実行されます。

# 5

## トラブルシューティング

この章では、システムの運用中に発生したトラブルに対処する方法について説明します。

---

5.1 対処の手順

---

5.2 取得が必要な資料の種類

---

5.3 資料の取得

---

5.4 資料の個別取得

---

5.5 資料の調査

---

5.6 トラブルへの対処と回復

---

5.7 トラブルシューティングに関連する留意事項

---

## 5.1 対処の手順

この節では、Cosminexus のシステムの運用中にトラブルが発生した場合の、対処の手順について説明します。

なお、トラブル発生時に取得する情報のうち、次に示す情報を取得するためには、運用開始前の準備が必要です。

日立 JavaVM 固有のログ（日立 JavaVM ログファイル）

このファイルには、JavaVM のガーベージコレクションのログも出力されます。

ユーザダンプ（Windows の場合）

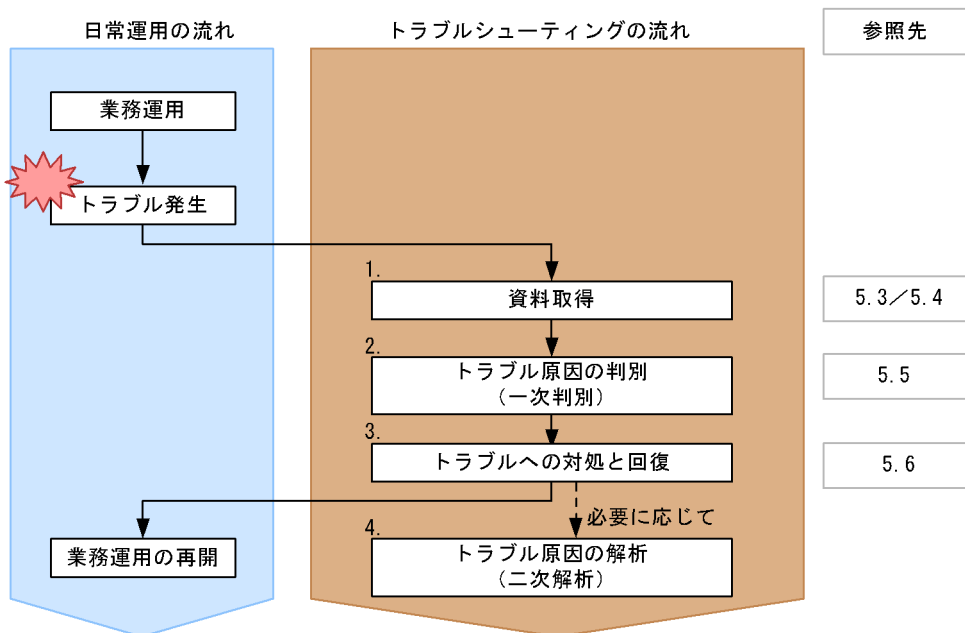
core ダンプ（UNIX の場合）

OS の統計情報（Windows の場合）

これらの情報については、必要に応じてシステム構築時に取得準備をしてください。詳細は、マニュアル「Cosminexus システム構築ガイド」のトラブルシューティングの資料取得の設定に関する説明を参照してください。

システムの運用中にトラブルが発生した場合、次の図に示す手順で対処します。

図 5-1 トラブル発生時の対処の流れ



### 1. 資料取得

トラブルシューティングに必要な資料をすべて取得します。

取得が必要な資料については、「5.2 取得が必要な資料の種類」を参照してください

い。なお、メモリダンプは、J2EE サーバまたは CORBA ネーミングサービスを再起動する場合にだけ必要な資料です。

また、資料の取得方法については、「5.3 資料の取得」、または「5.4 資料の個別取得」を参照してください。

---

#### 参考

- 資料の取得には、snapshot ログとして一括して取得する方法と、個別のログを取得する方法があります。
- snapshot ログの取得には、運用管理コマンドを使用して取得する方法、および設定によって自動で取得する方法があります。
- 論理サーバに異常が発生した場合の資料取得を自動化できます。詳細は、「5.3.2 トラブル発生時に自動的に取得できる資料」を参照してください。

---

#### 2. トラブル原因の判別（一次判別）

トラブルの原因を判別します。判別には、取得した資料を使用します。資料の調査方法については、「5.5 資料の調査」を参照してください。

#### 3. トラブルへの対処と回復

トラブルに対処して、回復します。

トラブルの種類ごとの対処方法については、「5.6 トラブルへの対処と回復」を参照してください。

#### 4. トラブル原因の解析（二次解析）

必要に応じてトラブル原因をさらに分析します。なお、この解析作業は保守員が実施します。

## 5.2 取得が必要な資料の種類

---

ここでは、トラブル発生時に取得が必要な資料の種類を示します。

取得が必要な資料は、発生したトラブルの種別によって異なります。ここでは、Cosminexus のシステムで想定されるトラブルの種別と、それに応じて取得が必要な資料の一覧を示します。

なお、トラブルの内容によっては、保守員に連絡する必要があります。保守員とは、ご購入契約に基づくお問い合わせ窓口のことです。

次のような場合は、保守員に連絡してください。

保守員に連絡が必要なトラブル

- 出力されたエラーメッセージの対処に「保守員に連絡してください。」とある場合
- トラブルの要因がわからない場合
- エラーメッセージの内容に対処できない場合

保守員に連絡する場合には、次の内容をできるだけ正確に伝えてください。

- エラーの発生時期
- エラー発生直前に実行した操作
- エラー発生時の画面の動作
- トラブル発生時に取得した資料

取得した資料を保守員に送付する場合、取得した資料は、次の2種類に分類できます。

メールなどで早急に送付する資料（一次送付資料）

メールの添付資料として送付できるファイルサイズの資料が該当します。

別途送付する資料（二次送付資料）

ファイルサイズが比較的大きく、保守員への送付に時間が掛かるため、メール以外の方法などで別途送付する必要がある資料が該当します。なお、二次送付資料には、一次送付資料の内容が含まれます。

### 5.2.1 トラブルの種別と取得が必要な資料

Cosminexus で想定されるトラブルには、次に示す種別があります。

エラーメッセージ出力

エラーメッセージが出力されます。

システムダウン

Cosminexus のサーバプロセスが異常終了します。

ハングアップ（無応答）

アプリケーション呼び出しの応答が返らなくなります。

スローダウン



アプリケーション呼び出しの応答は返りますが、応答時間が長くなります。

トラブル種別ごとに取得する資料を、次に示します。

表 5-1 トラブル種別ごとの取得資料一覧

項番	資料の種類	トラブル種別			
		エラーメッセージ出力	システムダウン	ハングアップ(無応答)	スローダウン
1	メッセージログ <sup>1</sup>				
2	ユーザログ <sup>1</sup>				
3	例外ログ <sup>1</sup>				
4	保守用ログ <sup>1</sup>				
5	性能解析トレース				
6	スレッドダンプ				
7	ガーベージコレクションのログ				
8	メモリダンプ	-			
9	日立 JavaVM ログファイル <sup>2</sup>	-			
10	エラーレポートファイル	-		-	-
11	OSの状態・ログ				
12	OSの統計情報				
13	定義情報				
14	作業ディレクトリ				
15	リソースの設定				
16	Web サーバのログ				
17	JavaVM のスタックトレース <sup>3</sup>	-		-	-

(凡例)

: 必須

: 取得するかどうかは、実際のエラーの内容によって異なる

- : 取得しない

注 1

Cosminexus の構成ソフトウェアが出力するログです。

注 2

usrconf.cfg の add.jvm.arg キーに特定のオプションを指定している場合に出力され

## 5. トラブルシューティング

ます。取得するためのオプションについては、「5.4.8 日立固有の JavaVM ログの取得（日立 JavaVM ログファイル）」を参照してください。また、オプションの詳細については、マニュアル「Cosminexus リファレンス 定義編」を参照してください。

### 注 3

UNIX を使用している場合だけ出力されます。

## 5.2.2 取得が必要な資料の一覧

Cosminexus でトラブルシューティングに必要な資料の一覧を次の表に示します。また、ここではそれぞれの資料を保守員に送付する場合の送付タイミングについても示します。

表 5-2 取得資料一覧

項番	資料の種類	説明	送付タイミング
1	メッセージログ <sup>1</sup>	J2EE サーバなどの稼働状態およびエラー情報が出力されます。	一次 / 二次
2	ユーザログ <sup>1</sup>	アプリケーション中で出力される標準出力および標準エラー出力の情報が出力されます。	一次
3	例外ログ <sup>1</sup>	システムでトラブルが発生したときの例外情報が出力されます。	一次
4	保守用ログ <sup>1</sup>	システムでトラブルが発生したときの障害保守情報が出力されます。保守員が障害解析用に使います。	一次 / 二次
5	性能解析トレース	リクエストの一連の処理で出力されるトレース情報が出力されます。	二次
6	JavaVM のスレッドダンプ	JavaVM の稼働情報やスレッドのスタック状態が出力されます。	一次
7	JavaVM のガーベージコレクションのログ	JavaVM のガーベージコレクションの活動状態が出力されます。	一次
8	メモリダンプ	プロセスのメモリイメージがダンプ出力されます。	二次
9	日立 JavaVM ログファイル <sup>2</sup>	Cosminexus Developer's Kit for Java で提供される日立 JavaVM の固有のログが出力されます。	一次
10	エラーレポートファイル	JavaVM がダウンした場合に出力されるログファイルです。	一次
11	OS の状態・ログ	OS の稼働情報の記録です。	一次 / 二次
12	OS の統計情報	OS の統計情報の記録です。	一次
13	定義情報	Cosminexus の各種設定の定義情報です。	一次
14	作業ディレクトリ	Cosminexus 稼働時に作業ファイルが格納されているディレクトリです。	一次 / 二次

項番	資料の種類	説明	送付タイミング
15	リソースの設定	Cosminexus 上で使用するリソース（データソース、リソースアダプタ）の設定情報です。	二次
16	Web サーバのログ	Hitachi Web Server, Microsoft IIS のログです。	一次
17	JavaVM のスタックトレース <sup>3</sup>	JavaVM が異常終了した場合の原因究明に使用する情報です。	一次 / 二次

（凡例）

一次：一次送付資料

二次：二次送付資料

注 1

Cosminexus の構成ソフトウェアが出力するログです。

注 2

usrconf.cfg の add.jvm.arg キーに特定のオプションを指定している場合に出力されます。取得するためのオプションについては、「5.4.8 日立固有の JavaVM ログの取得（日立 JavaVM ログファイル）」を参照してください。また、オプションの詳細について、マニュアル「Cosminexus リファレンス 定義編」を参照してください。

注 3

UNIX を使用している場合だけ出力されます。

ログファイルは、各構成ソフトウェアの環境設定ファイルで指定したディレクトリに自動的に作成されます。ファイル名には、1 ~ 指定したログファイルの面数の通し番号が付きます。ログファイルのファイルサイズが一面当たりの最大サイズに達すると、次のログファイルに出力先を切り替えます。指定したログファイルの面数分のファイルに出力したあとは、ファイル名の番号が 1 のログファイルに再度ログを出力します。なお、ログファイルの 1 レコードの出力が途切れないようにするために、ログファイルのファイルサイズが環境設定ファイルに指定したサイズより若干大きくなる場合があります。1 レコードの最大長は約 4 キロバイトです。

### 5.2.3 取得方法および調査方法との対応

ここでは、取得が必要な資料の取得方法と調査方法の参照先を示します。また、snapshot ログとして取得できる情報の種類についても示します。

## 5. トラブルシューティング

表 5-3 取得が必要な資料の取得方法と調査方法の参照先

項番	資料の種類	デフォルト設定の snapshot ログで 取得可 <sup>1</sup>	参照先	
			個別の収集 方法	調査方法
1	メッセージログ		5.4.1	5.5.1
2	ユーザログ		5.4.2	5.5.2
3	例外ログ		5.4.3	5.5.3
4	保守用ログ			
5	性能解析トレース		5.4.4	3.3 5.5.4
6	JavaVM のスレッドダンプ	2	5.4.5	5.5.5
7	JavaVM のガーベージコレクションの ログ	×	5.4.6	5.5.6
8	メモリダンプ	×	5.4.7	- 3
9	日立 JavaVM ログファイル <sup>4</sup>	×	5.4.8	5.5.7
10	エラーレポートファイル	×	5.4.9	5.5.8
11	OS の状態・ログ		5.4.10	5.5.9
12	OS の統計情報	×	5.4.11	- 3
13	定義情報		5.4.12	- 3
14	作業ディレクトリ	×	5.4.13	- 3
15	リソースの設定	×	5.4.14	- 3
16	Web サーバのログ	×	5.4.15	- 3
17	JavaVM のスタックトレース <sup>3 5</sup>	×	5.4.16	5.5.10

(凡例) : 取得できます。 × : 取得できません。

注 1

snapshot ログの収集対象ディレクトリを追加すれば取得できます。

注 2

あらかじめ障害検知時コマンドなどによって出力されている場合だけ収集できます ( snapshot ログ収集時には、これらのダンプを出力するコマンドは実行されません )。

注 3

保守員が使用する資料です。

注 4

usrconf.cfg の add.jvm.arg キーに特定のオプションを指定している場合に出力されます。取得するためのオプションについては、「5.4.8 日立固有の JavaVM ログの取得 (日立 JavaVM ログファイル)」を参照してください。また、各オプションの詳細について、マニュアル「Cosminexus リファレンス 定義編」を参照してください。

注 5

UNIX を使用している場合だけ出力されます。

## 5.3 資料の取得

---

ここでは、トラブル発生時の資料の取得方法について説明します。

### 5.3.1 資料の取得方法

トラブルシュートに必要な資料は、次のどれかの方法で取得できます。

トラブル発生時に自動的に取得する

運用管理コマンドを利用して一括取得する

それぞれの情報を個別に収集する

トラブルシュートに必要な情報は、snapshot ログとしてまとめて収集できます。snapshot ログとは、システムの構成ソフトウェアのトラブル発生時に、そのときの状態を出力したログのことです。Cosminexus で構築したシステムでは、各種構成ソフトウェアの snapshot ログを一括して収集して、ZIP 形式のログファイルとして出力できます。また、snapshot ログは、一次送付資料と二次送付資料に分けて取得できます。

snapshot ログは、次に示すタイミングで自動収集されます。

トラブルが発生した論理サーバが停止する直前

運用管理ドメインで管理している J2EE サーバが自動再起動される直前

J2EE サーバを手動で一括再起動する直前

また、運用管理コマンド (mngsvrutil) を使用して、任意のタイミングで snapshot ログを収集することもできます。

それぞれの情報を個別に収集する方法、および snapshot ログとして取得しない情報については、「5.4.1 Cosminexus のログの取得 (J2EE アプリケーションを実行するシステム)」または「5.4.2 Cosminexus のログの取得 (パッチアプリケーションを実行するシステム)」に示す方法で、個別に収集してください。

### 5.3.2 トラブル発生時に自動的に取得できる資料

Cosminexus システムでは、論理サーバに異常が発生した場合に、自動的にトラブルシュートに必要な情報を取得、収集できます。

ポイント

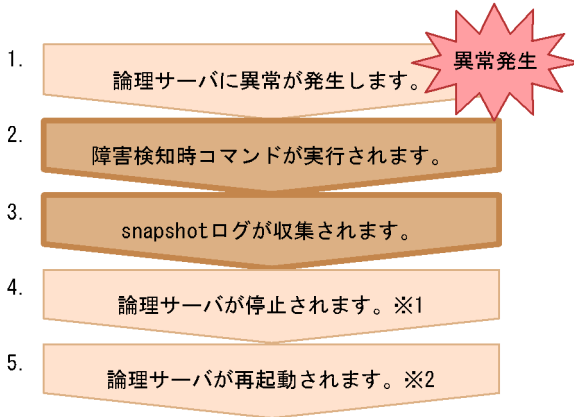
---

トラブル発生時に資料を自動で取得するためには、システムを構築する段階に必要な設定をしておく必要があります。トラブルシューティングの資料取得の設定については、マニュアル「Cosminexus システム構築ガイド」を参照してください。


---

自動で資料を取得する場合の、トラブル発生から資料の取得と収集、システムの回復までの流れを次の図に示します。

図 5-2 自動で資料を収集する場合の流れ



(凡例)

 : トラブルシューティングで使用する資料が取得されるタイミングです。

注 : 06-00との互換用の機能として、snapshotログを4.と5.の間に取得するように設定することもできます。ただし、その場合、論理サーバのうち、J2EEサーバにトラブルが発生して再起動するときだけsnapshotログが収集されます。

注※1 : 論理サーバがハングアップの場合だけ実行されます。

注※2 : 論理サーバがハングアップの場合およびプロセスが異常終了した場合だけ実行されます。ただし事前に再起動するための設定が必要です。

2.の障害検知時コマンドによって、トラブルシュートに必要な情報が出力されます。このコマンドで出力した情報、およびそれ以外のトラブルシュートに必要な情報を、3.でsnapshotログとしてまとめて収集します。

このほか、設定によっては、snapshotログは、運用管理ドメインで管理しているJ2EEサーバを手動で一括再起動する直前にも自動収集されます。

#### ポイント

障害検知時コマンドの実行とsnapshotログの収集によって、トラブルシューティングに必要な情報は、次の情報を除いてすべて自動収集するように設定できます。

- OSの統計情報 (Windowsの場合)

デフォルトの設定で収集されない情報については、次のように設定して収集します。

1. ユーザ作成の障害検知時コマンド内で、必要な情報を出力します。
2. 1.の情報の出力先を、snapshotログの収集対象のパスに追加します。

snapshotログ取得の設定については、マニュアル「Cosminexus システム構築ガイド」を参照してください。

## 5. トラブルシューティング

ここでは、2. の障害検知時コマンド実行時と、3. の snapshot ログ収集時に取得できる資料の種類について説明します。

### ！ 注意事項

snapshot ログで取得できる構成ソフトウェアのログのうち、次の表に示すログの取得については、このマニュアルでは説明していません。取得方法については、次の表の参照先マニュアルを参照してください。

表 5-4 このマニュアルでログの取得方法を記載していない構成ソフトウェア

ログの種類	参照先マニュアル名
Cosminexus Component Container の SOAP アプリケーション実行基盤のログ	Cosminexus SOAP アプリケーション開発ガイド
Cosminexus TPBroker のログ	TPBroker ユーザーズガイド
Cosminexus Web Services - Security のログ	Cosminexus Web サービスセキュリティ 使用の手引

### (1) 障害検知時コマンドで出力できる資料

障害検知時コマンドは、Management Server で管理している論理サーバにトラブルが発生した場合に、システムによって実行されます。

障害検知時コマンドには、次の 2 種類があります。

システムで提供されている障害検知時コマンド

ユーザが作成する障害検知時コマンド

ここでは、それぞれで取得できる情報について説明します。

#### ポイント

障害検知時コマンドは、システム構築時に必要な設定がされていない場合、実行されません。次の設定内容を確認してください。

- mserver.properties (Management Server 環境設定ファイル) の com.cosminexus.mngsvr.sys\_cmd.abnormal\_end.enabled キーに「true」が設定されている場合、システムで提供されている障害検知時コマンドが実行されます。
- mserver.properties (Management Server 環境設定ファイル) の com.cosminexus.mngsvr.usr\_cmd.abnormal\_end.enabled キーに「true」が設定されている場合、ユーザが作成した障害検知時コマンドが実行されます。

障害検知時コマンドによる資料取得の設定については、マニュアル「Cosminexus システム構築ガイド」を参照してください。



## (a) システムで提供されている障害検知時コマンドで出力できる資料

システムで提供されている障害検知時コマンドで出力できる資料について、次の表に示します。障害検知時コマンドで出力できる資料は、障害の種類（システムダウン、またはハングアップ）および使用している OS によって異なります。

なお、snapshot ログでこれらの情報を取得するためには、`adminagent.properties`（運用管理エージェントプロパティファイル）に必要な設定がされていることが必要です。

障害検知時コマンドで出力できる資料を次に示します。

表 5-5 障害検知時コマンドで出力できる資料

論理サーバ	OS	取得できる情報	
		プロセスダウンを検出した場合	ハングアップを検出した場合
論理パフォーマンストレーサ	Windows UNIX	<ul style="list-style-type: none"> <li>バッファの内容を出力した PRF トレースファイル</li> </ul>	<ul style="list-style-type: none"> <li>-</li> </ul>
論理 J2EE サーバ	Windows	<ul style="list-style-type: none"> <li>性能解析トレースファイル</li> </ul>	<ul style="list-style-type: none"> <li>スレッドダンプ</li> <li>性能解析トレースファイル</li> </ul>
	UNIX	<ul style="list-style-type: none"> <li>JavaVM のスタックトレース情報</li> <li>性能解析トレースファイル</li> </ul>	<ul style="list-style-type: none"> <li>スレッドダンプ</li> <li>性能解析トレースファイル</li> </ul>
論理 SFO サーバ	Windows	<ul style="list-style-type: none"> <li>性能解析トレースファイル</li> </ul>	<ul style="list-style-type: none"> <li>スレッドダンプ</li> <li>性能解析トレースファイル</li> </ul>
	UNIX	<ul style="list-style-type: none"> <li>JavaVM のスタックトレース情報</li> <li>性能解析トレースファイル</li> </ul>	<ul style="list-style-type: none"> <li>スレッドダンプ</li> <li>性能解析トレースファイル</li> </ul>
論理 Web サーバ	Windows UNIX	<ul style="list-style-type: none"> <li>性能解析トレースファイル</li> </ul>	<ul style="list-style-type: none"> <li>内部トレース</li> <li>性能解析トレースファイル</li> </ul>
ほかの論理サーバ	Windows UNIX	<ul style="list-style-type: none"> <li>性能解析トレースファイル</li> </ul>	<ul style="list-style-type: none"> <li>性能解析トレースファイル</li> </ul>

（凡例） - : 何も出力されません。

トラブルが発生した J2EE サーバまたは SFO サーバのスレッドダンプ `adminagent.properties`（運用管理エージェントプロパティファイル）の `adminagent.j2ee.sys_cmd.abnormal_end.threaddump` キーに「true」が設定されている場合に取得できます。

## 5. トラブルシューティング

### 性能解析トレース

adminagent.properties (運用管理エージェントプロパティファイル) の adminagent.sys\_cmd.abnormal\_end.prftrace キーに「true」が設定されている場合に取得できます。

#### ! 注意事項

トラブルが発生した J2EE サーバまたは SFO サーバのプロセスが存在しないと判断された場合、スレッドダンプは出力されません。また、パフォーマンストレーサのプロセスが存在しないと判断された場合、性能解析トレースは出力されません。

#### (b) ユーザが作成した障害検知時コマンドで出力できる情報

ユーザが作成した障害検知時コマンドとして、資料取得に必要な任意の処理を記述したバッチファイルまたはシェルスクリプトを実行できます。例えば、ユーザダンプまたは core ダンプは、このコマンドの中で drwtsn32 コマンドまたは cjdumpsv コマンドなどを実行して取得できます。障害検知時コマンドの作成方法については、マニュアル「Cosminexus システム構築ガイド」の障害検知時コマンドによる資料取得の設定に関する説明を参照してください。

なお、ユーザ作成の障害検知時コマンドは、adminagent.properties (運用管理エージェントプロパティファイル) の adminagent.<serverkind>.usr\_cmd.abnormal\_end キーに指定されているものが実行されます。

#### (2) snapshot ログとして収集できる資料

snapshot ログとして収集できる情報を示します。snapshot ログには、システム保守に必要な情報とアプリケーション保守に必要な情報が含まれます。

なお、snapshot ログとして収集する情報は、リストファイルの設定によって変更できます。例えば、ユーザ作成の障害検知時コマンドで出力した情報を収集したい場合には、設定を変更すれば収集対象に加えられます。リストファイルの設定はシステム構築時に実行します。

実際の運用環境での収集対象は、次のファイルで確認できます。

snapshotlog.conf

一次送付資料として取得する内容が定義されています。

snapshotlog.conf の格納場所を次に示します。

- Windows の場合

<Cosminexus のインストールディレクトリ >%manager%config%snapshotlog.conf

- UNIX の場合

/opt/Cosminexus/manager/config/snapshotlog.conf

snapshotlog.2.conf

二次送付資料として取得する内容が定義されています。

snapshotlog.2.conf の格納場所を次に示します。

- Windows の場合  
 <Cosminexus のインストールディレクトリ  
 >¥manager¥config¥snapshotlog.2.conf
- UNIX の場合  
 /opt/Cosminexus/manager/config/snapshotlog.2.conf

ただし、adminagent.properties（運用管理エージェントプロパティファイル）の指定内容によっては、収集できる情報が異なる場合があります。snapshot ログ取得の設定については、マニュアル「Cosminexus システム構築ガイド」を参照してください。

デフォルトの設定の場合、トラブル発生時に自動的に取得される snapshot ログには、次の情報が含まれます。

#### 一次送付資料

- snapshot ログの一次送付資料の収集対象として定義されているディレクトリ下の情報 <sup>1</sup>
- J2EE サーバおよび SFO サーバのスレッドダンプ <sup>2</sup>
- インストール情報
- OS の状態ログ

#### 二次送付資料

- 性能解析トレース <sup>2</sup>
- snapshot ログの二次送付資料の収集対象として定義されているディレクトリ下の情報 <sup>1</sup>
- インストール情報
- OS の状態ログ

注 1 snapshot ログの収集対象のディレクトリについては、「付録 B snapshot ログの収集対象一覧」を参照してください。

注 2 システムで提供されている障害検知時コマンドで、スレッドダンプおよび性能解析トレースを出力する設定にしている場合に収集できます。

次に、デフォルトの状態を取得できる情報と取得できない情報の例を示します。

#### (a) デフォルトの状態を取得できる情報

デフォルトの状態を取得できる情報の例を次に示します。

#### システム保守に必要な情報

システム保守に必要な情報として、次に示す構成ソフトウェアの定義ファイルおよびログファイルが収集されます。

- Cosminexus Component Container
- Cosminexus Component Transaction Monitor
- Cosminexus DABroker Library

## 5. トラブルシューティング

- Cosminexus Developer's Kit for Java
- Cosminexus Performance Tracer
- Cosminexus TPBroker
- Cosminexus Web Services - Security
- Hitachi Web Server
- Cosminexus cFramework

このほか、日立トレース共通ライブラリのログ、およびプログラムプロダクト情報（UNIX の場合）も収集します。

注 SOAP アプリケーション実行基盤の情報も含まれます。

### アプリケーション保守に必要な情報

アプリケーション保守に必要な情報として、Cosminexus Component Container のメッセージログおよびユーザログを収集します。

なお、デフォルトの状態では、構成ソフトウェアのインストール時にデフォルトで作成されたディレクトリが snapshot ログの収集対象として定義されています。ログの出力先を変更している場合は収集先も変更するようにしてください。

snapshot ログ取得の設定については、マニュアル「Cosminexus システム構築ガイド」、およびマニュアル「Cosminexus リファレンス 定義編」を参照してください。

### (b) デフォルトの状態では収集できない情報

次に示す情報は、デフォルトの状態では運用している場合は収集されません。必要に応じて、取得するように設定してください。snapshot ログ取得の設定については、マニュアル「Cosminexus システム構築ガイド」を参照してください。

### Windows の場合

#### Cosminexus Component Container 関連

- <Cosminexus のインストールディレクトリ>\%CC%\server\public\%jb%\<サーバ名称>\import 下すべて
- <Cosminexus のインストールディレクトリ>\%CC%\server\public\%jb%\<サーバ名称>\rars 下すべて
- <Cosminexus のインストールディレクトリ>\%CC%\server\public 下すべて
- Ear/Jar ファイル（デプロイ/インポートできなかった場合）

#### Microsoft IIS 関連

Microsoft IIS と連携している場合、次の資料を取得します。

- C:\%WINNT%\system32\LogFiles\W3SVC1（C: の部分にはシステムドライブを指定します）

#### OS 関連

- システムモニタ関連の資料一式（5.4.11 参照）
- イベントログ（アプリケーション、システム）

- OS の稼働資料  
winmsd 起動画面から [ 操作 ] - [ テキストファイルとして保存 ] メニューで資料採取します ( 5 分から 10 分掛かる場合があります )

## UNIX の場合

### Cosminexus Component Container 関連

- /opt/Cosminexus/CC/server/public/ejb/<サーバ名称>/import 下すべて
- /opt/Cosminexus/CC/server/public/ejb/<サーバ名称>/rars 下すべて
- /opt/Cosminexus/CC/server/public 下すべて
- Ear/Jar ファイル ( デプロイ / インポートできなかった場合 )

### OS 関連

- OS の稼働資料

## ( 3 ) 取得した情報の格納先

snapshot ログの収集によって自動で取得された情報は、adminagent.properties ( 運用管理エージェントプロパティファイル ) のキー「adminagent.snapshotlog.log\_dir」で指定したディレクトリに次の名称で格納されています。

一次送付資料として収集したファイル

snapshot-<論理サーバ名>-log-<yyyyMMddHHmmss >.zip

二次送付資料として収集したファイル

snapshot-<論理サーバ名>-log-<yyyyMMddHHmmss >.2.zip

注 yyyy : 西暦 MM : 月 dd : 日 HH : 時 mm : 分 ss : 秒

なお、出力先ディレクトリのデフォルトは、次のとおりです。

- Windows の場合  
<Manager のログ出力ディレクトリ >%snapshot
- UNIX の場合  
<Manager のログ出力ディレクトリ >/snapshot

## 参考

snapshot ログとして収集しようとしたファイルの一覧，および実際に収集したファイルの一覧は，次のファイルで確認できます。これらのファイルは，アーカイブされた zip ファイルに含まれています。

- 収集しようとしたファイルの一覧

```
snapshot-<ホスト名>-log-<yyyyMMddHHmmss >/filelist_pre.txt
snapshot-<ホスト名>-log-<yyyyMMddHHmmss >./filelist_pre.txt
```

- 実際に収集したファイルの一覧

```
snapshot-<ホスト名>-log-<yyyyMMddHHmmss >/filelist_post.txt
snapshot-<ホスト名>-log-<yyyyMMddHHmmss >./filelist_post.txt
```

注 yyy: 西暦 MM: 月 dd: 日 HH: 時 mm: 分 ss: 秒

### 5.3.3 snapshot ログの収集

snapshot ログは，自動実行されるほか，運用管理コマンド (mngsvrutil) を使用して任意のタイミングで収集できます。

なお，取得できる情報の種類は，トラブル発生時に自動的に収集される snapshot ログと同じです。「5.3.2(2) snapshot ログとして収集できる資料」を参照してください。

ログの出力先については，adminagent.properties ファイル (運用管理エージェントプロパティファイル) の adminagent.snapshotlog.log\_dir キーの値を参照してください。デフォルトの出力先は次のディレクトリです。

- Windows の場合  
<Manager のログ出力ディレクトリ>%snapshot
- UNIX の場合  
<Manager のログ出力ディレクトリ>/snapshot

snapshot ログを収集するには，運用管理コマンド (mngsvrutil) にサブコマンド「collect」を指定して実行します。-t オプションで指定したホスト内の snapshot ログが収集できます。

コマンドの実行形式と実行例を次に示します。この例では，一次送付ファイルと二次送付ファイルの両方を収集します。<n> には，一次送付資料または二次送付資料のどちらを収集するかを指定します。

#### 実行形式

```
mngsvrutil -m <Management Serverのホスト名> -u <管理ユーザID> -p <管理パスワード> -t <ホスト名> -k host collect snapshot <n>
```

#### 実行例

```
mngsvrutil -m mnghost -u user01 -p pw1 -t host01 -k host collect  
snapshot 1  
mngsvrutil -m mnghost -u user01 -p pw1 -t host01 -k host collect  
snapshot 2
```

## 5.4 資料の個別取得

Cosminexus の構成ソフトウェアが出力するログなどの資料を、snapshot ログを使用しないで個別に取得することもできます。ここでは、トラブル発生時に必要な資料を個別に取得する場合の方法について説明します。

snapshot ログを使用しない場合に個別に取得する必要がある資料と参照先を次の表に示します。

表 5-6 取得が必要な資料

取得する資料	参照先
Cosminexus のログ ( J2EE アプリケーションを実行するシステムの場合 )	5.4.1
Cosminexus のログ ( バッチアプリケーションを実行するシステムの場合 )	5.4.2
性能解析トレース	5.4.4
JavaVM のスレッドダンプ	5.4.5
JavaVM のガーベジコレクションログ	5.4.6
メモリダンプ	5.4.7
日立固有の JavaVM ログ ( 日立 JavaVM ログファイル )	5.4.8
JavaVM 出力メッセージログ ( 標準出力またはエラーレポートファイル )	5.4.9
OS の状態情報, OS のログ	5.4.10
OS の統計情報	5.4.11
Cosminexus の定義情報	5.4.12
Cosminexus の作業ディレクトリの内容	5.4.13
Cosminexus のリソース設定情報	5.4.14
Web サーバログ	5.4.15
JavaVM のスタックトレース情報	5.4.16

なお、EJB クライアントアプリケーションを実行するシステムの場合は、EJB クライアントアプリケーションのシステムログも取得してください。EJB クライアントアプリケーションのシステムログの取得方法については「5.4.3 EJB クライアントアプリケーションのシステムログの取得」を参照してください。

### 5.4.1 Cosminexus のログの取得 ( J2EE アプリケーションを実行するシステム )

ここでは、Cosminexus の構成ソフトウェアが出力するログを手動で取得する方法につ



いて説明します。なお、snapshot ログとして Cosminexus のログをすでに収集している場合は、ここで説明する操作は不要です。

ここでは、次のログの取得方法について説明します。

- Cosminexus Component Container のログ
- Cosminexus Performance Tracer のログ
- Cosminexus Component Transaction Monitor のログ
- Cosminexus DABroker Library のログ
- 監査ログで出力するログ
- アプリケーションのユーザログ
- 構築済み実行環境の切り替えで出力するログ

#### 参考

---

Hitachi Web Server のログの取得方法については、「5.4.15 Web サーバログの取得」で説明します。

---

### (1) Cosminexus Component Container のログの取得

Cosminexus Component Container のログの種類とログの出力先について説明します。Cosminexus Component Container のログには、次の 3 種類のログがあります。

- J2EE サーバ・リダイレクタ・サーバ管理コマンドのログ
- 運用管理エージェント・運用監視エージェント・Management Server のログ
- 統合ユーザ管理のログ

それぞれのログの出力先について説明します。

#### ! 注意事項

Web Redirector を使用する場合は、リダイレクタのログの格納ディレクトリの「<Cosminexus のインストールディレクトリ>¥CC」を、「<Cosminexus のインストールディレクトリ>¥CCR」と読み替えてください。

---

#### (a) J2EE サーバ・リダイレクタ・サーバ管理コマンドのログの取得

J2EE サーバ・リダイレクタ・サーバ管理コマンドのログの取得方法について説明します。

また、Cosminexus Component Container では、これらのログに加えて、移行コマンドのログが出力されます。インプロセス HTTP サーバを使用している場合はインプロセス HTTP サーバのログが、リソース枯渇監視機能を使用している場合はリソース枯渇監視ログが、Server Plug-in を使用しているときは Server Plug-in のログが出力されます。

J2EE サーバおよびリダイレクタのログには、メッセージログ、ユーザログ、例外ロ

## 5. トラブルシューティング

グ、および保守用ログの 4 種類があります。なお、J2EE サーバでは、これらのログに加えて、起動、停止および異常終了時にイベントログまたは syslog を出力します。

サーバ管理コマンドのログには、メッセージログ、例外ログ、および保守用ログの 3 種類があります。

リソースアダプタのバージョンアップコマンド (cjrupdate) のログには、メッセージログ、例外ログ、および保守用ログの 3 種類があります。

インプロセス HTTP サーバのログには、アクセスログ、スレッドトレース、および通信トレースの 3 種類があります。

移行コマンドのログには、メッセージログ、例外ログ、および保守用ログの 3 種類があります。

次にそれぞれのログについて説明します。

### メッセージログ

J2EE サーバ、サーバ管理コマンド、移行コマンドなどの稼働状態が出力されます。各種サーバおよびコマンドの稼働監視の情報として使用します。

### ユーザログ

アプリケーション中で出力される標準出力および標準エラー出力の情報が出力されます。アプリケーションの開発時の動作確認用に使用します。Solaris 以外の場合、JavaVM のメモリ関連ログも含まれます。

### 例外ログ

システムでトラブルが発生したときの Cosminexus Component Container の例外情報が出力されます。なお、例外ログは日常的な運用で監視する必要はありません。ログにメッセージが出力された場合に、例外情報を参照するときにご利用ください。

### 保守用ログ

システムでトラブルが発生したときの Cosminexus Component Container の障害保守情報が出力されます。保守員が Cosminexus Component Container の障害解析用に使用します。

### イベントログ (Windows の場合)

J2EE サーバが起動、停止または異常終了したことを示す情報が出力されます。出力先は Windows のイベントログの設定によって異なります。

なお、イベントログは、J2EE サーバの停止のしかたによっては、出力されません。次の場合は、正しくログが出力されないことがあります。

- J2EE サーバが動作している JavaVM 自体に問題が発生した場合
  - J2EE サーバのプロセスを TerminateProcess によって外部から停止した場合
  - JavaVM の起動オプションとして -XX:+HitachiOutOfMemoryAbort オプションを指定している場合にメモリ不足によって J2EE サーバが異常終了したとき
- なお、-XX:+HitachiOutOfMemoryAbort オプションは、デフォルトで設定されて

いるオプションです。

#### syslog ( UNIX の場合 )

J2EE サーバが起動、停止または異常終了したことを示す情報が出力されます。出力先は UNIX の syslog の設定によって異なります。

なお、syslog は、J2EE サーバの停止のしかたによっては、出力されません。次の場合は、正しくログが出力されないことがあります。

- J2EE サーバが動作している JavaVM 自体に問題が発生した場合
- J2EE サーバのプロセスを SIGKILL シグナル ( kill -9 など ) によって外部から停止した場合
- JavaVM の起動オプションとして -XX:+HitachiOutOfMemoryAbort オプションを指定している場合にメモリ不足によって J2EE サーバが異常終了したとき  
-XX:+HitachiOutOfMemoryAbort オプションは、デフォルトで設定されているオプションです。

#### インプロセス HTTP サーバのログ

インプロセス HTTP サーバを使用している場合に、保守情報およびリクエストの処理結果が出力されます。保守情報は、保守員が Cosminexus Component Container の障害解析用に使います。リクエストの処理結果はアクセスログとして、インプロセス HTTP サーバのパフォーマンス情報、セッショントラッキング情報などの分析に使います。

#### リソース枯渇監視ログ

リソース枯渇監視機能を使用している場合に、監視対象のリソースについてのリソース枯渇監視情報が出力されます。リソースの使用率または使用数がしきい値を超えた場合の原因調査に使います。

ログは、若い面番号の付いたログファイルから順に記録されます。一つのログファイルのサイズが 1 面当たりの最大サイズに達すると、ログは次の面番号の付いたログファイルに記録されます。最後のログファイル ( 面数の番号が付いたログファイル ) のサイズが 1 面当たりの最大サイズに達すると、面の番号 1 のログファイルを空にし、そこへログを記録していきます。以降、ログファイルを空にしながら、面番号の順にログファイルへログを記録していきます。

ログの出力先のデフォルトを次の表に示します。Cosminexus Component Container のログは、サーバ単位またはコマンド単位に取得できます。

ログの出力先に示す < 作業ディレクトリ > は、簡易構築定義ファイルの論理 J2EE サーバ ( j2ee-server ) の < configuration > タグ内に、ejb.public.directory パラメータで指定したディレクトリを指します。デフォルト値は、< Cosminexus のインストールディレクトリ >¥CC¥server¥public ( Windows の場合 )、または /opt/Cosminexus/CC/server/public ( UNIX の場合 ) です。

5. トラブルシューティング

J2EE サーバおよびリダイレクタのログ

表 5-7 J2EE サーバおよびリダイレクタのログの出力先 (デフォルト)

分類	内容	ログ出力先およびログファイル名 <sup>1</sup>	デフォルトのサイズ × 面数	チャンネル名
メッセージログ	稼働ログ	<ul style="list-style-type: none"> <li>Windows の場合 &lt;ejb.server.log.directory&gt; <sup>2</sup>¥cjmessage[n].log</li> <li>UNIX の場合 &lt;ejb.server.log.directory&gt; <sup>2</sup>/ cjmessage[n].log</li> </ul>	1MB × 2	MessageLog File
	ログ稼働ログ	<ul style="list-style-type: none"> <li>Windows の場合 &lt;ejb.server.log.directory&gt; <sup>2</sup>¥cjlogger.log</li> <li>UNIX の場合 &lt;ejb.server.log.directory&gt; <sup>2</sup>/ cjlogger.log</li> </ul>	-	MessageLog File
	リダイレクタのメッセージログ	ディレクトリ <sup>3</sup> <ul style="list-style-type: none"> <li>Windows の場合 &lt;Cosminexus のインストール ディレクトリ &gt;¥CC¥web¥redirector¥logs</li> <li>UNIX の場合 /opt/Cosminexus/CC/web/ redirector/logs</li> </ul>	4MB × 5	-
		ファイル名 <sup>3</sup> <ul style="list-style-type: none"> <li>Windows の場合 isapi_redirect[n].log ( Microsoft IIS ) hws_redirect[n].log ( Hitachi Web Server )</li> <li>UNIX の場合 hws_redirect[n].log ( Hitachi Web Server )</li> </ul>		

分類	内容	ログ出力先およびログファイル名 1	デフォルトのサイズ × 面数	チャネル名
	J2EE リソースアダプタとしてデプロイして使用するリソースアダプタの稼働ログ <sup>4</sup>	<ul style="list-style-type: none"> <li>• Windows の場合 ( Connector 1.0 仕様のリソースアダプタ ) &lt;ejb.server.log.directory&gt; <sup>2</sup>¥connectors¥&lt; リソースアダプタの表示名 &gt;[n].log ( Connector 1.5 仕様のリソースアダプタ ) &lt;ejb.server.log.directory&gt; <sup>2</sup>¥connectors¥&lt; リソースアダプタの表示名 &gt;_&lt; コネクション定義の並び順 &gt;_[n].log</li> <li>• UNIX の場合 ( Connector 1.0 仕様のリソースアダプタ ) &lt;ejb.server.log.directory&gt; <sup>2</sup>/ connectors/&lt; リソースアダプタの表示名 &gt;[n].log ( Connector 1.5 仕様のリソースアダプタ ) &lt;ejb.server.log.directory&gt; <sup>2</sup>/ connectors/&lt; リソースアダプタの表示名 &gt;_&lt; コネクション定義の並び順 &gt;_[n].log</li> </ul>	1MB × 2	-

5. トラブルシューティング

分類	内容	ログ出力先およびログファイル名 1	デフォルトのサイズ × 面数	チャネル名
	J2EE アプリケーションに含めて使用するリソースアダプタの稼働ログ <sup>4</sup>	<ul style="list-style-type: none"> <li>• Windows の場合 (通常モード / Connector 1.0 仕様のリソースアダプタ) &lt;ejb.server.log.directory&gt; <sup>2</sup>¥connectors¥&lt;J2EE アプリケーション名&gt;¥&lt;リソースアダプタの表示名&gt;[n].log (テストモード / Connector 1.0 仕様のリソースアダプタ) &lt;ejb.server.log.directory&gt; <sup>2</sup>¥connectors¥test#&lt;J2EE アプリケーション名&gt;¥&lt;リソースアダプタの表示名&gt;[n].log (通常モード / Connector 1.5 仕様のリソースアダプタ) &lt;ejb.server.log.directory&gt; <sup>2</sup>¥connectors¥&lt;J2EE アプリケーション名&gt;¥&lt;リソースアダプタの表示名&gt;_&lt;コネクション定義の並び順&gt;_[n].log (テストモード / Connector 1.5 仕様のリソースアダプタ) &lt;ejb.server.log.directory&gt; <sup>2</sup>&gt;¥connectors¥test#&lt;J2EE アプリケーション名&gt;¥&lt;リソースアダプタの表示名&gt;_&lt;コネクション定義の並び順&gt;_[n].log</li> </ul>	1MB × 2	-

分類	内容	ログ出力先およびログファイル名 1	デフォルトのサイズ × 面数	チャンネル名
		<ul style="list-style-type: none"> <li>• UNIX の場合 (通常モード / Connector 1.0 仕様のリソースアダプタ) &lt;ejb.server.log.directory&gt; 2/ connectors/&lt;J2EE アプリケーション名&gt;/&lt;リソースアダプタの表示名&gt;[n].log</li> <li>(テストモード / Connector 1.0 仕様のリソースアダプタ) &lt;ejb.server.log.directory&gt; 2/ connectors/test#&lt;J2EE アプリケーション名&gt;/&lt;リソースアダプタの表示名&gt;[n].log</li> <li>(通常モード / Connector 1.5 仕様のリソースアダプタ) &lt;ejb.server.log.directory&gt; 2/ connectors/&lt;J2EE アプリケーション名&gt;/&lt;リソースアダプタの表示名&gt;_&lt;コネクション定義の並び順&gt;_[n].log</li> <li>(テストモード / Connector 1.5 仕様のリソースアダプタ) &lt;ejb.server.log.directory&gt; 2/ connectors/test#&lt;J2EE アプリケーション名&gt;/&lt;リソースアダプタの表示名&gt;_&lt;コネクション定義の並び順&gt;_[n].log</li> </ul>		
ユーザログ	Web サブレットログ 5	<ul style="list-style-type: none"> <li>• Windows の場合 &lt;ejb.server.log.directory&gt; 2¥web_servlet[n].log</li> <li>• UNIX の場合 &lt;ejb.server.log.directory&gt; 2/ web_servlet[n].log</li> </ul>	4MB × 4	WebSe rvletL ogFile
	ユーザ出力ログ	<ul style="list-style-type: none"> <li>• Windows の場合 &lt;ejb.server.log.directory&gt; 2¥user_out[n].log</li> <li>• UNIX の場合 &lt;ejb.server.log.directory&gt; 2/ user_out[n].log</li> </ul>	1MB × 2	UserO utLog File

## 5. トラブルシューティング

分類	内容	ログ出力先およびログファイル名 <sup>1</sup>	デフォルトのサイズ × 面数	チャンネル名
	ユーザエラーログ	<ul style="list-style-type: none"> <li>Windows の場合 &lt;ejb.server.log.directory&gt; <sup>2</sup>¥user_err[n].log</li> <li>UNIX の場合 &lt;ejb.server.log.directory&gt; <sup>2</sup>/ user_err[n].log</li> </ul>	1MB × 2	UserErrorLogFile
	JavaVM の保守情報およびガーベージコレクションのログ	<ul style="list-style-type: none"> <li>Windows の場合 &lt;ejb.server.log.directory&gt; <sup>2</sup>¥javalog[nn].log</li> <li>UNIX の場合 &lt;ejb.server.log.directory&gt; <sup>2</sup>/ javalog[nn].log</li> </ul>	256KB × 4	-
例外ログ	障害発生時の例外情報	<ul style="list-style-type: none"> <li>Windows の場合 &lt;ejb.server.log.directory&gt; <sup>2</sup>¥cjexception[n].log</li> <li>UNIX の場合 &lt;ejb.server.log.directory&gt; <sup>2</sup>/ cjexception[n].log</li> </ul>	1MB × 2	ExceptionLogFile
保守用ログ	保守情報	<ul style="list-style-type: none"> <li>Windows の場合 &lt;ejb.server.log.directory&gt; <sup>2</sup>¥CC¥maintenance¥cjmaintenance[n].log</li> <li>UNIX の場合 &lt;ejb.server.log.directory&gt; <sup>2</sup>/ CC/maintenance/ cjmaintenance[n].log</li> </ul>	16MB × 4	MaintenanceLogFile
	コンソールメッセージ	<ul style="list-style-type: none"> <li>Windows の場合 &lt;ejb.server.log.directory&gt; <sup>2</sup>¥CC¥maintenance¥cjconsole[n].log</li> <li>UNIX の場合 &lt;ejb.server.log.directory&gt; <sup>2</sup>/ CC/maintenance/ cjconsole[n].log</li> </ul>	1MB × 2	ConsoleLogFile
	EJB コンテナの保守情報	<ul style="list-style-type: none"> <li>Windows の場合 &lt;ejb.server.log.directory&gt; <sup>2</sup>¥CC¥maintenance¥cjejbcontainer[n].log</li> <li>UNIX の場合 &lt;ejb.server.log.directory&gt; <sup>2</sup>/ CC/maintenance/ cjejbcontainer[n].log</li> </ul>	1MB × 2	EJBContainerLogFile



分類	内容	ログ出力先およびログファイル名 <sup>1</sup>	デフォルトのサイズ×面数	チャンネル名
	Web コンテナの保守情報	<ul style="list-style-type: none"> <li>Windows の場合 &lt;ejb.server.log.directory&gt; <sup>2</sup>¥CC¥maintenance¥cjwebcontainer[n].log</li> <li>UNIX の場合 &lt;ejb.server.log.directory&gt; <sup>2</sup>/ CC/maintenance/ cjwebcontainer[n].log</li> </ul>	1MB × 2	WebContainerLogFile
	起動プロセス標準出力情報 <sup>6</sup>	<ul style="list-style-type: none"> <li>Windows の場合 &lt;ejb.server.log.directory&gt; <sup>2</sup>¥CC¥maintenance¥cjstdout.log</li> <li>UNIX の場合 &lt;ejb.server.log.directory&gt; <sup>2</sup>/ CC/maintenance/cjstdout.log</li> </ul>	-	-
	起動プロセス標準エラー情報 <sup>6</sup>	<ul style="list-style-type: none"> <li>Windows の場合 &lt;ejb.server.log.directory&gt; <sup>2</sup>¥CC¥maintenance¥cjstderr.log</li> <li>UNIX の場合 &lt;ejb.server.log.directory&gt; <sup>2</sup>/ CC/maintenance/cjstderr.log</li> </ul>	-	-
	終了プロセス情報	<ul style="list-style-type: none"> <li>Windows の場合 &lt;ejb.server.log.directory&gt; <sup>2</sup>¥CC¥maintenance¥cj_shutdown[n].log</li> <li>UNIX の場合 &lt;ejb.server.log.directory&gt; <sup>2</sup>/ CC/maintenance/ cj_shutdown[n].log</li> </ul>	4KB × 2 <sup>7</sup>	-
	リダイレクタの保守用トレースログ	<p>ディレクトリ<sup>3</sup></p> <ul style="list-style-type: none"> <li>Windows の場合 &lt;Cosminexus のインストールディレクトリ &gt;¥CC¥web¥redirector¥logs</li> <li>UNIX の場合 /opt/Cosminexus/CC/web/ redirector/logs/</li> </ul>	16MB × 4	-

## 5. トラブルシューティング

分類	内容	ログ出力先およびログファイル名 <sup>1</sup>	デフォルトのサイズ × 面数	チャンネル名
		ファイル名 <sup>3</sup> <ul style="list-style-type: none"> <li>Windows の場合 iis_rd_trace[n].log ( Microsoft IIS ) hws_rd_trace[n].log ( Hitachi Web Server )</li> <li>UNIX の場合 hws_rd_trace[n].log ( Hitachi Web Server )</li> </ul>		
イベントログ	J2EE サーバの起動、停止または異常終了を示すログ	Windows のイベントビューアのアプリケーションログ <sup>8</sup>	-	-
syslog	J2EE サーバの起動、停止または異常終了を示すログ	UNIX の syslog の設定に依存します。 <sup>9</sup>	-	-

### 注

チャンネル名は、ログの出力先を識別する名称です。ログの属性（サイズ、面数）を変更する場合のキー値として使用します。

### 注 1

ログファイル名の [n] の部分には、面の番号（1 から面数（最大 16）まで）が付きます。ただし、リダイレクタで Microsoft IIS 6.0 を使用する場合、または EJB クライアントアプリケーションのサブディレクトリ共有モードを使用する場合は、最大面数が 64 になります。また、[nn] の部分には、01 ~ 99 の通し番号が付きます。

### 注 2

<ejb.server.log.directory> は、簡易構築定義ファイルの論理 J2EE サーバ ( j2ee-server ) の <configuration> タグ内に、ejb.server.log.directory パラメータで指定したディレクトリを指します。デフォルト値は、<Cosminexus のインストールディレクトリ>¥CC¥server¥public¥ejb¥<サーバ名称>¥logs です。

簡易構築定義ファイルの ejb.server.log.directory パラメータの詳細については、マニュアル「Cosminexus 簡易構築・運用ガイド」を参照してください。

### 注 3

ディレクトリ、ファイル名は、isapi\_redirect.conf ( Microsoft IIS 用リダイレクタ動作定義ファイル )、または mod\_jk.conf ( Hitachi Web Server 用リダイレクタ動作定義ファイル ) で設定できます。リダイレクタのログ取得の設定については、マニュアル「Cosminexus システム構築ガイド」を参照してください。各ファイルについては、マニュアル「Cosminexus リファレンス 定義編」を参照してください。

リダイレクタのメッセージログに出力されるメッセージのうち、KDJE21000-Z ~ KDJE21999-Z ( Z は変数で、E ( エラーメッセージ )、W ( 警告メッセージ )、I ( 通知メッセージ ) を示します ) は、保守用のメッセージとなります。

### 注 4

リソースアダプタのログを取得するかどうかは、サーバ管理コマンド、または Server Plug-in で指定されている内容に従います。また、リソースアダプタのログは、簡易構築定義ファイルでサイズおよび面数を変更できます。リソースアダプタのログ取得の設定については、マニュアル「Cosminexus システム構築ガイド」を参照してください。

注 5

サブレット、JSP で発生した例外のスタックトレースについても出力されます。

注 6

起動プロセス情報だけを取得するログです。主に J2EE サーバの起動または終了時に出力されるため、オンライン中に出力されることはほとんどありません。ファイルのサイズが上限に達したときは、<作業ディレクトリ>%ejb%<サーバ名称>%logs 下 (Windows の場合)、または <作業ディレクトリ>%ejb/<サーバ名称>/logs 下 (UNIX の場合) の cjstdout\_save.log または cjstderr\_save.log に退避されます。すでに cjstdout\_save.log または cjstderr\_save.log があつたときは、上書きされます。

注 7

サイズおよび面数は、変更できません。

注 8

ログファイルの出力先は Windows のイベントログの設定によって異なります。

注 9

J2EE サーバの起動、停止および異常終了のメッセージを syslog に出力するためには、syslog の設定で、facility 「daemon」に対する priority を「info」または「debug」に設定する必要があります。syslog の設定に関する詳細については、OS 付属のマニュアルを参照してください。

### ! 注意事項

- Web サーバとして Hitachi Web Server (UNIX) を使用している場合、リダイレクタのログファイルは、Web サーバの起動時に、mod\_jk.conf で指定したファイルサイズで、指定した面数分、すべて作成されます。このとき、トレースファイルは空白 (0x20) で初期化されます。デフォルトのファイルサイズと面数で運用している場合、ログの出力先に指定したディレクトリに、84 メガバイトのディスク容量が必要になります。
- Web サーバとして Microsoft IIS 6.0 を使用している場合、リダイレクタのログファイルは、Web サーバへの初回アクセス時に、isapi\_redirect.conf で指定したファイルサイズで、指定した面数分、すべて作成されます。このとき、トレースファイルは空白 (0x20) で初期化されます。デフォルトのファイルサイズと面数で運用している場合、ログの出力先に指定したディレクトリに、84 メガバイトのディスク容量が必要になります。
- UNIX の場合、リダイレクタの保守用トレースログを取得する前に次のコマンドを実行してください。

```
/opt/hitachi/HNTRLib2/bin/hntr2sync <JkTraceLogFileDir の指定値 >/
<JkTraceLogFilePrefix の指定値 >
```

なお、JkTraceLogFileDir および JkTraceLogFilePrefix は mod\_jk.conf (Hitachi Web Server 用リダイレクタ動作定義ファイル) に指定するキーです。

## 5. トラブルシューティング

### 参考

セッションフェイルオーバ機能を使用している場合、セッションフェイルオーバについてのログは次の出力先に出力されます。

- セッションフェイルオーバ機能のログ  
セッションフェイルオーバ機能を使用している J2EE サーバのログとして出力されます。
- SFO サーバのログ  
SFO サーバとして動作している J2EE サーバのログは、J2EE サーバと同じディレクトリに出力されます。また、SFO サーバ上で動作する SFO サーバアプリケーションのログも、SFO サーバアプリケーションが動作している J2EE サーバのログに出力されます。

### サーバ管理コマンドのログ

表 5-8 サーバ管理コマンドのログの出力先 (デフォルト)

分類	内容	ログ出力先およびログファイル名 <sup>1</sup>	デフォルトのサイズ × 面数	チャネル名
メッセージログ	稼働ログ <sup>2 3</sup>	<ul style="list-style-type: none"> <li>• Windows の場合 &lt;Cosminexus のインストールディレクトリ&gt;%CC%\admin\logs\%cjmessage[n].log</li> <li>• UNIX の場合 /opt/Cosminexus/CC/admin/logs/cjmessage[n].log</li> </ul>	1024KB × 3	MessageLog File
	ログ稼働ログ <sup>2</sup>	<ul style="list-style-type: none"> <li>• Windows の場合 &lt;Cosminexus のインストールディレクトリ&gt;%CC%\admin\logs\%cjlogger.log</li> <li>• UNIX の場合 /opt/Cosminexus/CC/admin/logs/cjlogger.log</li> </ul>	1024KB × 2	-
例外ログ	障害発生時の例外情報 <sup>2 3</sup>	<ul style="list-style-type: none"> <li>• Windows の場合 &lt;Cosminexus のインストールディレクトリ&gt;%CC%\admin\logs\%cjexception[n].log</li> <li>• UNIX の場合 /opt/Cosminexus/CC/admin/logs/cjexception[n].log</li> </ul>	1024KB × 6	ExceptionLog File

分類	内容	ログ出力先およびログファイル名 <sup>1</sup>	デフォルトのサイズ × 面数	チャンネル名
保守用ログ	保守情報 <sup>2</sup>	<ul style="list-style-type: none"> <li>Windows の場合 &lt;Cosminexus のインストールディレクトリ&gt;%CC%\admin¥logs¥CC¥maintenance¥cjmaintenance[n].log</li> <li>UNIX の場合 /opt/Cosminexus/CC/admin/logs/CC/maintenance/cjmaintenance[n].log</li> </ul>	1024KB × 3	MaintenanceLogFile
	コンソールメッセージ <sup>2</sup>	<ul style="list-style-type: none"> <li>Windows の場合 &lt;Cosminexus のインストールディレクトリ&gt;%CC%\admin¥logs¥CC¥maintenance¥cjconsole[n].log</li> <li>UNIX の場合 /opt/Cosminexus/CC/admin/logs/CC/maintenance/cjconsole[n].log</li> </ul>	32KB × 3	ConsoleLogFile
	サーバ管理コマンドの保守情報 <sup>2</sup>	<ul style="list-style-type: none"> <li>Windows の場合 &lt;Cosminexus のインストールディレクトリ&gt;%CC%\admin¥logs¥CC¥maintenance¥cjserveradmin[n].log</li> <li>UNIX の場合 /opt/Cosminexus/CC/admin/logs/CC/maintenance/cjserveradmin[n].log</li> </ul>	32KB × 3	ServerAdminLogFile

## 注

チャンネル名は、ログの出力先を識別する名称です。ログの属性（サイズ、面数）を変更する場合のキー値として使用します。

## 注 1

ログファイル名の [n] の部分には、面の番号（1 から面数（最大 16）まで）が付きます。

## 注 2

日立トレース共通ライブラリ形式の出力メッセージ（アプリケーション識別名）には、コマンド名が表示されます。

## 注 3

互換モードの場合、稼働ログと障害発生時の例外情報の出力先は標準モードと異なります。互換モードの場合の出力先とデフォルトのサイズ・面数は次のようになります。

## 5. トラブルシューティング

表 5-9 サーバ管理コマンドのログの出力先（互換モード）

内容	ログ出力先およびログファイル名	デフォルトのサイズ×面数
稼働ログ	<ul style="list-style-type: none"> <li>Windows の場合 &lt;Cosminexus インストールディレクトリ&gt;%CC%admin¥logs¥&lt; コマンド名称 &gt;message[n].log</li> <li>UNIX の場合 /opt/Cosminexus/CC/admin/logs/&lt; コマンド名称 &gt;message[n].log</li> </ul>	128KB × 2
障害発生時の例外情報	<ul style="list-style-type: none"> <li>Windows の場合 &lt;Cosminexus インストールディレクトリ&gt;%CC%admin¥logs¥&lt; コマンド名称 &gt;exception[n].log</li> <li>UNIX の場合 /opt/Cosminexus/CC/admin/logs/&lt; コマンド名称 &gt;exception[n].log</li> </ul>	256KB × 2

### 注

ログファイル名の [n] の部分には、面の番号（1 から面数（最大 16）まで）が付きま

す。サーバ管理コマンドのメッセージログに出力されるメッセージには、メッセージ ID フィールドが空白で、メッセージテキストフィールドにメッセージ ID（KDJEnnnnn-Y など）が含まれる場合があります。それらはサーバ側で発生したメッセージで、前後に出力されるメッセージの付加情報となります。

### リソースアダプタバージョンアップコマンド（cjrupdate）のログ

表 5-10 リソースアダプタバージョンアップコマンド（cjrupdate）のログの出力先

分類	内容	ログ出力先およびログファイル名	デフォルトのサイズ×面数	チャネル名
メッセージログ	稼働ログ	<ul style="list-style-type: none"> <li>Windows の場合 &lt;Cosminexus のインストールディレクトリ&gt;%CC%logs¥cjrupdatemessage[n].log</li> <li>UNIX の場合 /opt/Cosminexus/CC/logs/cjrupdatemessage[n].log</li> </ul>	1MB × 2	-
例外ログ	障害発生時の例外情報	<ul style="list-style-type: none"> <li>Windows の場合 &lt;Cosminexus のインストールディレクトリ&gt;%CC%logs¥cjrupdateexception[n].log</li> <li>UNIX の場合 /opt/Cosminexus/CC/logs/cjrupdateexception[n].log</li> </ul>	1MB × 2	-

分類	内容	ログ出力先およびログファイル名	デフォルトのサイズ × 面数	チャンネル名
保守用ログ	保守情報	<ul style="list-style-type: none"> <li>Windows の場合 &lt;Cosminexus のインストールディレクトリ&gt;¥CC¥logs¥cjrarrupdatemaintenance[n].log</li> <li>UNIX の場合 /opt/Cosminexus/CC/logs/cjrarrupdatemaintenance[n].log</li> </ul>	1MB × 2	-

## 注

チャンネル名は、ログの出力先を識別する名称です。ログの属性（サイズ、面数）を変更する場合のキー値として使用します。

## 注

ログファイル名の [n] の部分には、面の番号（1 から面数（最大 16）まで）が付きます。

## インプロセス HTTP サーバのログ

表 5-11 インプロセス HTTP サーバのログの出力先（デフォルト）

分類	内容	ログ出力先およびログファイル名	デフォルトのサイズ × 面数	チャンネル名
アクセスログ	インプロセス HTTP サーバの処理結果	<ul style="list-style-type: none"> <li>Windows の場合 &lt;ejb.server.log.directory&gt; <sup>1</sup>¥http¥cjhttp_access.inprocess_http[n].log<sup>2</sup></li> <li>UNIX の場合 &lt;ejb.server.log.directory&gt; <sup>1</sup>/http/cjhttp_access.inprocess_http[n].log<sup>2</sup></li> </ul>	4MB × 16	-
スレッドトレース	スレッドトレースの情報	<ul style="list-style-type: none"> <li>Windows の場合 &lt;ejb.server.log.directory&gt; <sup>1</sup>¥http¥maintenance¥thr¥cjhttp_thr.&lt;時間情報&gt;.inprocess_http.mm<sup>3</sup></li> <li>UNIX の場合 &lt;ejb.server.log.directory&gt; <sup>1</sup>/http/maintenance/thr/cjhttp_thr.&lt;時間情報&gt;.inprocess_http.mm<sup>3</sup></li> </ul>	約 3.2MB <sup>4</sup> × 16	-

## 5. トラブルシューティング

分類	内容	ログ出力先およびログファイル名	デフォルトのサイズ×面数	チャンネル名
通信トレース	通信トレースの情報	<ul style="list-style-type: none"> <li>Windows の場合 &lt;ejb.server.log.directory&gt; <sup>1</sup>¥http¥maintenance¥comm¥cjhttp_comm.&lt;時間情報&gt;.inprocess_http.mm <sup>3</sup></li> <li>UNIX の場合 &lt;ejb.server.log.directory&gt; <sup>1</sup>/http/maintenance/comm/cjhttp_comm.&lt;時間情報&gt;.inprocess_http.mm <sup>3</sup></li> </ul>	約 16.6MB <sup>4</sup> × 16	-

### 注

チャンネル名は、ログの出力先を識別する名称です。ログの属性（サイズ、面数）を変更する場合のキー値として使用します。

### 注 1

<ejb.server.log.directory> は、簡易構築定義ファイルの論理 J2EE サーバ (j2ee-server) の <configuration> タグ内に、ejb.server.log.directory パラメタで指定したディレクトリを指します。デフォルト値は、<Cosminexus のインストールディレクトリ>¥CC¥server¥public¥ejb¥<サーバ名称>¥logs です。簡易構築定義ファイルの ejb.server.log.directory パラメタの詳細については、マニュアル「Cosminexus 簡易構築・運用ガイド」を参照してください。

### 注 2

[n] には、面の番号（1 から面数（最大 16）まで）が付きます。

### 注 3

<時間情報> には、ファイルを出力した日時が、「YYYYMMDDHHmmSSnnn」形式で表示されます。例えば、ファイルを出力した時間が、2005 年 1 月 20 日 10 時 50 分 30.158 秒だった場合、ファイル名は「cjhttp\_thr.20050120105030158.inprocess\_http.mm」または「cjhttp\_comm.20050120105030158.inprocess\_http.mm」になります。

### 注 4

起動するたびに新規にトレースファイルが作成されます。面数が最大値（デフォルト値は 16）を超えると、最も古い日時のファイルを再利用します。ファイルサイズは次のように決定されます。  
スレッドトレースのファイルサイズ =  $((A+B) \times 32786) + 32914$  バイト  
通信トレースのファイルサイズ =  $((A+B) \times 172050) + 128$  バイト  
A = webserver.connector.inprocess\_http.max\_connections の値  
B = webserver.connector.inprocess\_http.send\_timeout の値が 0 の場合 0, 0 以外の場合 1



**!** 注意事項

インプロセス HTTP サーバのスレッドトレースと通信トレースを取得する前に、次のコマンドを実行してください。

Windows の場合

```
<Cosminexus のインストールディレクトリ >%CC%\admin\bin\cjtracesync  
[-thr|-comm] <サーバ名称 >
```

UNIX の場合

```
/opt/Cosminexus/CC/admin/bin/cjtracesync [-thr|-comm] <サーバ名称 >  
cjtracesync コマンドの詳細については、マニュアル「Cosminexus リファレンス コマ  
ンド編」の cjtracesync コマンドに関する説明を参照してください。
```

## 移行コマンド (cjenvupdate) のログ

表 5-12 移行コマンド (cjenvupdate) のログの出力先

分類	内容	ログ出力先およびログファイル名	デフォルトのサイズ × 面数	チャンネル名
メッセージログ	cjenvupdate コマンドの稼働ログ	<ul style="list-style-type: none"> <li>Windows の場合 &lt;Cosminexus のインストールディレクトリ &gt;%CC%\logs\cjenvupdatemessage[n].log</li> <li>UNIX の場合 /opt/Cosminexus/CC/logs/cjenvupdatemessage[n].log</li> </ul>	4MB × 4	-
例外ログ	cjenvupdate コマンドの例外情報	<ul style="list-style-type: none"> <li>Windows の場合 &lt;Cosminexus のインストールディレクトリ &gt;%CC%\logs\cjenvupdateexception[n].log</li> <li>UNIX の場合 /opt/Cosminexus/CC/logs/cjenvupdateexception[n].log</li> </ul>	4MB × 4	-
保守用ログ	cjenvupdate コマンドの保守情報	<ul style="list-style-type: none"> <li>Windows の場合 &lt;Cosminexus のインストールディレクトリ &gt;%CC%\logs\cjenvupdatemaintenance[n].log</li> <li>UNIX の場合 /opt/Cosminexus/CC/logs/cjenvupdatemaintenance[n].log</li> </ul>	4MB × 4	-

注

[n] には、面の番号 (1 から面数 (最大 16) まで) が付きます。

## リソース枯渇監視のログ

表 5-13 リソース枯渇監視のログの出力先

監視対象 リソース	ログ取得場所およびログファイル名 <sup>1</sup>	デフォルトのサイズ×面数	チャンネル名 <sup>2</sup>
メモリ	<ul style="list-style-type: none"> <li>Windows の場合 &lt;ejb.server.log.directory&gt; <sup>3</sup>¥watch¥cjmemorywatch[n].log</li> <li>UNIX の場合 &lt;ejb.server.log.directory&gt; <sup>3</sup>/watch/ cjmemorywatch[n].log</li> </ul>	1MB × 2	MemoryWatchLogFile
ファイルディス クリプタ	<ul style="list-style-type: none"> <li>UNIX の場合<sup>4</sup> &lt;ejb.server.log.directory&gt; <sup>3</sup>/watch/ cjfiledescriptorwatch[n].log</li> </ul>	1MB × 2	FileDescriptorWatchLo gFile
スレッド	<ul style="list-style-type: none"> <li>Windows の場合 &lt;ejb.server.log.directory&gt; <sup>3</sup>¥watch¥cjthreadwatch[n].log</li> <li>UNIX の場合 &lt;ejb.server.log.directory&gt; <sup>3</sup>/watch/ cjthreadwatch[n].log</li> </ul>	1MB × 2	ThreadWatchLogFile
スレッドダンプ	<ul style="list-style-type: none"> <li>Windows の場合 &lt;ejb.server.log.directory&gt; <sup>3</sup>¥watch¥cjthreaddumpwatch[n].log</li> <li>UNIX の場合 &lt;ejb.server.log.directory&gt; <sup>3</sup>/watch/ cjthreaddumpwatch[n].log</li> </ul>	1MB × 2	ThreaddumpWatchLog File
HTTP リクエス ト実行待ち キュー	<ul style="list-style-type: none"> <li>Windows の場合 &lt;ejb.server.log.directory&gt; <sup>3</sup>¥watch¥cjrequestqueewatch[n].log</li> <li>UNIX の場合 &lt;ejb.server.log.directory&gt; <sup>3</sup>/watch/ cjrequestqueewatch[n].log</li> </ul>	1MB × 2	RequestQueueWatchLo gFile
HTTP セッショ ン数	<ul style="list-style-type: none"> <li>Windows の場合 &lt;ejb.server.log.directory&gt; <sup>3</sup>¥watch¥cjhttpsessionwatch[n].log</li> <li>UNIX の場合 &lt;ejb.server.log.directory&gt; <sup>3</sup>/watch/ cjhttpsessionwatch[n].log</li> </ul>	1MB × 2	HttpSessionWatchLogF ile
コネクション プール	<ul style="list-style-type: none"> <li>Windows の場合 &lt;ejb.server.log.directory&gt; <sup>3</sup>¥watch¥cjconnectionpoolwatch[n].log</li> <li>UNIX の場合 &lt;ejb.server.log.directory&gt; <sup>3</sup>/watch/ cjconnectionpoolwatch[n].log</li> </ul>	1MB × 2	ConnectionPoolWatchL ogFile

## 注 1

[n] には、面の番号（1 から面数（最大 16）まで）が付きます。

## 注 2

ログの出力先を識別する名称で、ログの属性（サイズ、面数）を変更する場合のキー値として使用します。

## 注 3

<ejb.server.log.directory> は、簡易構築定義ファイルの論理 J2EE サーバ (j2ee-server) の <configuration> タグ内に、ejb.server.log.directory パラメータで指定したディレクトリを指します。デフォルト値は、<Cosminexus のインストールディレクトリ>%CC%¥server¥public¥ejb¥<サーバ名称>¥logs です。

簡易構築定義ファイルの ejb.server.log.directory パラメータの詳細については、マニュアル「Cosminexus 簡易構築・運用ガイド」を参照してください。

## 注 4

Windows の場合、または AIX の場合は、ファイルディスクリプタを監視できません。

リソース枯渇監視ログファイルに出力される情報やログファイルの出力形式については、「2.5 リソース枯渇の監視」を参照してください。

#### ログの出力先を設定するユーザ定義ファイル

J2EE サーバ、リダイレクタ、およびサーバ管理コマンドのログの出力先を変更している場合は、次の表に示す、ログの出力先を設定するユーザ定義ファイルを参照して出力先を確認してください。なお、ログの出力先を変更した場合、変更後の出力先は snapshot ログの一括収集時に収集対象外となります。必要に応じて snapshot ログの収集先を変更してください。

表 5-14 ログの出力先を設定するユーザ定義ファイル

分類	ユーザ定義ファイル
J2EE サーバ	簡易構築定義ファイルの論理 J2EE サーバ (j2ee-server) の <configuration> タグ内に指定する、ejb.server.log.directory パラメータ デフォルトは、「<Cosminexus のインストールディレクトリ>%CC%¥server¥public¥ejb¥<サーバ名称>¥logs」(Windows の場合)、または「/opt/Cosminexus/CC/server/public/ejb/<サーバ名称>/logs」(UNIX の場合) です。
リダイレクタ	Hitachi Web Server の場合 <ul style="list-style-type: none"> <li>メッセージログ 簡易構築定義ファイルの論理 Web サーバ (web-server) の &lt;configuration&gt; タグ内に指定する、JkLogFileDir パラメータ デフォルトは、「logs」です。</li> <li>保守用トレースログ 簡易構築定義ファイルの論理 Web サーバ (web-server) の &lt;configuration&gt; タグ内に指定する、JkTraceLogFileDir パラメータ デフォルトは、「logs」です。</li> </ul>

## 5. トラブルシューティング

分類	ユーザ定義ファイル
	<p>Microsoft IIS の場合</p> <ul style="list-style-type: none"> <li>• メッセージログ &lt;Cosminexus のインストールディレクトリ &gt;%CC%\web\redirector\isapi_redirect.conf の log_file_dir キー キーのデフォルトは、「&lt;Cosminexus のインストールディレクトリ &gt;%CC%\web\redirector\logs」です。</li> <li>• 保守用トレースログ &lt;Cosminexus のインストールディレクトリ &gt;%CC%\web\redirector\isapi_redirect.conf の trace_log_file_dir キー キーのデフォルトは、「&lt;Cosminexus のインストールディレクトリ &gt;%CC%\web\redirector\logs」です。</li> </ul>
サーバ管理コマンド	<p>サーバ管理コマンドの usrconf.bat ( Windows の場合 ), または usrconf ( UNIX の場合 ) の ejbserver.log.directory キー キーのデフォルトは、「&lt;Cosminexus のインストールディレクトリ &gt;%CC%\admin\logs」( Windows の場合 ), または 「 /opt/Cosminexus/CC/ admin/logs」( UNIX の場合 ) です。 Management Server リモート管理機能から操作した場合は、サーバ管理コマ ンドのログ出力先は変更できません。</p>

ログの出力先の変更方法など、トラブルシューティングの資料取得の設定については、マニュアル「Cosminexus システム構築ガイド」を参照してください。

### (b) 運用管理エージェント・運用監視エージェント・Management Server のログの取得

ここでは、運用管理エージェント、運用監視エージェント、および Management Server のログの出力先について説明します。

運用管理エージェント・運用監視エージェント・Management Server のログには、個別に取得する以外に、統合ログとしてまとめて取得できるものがあります。統合ログには、次の種類があります。

#### 統合メッセージログ

Manager のメッセージログが統合されて出力されます。

#### 統合トレースログ

Manager のトレースログが統合されて出力されます。

#### コマンド保守ログ

運用管理コマンド、Smart Composer 機能で使用するコマンド、および snapshotlog コマンドのトレースログが統合されて出力されます。

注 Smart Composer 機能で使用するコマンドの詳細については、マニュアル「Cosminexus 簡易構築・運用ガイド」を参照してください。

統合ログの出力先を次の表に示します。

表 5-15 統合ログの出力先 (Windows の場合)

ファイル名	内容	出力先ディレクトリ
mngmessage[n]. <sup>1</sup> log	統合メッセージログ	<Manager のログ出力ディレクトリ >¥message <sup>2</sup>
mngtrace[n]. <sup>1</sup> log	統合トレースログ	<Manager のログ出力ディレクトリ >¥trace <sup>2</sup>
mngcmd[n]. <sup>1</sup> log	コマンド保守ログ	<Manager のログ出力ディレクトリ >¥maintenance <sup>2</sup>

注 1

ログファイル名の [n] の部分には、面の番号 (1 から面数 (最大 64) まで) が付きます。

注 2

<Manager のログ出力ディレクトリ > は、manager.cfg (Manager ログ設定ファイル) で指定されたディレクトリを指します。デフォルト値は、<Cosminexus のインストールディレクトリ >¥manager¥log です。manager.cfg の詳細はマニュアル「Cosminexus リファレンス 定義編」を参照してください。

表 5-16 統合ログの出力先 (UNIX の場合)

ファイル名	内容	出力先ディレクトリ
mngmessage[n]. <sup>1</sup> log	統合メッセージログ	<Manager のログ出力ディレクトリ >/ message <sup>2</sup>
mngtrace[n]. <sup>1</sup> log	統合トレースログ	<Manager のログ出力ディレクトリ >/trace <sup>2</sup>
mngcmd[n]. <sup>1</sup> log	コマンド保守ログ	<Manager のログ出力ディレクトリ >/ maintenance <sup>2</sup>

注 1

ログファイル名の [n] の部分には、面の番号 (1 から面数 (最大 64) まで) が付きます。

注 2

<Manager のログ出力ディレクトリ > は、manager.cfg (Manager ログ設定ファイル) で指定されたディレクトリを指します。デフォルト値は、/opt/Cosminexus/manager/log です。manager.cfg の詳細はマニュアル「Cosminexus リファレンス 定義編」を参照してください。

なお、統合ログに出力したログは、デフォルトでは個別のログとしても出力されます。

また、統合ログに出力した場合には個別に出力しない設定にできるログもあります。統合ログの出力に関する設定については、マニュアル「Cosminexus システム構築ガイド」の Cosminexus Manager のログ取得の設定に関する説明を参照してください。

個別に取得する場合の運用管理エージェント、運用監視エージェント、および Management Server のログの出力先、および統合メッセージログ・統合トレースログへ

## 5. トラブルシューティング

の出力の有無を次の表に示します。

なお、表中の <Manager のログ出力ディレクトリ> は、manager.cfg (Manager ログ設定ファイル) で指定されたディレクトリを指します。デフォルト値は、<Cosminexus のインストールディレクトリ>%manager¥log (Windows の場合)、または /opt/Cosminexus/manager/log (UNIX の場合) です。manager.cfg の詳細はマニュアル「Cosminexus リファレンス 定義編」を参照してください。

表 5-17 運用管理エージェント、運用監視エージェント、および Management Server のログを個別に取得する場合の出力先 (Windows の場合)

分類	ファイル名	内容	出力先ディレクトリ	統合メッセージログ/統合トレースログ
運用管理エージェント	adminagent.err.[1-16].log <sup>2</sup>	運用管理エージェントの標準エラー出力	<Manager のログ出力ディレクトリ>	-
	adminagent.out.[1-16].log <sup>2</sup>	運用管理エージェントの標準出力		-
	adminagent.err	運用管理エージェントの標準エラー出力コマンドライン		-
	adminagent[1-16].log	運用管理エージェントのログ		-
	adminagentctl.exe.[1-2].log	運用管理エージェントの起動・停止コマンドのログ		-
	adminagent[n] <sup>1</sup> .log	運用管理エージェントの保守ログ	<Manager のログ出力ディレクトリ>%maintenance	-
	mngirmi[n] <sup>1</sup> .log	運用管理エージェントが行う RMI 処理での保守ログ		-
	processConsole[n] <sup>1</sup> .log	コンソールログ	<Manager のログ出力ディレクトリ>	-
	adminagentsv.exe.[1-16].log <sup>2</sup>	運用管理エージェントサービスのログ		-
	adminagentsv.exe.out <sup>2</sup>	運用管理エージェントサービスの標準出力		-
adminagentsv.exe.err <sup>2</sup>	運用管理エージェントサービスの標準エラー出力		-	

分類	ファイル名	内容	出力先ディレクトリ	統合メッセージ ログ/統合ト レースログ
運用監視 エージェント	mngagent<ド メイン名><Agent 名>.[n] 1.log 3	<ul style="list-style-type: none"> <li>運用監視エー ジェントのログ・ト レース</li> <li>J2EE サーバ用シ ステム JP1 イベント および J2EE サーバ用ユーザ JP1 イベントのロ グ 4</li> <li>Management イベ ント発行ログ 5</li> </ul>		-
Manage ment Server	mngsvr.exe.[1-2 .log	Management Server サービスのログ		-
	mngsvr.exe.err 2	Management Server サービスの標準エ ラー出力		-
	mngsvr.exe.out 2	Management Server サービスの標準出力		-
	mngsvrctl.exe.[1- 2].log	Management Server サービス起動・停止 コマンドのログ		-
	mngsvr[n] 1.log	Management Server のログ Cosminexus の運用 管理サーバのシステ ム JP1 イベントのロ グ 4		
	mngsvr[n] 1.log	Management Server の保守ログ	<Manager のログ出力 ディレクトリ >¥maintenance	-

(凡例) : 統合ログに出力されます - : 統合ログには出力されません

注 1

ファイル名の [n] の部分には、1 ~ 指定したログの面数の通し番号が付きます。

注 2

日立トレース共通ライブラリ形式以外の形式でログが出力されます。

注 3

運用監視エージェントのログ・トレースの出力先は変更できます。運用監視エージェントのログ・トレースの出力先を変更している場合は、mngagent.properties ファイル (運用監視エージェントプロパティファイル) の mngagent.log.filename キーの値を参照してください。

注 4

JP1 と連携して Cosminexus で構築したシステムを運用する場合に出力されます。

## 5. トラブルシューティング

注 5

Management イベントを使用している場合に出力されます。

表 5-18 運用管理エージェント，運用監視エージェント，および Management Server のログを個別に取得する場合の出力先（UNIX の場合）

分類	ファイル名	内容	出力先ディレクトリ	統合メッセージログ/統合トレースログ
運用管理エージェント	adminagent.err.[1-16].log <sup>2</sup>	運用管理エージェントの標準エラー出力	<Manager のログ出力ディレクトリ>	-
	adminagent.out.[1-16].log <sup>2</sup>	運用管理エージェントの標準出力		-
	adminagent.err	運用管理エージェントの標準エラー出力コマンドライン		-
	adminagent[1-16].log	運用管理エージェントのログ		-
	adminagentctl.[1-16].log	運用管理エージェントの起動・停止コマンドのログ		-
	adminagent[n] <sup>1</sup> .log	運用管理エージェントの保守ログ	<Manager のログ出力ディレクトリ>/maintenance	-
	mnggrmi[n] <sup>1</sup> .log	運用管理エージェントが行う RMI 処理での保守ログ	-	-
	processConsole[n] <sup>1</sup> .log	コンソールログ	<Manager のログ出力ディレクトリ>	-
運用監視エージェント	mngagent-<ドメイン名>-<Agent 名>.[n] <sup>1</sup> .log <sup>3</sup>	<ul style="list-style-type: none"> <li>運用監視エージェントのログ・トレース</li> <li>J2EE サーバ用システム JP1 イベントおよび J2EE サーバ用ユーザ JP1 イベントのログ<sup>4</sup></li> <li>Management イベント発行ログ<sup>5</sup></li> </ul>	<Manager のログ出力ディレクトリ>	-
Management Server	mngsvrctlstart.[1-2].log	Management Server 起動コマンド	<Manager のログ出力ディレクトリ>	-
	mngsvrctlstop.[1-2].log	Management Server 停止コマンド	<Manager のログ出力ディレクトリ>	-
	mngsvrctlsetup.[1-2].log	Management Server セットアップコマンド	<Manager のログ出力ディレクトリ>	-



分類	ファイル名	内容	出力先ディレクトリ	統合メッセージ ログ / 統合ト レースログ
	mngsvr[n].log	Management Server のロ グ Cosminexus の運用管理 サーバのシステム JP1 イ ベントのログ <sup>4</sup>		
	mngenvsetup.[ 1-2].log	mngenvsetup コマンドの 実行ログ	<Manager のログ出力 ディレクトリ >/ maintenance	-
	mngsvr[n] 1.log	Management Server の保 守ログ	<Manager のログ出力 ディレクトリ >/ maintenance	-

(凡例) : 統合ログに出力されます - : 統合ログには出力されません

注 1

ファイル名の [n] の部分には、1 ~ 指定したログの面数の通し番号が付きます。

注 2

日立トレース共通ライブラリ形式以外の形式でログが出力されます。

注 3

運用監視エージェントのログ・トレースの出力先は変更できます。運用監視エージェントのログ・トレースの出力先を変更している場合は、mngagent.properties ファイル (運用監視エージェントプロパティファイル) の mngagent.log.filename キーの値を参照してください。

注 4

JP1 と連携して Cosminexus で構築したシステムを運用する場合に出力されます。

注 5

Management イベントを使用している場合に出力されます。

### ! 注意事項

コンソールログには、運用管理エージェントが起動したサーバプロセスの標準出力、および標準エラー出力が出力されます。コンソールログについての注意事項を次に示します。

- Windows の場合、次のプロセスに対しては、コンソールログは出力されません。
  - 論理パフォーマンストレーサ
  - 論理 Web サーバ
  - 論理 CTM ドメインマネージャ
  - 論理 CTM
  - 間接起動の論理ユーザサーバ
- 論理ユーザサーバの起動種別については、マニュアル「Cosminexus システム構築ガイド」を参照してください。
- コンソールログに出力する情報が一度に複数行出力された場合、コンソールログには 1 行にまとめて表示されることがあります。
- コンソール情報に出力する情報の文字数が 2039 文字を超えてしまう場合、2039 文字以降の情報は次の行に分割されて出力されます。

#### (c) 統合ユーザ管理のログの取得

統合ユーザ管理のトレースファイルは、ua.conf ファイル（統合ユーザ管理のコンフィグレーションファイル）の com.cosminexus.admin.auth.trace.prefix オプションの設定に応じて出力されます。ua.conf ファイルの詳細については、マニュアル「Cosminexus リファレンス 定義編」を参照してください。

#### (d) ログ以外に取得が必要な情報

ここでは、ログ以外に取得する必要がある情報について説明します。

##### インプロセストランザクションサービスを使用している場合

インプロセストランザクションサービスを使用している場合、インプロセストランザクションサービスのステータスファイルを取得する必要があります。なお、ステータスファイルを二重化している場合は、予備のステータスファイルも取得してください。

ステータスファイルは、簡易構築定義ファイルの論理 J2EE サーバ（j2ee-server）の <configuration> タグ内に、ejbserver.distributedtx.ots.status.directory1 パラメータ、および ejbserver.distributedtx.ots.status.directory2 パラメータ（二重化している場合）で指定したパスに格納されています。

## (2) Cosminexus Performance Tracer のログの取得

Cosminexus Performance Tracer のログの種類とログの出力先について説明します。

#### (a) Cosminexus Performance Tracer のログの種類

Cosminexus Performance Tracer では、PRF 識別子ごとに PRF デモン、PRF コマンドのログを出力します。また、システムでトラブルが発生したときの保守員による障害解析用に、モジュールトレースおよび構造化例外発生ログも出力します。

## (b) Cosminexus Performance Tracer のログの出力先

Cosminexus Performance Tracer のログの出力先を次に示します。

表 5-19 Cosminexus Performance Tracer のログの出力先

内容	出力先ディレクトリ
PRF デモンおよび PRF コマンドのログ	<ul style="list-style-type: none"> <li>Windows の場合 &lt;環境変数 PRFSPOOL の設定ディレクトリ &gt;¥log¥&lt;PRF 識別子 &gt;¥ctmlog[n]</li> <li>UNIX の場合 \$PRFSPOOL/log/&lt;PRF 識別子 &gt;/ctmlog[n]</li> </ul>
モジュールトレース	<ul style="list-style-type: none"> <li>Windows の場合 &lt;環境変数 PRFSPOOL の設定ディレクトリ &gt;¥utt¥umt</li> <li>UNIX の場合 \$PRFSPOOL/utt/umt</li> </ul>
構造化例外発生ログ	<ul style="list-style-type: none"> <li>Windows の場合 &lt;環境変数 PRFSPOOL の設定ディレクトリ &gt;¥osltrc</li> <li>UNIX の場合 \$PRFSPOOL/osltrc</li> </ul>
保守情報	<ul style="list-style-type: none"> <li>Windows の場合 &lt;環境変数 PRFSPOOL の設定ディレクトリ &gt;</li> <li>UNIX の場合 \$PRFSPOOL/</li> </ul>

注

[n] には「01」または「02」が表示されます。

## (3) Cosminexus Component Transaction Monitor のログの取得

ここでは、Cosminexus Component Transaction Monitor のログの出力先について説明します。

Cosminexus Component Transaction Monitor のログは「<環境変数 CTMSPOOL の設定ディレクトリ >¥log」(Windows の場合)、または「\$CTMSPOOL/log」(UNIX の場合)に出力されます。デフォルトでは、「<Cosminexus のインストールディレクトリ >¥CTM¥spool¥log」(Windows の場合)、または「/opt/Cosminexus/CTM/spool/log」(UNIX の場合)に出力されます。また、システムでトラブルが発生したときの保守員による障害解析用に、各種保守情報を環境変数 CTMSPOOL の設定ディレクトリに出力します。ctmlogcat コマンドを実行することで出力メッセージを確認できます。

表 5-20 Cosminexus Component Transaction Monitor のログの出力先

内容	出力先ディレクトリ
CTM デーモンおよび CTM コマンドのログ	<ul style="list-style-type: none"> <li>Windows の場合 &lt;環境変数 CTMSPOOL の設定ディレクトリ &gt;¥log¥&lt;CTM デーモン識別子 &gt;¥ctmlog[n]</li> <li>UNIX の場合 \$CTMSPOOL/log/&lt;CTM デーモン識別子 &gt;/ctmlog[n]</li> </ul>
保守情報	<ul style="list-style-type: none"> <li>Windows の場合 &lt;環境変数 CTMSPOOL の設定ディレクトリ &gt;</li> <li>UNIX の場合 \$CTMSPOOL</li> </ul>

#### (4) Cosminexus DABroker Library のログの取得

Cosminexus DABroker Library のログの取得方法について説明します。Cosminexus DABroker Library では、トラブルシューティングに使用するログとして、スプール情報を取得します。スプール情報として取得できるのは、次の情報です。

メッセージログ

拡張データベースアクセストレース

エラートレース

Exception トレースログ

イベントトレース (DABroker for C++ をインストールしている場合)

XA トレース

「<Cosminexus DABroker Library の運用ディレクトリ >¥spool 下」(Windows の場合)、または「<Cosminexus DABroker Library の運用ディレクトリ >/spool 下」(UNIX の場合)のすべてのファイルをコピーして保存してください。<Cosminexus DABroker Library の運用ディレクトリ >とは、「<Cosminexus のインストールディレクトリ >¥DAB」(Windows の場合)、または「/opt/DABroker」(UNIX の場合)のことです。なお、環境設定によって出力先を変更している場合は、変更した出力先のファイルをコピーして保存してください。

なお、Cosminexus DABroker Library のスプール情報のうち、Exception トレースログは、運用開始前に取得のための設定をしておく必要があります。Exception トレースログ取得の設定については、マニュアル「Cosminexus システム構築ガイド」を参照してください。

#### (5) 監査ログで出力するログの取得

ここでは、監査ログの機能を使用した場合に出力されるログファイルについて説明します。

監査ログで出力するログの種類と出力先を次の表に示します。

表 5-21 監査ログで出力するのログの出力先（デフォルト）

分類	内容	ログ出力先およびログファイル名	デフォルトのサイズ×面数
メッセージログ	監査ログのメッセージログ	<ul style="list-style-type: none"> <li>Windows の場合 &lt;Cosminexus のインストールディレクトリ&gt;¥auditlog¥rasmessage[n].log</li> <li>UNIX の場合 /opt/Cosminexus/auditlog/rasmessage[n].log</li> </ul>	1MB × 4
例外ログ	監査ログの例外情報	<ul style="list-style-type: none"> <li>Windows の場合 &lt;Cosminexus のインストールディレクトリ&gt;¥auditlog¥rasexception[n].log</li> <li>UNIX の場合 /opt/Cosminexus/auditlog/rasexception[n].log</li> </ul>	1MB × 8

注

ログファイル名の [n] の部分には、面の番号（1 から面数（最大 64）まで）が付きます。

## （6）アプリケーションのユーザログの取得

アプリケーションのユーザログは、日立トレース共通ライブラリ形式で出力するように設定している場合に取得できます。ユーザログには、次の 2 種類があります。

- J2EE サーバ上で動作する J2EE アプリケーション（J2EE コンポーネント）が出力するユーザログ
- EJB クライアントアプリケーションが出力するユーザログ

### （a）J2EE アプリケーションのユーザログの取得

J2EE アプリケーションのログの出力先については、次のディレクトリに、簡易構築定義ファイルの論理 J2EE サーバ（j2ee-server）の <configuration> タグ内に ejbserver.application.userlog.CJLogHandler.<ハンドラ名>.path パラメタで指定したプリフィックスを持つファイル名で出力されます。なお、<ハンドラ名>には、キーの値を区別するためのハンドラ名が指定されています。

- Windows の場合  
<ログ出力先ルート（ejb.server.log.directory の値）>¥user¥（デフォルトは <Cosminexus 作業ディレクトリ>¥ejb¥<J2EE サーバ名>¥logs¥user）
- UNIX の場合  
<ログ出力先ルート（ejb.server.log.directory の値）>/user/（デフォルトは <Cosminexus 作業ディレクトリ>/ejb/<J2EE サーバ名>/logs/user）

なお、J2EE アプリケーションのユーザログ出力の設定については、マニュアル「Cosminexus システム構築ガイド」を参照してください。

## 5. トラブルシューティング

### (b) EJB クライアントアプリケーションのユーザログの取得

EJB クライアントアプリケーションのユーザログの出力先については、次のディレクトリに EJB クライアントアプリケーションのシステムプロパティの `ejbserver.application.userlog.CJLogHandler.<ハンドラ名>.path` キーの値で指定したプリフィックスを持つファイル名で出力されます。なお、<ハンドラ名>には、キーの値を区別するためのハンドラ名が指定されています。

- Windows の場合  
    <ログ出力先ルート ( `ejb.server.log.directory` の値 ) >¥user¥ ( デフォルトは  
    <Cosminexus 作業ディレクトリ >¥ejb¥<J2EE サーバ名 >¥logs¥user )
- UNIX の場合  
    <ログ出力先ルート ( `ejb.server.log.directory` の値 ) >/user/ ( デフォルトは  
    <Cosminexus 作業ディレクトリ >/ejb/<J2EE サーバ名 >/logs/user )

EJB クライアントアプリケーションでユーザログを出力するためのシステムプロパティの設定方法については、マニュアル「Cosminexus システム構築ガイド」の EJB クライアントアプリケーションのログ出力の設定に関する説明、またはマニュアル「Cosminexus 機能解説」の EJB クライアントアプリケーションでのユーザログ出力の実装に関する説明を参照してください。

### (7) 構築済み実行環境の切り替えで出力するログの取得

構築済み実行環境を `cosmienv` コマンドで切り替えた場合に出力されるログファイルについて説明します。`cosmienv` コマンドで出力するログファイルは、コマンド実行時に指定する引数 (機能) によって異なります。各ログファイルの出力先は「`/opt/Cosminexus/env/log`」です。ログファイルのファイルサイズは 100 キロバイト以内です。また、ファイル面数は 2 面です。コマンド実行時に指定する引数 (機能) ごとのログファイルを次に示します。

- 引数に `-s` を指定した場合 (退避機能の場合)  
    `cosmienv_s[2].log`
- 引数に `-d` を指定した場合 (回復機能の場合)  
    `cosmienv_d[2].log`
- 引数に `-r` を指定した場合 (削除機能の場合)  
    `cosmienv_r[2].log`

ログファイルの出力形式および出力項目については、「5.5.1 Cosminexus のログの調査」を参照してください。

## 5.4.2 Cosminexus のログの取得 (バッチアプリケーションを実行するシステム)

ここでは、Cosminexus の構成ソフトウェアが出力するログを手動で取得する方法につ

いて説明します。なお、snapshot ログとして Cosminexus のログをすでに収集している場合は、ここで説明する操作は不要です。

ここでは、次のログの取得方法について説明します。

- Cosminexus Component Container のログ
- アプリケーションのユーザログ

また、バッチアプリケーションを実行するシステムでは、Cosminexus の構成ソフトウェアが出力するログを手動で取得する場合、次のログも取得してください。

- Cosminexus Performance Tracer のログ
- Cosminexus DABroker Library のログ
- 監査ログで出力するログ
- 構築済み実行環境の切り替えで出力するログ

これらのログの取得方法については、「5.4.1 Cosminexus のログの取得 (J2EE アプリケーションを実行するシステム)」を参照してください。それぞれの参照先の詳細を次に示します。

- Cosminexus Performance Tracer のログ  
「5.4.1 Cosminexus のログの取得 (J2EE アプリケーションを実行するシステム)」の「(2) Cosminexus Performance Tracer のログの取得」
- Cosminexus DABroker Library のログ  
「5.4.1 Cosminexus のログの取得 (J2EE アプリケーションを実行するシステム)」の「(4) Cosminexus DABroker Library のログの取得」
- 監査ログで出力するログ  
「5.4.1 Cosminexus のログの取得 (J2EE アプリケーションを実行するシステム)」の「(5) 監査ログで出力するログの取得」
- 構築済み実行環境の切り替えで出力するログ  
「5.4.1 Cosminexus のログの取得 (J2EE アプリケーションを実行するシステム)」の「(7) 構築済み実行環境の切り替えで出力するログの取得」

### (1) Cosminexus Component Container のログの取得

Cosminexus Component Container のログの種類とログの出力先について説明します。Cosminexus Component Container のログには、次の2種類のログがあります。

- バッチサーバ・サーバ管理コマンド・バッチアプリケーションのログ
- 運用管理エージェント・運用監視エージェント・Management Server のログ

それぞれのログの出力先について説明します。

#### (a) バッチサーバ・サーバ管理コマンド・バッチアプリケーションのログの取得

バッチサーバ、サーバ管理コマンド、およびバッチアプリケーションのログの取得方法について説明します。

## 5. トラブルシューティング

また、Cosminexus Component Container では、これらのログに加えて、移行コマンドのログが出力されます。また、リソース枯渇監視機能を使用している場合はリソース枯渇監視ログが、Server Plug-in を使用しているときは Server Plug-in のログが出力されます。

バッチサーバのログには、メッセージログ、ユーザログ、例外ログ、および保守用ログの 4 種類があります。なお、バッチサーバでは、これらのログに加えて、起動、停止および異常終了時にイベントログまたは syslog を出力します。

サーバ管理コマンドのログには、メッセージログ、例外ログ、および保守用ログの 3 種類があります。

バッチアプリケーションのログは、メッセージログの 1 種類です。

リソースアダプタのバージョンアップコマンド (cjrupdate) のログには、メッセージログ、例外ログ、および保守用ログの 3 種類があります。

移行コマンドのログには、メッセージログ、例外ログ、および保守用ログの 3 種類があります。

次にそれぞれのログについて説明します。

### メッセージログ

バッチサーバ、サーバ管理コマンド、移行コマンドなどの稼働状態が出力されます。各種サーバおよびコマンドの稼働監視の情報として使用します。

### ユーザログ

アプリケーション中で出力される標準出力および標準エラー出力の情報が出力されます。アプリケーションの開発時の動作確認用に使用します。Solaris 以外の場合、JavaVM のメモリ関連ログも含まれます。

### 例外ログ

システムでトラブルが発生したときの Cosminexus Component Container の例外情報が出力されます。なお、例外ログは日常的な運用で監視する必要はありません。ログにメッセージが出力された場合に、例外情報を参照するときにご利用ください。

### 保守用ログ

システムでトラブルが発生したときの Cosminexus Component Container の障害保守情報が出力されます。保守員が Cosminexus Component Container の障害解析に使用します。

### イベントログ (Windows の場合)

バッチサーバが起動、停止または異常終了したことを示す情報が出力されます。出力先は Windows のイベントログの設定によって異なります。

なお、イベントログは、バッチサーバの停止のしかたによっては、出力されません。次の場合は、正しくログが出力されないことがあります。

- バッチサーバが動作している JavaVM 自体に問題が発生した場合



- バッチサーバのプロセスを TerminateProcess によって外部から停止した場合
- JavaVM の起動オプションとして -XX:+HitachiOutOfMemoryAbort オプションを指定している場合にメモリ不足によってバッチサーバが異常終了したとき  
なお、-XX:+HitachiOutOfMemoryAbort オプションは、デフォルトで設定されているオプションです。

#### syslog (UNIX の場合)

バッチサーバが起動、停止または異常終了したことを示す情報が出力されます。出力先は UNIX の syslog の設定によって異なります。

なお、syslog は、バッチサーバの停止のしかたによっては、出力されません。次の場合は、正しくログが出力されないことがあります。

- バッチサーバが動作している JavaVM 自体に問題が発生した場合
- バッチサーバのプロセスを SIGKILL シグナル (kill -9 など) によって外部から停止した場合
- JavaVM の起動オプションとして -XX:+HitachiOutOfMemoryAbort オプションを指定している場合にメモリ不足によってバッチサーバが異常終了したとき  
-XX:+HitachiOutOfMemoryAbort オプションは、デフォルトで設定されているオプションです。

#### リソース枯渇監視ログ

リソース枯渇監視機能を使用している場合に、監視対象のリソースについてのリソース枯渇監視情報が出力されます。リソースの使用率または使用数がしきい値を超えた場合の原因調査に使用します。

ログは、若い面番号の付いたログファイルから順に記録されます。一つのログファイルのサイズが 1 面当たりの最大サイズに達すると、ログは次の面番号の付いたログファイルに記録されます。最後のログファイル (面数の番号が付いたログファイル) のサイズが 1 面当たりの最大サイズに達すると、面の番号 1 のログファイルを空にし、そこへログを記録していきます。以降、ログファイルを空にししながら、面番号の順にログファイルへログを記録していきます。

ログの出力先のデフォルトを次の表に示します。Cosminexus Component Container のログは、サーバ単位またはコマンド単位に取得できます。

ログの出力先に示す <作業ディレクトリ> は、簡易構築定義ファイルの論理 J2EE サーバ (j2ee-server) の <configuration> タグ内に、ejb.public.directory パラメタで指定したディレクトリを指します。デフォルト値は、<Cosminexus のインストールディレクトリ>¥CC¥server¥public (Windows の場合)、または /opt/Cosminexus/CC/server/public (UNIX の場合) です。

5. トラブルシューティング

バッチサーバのログ

表 5-22 バッチサーバのログの出力先 (デフォルト)

分類	内容	ログ出力先およびログファイル名 <sup>1</sup>	デフォルトのサイズ × 面数	チャンネル名
メッセージログ	稼働ログ	<ul style="list-style-type: none"> <li>Windows の場合 &lt;ejb.server.log.directory&gt; <sup>2</sup>¥cjmessage[n].log</li> <li>UNIX の場合 &lt;ejb.server.log.directory&gt; <sup>2</sup>/ cjmessage[n].log</li> </ul>	1MB × 2	MessageLog File
	ログ稼働ログ	<ul style="list-style-type: none"> <li>Windows の場合 &lt;ejb.server.log.directory&gt; <sup>2</sup>¥cjlogger.log</li> <li>UNIX の場合 &lt;ejb.server.log.directory&gt; <sup>2</sup>/ cjlogger.log</li> </ul>	-	MessageLog File
	バッチサーバにデプロイして使用するリソースアダプタの稼働ログ <sup>3</sup>	<ul style="list-style-type: none"> <li>Windows の場合 &lt;ejb.server.log.directory&gt; <sup>2</sup>¥connectors¥&lt;リソースアダプタの表示名&gt;[n].log</li> <li>UNIX の場合 &lt;ejb.server.log.directory&gt; <sup>2</sup>/ connectors/&lt;リソースアダプタの表示名&gt;[n].log</li> </ul>	1MB × 2	-
ユーザログ	ユーザ出力ログ	<ul style="list-style-type: none"> <li>Windows の場合 &lt;ejb.server.log.directory&gt; <sup>2</sup>¥user_out[n].log</li> <li>UNIX の場合 &lt;ejb.server.log.directory&gt; <sup>2</sup>/ user_out[n].log</li> </ul>	1MB × 2	UserOutputLogFile
	ユーザエラーログ	<ul style="list-style-type: none"> <li>Windows の場合 &lt;ejb.server.log.directory&gt; <sup>2</sup>¥user_err[n].log</li> <li>UNIX の場合 &lt;ejb.server.log.directory&gt; <sup>2</sup>/ user_err[n].log</li> </ul>	1MB × 2	UserErrorLogFile
	JavaVM の保守情報およびガーベージコレクションのログ	<ul style="list-style-type: none"> <li>Windows の場合 &lt;ejb.server.log.directory&gt; <sup>2</sup>¥javalog[nn].log</li> <li>UNIX の場合 &lt;ejb.server.log.directory&gt; <sup>2</sup>/ javalog[nn].log</li> </ul>	256KB × 4	-

分類	内容	ログ出力先およびログファイル名 <sup>1</sup>	デフォルトのサイズ × 面数	チャンネル名
例外ログ	障害発生時の例外情報	<ul style="list-style-type: none"> <li>Windows の場合 &lt;ejb.server.log.directory&gt; <sup>2</sup>¥cjexception[n].log</li> <li>UNIX の場合 &lt;ejb.server.log.directory&gt; <sup>2</sup>/ cjexception[n].log</li> </ul>	1MB × 2	ExceptionLogFile
保守用ログ	保守情報	<ul style="list-style-type: none"> <li>Windows の場合 &lt;ejb.server.log.directory&gt; <sup>2</sup>¥CC¥maintenance¥cjmaintenance[n].log</li> <li>UNIX の場合 &lt;ejb.server.log.directory&gt; <sup>2</sup>/ CC/maintenance/ cjmaintenance[n].log</li> </ul>	16MB × 4	MaintenanceLogFile
	コンソールメッセージ	<ul style="list-style-type: none"> <li>Windows の場合 &lt;ejb.server.log.directory&gt; <sup>2</sup>¥CC¥maintenance¥cjconsole[n].log</li> <li>UNIX の場合 &lt;ejb.server.log.directory&gt; <sup>2</sup>/ CC/maintenance/ cjconsole[n].log</li> </ul>	1MB × 2	ConsoleLogFile
	EJB コンテナの保守情報	<ul style="list-style-type: none"> <li>Windows の場合 &lt;ejb.server.log.directory&gt; <sup>2</sup>¥CC¥maintenance¥cjejbcontainer[n].log</li> <li>UNIX の場合 &lt;ejb.server.log.directory&gt; <sup>2</sup>/ CC/maintenance/ cjejbcontainer[n].log</li> </ul>	1MB × 2	EJBContainerLogFile
	起動プロセス標準出力情報 <sup>4</sup>	<ul style="list-style-type: none"> <li>Windows の場合 &lt;ejb.server.log.directory&gt; <sup>2</sup>¥CC¥maintenance¥cjstdout.log</li> <li>UNIX の場合 &lt;ejb.server.log.directory&gt; <sup>2</sup>/ CC/maintenance/cjstdout.log</li> </ul>	-	-

## 5. トラブルシューティング

分類	内容	ログ出力先およびログファイル名 <sup>1</sup>	デフォルトのサイズ × 面数	チャンネル名
	起動プロセス標準エラー情報 <sup>4</sup>	<ul style="list-style-type: none"> <li>Windows の場合 &lt;ejb.server.log.directory&gt; ^¥CC¥maintenance¥cjstderr.log</li> <li>UNIX の場合 &lt;ejb.server.log.directory&gt;<sup>2/</sup> CC/maintenance/cjstderr.log</li> </ul>	-	-
	終了プロセス情報	<ul style="list-style-type: none"> <li>Windows の場合 &lt;ejb.server.log.directory&gt; ^¥CC¥maintenance¥cj_shutdown[n].log</li> <li>UNIX の場合 &lt;ejb.server.log.directory&gt;<sup>2/</sup> CC/maintenance/ cj_shutdown[n].log</li> </ul>	4KB × 2 5	-
イベントログ	バッチサーバの起動、停止または異常終了を示すログ	Windows のイベントビューアのアプリケーションログ <sup>6</sup>	-	-
syslog	バッチサーバの起動、停止または異常終了を示すログ	UNIX の syslog の設定に依存します。 <sup>7</sup>	-	-

(凡例) - : 該当なし

注

チャンネル名は、ログの出力先を識別する名称です。ログの属性（サイズ、面数）を変更する場合のキー値として使用します。

注 1

ログファイル名の [n] の部分には、面の番号（1 から面数（最大 16）まで）が付きます。また、[nn] の部分には、01 ~ 99 の通し番号が付きます。

注 2

<ejb.server.log.directory> は、簡易構築定義ファイルの論理 J2EE サーバ (j2ee-server) の <configuration> タグ内に、ejb.server.log.directory パラメータで指定したディレクトリを指します。デフォルト値は、<Cosminexus のインストールディレクトリ>¥CC¥server¥public¥ejb¥<サーバ名称>¥logs です。

簡易構築定義ファイルの ejb.server.log.directory パラメータの詳細については、マニュアル「Cosminexus 簡易構築・運用ガイド」を参照してください。

注 3

リソースアダプタのログを取得するかどうかは、サーバ管理コマンド、または Server Plug-in で指定されている内容に従います。また、リソースアダプタのログは、簡易構築定義ファイルでサイズおよび面数を変更できます。リソースアダプタのログ取得の設定については、マニュアル「Cosminexus システム構築ガイド」を参照してください。

## 注 4

起動プロセス情報だけを取得するログです。主にバッチサーバの起動または終了時に出力されるため、オンライン中に出力されることはほとんどありません。ファイルのサイズが上限に達したときは、<作業ディレクトリ>%ejb%<サーバ名称>%logs 下 (Windows の場合)、または<作業ディレクトリ>/ejb/<サーバ名称>/logs 下 (UNIX の場合) の cjstdout\_save.log または cjstderr\_save.log に退避されます。すでに cjstdout\_save.log または cjstderr\_save.log があつたときは、上書きされます。

## 注 5

サイズおよび面数は、変更できません。

## 注 6

ログファイルの出力先は Windows のイベントログの設定によって異なります。

## 注 7

バッチサーバの起動、停止および異常終了のメッセージを syslog に出力するためには、syslog の設定で、facility 「daemon」 に対する priority を 「info」 または 「debug」 に設定する必要があります。syslog の設定に関する詳細については、OS 付属のマニュアルを参照してください。

## サーバ管理コマンドのログ

表 5-23 サーバ管理コマンドのログの出力先 (デフォルト)

分類	内容	ログ出力先およびログファイル名 <sup>1</sup>	デフォルトのサイズ × 面数	チャンネル名
メッセージログ	稼働ログ <sup>2 3</sup>	<ul style="list-style-type: none"> <li>Windows の場合 &lt;Cosminexus のインストールディレクトリ&gt;%CC%¥admin¥logs¥cjmessage[n].log</li> <li>UNIX の場合 /opt/Cosminexus/CC/admin/logs/cjmessage[n].log</li> </ul>	1024K B × 3	MessageLog File
	ログ稼働ログ <sup>2</sup>	<ul style="list-style-type: none"> <li>Windows の場合 &lt;Cosminexus のインストールディレクトリ&gt;%CC%¥admin¥logs¥cjlogger.log</li> <li>UNIX の場合 /opt/Cosminexus/CC/admin/logs/cjlogger.log</li> </ul>	1024K B × 2	-
例外ログ	障害発生時の例外情報 <sup>2 3</sup>	<ul style="list-style-type: none"> <li>Windows の場合 &lt;Cosminexus のインストールディレクトリ&gt;%CC%¥admin¥logs¥cjexception[n].log</li> <li>UNIX の場合 /opt/Cosminexus/CC/admin/logs/cjexception[n].log</li> </ul>	1024K B × 6	ExceptionLog File

## 5. トラブルシューティング

分類	内容	ログ出力先およびログファイル名 <sup>1</sup>	デフォルトのサイズ × 面数	チャンネル名
保守用ログ	保守情報 <sup>2</sup>	<ul style="list-style-type: none"> <li>Windows の場合 &lt;Cosminexus のインストールディレクトリ &gt;%CC%\admin\logs\%CC%\maintenance\cjmaintenance[n].log</li> <li>UNIX の場合 /opt/Cosminexus/CC/admin/logs/CC/maintenance/cjmaintenance[n].log</li> </ul>	1024KB × 3	MaintenanceLogFile
	コンソールメッセージ <sup>2</sup>	<ul style="list-style-type: none"> <li>Windows の場合 &lt;Cosminexus のインストールディレクトリ &gt;%CC%\admin\logs\%CC%\maintenance\cjconsole[n].log</li> <li>UNIX の場合 /opt/Cosminexus/CC/admin/logs/CC/maintenance/cjconsole[n].log</li> </ul>	32KB × 3	ConsoleLogFile
	サーバ管理コマンドの保守情報 <sup>2</sup>	<ul style="list-style-type: none"> <li>Windows の場合 &lt;Cosminexus のインストールディレクトリ &gt;%CC%\admin\logs\%CC%\maintenance\cjserveradmin[n].log</li> <li>UNIX の場合 /opt/Cosminexus/CC/admin/logs/CC/maintenance/cjserveradmin[n].log</li> </ul>	32KB × 3	ServerAdminLogFile

(凡例) - : 該当なし

注

チャンネル名は、ログの出力先を識別する名称です。ログの属性（サイズ、面数）を変更する場合はキー値として使用します。

注 1

ログファイル名の [n] の部分には、面の番号（1 から面数（最大 16）まで）が付きます。

注 2

日立トレース共通ライブラリ形式の出力メッセージ（アプリケーション識別名）には、コマンド名が表示されます。

注 3

互換モードの場合、稼働ログと障害発生時の例外情報の出力先は標準モードと異なります。互換モードの場合の出力先とデフォルトのサイズ・面数は次のようになります。

表 5-24 サーバ管理コマンドのログの出力先（互換モード）

内容	ログ出力先およびログファイル名	デフォルトのサイズ×面数
稼働ログ	<ul style="list-style-type: none"> <li>Windows の場合 &lt;Cosminexus インストールディレクトリ&gt;%CC%\admin\logs&lt; コマンド名称 &gt;message[n].log</li> <li>UNIX の場合 /opt/Cosminexus/CC/admin/logs/&lt; コマンド名称 &gt;message[n].log</li> </ul>	128KB × 2
障害発生時の例外情報	<ul style="list-style-type: none"> <li>Windows の場合 &lt;Cosminexus インストールディレクトリ&gt;%CC%\admin\logs&lt; コマンド名称 &gt;exception[n].log</li> <li>UNIX の場合 /opt/Cosminexus/CC/admin/logs/&lt; コマンド名称 &gt;exception[n].log</li> </ul>	256KB × 2

## 注

ログファイル名の [n] の部分には、面の番号（1 から面数（最大 16）まで）が付きます。

サーバ管理コマンドのメッセージログに出力されるメッセージには、メッセージ ID フィールドが空白で、メッセージテキストフィールドにメッセージ ID（KDJE<sup>n</sup>nnnn-Y など）が含まれる場合があります。それらはサーバ側で発生したメッセージで、前後に出力されるメッセージの付加情報となります。

## バッチアプリケーションのログ

表 5-25 バッチアプリケーションのログの出力先（デフォルト）

分類	内容	ログ出力先およびログファイル名 <sup>1</sup>	デフォルトのサイズ×面数
メッセージログ	cexecjob コマンドおよび ckilljob コマンドの稼働ログ	<ul style="list-style-type: none"> <li>Windows の場合 &lt;batch.log.directory&gt;<sup>2</sup>%cjmessage[n].log</li> <li>UNIX の場合 &lt;batch.log.directory&gt;<sup>2</sup>/cjmessage[n].log</li> </ul>	1MB × 2

## 注 1

ログファイル名の [n] の部分には、面の番号（1 から面数（最大 16）まで）が付きます。

## 注 2

<batch.log.directory> は、バッチアプリケーション用オプション定義ファイル（usrconf.cfg ファイル）の batch.log.directory で指定されたディレクトリを指します。デフォルト値は、次のとおりです。

Windows の場合

<Cosminexus のインストールディレクトリ>%CC%\batch\logs

UNIX の場合

## 5. トラブルシューティング

/opt/Cosminexus/CC/batch/logs

リソースアダプタバージョンアップコマンド (cjrupdate) のログ

表 5-26 リソースアダプタバージョンアップコマンド (cjrupdate) のログの出力先

分類	内容	ログ出力先およびログファイル名	デフォルトのサイズ × 面数	チャネル名
メッセージログ	稼働ログ	<ul style="list-style-type: none"> <li>Windows の場合 &lt;Cosminexus のインストールディレクトリ&gt;%CC%\logs\%cjrupdatemessage[n].log</li> <li>UNIX の場合 /opt/Cosminexus/CC/logs/cjrupdatemessage[n].log</li> </ul>	1MB × 2	-
例外ログ	障害発生時の例外情報	<ul style="list-style-type: none"> <li>Windows の場合 &lt;Cosminexus のインストールディレクトリ&gt;%CC%\logs\%cjrupdateexception[n].log</li> <li>UNIX の場合 /opt/Cosminexus/CC/logs/cjrupdateexception[n].log</li> </ul>	1MB × 2	-
保守用ログ	保守情報	<ul style="list-style-type: none"> <li>Windows の場合 &lt;Cosminexus のインストールディレクトリ&gt;%CC%\logs\%cjrupdatemaintenance[n].log</li> <li>UNIX の場合 /opt/Cosminexus/CC/logs/cjrupdatemaintenance[n].log</li> </ul>	1MB × 2	-

(凡例) - : 該当なし

注

チャネル名は、ログの出力先を識別する名称です。ログの属性 (サイズ, 面数) を変更する場合のキー値として使用します。

注

ログファイル名の [n] の部分には、面の番号 (1 から面数 (最大 16) まで) が付きます。



## 移行コマンド (cjenvupdate) のログ

表 5-27 移行コマンド (cjenvupdate) のログの出力先

分類	内容	ログ出力先およびログファイル名	デフォルトのサイズ × 面数	チャンネル名
メッセージログ	cjenvupdate コマンドの稼働ログ	<ul style="list-style-type: none"> <li>Windows の場合 &lt;Cosminexus のインストールディレクトリ&gt;%CC%\logs\cjenvupdatemessage[n].log</li> <li>UNIX の場合 /opt/Cosminexus/CC/logs/cjenvupdatemessage[n].log</li> </ul>	4MB × 4	-
例外ログ	cjenvupdate コマンドの例外情報	<ul style="list-style-type: none"> <li>Windows の場合 &lt;Cosminexus のインストールディレクトリ&gt;%CC%\logs\cjenvupdateexception[n].log</li> <li>UNIX の場合 /opt/Cosminexus/CC/logs/cjenvupdateexception[n].log</li> </ul>	4MB × 4	-
保守用ログ	cjenvupdate コマンドの保守情報	<ul style="list-style-type: none"> <li>Windows の場合 &lt;Cosminexus のインストールディレクトリ&gt;%CC%\logs\cjenvupdatemaintenance[n].log</li> <li>UNIX の場合 /opt/Cosminexus/CC/logs/cjenvupdatemaintenance[n].log</li> </ul>	4MB × 4	-

(凡例) - : 該当なし

注

[n] には、面の番号 (1 から面数 (最大 16) まで) が付きます。

## リソース枯渇監視のログ

表 5-28 リソース枯渇監視のログの出力先

監視対象リソース	ログ取得場所およびログファイル名 <sup>1</sup>	デフォルトのサイズ × 面数	チャンネル名 <sup>2</sup>
メモリ	<ul style="list-style-type: none"> <li>Windows の場合 &lt;ejb.server.log.directory&gt; <sup>3</sup>%watch%\cjmemorywatch[n].log</li> <li>UNIX の場合 &lt;ejb.server.log.directory&gt; <sup>3</sup>/watch/ cjmemorywatch[n].log</li> </ul>	1MB × 2	MemoryWatchLogFile

## 5. トラブルシューティング

監視対象リソース	ログ取得場所およびログファイル名 <sup>1</sup>	デフォルトのサイズ×面数	チャンネル名 <sup>2</sup>
ファイルディスクリプタ	<ul style="list-style-type: none"> <li>UNIX の場合 <sup>4</sup> &lt;ejb.server.log.directory&gt; <sup>3</sup>/watch/ cjfiledescriptorwatch[n].log</li> </ul>	1MB × 2	FileDescriptorWatchLogFile
スレッド	<ul style="list-style-type: none"> <li>Windows の場合 &lt;ejb.server.log.directory&gt; <sup>3</sup>¥watch¥cjthreadwatch[n].log</li> <li>UNIX の場合 &lt;ejb.server.log.directory&gt; <sup>3</sup>/watch/ cjthreadwatch[n].log</li> </ul>	1MB × 2	ThreadWatchLogFile
スレッドダンプ	<ul style="list-style-type: none"> <li>Windows の場合 &lt;ejb.server.log.directory&gt; <sup>3</sup>¥watch¥cjthreaddumpwatch[n].log</li> <li>UNIX の場合 &lt;ejb.server.log.directory&gt; <sup>3</sup>/watch/ cjthreaddumpwatch[n].log</li> </ul>	1MB × 2	ThreaddumpWatchLogFile
HTTP リクエスト実行待ちキュー	<ul style="list-style-type: none"> <li>Windows の場合 &lt;ejb.server.log.directory&gt; <sup>3</sup>¥watch¥cjrequestqueewatch[n].log</li> <li>UNIX の場合 &lt;ejb.server.log.directory&gt; <sup>3</sup>/watch/ cjrequestqueewatch[n].log</li> </ul>	1MB × 2	RequestQueueWatchLogFile
HTTP セッション数	<ul style="list-style-type: none"> <li>Windows の場合 &lt;ejb.server.log.directory&gt; <sup>3</sup>¥watch¥cjhttpsessionwatch[n].log</li> <li>UNIX の場合 &lt;ejb.server.log.directory&gt; <sup>3</sup>/watch/ cjhttpsessionwatch[n].log</li> </ul>	1MB × 2	HttpSessionWatchLogFile
コネクションプール	<ul style="list-style-type: none"> <li>Windows の場合 &lt;ejb.server.log.directory&gt; <sup>3</sup>¥watch¥cjconnectionpoolwatch[n].log</li> <li>UNIX の場合 &lt;ejb.server.log.directory&gt; <sup>3</sup>/watch/ cjconnectionpoolwatch[n].log</li> </ul>	1MB × 2	ConnectionPoolWatchLogFile

注 1

[n]には、面の番号（1から面数（最大16）まで）が付きます。

注 2

ログの出力先を識別する名称で、ログの属性（サイズ、面数）を変更する場合のキー値として使用します。

注 3

<ejb.server.log.directory> は、簡易構築定義ファイルの論理 J2EE サーバ（j2ee-server）の <configuration> タグ内に、ejb.server.log.directory パラメタで指定したディレクトリを指しま

す。デフォルト値は、<Cosminexus のインストールディレクトリ  
>¥CC¥server¥public¥ejb¥<サーバ名称>¥logs です。

簡易構築定義ファイルの ejb.server.log.directory パラメタの詳細については、マニュアル  
「Cosminexus 簡易構築・運用ガイド」を参照してください。

注 4

Windows の場合、または AIX の場合は、ファイルディスクリプタを監視できません。

リソース枯渇監視ログファイルに出力される情報やログファイルの出力形式については、  
「2.5 リソース枯渇の監視」を参照してください。

ログの出力先を設定するユーザ定義ファイル

バッチサーバおよびサーバ管理コマンドのログの出力先を変更している場合は、次の表  
に示す、ログの出力先を設定するユーザ定義ファイルを参照して出力先を確認してくだ  
さい。なお、ログの出力先を変更した場合、変更後の出力先は snapshot ログの一括収集  
時に収集対象外となります。必要に応じて snapshot ログの収集先を変更してください。

表 5-29 ログの出力先を設定するユーザ定義ファイル

分類	ユーザ定義ファイルの指定箇所
バッチサーバ	簡易構築定義ファイルの論理 J2EE サーバ (j2ee-server) の <configuration> タグ内に指定する、ejb.server.log.directory パラメタ。 デフォルトは、「<Cosminexus のインストールディレクトリ >¥CC¥server¥public¥ejb¥<サーバ名称>¥logs」(Windows の場合)、また は「/opt/Cosminexus/CC/server/public/ejb/<サーバ名称>/logs」(UNIX の 場合)です。
サーバ管理コマンド	サーバ管理コマンドの usrconf.bat (Windows の場合)、または usrconf (UNIX の場合) の ejbserver.log.directory キー。 キーのデフォルトは、「<Cosminexus のインストールディレクトリ >¥CC¥admin¥logs」(Windows の場合)、または「/opt/Cosminexus/CC/ admin/logs」(UNIX の場合)です。 Management Server リモート管理機能から操作した場合は、サーバ管理コマ ンドのログ出力先は変更できません。

ログの出力先の変更方法など、トラブルシューティングの資料取得の設定については、  
マニュアル「Cosminexus システム構築ガイド」を参照してください。

(b) 運用管理エージェント・運用監視エージェント・Management Server のログの取得

ここでは、運用管理エージェント、運用監視エージェント、および Management Server  
のログの出力先について説明します。

運用管理エージェント・運用監視エージェント・Management Server のログには、個別  
に取得する以外に、統合ログとしてまとめて取得できるものがあります。

運用管理エージェント・運用監視エージェント・Management Server のログを統合ログ  
としてまとめて取得する場合、および個別に取得する場合の出力先については、「5.4.1  
Cosminexus のログの取得 (J2EE アプリケーションを実行するシステム)」の「(1)(b)

## 5. トラブルシューティング

運用管理エージェント・運用監視エージェント・Management Server のログの取得」を参照してください。

### (c) ログ以外に取得が必要な情報

ログ以外に取得する必要がある情報については、「5.4.1 Cosminexus のログの取得 (J2EE アプリケーションを実行するシステム)」の「(1)(d) ログ以外に取得が必要な情報」を参照してください。

## (2) アプリケーションのユーザログの取得

バッチアプリケーションのユーザログは、日立トレース共通ライブラリ形式で出力するように設定している場合に取得できます。

バッチアプリケーションのユーザログの出力先については、次のディレクトリに、簡易構築定義ファイルの論理 J2EE サーバ (j2ee-server) の <configuration> タグ内に ejbserver.application.userlog.CJLogHandler.<ハンドラ名>.path パラメタで指定したプリフィックスを持つファイル名で出力されます。なお、<ハンドラ名>には、キーの値を区別するためのハンドラ名が指定されています。

- Windows の場合  
<ログ出力先ルート (ejb.server.log.directory の値)>¥user¥ (デフォルトは <Cosminexus 作業ディレクトリ>¥ejb¥<バッチサーバ名>¥logs¥user)
- UNIX の場合  
<ログ出力先ルート (ejb.server.log.directory の値)>/user/ (デフォルトは <Cosminexus 作業ディレクトリ>/ejb/<バッチサーバ名>/logs/user)

なお、バッチアプリケーションのユーザログ出力の設定については、マニュアル「Cosminexus システム構築ガイド」を参照してください。

## 5.4.3 EJB クライアントアプリケーションのシステムログの取得

EJB クライアントアプリケーションのシステムログの種類と出力先について説明します。

なお、EJB クライアントアプリケーションのシステムログを運用するときには留意することについては、「5.7.1 EJB クライアントアプリケーションのシステムログに関する留意事項」を参照してください。

### ! 注意事項

uCosminexus Client を使用する場合は、EJB クライアントアプリケーションのログの格納ディレクトリの「<Cosminexus のインストールディレクトリ>¥CC」を、「<Cosminexus のインストールディレクトリ>¥CCL」と読み替えてください。

## (1) EJB クライアントアプリケーションのシステムログの種類

EJB クライアントアプリケーションでは、EJB クライアントアプリケーションのプロセス単位にシステムログを出力します。EJB クライアントアプリケーションのシステムログの種類を次に示します。

表 5-30 EJB クライアントアプリケーションのシステムログの種類

種類	ログの内容	ファイル名	デフォルトのサイズ×面数	チャンネル名
メッセージログ	稼働ログ	cjclmessage[n].log <sup>1</sup>	1MB × 2	ClientMessageLogFile
	cjclstartap コマンドの稼働ログ	cjclstartap[n].log <sup>2</sup>	1MB × 2	-
	cjcldellog コマンドの稼働ログ	cjcldellog.log	1MB × 2 <sup>3</sup>	-
ユーザログ	ユーザ出力ログ	user_out[n].log <sup>1</sup>	1MB × 2	UserOutLogFile
	ユーザエラーログ	user_err[n].log <sup>1</sup>	1MB × 2	UserErrLogFile
Java ログ	JavaVM の保守情報、ガーベージコレクションのログ	java[n].log <sup>1</sup>	256KB × 4	-
例外ログ	障害発生時の例外情報	cjclexception[n].log <sup>1</sup>	1MB × 2	ClientExceptionLogFile
保守用ログ <sup>4</sup>	保守情報	cjclmaintenance[n].log <sup>1</sup>	1MB × 2	ClientMaintenanceLogFile
	EJB コンテナの保守情報	cjejbcontainer[n].log <sup>1</sup>	1MB × 2	EJBContainerLogFile
	起動プロセス標準出力情報	cjstdout[n].log <sup>2</sup>	1MB × 2	-
	起動プロセス標準エラー情報	cjstderr[n].log <sup>2</sup>	1MB × 2	-
	ログの稼働情報	cjlogger.log	1MB × 2 <sup>3</sup>	-

(凡例) - : 該当なし

注 1

ファイル名の [n] の部分には、1 ~ 指定したログの面数の通し番号が付きます。

注 2

ファイル名の [n] の部分には指定したログの面数の番号 (1 または 2) が付きます。

注 3

cjcldellog.log のサイズが 1MB を超えた場合、cjcldellog\_save.log というバックアップファイル名称のログファイルになります。

## 5. トラブルシューティング

### 注 4

このログは、保守員へ送付する場合に必要に応じて収集してください。

### (2) EJB クライアントアプリケーションのシステムログの出力先

EJB クライアントアプリケーションのシステムログは、次のキーに指定したディレクトリに出力されます。

- Java アプリケーション用オプション定義ファイル (usrconf.cfg) の `ejb.client.log.directory` キー、および `ejb.client.ejb.log` キーに指定したディレクトリ
- システムプロパティの `ejbserver.client.log.directory` キーおよび `ejbserver.client.ejb.log` キーに指定したディレクトリ

Java アプリケーション用オプション定義ファイル (usrconf.cfg) とシステムプロパティの両方に異なる値が指定された場合、システムプロパティの設定が有効になります。

なお、システムプロパティで出力先を設定できるのは、次のログだけです。

- 稼働ログ (cjclmessage[n].log)
- 障害発生時の例外情報 (cjclexception[n].log)
- 保守情報 (cjclmaintenance[n].log)
- ログの稼働情報 (cjlogger.log)

`ejcldellog` コマンドの稼働ログ (`ejcldellog.log`) および `ejclstartap` コマンドの稼働ログ (`ejclstartap[n].log`) は、「<Cosminexus のインストールディレクトリ>¥CC¥client¥logs」(Windows の場合)、または「/opt/Cosminexus/CC/client/logs」(UNIX の場合)の直下に出力されます。

### ! 注意事項

- `ejclstartap` コマンドはサブディレクトリ共有モードだけで使用してください。サブディレクトリ専有モードで `ejclstartap` コマンドを実行して EJB クライアントを実行した場合、`ejclstartap` コマンドの稼働ログ (`ejclstartap[n].log`) が正しく表示されません。
- EJB クライアントアプリケーションの標準出力、および標準エラーの内容は、ログファイルには出力されません。ログファイルに出力する場合は、ユーザログ機能を使用するか、リダイレクトしてください。

EJB クライアントアプリケーションのシステムログの出力方法には、次の 2 種類のモードがあります。モードによって、出力先になるサブディレクトリの構成が異なります。これらのモードは、クラスパスの指定によって選択します。

#### サブディレクトリ共有モード

複数のプロセスのログを一つのディレクトリ (デフォルトは `ejbel` ディレクトリ) 下に出力するモードです。

#### サブディレクトリ専有モード

それぞれのプロセスのログを、プロセス単位に作成されたサブディレクトリに別々に出力するモードです。

Cosminexus の 06-50 よりも前のバージョンとの互換用に使用するモードです。

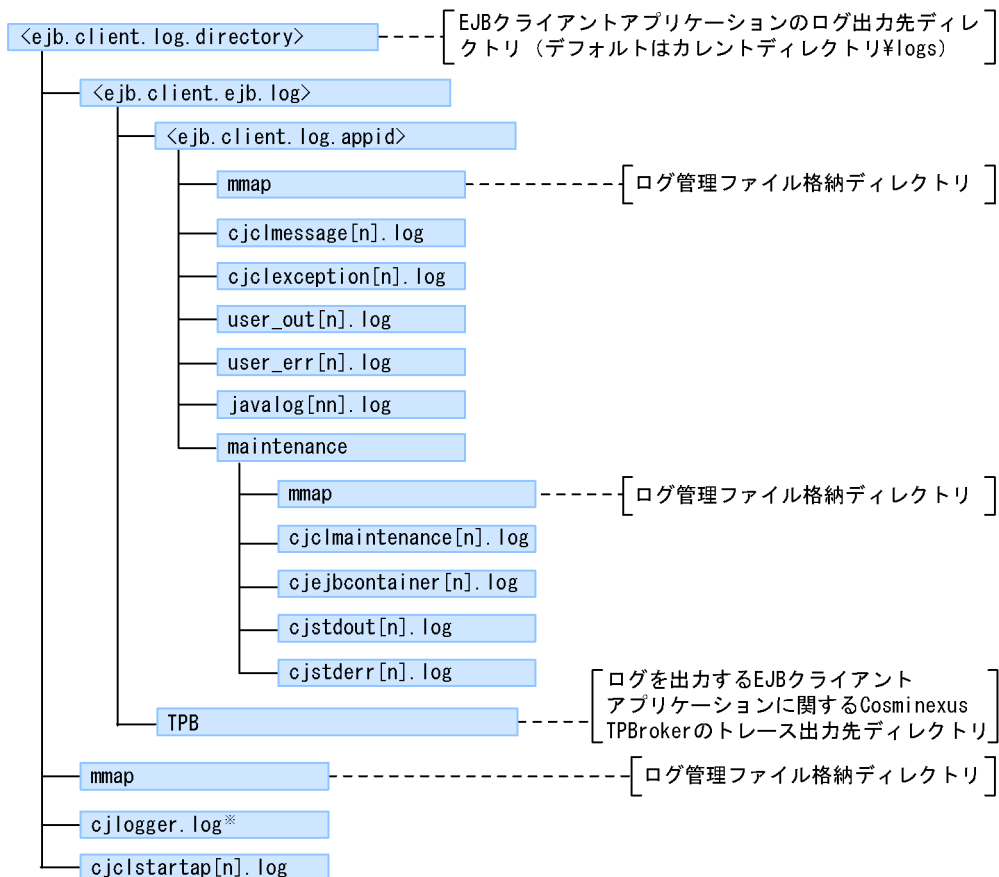
サブディレクトリ共有モードとサブディレクトリ専有モードの違いの詳細については、マニュアル「Cosminexus システム構築ガイド」の EJB クライアントアプリケーションのログの種類に関する説明を参照してください。

(a) サブディレクトリ共有モードの場合

EJB クライアントアプリケーションのシステムログの出力先を次の図に示します。

## 5. トラブルシューティング

図 5-3 サブディレクトリ共有モードの EJB クライアントアプリケーションのシステムログの出力先 (Java アプリケーション用オプション定義ファイルで指定した場合)



### 注

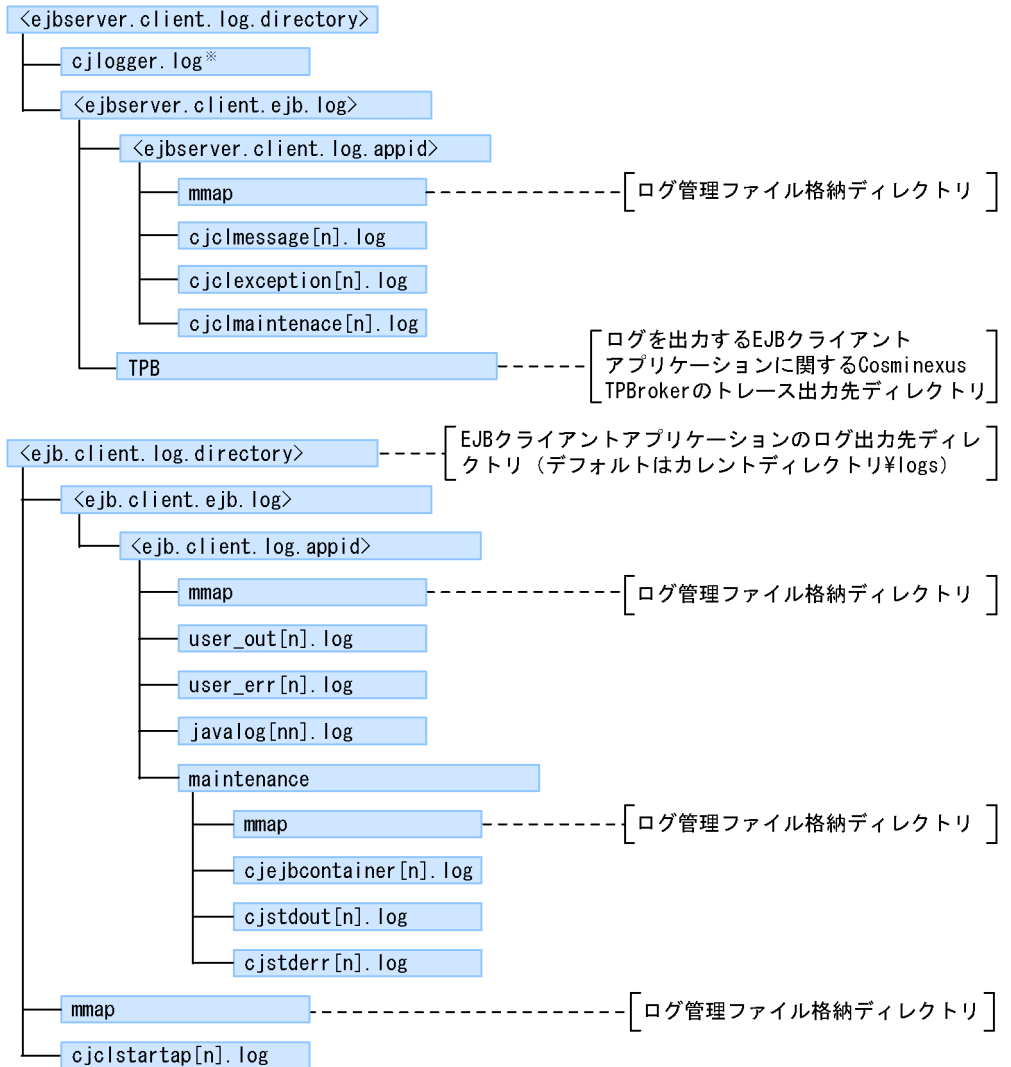
ファイル名の[n]は、1～(指定したログの面数)の通し番号になります。

### 注※

cjlogger.logは、EJBクライアントアプリケーションのプロパティの設定などにエラーがある場合だけ出力されます。



図 5-4 サブディレクトリ共有モードの EJB クライアントアプリケーションのシステムログの出力先 (システムプロパティで指定した場合)



注

ファイル名の[n]は、1～(指定したログの面数)の通し番号になります。

注※

cjlogger.logは、EJBクライアントアプリケーションのプロパティの設定などにエラーがある場合だけ出力されます。

#### (b) サブディレクトリ専有モードの場合

EJB クライアントアプリケーションのシステムログは、ejbserver.client.log.directory キーに指定したディレクトリ下の、EJB クライアントアプリケーションのプロセス単位に作成されるサブディレクトリ下に格納されます。EJB クライアントアプリケーション

## 5. トラブルシューティング

のプロセス単位のサブディレクトリは、ejbserver.client.log.directory キーに指定したディレクトリ下の、ejbserver.client.ejb.log キーに指定したディレクトリ下に作成されます。デフォルトでは、「<Cosminexus のインストールディレクトリ>¥CC¥client¥logs¥ejb」(Windows の場合)、または「/opt/Cosminexus/CC/client/logs/ejb」(UNIX の場合)の下にサブディレクトリが作成されます。

EJB クライアントアプリケーションのシステムログ出力先のサブディレクトリ名を次に示します。

表 5-31 EJB クライアントアプリケーションのシステムログ出力先のサブディレクトリ名

コマンド名	サブディレクトリ名
vbj コマンド	cjcl<日時>_<プロセス ID>

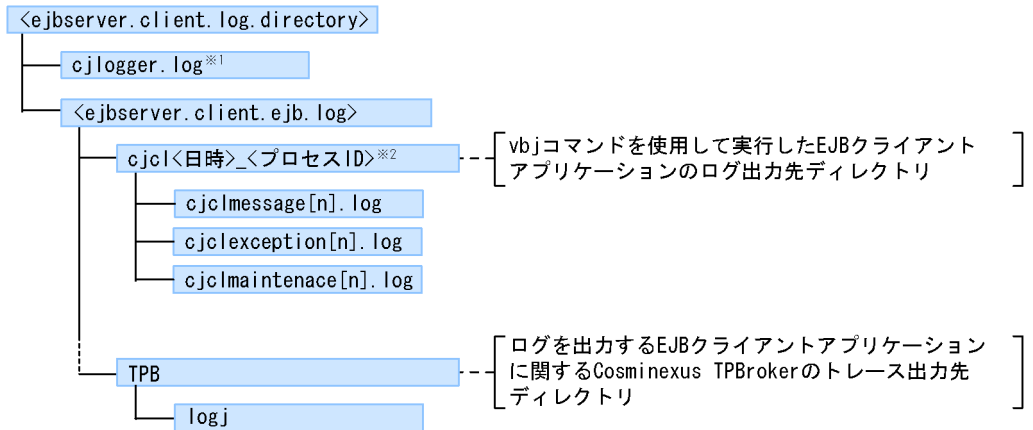
### 注

<日時>には、EJB クライアントアプリケーションを開始した日時が「mddhhmmssmmm」の形式で表示されます。日時に表示される項目は次のとおりです。なお、月以外は 10 進数で表示されます。

m : 月 (16 進数) dd : 日付 hh : 時間 mm : 分 ss : 秒 mmm : ミリ秒

EJB クライアントアプリケーションのシステムログ出力先のディレクトリ構成例を次に示します。

図 5-5 サブディレクトリ専有モードの EJB クライアントアプリケーションのシステムログの出力先



注

ファイル名の [n] は、1～（指定したログの面数）の通し番号になります。

注※1

cjlogger.log は、EJB クライアントアプリケーションのプロパティの設定などにエラーがある場合だけ出力されます。

注※2

EJB クライアントアプリケーションのシステムプロパティの `ejbserver.client.log.appid` キーにサブディレクトリ名を指定している場合、指定した名称でサブディレクトリが作成されます。

サブディレクトリは、次のように作成および運用されます。

- EJB クライアントアプリケーション単位に作成するサブディレクトリは、上限値を設定できます。EJB クライアントアプリケーションが終了すると、サブディレクトリに `client_process_end` という空のファイルが作成されます。上限値を超えた場合は、このファイルの作成時刻が古い順にサブディレクトリが削除されます。なお、サブディレクトリにログを出力している EJB クライアントアプリケーションがどれも終了していない（`client_process_end` ファイルが作成されていない）場合は、上限値を超えてサブディレクトリが作成されます。作成されたサブディレクトリがあとで上限値まで削除されることはありません。
- `ejbserver.client.log.directory` キーに指定したディレクトリ下の、`ejbserver.client.ejb.log` キーに指定したディレクトリ下にユーザ作成のサブディレクトリがある場合、EJB クライアントアプリケーション単位に作成するサブディレクトリと同様に、サブディレクトリとして数えられます。
- EJB クライアントアプリケーションが異常終了すると、`client_process_end` ファイルが作成されないことがあるため、異常終了した EJB クライアントアプリケーションのサブディレクトリは上限値を超えても削除されないことがあります。

## 5. トラブルシューティング

- 一つの EJB クライアントアプリケーションを一つだけ起動する場合（同時実行しない場合）には、その EJB クライアントアプリケーション固有のシステムログ出力先ディレクトリを作成して、常にそのディレクトリに出力する運用ができます。このサブディレクトリは、上限値の指定に関係なく、削除されません。
- サブディレクトリを削除する場合、`ejcldellog` コマンドを使用すると、指定した更新日または更新時間より前に作成されたサブディレクトリを一括して削除できます。なお、`ejcldellog` コマンドは、Windows の場合は Windows のタスク、UNIX の場合は `at` コマンドまたは `crontab` コマンドを利用して定期的に行ってください。

EJB クライアントアプリケーションのログ出力の設定については、マニュアル「Cosminexus システム構築ガイド」を参照してください。

### 5.4.4 性能解析トレースの取得

性能解析トレースの収集方法については、「3. システムの処理性能の解析」を参照してください。性能解析トレースには、セッショントレースも出力されている場合があります。

性能解析トレースの格納場所は次のとおりです。

- Windows の場合  
<環境変数 PRFSPOOL の設定ディレクトリ>\%utt%\prf\<PRF 識別子>
- UNIX の場合  
\$PRFSPOOL/utt/prf/<PRF 識別子>

### 5.4.5 JavaVM のスレッドダンプの取得

ここでは、JavaVM のスレッドダンプを取得する方法について説明します。

JavaVM のスレッドダンプは、次の方法で取得できます。

- 運用管理コマンド (`mngsvrutil`) を使用して、J2EE サーバ、SFO サーバ、CORBA ネーミングサービス、および CTM のスレッドダンプを取得する。
- 個別のコマンドを使用して、J2EE サーバ、CORBA ネーミングサービス、および EJB クライアントアプリケーションのスレッドダンプを取得する。

それぞれの方法について次に説明します。

#### (1) 運用管理コマンドを使用する場合

運用管理コマンド (`mngsvrutil`) を利用した JavaVM のスレッドダンプの取得は、`mngsvrutil` コマンドのサブコマンド「`dump`」の引数に `server` を指定して実行します。

次のスレッドダンプを取得できます。

- J2EE サーバ（クラスタを含む）

- SFO サーバ
- CORBA ネーミングサービスおよび CTM

mngsvrutil コマンドおよびそのサブコマンドの詳細については、マニュアル「Cosminexus リファレンス コマンド編」を参照してください。

実行形式，実行例，および出力先を次に示します。

#### 実行形式

```
mngsvrutil -m <Management Serverのホスト名> -u <管理ユーザID> -p <管理パスワード> -t
<論理サーバ名> dump server
```

#### 実行例

```
mngsvrutil -m mnghost -u user01 -p pwl -t myserver dump server
```

#### 出力先

##### J2EE サーバを対象にする場合

- Windows の場合  
<作業ディレクトリ> %ejb%\<サーバ名称>\javacore\*.txt
- UNIX の場合  
<作業ディレクトリ> /ejb/<サーバ名称>/javacore\*.txt

注 <作業ディレクトリ> は、J2EE サーバのユーザ定義（usrconf.cfg ファイル中の ejb.public.directory）で指定されたディレクトリを指します。デフォルト値は、次のとおりです。

- Windows の場合  
<Cosminexus のインストールディレクトリ>%CC%\server\public
- UNIX の場合  
/opt/Cosminexus/CC/server/public

##### SFO サーバを対象にする場合

- Windows の場合  
<Cosminexus のインストールディレクトリ>%CC%\server\public\ejb%\<サーバ名称>\javacore\*.txt
- UNIX の場合  
/opt/Cosminexus/CC/server/public/ejb/<サーバ名称>/javacore\*.txt

##### CORBA ネーミングサービスおよび CTM を対象にする場合

- Windows の場合  
<Cosminexus のインストールディレクトリ>%TPB%\log\javacore\*.txt
- UNIX の場合

## 5. トラブルシューティング

/opt/Cosminexus/TPB/logj/javacore\*.txt

### (2) 個別のコマンドを使用する場合

スレッドダンプの出力のされ方は、指定している J2EE サーバを実行する JavaVM の起動オプションによって異なります。

-XX:+HitachiThreadDump が設定されている場合、拡張スレッドダンプを取得できません。このオプションはデフォルトで設定されています。

-XX:+HitachiThreadDumpToStdout が設定されている場合、スレッドダンプは標準出力にも出力されます。このオプションはデフォルトでは設定されていません。必要に応じて設定してください。

-XX:+HitachiOutOfMemoryAbort が設定されている場合に、  
-XX:+HitachiOutOfMemoryAbortThreadDump が設定されていると、  
OutOfMemoryError によって強制終了した時に、スレッドダンプが出力されます。ただし、次の場合は除きます。

- J2SE クラスライブラリで C ヒープ不足が発生した場合
- JavaVM の処理中に C ヒープ不足が発生した場合

#### (a) J2EE サーバのスレッドダンプの取得

J2EE サーバプロセス (cjstartsv) が存在する場合、J2EE サーバのスレッドダンプは、cjdumpsv コマンドを実行して取得します。cjdumpsv コマンドの実行例を次に示します。時間の経過に応じた各スレッドの状態遷移を確認するため、複数回 cjdumpsv コマンドを実行します。目安として 3 秒おきに 10 回程度実行します。

- Windows の場合

```
<Cosminexusのインストールディレクトリ>%CC%server%bin%cjdumpsv <J2EEサーバ名>
```

- UNIX の場合

```
/opt/Cosminexus/CC/server/bin/cjdumpsv <J2EEサーバ名>
```

cjdumpsv コマンドを実行すると、次に示すファイルに JavaVM のスレッドダンプが出力されます。

- サーバ標準出力ログ
- <作業ディレクトリ>%ejb%<サーバ名称>%javacore<プロセス番号><コマンド実行日時>.txt (Windows の場合)
- <作業ディレクトリ>/ejb/<サーバ名称>/javacore<プロセス番号><コマンド実行日時>.txt (UNIX の場合)

サーバ標準出力ログのデフォルトの出力先は

「<ejb.server.log.directory>%CC%maintenance%cjstdout.log」(Windows の場合), または「<ejb.server.log.directory>/CC/maintenance/cjstdout.log」(UNIX の場合)です。出力先を変更している場合は、「5.4.1 Cosminexus のログの取得 (J2EE アプリケーションを実行するシステム)」または「5.4.2 Cosminexus のログの取得 (バッチアプリケーションを実行するシステム)」を参照してください。なお、作業ディレクトリのデフォルトのディレクトリパスは「<Cosminexus のインストールディレクトリ>%CC%server%public」(Windows の場合), または「/opt/Cosminexus/CC/server/public」(UNIX の場合)です。

また、javacore<プロセス番号>.<コマンド実行日時>.txt ファイルの出力先は、環境変数 JAVACOREDIREC で変更できます。ただし、指定したディレクトリへの書き込みに失敗した場合は、デフォルトの出力先に出力されます。このディレクトリにも出力できなかった場合は、標準エラー出力にだけ出力されます。

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

#### 参考

スレッドダンプが出力されると、標準出力に、次のメッセージが出力され、java プログラムの実行が継続されます。このメッセージは、-XX:+HitachiThreadDumpToStdout の設定とは関係なく出力されます。

```
Writing Java core to <ファイル名(フルパス)>...OK
```

#### (b) CORBA ネーミングサービスのスレッドダンプの取得

Windows の場合、CORBA ネーミングサービスのプロセス (nameserv) が存在するときは、CORBA ネーミングサービスを起動したコマンドプロンプトに対して Ctrl+Break キーを押します。時間の経過に応じた各スレッドの状態遷移を確認するため、複数回実行します。目安として 3 秒おきに 10 回程度実行します。なお、CORBA ネーミングサービスを Management Server から監視している場合はスレッドダンプを取得できません。

UNIX の場合、CORBA ネーミングサービスのプロセス (java) が存在するときは、kill コマンドを実行して CORBA ネーミングサービスのスレッドダンプを取得します。なお、CORBA ネーミングサービスを Management Server から監視している場合はスレッドダンプを取得できません。

UNIX での CORBA ネーミングサービスのスレッドダンプの取得手順を次に示します。

## 5. トラブルシューティング

### 1. CORBA ネーミングサービスのプロセス ID を取得します。

CORBA ネーミングサービスのプロセス ID の取得方法は、次の場合によって異なります。

java プロセスがほかに起動していない場合

```
ps -ef | grep java
```

java プロセスが複数起動している場合

CORBA ネーミングサービスの起動用シェルスクリプトを使用すると、カレントワーキングディレクトリに生成される `namesv_pid` ファイルに CORBA ネーミングサービスのプロセス ID を出力できます。

CORBA ネーミングサービスの起動用シェルスクリプトの例を次に示します。

```
#!/bin/sh
export VBROKER_ADM=/opt/Cosminexus/TPB/adm
export SHLIB_PATH="${SHLIB_PATH}:/opt/Cosminexus/TPB/lib"

# start name server process
exec /opt/Cosminexus/TPB/bin/nameserv ¥
-J-Dvbroker.agent.enableLocator=false ¥
-J-Djava.security.policy=/opt/Cosminexus/CC/server/sysconf/cli.policy ¥
-J-Dvbroker.se.iiop_tp.scm.iiop_tp.listener.port=900 &

# save background java process pid
echo $! > ./namesv_pid
```

### 2. 取得したプロセス ID を指定して、kill コマンドを実行します。

```
kill -3 `cat namesv_pid`
```

#### (c) EJB クライアントアプリケーションのスレッドダンプの取得

EJB クライアントアプリケーションのスレッドダンプは、`cjldumpap` コマンドを実行して取得します。

`cjldumpap` コマンドを実行すると、`cjelstartap` コマンドを実行して起動した EJB クライアントアプリケーションのスレッドダンプが出力されます。また、特定のプロセスのスレッドダンプを出力することもできます。`cjldumpap` コマンドの詳細については、マニュアル「Cosminexus リファレンス コマンド編」を参照してください。

`cjldumpap` コマンドの実行形式、実行例、およびスレッドダンプの出力先を次に示します。

#### 実行形式

`cjelstartap` コマンドを使用して起動した EJB クライアントアプリケーションのスレッドダンプを出力する場合



```
cjcldumpap
```

特定のプロセスのスレッドダンプを出力する場合

```
cjcldumpap <プロセスID>
```

#### 実行例

cjclstartap コマンドを使用して起動した EJB クライアントアプリケーションのスレッドダンプを出力する場合

```
cjcldumpap
```

特定のプロセスのスレッドダンプを出力する場合

```
cjcldumpap 3264
```

#### 出力先

cjclstartap コマンドを実行しているカレントディレクトリ

## 5.4.6 JavaVM のガーベージコレクションログの取得

JavaVM のガーベージコレクションログは、JavaVM または J2EE サーバ起動前にログ出力先が設定されている場合にだけ取得できます。

ガーベージコレクションログの出力先は、簡易構築定義ファイルの論理 J2EE サーバ (j2ee-server) の <configuration> タグ内に、ejb.server.log.directory パラメタで指定します。

また、実行中の Java プロセスでガーベージコレクションを発生させたい場合は、javagc コマンドを実行してください。javagc コマンドは、トラブル発生時の要因調査のほか、1 トランザクション当たりのメモリ使用量の測定、メモリリークの調査、アプリケーションのデバッグなどの目的でも使用できます。

実行中の Java プロセスを指定してガーベージコレクションを発生させる場合の javagc コマンドの実行形式を次に示します。なお、これ以外に指定できるオプションについては、マニュアル「Cosminexus リファレンス コマンド編」を参照してください。

#### 実行形式

```
javagc -p <プロセスID>
```

実行結果として、次のログが出力されます。なお、これは、オプションとして、-XX:+HitachiVerboseGCPrintCause が指定されている場合の例です。

## 5. トラブルシューティング

```
[VGC]<Wed Mar 17 00:42:30 2004>(Skip Full:0,Copy:0)[ Full GC
149K->149K(1984K),0.0786038
secs] [ DefNew::Eden:264K->0K(512K)] [ DefNew::Survivor:0K->63K(64K)] [
Tenured:
85K->149K(1408K)] [ Perm:1300K->1300K(1536K)] [ cause:JavaGC Command]
```

### 5.4.7 メモリダンプの取得

J2EE サーバまたは CORBA ネーミングサービスを再起動する場合は、メモリダンプとして次のファイルを取得します。

ユーザダンプ (Windows の場合)

J2EE サーバのメモリダンプ

CORBA ネーミングサービスのメモリダンプ

これらのファイルは、システムでトラブルが発生したときに、保守員が障害を解析するために使用します。

#### (1) ユーザダンプの取得 (Windows の場合)

プロセスダウンが発生した場合に次のファイルがあれば採取します。

<ワトソン博士のクラッシュダンプ出力先指定ディレクトリ>¥user.dmp

<ワトソン博士のクラッシュダンプ出力先指定ディレクトリ>は、ワトソン博士 (drwtsn32) を実行してクラッシュダンプ欄のパスを確認してください。

プロセスのハングアップやスローダウンが発生した場合にユーザダンプを取得するときは、プロセス ID を確認して userdump.exe コマンドを実行します。userdump.exe は、Microsoft によって提供されているコマンドです。

```
userdump.exe <プロセスID>
```

userdump.exe コマンドでユーザダンプを取得した場合、ユーザダンプは userdump.exe を実行したカレントディレクトリに出力されます。

#### (2) J2EE サーバのメモリダンプの取得

J2EE サーバのメモリダンプの取得について、OS ごとに説明します。

##### (a) Windows の場合

J2EE サーバが稼働している場合 (cjstartsv プロセスが存在する場合)、タスクマネージャから cjstartsv のプロセス ID を取得します。userdump.exe <プロセス ID> を実行し、メモリダンプ (user.dump) を採取します。

## (b) UNIX の場合

cjstartsv プロセスがダウンした場合、「<作業ディレクトリ>/ejb/<サーバ名称>」に出力された core ダンプを取得します。

core ダンプのファイル名は、cjstartsv プロセスの再起動時に「core.<出力日時 >」(AIX, HP-UX, Solaris の場合)または「core.<プロセス ID>.<出力日時 >」(Linux の場合)にリネームされます。cjstartsv プロセスの再起動時に core ダンプが上書き保存されないため、障害発生時の core ダンプを保存できます。

## 注

出力日時は「YYMMDDhhmmss」の形式で出力されます。

YY: 西暦(下2けた) MM: 月(2けた) DD: 日(2けた)

hh: 時(24時間表記で2けた) mm: 分(2けた) ss: 秒(2けた)

なお、保存する core ダンプは上限値を設定できます。Windows の場合、cjstartsv プロセスの再起動時、および javacore コマンド実行時に、出力日時が古い順に削除されます。UNIX の場合、「<作業ディレクトリ>/ejb/<サーバ名称>」に出力される core ダンプファイルの合計が上限値を超えたときに、出力日時が古い順に削除されます。上限値は、簡易構築定義ファイルの論理 J2EE サーバ(j2ee-server)の<configuration>タグ内に、J2EE サーバの拡張パラメタの ejb.server.corefilenum で指定します。なお、ファイルの削除は cjstartsv プロセスの再起動時に実行されます。core ダンプファイル数の上限値の設定については、マニュアル「Cosminexus システム構築ガイド」を参照してください。

core ダンプを取得したあとに、core ダンプからスタックトレース情報だけを取得する場合は、javatrace コマンドを実行してください。スタックトレース情報は、JavaVM の異常終了の原因を究明するために必要な情報です。スタックトレース情報の取得方法については、「5.4.16 JavaVM のスタックトレース情報の取得」を参照してください。

core ダンプは次のような場合にも取得できます。それぞれの場合の取得方法について説明します。

cjstartsv プロセスが稼働している状態で core ダンプを取得する場合

cjstartsv プロセス(J2EE サーバ)が稼働している状態で core ダンプを取得する場合、cjstartsv プロセスのプロセス ID を確認して、kill コマンドを実行します。kill コマンドは次の形式で実行してください。なお、kill コマンドを実行するとプロセスが終了するため、再起動する前に kill コマンドを実行することを推奨します。

```
kill -6 <cjstartsvのプロセスID>
```

実行中の Java プロセスで core ダンプとスレッドダンプを同時に取得する場合

## 5. トラブルシューティング

実行中の Java プロセスで core ダンプとスレッドダンプを同時に取得する場合は、javacore コマンドを実行してください。javacore コマンドの実行形式を次に示します。なお、指定できるオプションについては、マニュアル「Cosminexus リファレンス コマンド編」を参照してください。

```
javacore <-プロセスID>
```

上記の形式でコマンドを実行すると、次のメッセージが出力されます。

```
send SIGQUIT to 8662: ? (y/n)
```

「y」を入力すると、実行中の Java プログラムのカレントディレクトリに、"javacore<プロセス ID>.<出力日時>.core" (core ダンプ)、および "javacore<プロセス ID>.<出力日時>.txt" (スレッドダンプ) が出力されます。「n」を入力すると、core ダンプおよびスレッドダンプを取得しないでコマンドの実行を終了します。core ダンプおよびスレッドダンプを取得すると、実行中の Java プログラムには、次のメッセージが出力されます。なお、斜体の部分は実際には表示されません。

```
Now generating core file (javacore8662.030806215140.core)...  
done
```

```
(core ダンプおよびスレッドダンプの出力終了)
```

```
Writing Java core to javacore8662.030806215140.txt... OK
```

### (3) CORBA ネーミングサービスのメモリダンプの取得

CORBA ネーミングサービスのメモリダンプの取得について OS ごとに説明します。

#### (a) Windows の場合

CORBA ネーミングサービスが稼働している場合 (CORBA ネーミングサービスのプロセスが存在する場合)、userdump.exe <プロセス ID> を実行し、メモリダンプ (user.dump) を採取します。CORBA ネーミングサービスのプロセス ID はあらかじめタスクマネージャで確認します。

#### (b) UNIX の場合

CORBA ネーミングサービスが稼働している場合 (CORBA ネーミングサービスのプロセスが存在する場合)、CORBA ネーミングサービスのプロセス ID を確認して、kill コマンドを実行します。kill コマンドは次の形式で実行してください。なお、kill コマンドを実行するとプロセスが終了するため、再起動する前に kill コマンドを実行することを推奨します。

```
ps -ef | grep java
kill -6 <CORBAネーミングサービスのプロセスID>
```

## 5.4.8 日立固有の JavaVM ログの取得（日立 JavaVM ログファイル）

日立固有の JavaVM ログとは、日立が標準の JavaVM に追加した拡張オプションを使用して取得できるログです。標準の JavaVM よりも、多くのトラブルシュート情報が取得できます。日立固有の JavaVM ログは、次のどれかのオプションを指定した場合に、ログファイルに出力されます。なお、このログファイルを、日立 JavaVM ログファイルといいます。

`-XX:+HitachiOutOfMemoryStackTrace`

なお、このオプションを指定した場合に同時に指定される、

`-XX:+HitachiOutOfMemorySize` および `-XX:+HitachiOutOfMemoryCause` が指定された場合も、日立 JavaVM ログファイルが出力されます。

`-XX:+HitachiVerboseGC`

`-XX:+HitachiJavaClassLibTrace`

このほか、出力内容や出力方法についても、オプションに指定しておく必要があります。JavaVM の資料取得の設定については、マニュアル「Cosminexus システム構築ガイド」を参照してください。

日立 JavaVM ログファイルは、`-XX:HitachiJavaLog: <パスおよびファイル名称> オプション`に指定した出力先に、指定したファイル名称で出力されます。指定を省略した場合は、簡易構築定義ファイルの論理 J2EE サーバ（`j2ee-server`）の `<configuration>` タグ内に、`ejb.server.log.directory` パラメタで指定したディレクトリに、「`javalogxx.log`」という名称で出力されます。xx は 01 から始まる 2 けたの通番です。

## 5.4.9 JavaVM 出力メッセージログの取得（標準出力またはエラーレポートファイル）

ここでは、JavaVM が出力するメッセージログの取得について説明します。

JavaVM でクラッシュが発生した場合、JavaVM はデバッグ情報を標準出力とエラーレポートファイルに出力します。

エラーレポートファイルは、次の場合に出力されます。

JNI の中でシグナルが発生したとき

JavaVM で C ヒープ不足が発生したとき

JavaVM で予期しないシグナルが発生したとき

## 5. トラブルシューティング

JavaVM で Internal Error ( 内部論理エラー ) が発生したとき

ただし、エラーレポートファイルの作成時にシグナルやメモリ不足が発生した場合、このファイルは作成されません。

### ( 1 ) Windows の場合

エラーレポートファイルの出力先および出力ファイル名は次のとおりです。

< 作業ディレクトリ >%ejb%< サーバ名 >%hs\_err\_pid< サーバプロセスのプロセス ID>.log

#### 参考

---

C ヒープが不足した場合、次の順序でメッセージ出力およびダンプ出力が実行されます。必要な情報を取得してください。

1. C ヒープ不足を示すメッセージログが、エラーレポートファイルおよび標準出力に出力されます。
  2. 1. の実行中にメモリ不足が発生した場合、簡易メッセージが標準出力に出力されます。
  3. ワトソンログが起動され、ダンプ出力が実行されます。
- 

### ( 2 ) UNIX の場合

エラーレポートファイルの出力先および出力ファイル名は次のとおりです。

< 作業ディレクトリ >/ejb/< サーバ名 >/hs\_err\_pid< サーバプロセスのプロセス ID>.log

#### 参考

---

C ヒープが不足した場合、次の順序でメッセージ出力および core ダンプの生成が実行されます。必要な情報を取得してください。

1. C ヒープ不足を示すメッセージログが、エラーレポートファイルおよび標準出力に出力されます。
  2. 1. の実行中にメモリ不足が発生した場合、簡易メッセージが標準出力に出力されます。
  3. 簡易メッセージの出力処理中にさらにメモリ不足が発生した場合、メッセージおよびエラーログファイルの出力処理を中止して、core ダンプを生成します。
- 

## 5.4.10 OS の状態情報と OS のログの取得

トラブルシュート情報として必要な OS のログ情報を次に示します。

### ( 1 ) OS の状態情報の取得

トラブルシュート情報として必要な OS の状態情報の取得について、OS ごとに説明します。

## (a) Windows の場合

OS の状態情報は、cjgetsysinfo コマンドを使用して取得できます。-f オプションを指定すると、OS 状態出力ファイルに出力できます。

コマンドは、次の形式で実行します。

```
cjgetsysinfo -f <OS状態出力ファイルパス>
```

このコマンドによって、OS の次のコマンドを実行した場合と同じ情報が取得できます。

```
netstat -e
netstat -s
netstat -a
set
```

なお、cjgetsysinfo コマンドを実行しない場合に、トラブルシューティング情報として取得する必要がある OS の状態情報を次に示します。各情報の取得はあらかじめディレクトリを作成し、そこにファイルを生成します。作成するディレクトリのパスは任意です。

表 5-32 トラブルシューティング情報として必要な OS の状態情報

情報の種類	デフォルトのファイル名
ネットワークの情報	プロトコルの統計情報と現在の TCP/IP ネットワーク接続の情報。次のコマンドで順次取得します。 <pre>netstat -e &gt; netstat_e.txt netstat -s &gt; netstat_s.txt netstat -a &gt; netstat_a.txt</pre>
環境変数	現在設定されている環境変数。次のコマンドで取得します。 <pre>set &gt; set.txt</pre>

## (b) UNIX の場合

OS の状態情報は、cjgetsysinfo コマンドを使用して取得できます。-f オプションを指定すると、OS 状態出力ファイルに出力できます。

コマンドは、次の形式で実行します。

```
cjgetsysinfo -f <OS状態出力ファイルパス>
```

このコマンドによって、OS の次の表に示すコマンドを実行した場合と同じ情報が取得できます。

5. トラブルシューティング

表 5-33 cjgetsysinfo コマンドの実行によって実行される OS のコマンド

AIX の場合	HP-UX の場合	Linux の場合	Solaris の場合
<ul style="list-style-type: none"> <li>• df -k</li> <li>• ps -elf</li> <li>• ps -A -m -o THREAD</li> <li>• vmstat -t 1 1</li> <li>• vmstat -s</li> <li>• lspss -s</li> <li>• netstat -i</li> <li>• netstat -m</li> <li>• netstat -a</li> <li>• iostat</li> <li>• svmon -P</li> <li>• svmon -G</li> <li>• sar -A 1</li> <li>• instfix -i</li> <li>• lslpp -hac</li> <li>• uname -a</li> <li>• env</li> <li>• set</li> <li>• ipcs -a</li> </ul>	<ul style="list-style-type: none"> <li>• df -kt</li> <li>• top -s 1 -d 1</li> <li>• ps -efl</li> <li>• vmstat</li> <li>• vmstat -s</li> <li>• netstat -i</li> <li>• netstat -a</li> <li>• iostat</li> <li>• sar -A 1</li> <li>• swapinfo -a</li> <li>• kmtune 1</li> <li>• kctune 1</li> <li>• swlist</li> <li>• swlist -l product</li> <li>• uname -a</li> <li>• env</li> <li>• set</li> <li>• ipcs -a</li> </ul>	<ul style="list-style-type: none"> <li>• df</li> <li>• ps -eflm</li> <li>• vmstat</li> <li>• netstat -s</li> <li>• netstat -a</li> <li>• iostat 2</li> <li>• top -b -n 1</li> <li>• sysctl -a</li> <li>• sar -A 1 2</li> <li>• rpm -qa</li> <li>• rpm -qai</li> <li>• uname -a</li> <li>• env</li> <li>• set</li> <li>• ipcs</li> <li>• ipcs -t</li> <li>• ipcs -p</li> <li>• ipcs -c</li> <li>• ipcs -u</li> <li>• ipcs -l</li> </ul>	<ul style="list-style-type: none"> <li>• df -k</li> <li>• ps -eflL</li> <li>• vmstat</li> <li>• vmstat -s</li> <li>• netstat -i</li> <li>• netstat -m</li> <li>• netstat -a</li> <li>• iostat</li> <li>• mpstat</li> <li>• swap -l</li> <li>• sar -A 1</li> <li>• sysdef</li> <li>• pkginfo</li> <li>• patchadd -p</li> <li>• uname -a</li> <li>• env</li> <li>• set</li> </ul>

注 1 HP-UX (IPF) の場合に実行されるコマンドです。

注 2 sar コマンドおよび iostat コマンドを実行するためには、Linux に含まれている sysstat パッケージをインストールする必要があります。

なお、cjgetsysinfo コマンドを実行しない場合に、トラブルシュート情報として取得する必要がある OS の状態を示す情報の取得方法 (コマンド) を次に示します。

AIX の場合

```
df -k > df_k`date +"%Y%m%d%H%M%S"`.txt
ps -elf > ps_elf`date +"%Y%m%d%H%M%S"`.txt
ps -A -m -o THREAD > ps_AmoTHREAD`date +"%Y%m%d%H%M%S"`.txt
vmstat -t 1 5 > vmstat`date +"%Y%m%d%H%M%S"`.txt
vmstat -s > vmstat_s`date +"%Y%m%d%H%M%S"`.txt
lspss -s > lspss_s`date +"%Y%m%d%H%M%S"`.txt
netstat -i > netstat_i`date +"%Y%m%d%H%M%S"`.txt
netstat -m > netstat_m`date +"%Y%m%d%H%M%S"`.txt
netstat -a > netstat_a`date +"%Y%m%d%H%M%S"`.txt
iostat 1 5 > iostat`date +"%Y%m%d%H%M%S"`.txt
svmon -P > svmon_P`date +"%Y%m%d%H%M%S"`.txt 1
svmon -G -i 1 5 > svmon_G`date +"%Y%m%d%H%M%S"`.txt 1
sar -A 1 5 > sar_A`date +"%Y%m%d%H%M%S"`.txt 1
/usr/samples/kernel/vmtune > vmtune`date +"%Y%m%d%H%M%S"`.txt
instfix -i > instfix_i`date +"%Y%m%d%H%M%S"`.txt
lslpp -hac > lslpp_hac`date +"%Y%m%d%H%M%S"`.txt
uname -a > uname_a`date +"%Y%m%d%H%M%S"`.txt
env > env`date +"%Y%m%d%H%M%S"`.txt
set > set`date +"%Y%m%d%H%M%S"`.txt
ipcs -a > ipcs_a`date +"%Y%m%d%H%M%S"`.txt
```



## HP-UX の場合

```
df -kt > df_kt`date +"%y%m%d%H%M%S"`.txt
top -s 1 -d 10 -f top`date +"%y%m%d%H%M%S"`.txt
ps -efl > ps`date +"%y%m%d%H%M%S"`.txt
vmstat 1 5 > vmstat`date +"%y%m%d%H%M%S"`.txt
vmstat -s > vmstat_s`date +"%y%m%d%H%M%S"`.txt
netstat -i > netstat_i`date +"%y%m%d%H%M%S"`.txt
netstat -a > netstat_a`date +"%y%m%d%H%M%S"`.txt
iostat 1 5 > iostat`date +"%y%m%d%H%M%S"`.txt
sar -A 1 5 > sar_A`date +"%y%m%d%H%M%S"`.txt 1
swapinfo -a > swapinfo_a`date +"%y%m%d%H%M%S"`.txt 1
kmtune > kmtune`date +"%y%m%d%H%M%S"`.txt
swlist > swlist`date +"%y%m%d%H%M%S"`.txt
swlist -l product > swlist_l_product`date +"%y%m%d%H%M%S"`.txt
uname -a > uname_a`date +"%y%m%d%H%M%S"`.txt
env > env`date +"%y%m%d%H%M%S"`.txt
set > set`date +"%y%m%d%H%M%S"`.txt
ipcs -a > ipcs_a`date +"%y%m%d%H%M%S"`.txt
```

## Linux の場合

```
df > df`date +"%y%m%d%H%M%S"`.txt
ps -eflm > ps`date +"%y%m%d%H%M%S"`.txt
vmstat 1 5 > vmstat`date +"%y%m%d%H%M%S"`.txt
netstat -s > netstat_s`date +"%y%m%d%H%M%S"`.txt
netstat -a > netstat_a`date +"%y%m%d%H%M%S"`.txt
iostat 1 5 > iostat`date +"%y%m%d%H%M%S"`.txt 2
top n 5 > top`date +"%y%m%d%H%M%S"`.txt
sar -A 1 > sar`date +"%y%m%d%H%M%S"`.txt 2
sysctl -a > sysctl`date +"%y%m%d%H%M%S"`.txt
rpm -qa > rpm_qa`date +"%y%m%d%H%M%S"`.txt
rpm -qai > rpm_qai`date +"%y%m%d%H%M%S"`.txt
uname -a > uname_a`date +"%y%m%d%H%M%S"`.txt
env > env`date +"%y%m%d%H%M%S"`.txt
set > set`date +"%y%m%d%H%M%S"`.txt
ipcs > ipcs`date +"%y%m%d%H%M%S"`.txt
ipcs -t > ipcs_t`date +"%y%m%d%H%M%S"`.txt
ipcs -p > ipcs_p`date +"%y%m%d%H%M%S"`.txt
ipcs -c > ipcs_c`date +"%y%m%d%H%M%S"`.txt
ipcs -u > ipcs_u`date +"%y%m%d%H%M%S"`.txt
ipcs -l > ipcs_l`date +"%y%m%d%H%M%S"`.txt
```

## Solaris の場合

```
df -k > df`date +"%y%m%d%H%M%S"`.txt
ps -eflL > ps_eflL`date +"%y%m%d%H%M%S"`.txt
vmstat 1 5 > vmstat`date +"%y%m%d%H%M%S"`.txt
vmstat -s > vmstat_s`date +"%y%m%d%H%M%S"`.txt
netstat -i > netstat_i`date +"%y%m%d%H%M%S"`.txt
netstat -m > netstat_m`date +"%y%m%d%H%M%S"`.txt
netstat -a > netstat_a`date +"%y%m%d%H%M%S"`.txt
iostat 1 5 > iostat`date +"%y%m%d%H%M%S"`.txt
mpstat 1 5 > mpstat`date +"%y%m%d%H%M%S"`.txt
swap -l > swap_l`date +"%y%m%d%H%M%S"`.txt 1
sar -A 1 5 > sar_A`date +"%y%m%d%H%M%S"`.txt
sysdef > sysdef`date +"%y%m%d%H%M%S"`.txt 1
pkginfo > pkginfo`date +"%y%m%d%H%M%S"`.txt
patchadd -p > patchadd_p`date +"%y%m%d%H%M%S"`.txt 1
uname -a > uname_a`date +"%y%m%d%H%M%S"`.txt
env > env`date +"%y%m%d%H%M%S"`.txt
set > set`date +"%y%m%d%H%M%S"`.txt
```

## 5. トラブルシューティング

### 注 1

コマンドの実行には root 権限が必要です。

### 注 2

sar コマンドおよび iostat コマンドを実行するためには、Linux に含まれている sysstat パッケージをインストールする必要があります。

## (2) OS のログの取得

トラブルシューティング情報として必要な OS のログの取得について、OS ごとに説明します。

### (a) Windows の場合

トラブルシューティング情報として必要な OS のログを次に示します。

表 5-34 トラブルシューティング情報として必要な OS のログ情報

情報の種類	デフォルトのファイル名
イベントログ	イベントビューアを開き、アプリケーションとシステムのログを保存します。
ワトソンログ	次のディレクトリに格納されます。 C:¥Documents and Settings¥All Users¥Documents¥DrWatson¥drwtsn32.log

### 注

「C:」は OS がインストールされているドライブ名です。

なお、ワトソンログの出力先はワトソン博士の設定画面で確認してください。

### (b) UNIX の場合

トラブルシューティング情報として必要な OS のログ ( syslog ) の格納場所を次に示します。

#### AIX の場合

/var/adm/ras 下すべて

#### HP-UX の場合

/var/adm/syslog 下すべて

#### Linux の場合

/var/log 下すべて

#### Solaris の場合

dmesg > dmesg`date +"%y%m%d%H%M%S"`.txt

( dmesg コマンドの実行結果を任意のファイルに格納します )

## 5.4.11 OS の統計情報の取得

OS の統計情報の取得について、OS ごとに説明します。

## (1) Windows の場合

障害発生後のパフォーマンスログを保存しておきます。「パフォーマンス」の操作の詳細については、OS 付属のマニュアルなどを参照してください。

## ポイント

OS の統計情報は、あらかじめ、OS 付属の「パフォーマンス」でパフォーマンスログの取得を開始している場合にだけ取得できます。

J2EE サーバ実行中に次のシステムモニタのログを 60 秒間隔で採取します。具体的な設定方法は、OS 付属のマニュアルなどを確認してください。

表 5-35 システムモニタの設定

パフォーマンス オブジェクト	インスタ ンス	項目名	説明
processor	-	%Processor Time	CPU 使用率 (非アイドル状態のスレッドを除く合計値)
		%Privileged Time	CPU 使用率 (カーネルモード分)
		%User Time	CPU 使用率 (ユーザモード分)
memory	-	Cache Bytes	ファイルシステム キャッシュが現在使用しているバイト数。
		Cache Faults/sec	1 秒間当たりのメモリの別の場所からの取り出し数、ディスクから取り出しの回数。
		Page Faults/sec	1 秒間当たりのページフォールトの数。
		Transition Faults/sec	1 秒間当たりのフォールトの数。

## 5. トラブルシューティング

パフォーマンス オブジェクト	インスタ ンス	項目名	説明
process	_Total	Handle Count	現在オープンしているハンドルの総数。
		Page Faults/sec	ページフォールトの発生率。
		Private Bytes	メモリ使用量 (バイト)。
		Virtual Bytes	仮想メモリ使用量 (バイト)。
		Working Set Bytes	実メモリ使用量 (バイト)。
	cjstartsv	%Processor Time	CPU 使用率 (非アイドル状態のスレッドを除く合計値)。
		%Privileged Time	CPU 使用率 (カーネルモード分)。
		%User Time	CPU 使用率 (ユーザモード分)。
		Page Faults/sec	ページフォールトの発生率。
		Thread Count	スレッド数。
		Private Bytes	メモリ使用量 (バイト)。
		Virtual Bytes	仮想メモリ使用量 (バイト)。
		Working Set Bytes	実メモリ使用量 (バイト)。

(凡例) - : 該当しません。

### (2) UNIX の場合

Cosminexus 起動中に次に示すコマンドを実行して、OS の統計情報を取得します。60 秒間隔で取得することを推奨しますが、ディスク容量に応じて取得間隔を決めてください。なお、取得間隔を長くすると、OS の統計情報の取得による性能劣化を少なくできませんが、OS の統計情報の精度が悪化することがあります。

#### AIX の場合

```
ps -efl
ps -A -m -o THREAD
vmstat -t
vmstat -s
lspcs -s
svmon -P <cjstartsv, cjstartwebのプロセスID> 1
svmon -G 1
sar -A 1 1
```

#### HP-UX の場合

```
ps -efl
top -d 1 -f {ファイル名}
vmstat -t
vmstat -s
swapinfo -a 1
sar -A 1 1
```

## Linux の場合

```
ps -eflm
top n 1
vmstat
sar -A 1 2
```

## Solaris の場合

```
ps -eflL
swap -l 1
vmstat
sar -A 1
```

## 注 1

コマンドの実行には root 権限が必要です。

## 注 2

sar コマンドを実行するためには、Linux に含まれている sysstat パッケージをインストールする必要があります。

## 5.4.12 Cosminexus の定義情報の取得

Cosminexus の定義を取得します。この情報を使用して、障害が発生したときに設定されていた定義内容を確認します。

### Cosminexus Component Container に関する定義情報

次のディレクトリ下に格納されているファイル一式を取得します。

#### Windows の場合

- <Cosminexus のインストールディレクトリ>%CC%\server\usrconf\ejb<サーバ名称>

#### UNIX の場合

- /opt/Cosminexus/CC/server/usrconf/ejb/<サーバ名称>

### Cosminexus DABroker Library に関する定義情報

次のファイルを取得します。

#### Windows の場合

- <Cosminexus DABroker Library 運用ディレクトリ>%conf%\dasys.conf  
(Cosminexus DABroker Library 動作環境定義ファイル)
- <Cosminexus DABroker Library 運用ディレクトリ>%conf%\dabenv (接続先データベース定義ファイル)

#### UNIX の場合

- <Cosminexus DABroker Library 運用ディレクトリ>/conf/dasys.conf  
(Cosminexus DABroker Library 動作環境定義ファイル)

## 5. トラブルシューティング

- <Cosminexus DABroker Library 運用ディレクトリ >/conf/dabenv ( 接続先データベース定義ファイル )

トラブルシューティング用に保存するファイルからは、パスワードなどの公開できない情報を削除することをお勧めします。

### 5.4.13 Cosminexus の作業ディレクトリの内容の取得

システムでトラブルが発生した場合に、保守員による原因究明のため作業ディレクトリを調査に使用することがあります。作業ディレクトリは、簡易構築定義ファイルの論理 J2EE サーバ ( j2ee-server ) の <configuration> タグ内に、ejb.public.directory パラメータで指定したディレクトリです。なお、作業ディレクトリのデフォルトのディレクトリパスは「<Cosminexus のインストールディレクトリ >¥CC¥server¥public」( Windows の場合 )、または「/opt/Cosminexus/CC/server/public」( UNIX の場合 ) です。

設定を確認した上で、トラブルが発生した場合に保存しておいてください。

### 5.4.14 Cosminexus のリソース設定情報の取得

コネクションプールの設定などリソースの設定を確認するために、リソースの設定情報を取得します。設定情報の格納場所は次のとおりです。

- Windows の場合
  - <作業ディレクトリ >¥ejb¥<サーバ名称 >¥import
  - <作業ディレクトリ >¥ejb¥<サーバ名称 >¥rars
- UNIX の場合
  - <作業ディレクトリ >/ejb/<サーバ名称 >/import
  - <作業ディレクトリ >/ejb/<サーバ名称 >/rars

### 5.4.15 Web サーバログの取得

システムで使用している Web サーバ ( Hitachi Web Server または Microsoft IIS ) のログを取得します。

Hitachi Web Server の場合

ログの格納場所を次に示します。

- Windows の場合
  - <Cosminexus のインストールディレクトリ >¥httpsd¥logs
- UNIX の場合
  - /opt/hitachi/httpsd/logs ( デフォルト )

Microsoft IIS の場合

ログの格納場所を次に示します。

C:¥WINNT¥system32¥LogFiles¥W3SVC1 ( C: の部分にはシステムドライブを指定

します)

## 5.4.16 JavaVM のスタックトレース情報の取得

UNIX では、JavaVM が異常終了して core ダンプが出力された場合、異常終了した原因の究明に必要な情報（スタックトレース情報）を、jvratrace コマンドで取得できます。jvratrace コマンドでは、出力された core ダンプから、スタックトレース情報を取得します。jvratrace コマンドは、「/opt/Cosminexus/jdk/jre/bin」にインストールされています。

jvratrace コマンドの実行形式を次に示します。

```
jvratrace coreダンプのファイル名称 coreダンプを生成した実行ファイルの名称
```

JavaVM が異常終了して、「core」というファイル名称で core ダンプが作成されているときにこのコマンドを実行すると、実行結果として、カレントディレクトリ下に「jvratrace.log」というファイルが出力されます。このファイルは保守員に送付してください。

### 実行例

JavaVM が異常終了して core ダンプが作成されたときに出力されるメッセージの例を、次に示します。

```
:
# You can get further information from jvratrace.log file generated
# by using jvratrace command.
# usage: jvratrace core-file-name loadmodule-name [out-file-name]
# [-l(library-name)...]
# Please use jvratrace command as follows and submit a bug report
# to Hitachi with jvratrace.log file:
# [/opt/Cosminexus/jdk/jre/bin/jvratrace core /opt/Cosminexus/CC/server/bin/
cjstasrsv]
#
```

メッセージ内に表示される jvratrace コマンドの文字列を実行します。この例の場合には、「/opt/Cosminexus/jdk/jre/bin/jvratrace core /opt/Cosminexus/CC/server/bin/cjstasrsv」を実行してください。実行結果として、カレントディレクトリに「jvratrace.log」というファイルが出力されます。

なお、OS によっては実際に出力される core ダンプのファイル名が「core. プロセス ID」になる場合があります。その場合は、実際に出力された core ダンプのファイル名を jvratrace の引数に指定してください。

## 5.5 資料の調査

---

この節では、取得した資料に出力される内容について説明します。これらの資料を基に、トラブルの原因を判別してください。

ここでは、次の資料について説明します。

Cosminexus のログ

EJB クライアントアプリケーションのログ

Cosminexus DABroker Library のスプール情報

性能解析トレース

JavaVM のスレッドダンプ

JavaVM のガーベージコレクションログ

日立 JavaVM 固有のログ (日立 JavaVM ログファイル)

JavaVM が出力するメッセージログ (標準出力およびエラーレポートファイル)

OS の状態情報および OS のログ

スタックトレース

なお、「5.3 資料の取得」で取得した資料のうち、OS の統計情報については、ご使用の OS 付属のマニュアルなどを参照してください。また、メモリダンプは、保守員が確認する情報なので、ここでは説明しません。

### ! 注意事項

Windows Vista を使用している場合、次の点に注意してください。

ログや PRF トレースに出力する内容に Unicode の補助文字が含まれている場合、その文字は正しく出力されません。ただし、それ以外の出力内容や、アプリケーションの動作に問題はありません。Unicode の補助文字は、Internet Explorer 7 などのクライアントから送信されたリクエストに含まれることがあります。

### 5.5.1 Cosminexus のログの調査

ここでは、Cosminexus のログの調査方法について説明します。

Cosminexus のログのメッセージログ・ユーザログを調査することでエラーの発生原因を調べることができます。また、プロセス障害などの場合も、処理の進行具合や障害の兆候を確認できます。

なお、Cosminexus のログとして収集する資料のうち、Cosminexus DABroker Library のスプール情報については形式が異なります。Cosminexus DABroker Library のスプー



ル情報については、「5.5.3 Cosminexus DABroker Library のスプール情報の調査」を参照してください。また、Cosminexus のログのうち、EJB クライアントアプリケーションのシステムログを参照する場合の留意事項については、「5.7.1 EJB クライアントアプリケーションのシステムログに関する留意事項」を参照してください。

ここでは、次のログの出力形式と出力項目について説明します。

日立トレース共通ライブラリ形式のログ

インプロセス HTTP サーバのアクセスログ

イベントログ

syslog

監査ログで出力するログ

構築済み実行環境の切り替えで出力するログ

また、Cosminexus のログの種別には、次の三つがあります。

日立トレース共通ライブラリ形式（シングルプロセス）

日立トレース共通ライブラリ形式（マルチプロセス）

独自形式

それぞれの Cosminexus のログの種別を、次の表に示します。

表 5-36 J2EE サーバおよびリダイレクタのログ種別

分類	内容	種別
メッセージログ	稼働ログ	日立トレース共通ライブラリ形式（シングルプロセス）
	ログ稼働ログ	日立トレース共通ライブラリ形式（シングルプロセス）
	リダイレクタのメッセージログ	<ul style="list-style-type: none"> <li>Windows で Hitachi Web Server を使用する場合 日立トレース共通ライブラリ形式（シングルプロセス）</li> <li>Windows で Microsoft IIS 6.0 を使用する場合または UNIX の場合 日立トレース共通ライブラリ形式（マルチプロセス）</li> </ul>
	J2EE リソースアダプタとしてデプロイして使用するリソースアダプタの稼働ログ	日立トレース共通ライブラリ形式（シングルプロセス）
	J2EE アプリケーションに含めて使用するリソースアダプタの稼働ログ	日立トレース共通ライブラリ形式（シングルプロセス）
ユーザログ	Web サブレットログ	日立トレース共通ライブラリ形式（シングルプロセス）

5. トラブルシューティング

分類	内容	種別
	ユーザ出力ログ	日立トレース共通ライブラリ形式 (シングルプロセス)
	ユーザエラーログ	日立トレース共通ライブラリ形式 (シングルプロセス)
	JavaVM の保守情報およびガーベージ コレクションのログ	日立トレース共通ライブラリ形式 (シングルプロセス)
例外ログ	障害発生時の例外情報	日立トレース共通ライブラリ形式 (シングルプロセス)
保守用ログ	保守情報	日立トレース共通ライブラリ形式 (シングルプロセス)
	コンソールメッセージ	日立トレース共通ライブラリ形式 (シングルプロセス)
	EJB コンテナの保守情報	日立トレース共通ライブラリ形式 (シングルプロセス)
	Web コンテナの保守情報	日立トレース共通ライブラリ形式 (シングルプロセス)
	起動プロセス標準出力情報	日立トレース共通ライブラリ形式 (シングルプロセス)
	起動プロセス標準エラー情報	日立トレース共通ライブラリ形式 (シングルプロセス)
	終了プロセス情報	日立トレース共通ライブラリ形式 (シングルプロセス)
	リダイレクタの保守用トレースログ	<ul style="list-style-type: none"> <li>Windows で Hitachi Web Server を使用する 場合 日立トレース共通ライブラリ形式 (シングルプロセス)</li> <li>Windows で Microsoft IIS 6.0 を使用する 場合または UNIX の場合 日立トレース共通ライブラリ形式 (マルチプロセス)</li> </ul>
イベントログ	J2EE サーバの起動，停止または異常 終了を示すログ	日立トレース共通ライブラリ形式 (シングルプロセス)
syslog	J2EE サーバの起動，停止または異常 終了を示すログ	日立トレース共通ライブラリ形式 (シングルプロセス)

表 5-37 サーバ管理コマンドのログ

分類	内容	種別
メッセージログ	稼働ログ	日立トレース共通ライブラリ形式 (マルチプロセス)
	ログ稼働ログ	独自形式
例外ログ	障害発生時の例外情報	日立トレース共通ライブラリ形式 (マルチプロセス)

分類	内容	種別
保守用ログ	保守情報	日立トレース共通ライブラリ形式 (マルチプロセス)
	コンソールメッセージ	日立トレース共通ライブラリ形式 (マルチプロセス)
	サーバ管理コマンドの保守情報	日立トレース共通ライブラリ形式 (マルチプロセス)

表 5-38 リソースアダプタバージョンアップコマンド (cjrupdate) のログ

分類	内容	種別
メッセージログ	稼働ログ	日立トレース共通ライブラリ形式 (マルチプロセス)
例外ログ	障害発生時の例外情報	日立トレース共通ライブラリ形式 (マルチプロセス)
保守用ログ	保守情報	日立トレース共通ライブラリ形式 (マルチプロセス)

表 5-39 インプロセス HTTP サーバのログ

分類	内容	種別
アクセスログ	インプロセス HTTP サーバの処理結果	独自形式
スレッドトレース	スレッドトレースの情報	独自形式
通信トレース	通信トレースの情報	独自形式

表 5-40 移行コマンド (cjenvupdate) のログ

分類	内容	種別
メッセージログ	cjenvupdate コマンドの稼働ログ	日立トレース共通ライブラリ形式 (マルチプロセス)
例外ログ	cjenvupdate コマンドの例外情報	日立トレース共通ライブラリ形式 (マルチプロセス)
保守用ログ	cjenvupdate コマンドの保守情報	日立トレース共通ライブラリ形式 (マルチプロセス)

表 5-41 リソース枯渇監視のログ

監視対象 リソース	種別
メモリ	日立トレース共通ライブラリ形式 (シングルプロセス)
ファイルディסקリプタ	日立トレース共通ライブラリ形式 (シングルプロセス)

5. トラブルシューティング

監視対象リソース	種別
スレッド	日立トレース共通ライブラリ形式（シングルプロセス）
スレッドダンプ	日立トレース共通ライブラリ形式（シングルプロセス）
HTTP リクエスト実行待ちキュー	日立トレース共通ライブラリ形式（シングルプロセス）
HTTP セッション数	日立トレース共通ライブラリ形式（シングルプロセス）
コネクションプール	日立トレース共通ライブラリ形式（シングルプロセス）

表 5-42 運用管理エージェント・運用監視エージェント・Management Server のログ

分類	内容	種別
統合ログ	統合メッセージログ	日立トレース共通ライブラリ形式（マルチプロセス）
	統合トレースログ	日立トレース共通ライブラリ形式（マルチプロセス）
	コマンド保守ログ	日立トレース共通ライブラリ形式（マルチプロセス）
運用管理エージェント	運用管理エージェントの標準エラー出力	日立トレース共通ライブラリ形式（シングルプロセス）
	運用管理エージェントの標準出力	日立トレース共通ライブラリ形式（シングルプロセス）
	運用管理エージェントの標準エラー出力コマンドライン	独自形式
	運用管理エージェントのログ	日立トレース共通ライブラリ形式（シングルプロセス）
	運用管理エージェントの起動・停止コマンドのログ	日立トレース共通ライブラリ形式（シングルプロセス）
	運用管理エージェントの保守ログ	日立トレース共通ライブラリ形式（シングルプロセス）
	コンソールログ	日立トレース共通ライブラリ形式（シングルプロセス）
	運用管理エージェントサービスのログ	日立トレース共通ライブラリ形式（シングルプロセス）
	運用管理エージェントサービスの標準出力	独自形式
	運用管理エージェントサービスの標準エラー出力	独自形式

分類	内容	種別
運用監視エージェント	運用監視エージェントのログ・ トレース J2EE サーバ用システム JP1 イ ベントおよび J2EE サーバ用 ユーザ JP1 イベントのログ Management イベント発行ログ	日立トレース共通ライブラリ形 式（シングルプロセス）
Management Server	Management Server サービスの ログ	日立トレース共通ライブラリ形 式（シングルプロセス）
	Management Server サービスの 標準エラー出力	独自形式
	Management Server サービスの 標準出力	独自形式
	Management Server サービス起 動・停止コマンド	日立トレース共通ライブラリ形 式（シングルプロセス）
	Management Server のログ Cosminexus の運用管理サーバ のシステム JP1 イベントのログ	日立トレース共通ライブラリ形 式（シングルプロセス）
	mngenvsetup コマンドの実行ロ グ	日立トレース共通ライブラリ形 式（シングルプロセス）
	Management Server の保守ログ	日立トレース共通ライブラリ形 式（シングルプロセス）

表 5-43 Cosminexus Performance Tracer のログ

内容	種別
PRF デモンおよび PRF コマンドのログ	独自形式
モジュールトレース	独自形式
構造化例外発生ログ	独自形式
保守情報	独自形式

表 5-44 Cosminexus Component Transaction Monitor のログ

内容	種別
CTM デモンおよび CTM コマンドのログ	独自形式
保守情報	独自形式

表 5-45 Cosminexus DABroker Library のログ

内容	種別
メッセージログ	日立トレース共通ライブラリ形式（マルチプロセ ス）

## 5. トラブルシューティング

内容	種別
拡張データベースアクセストレース	日立トレース共通ライブラリ形式 (マルチプロセス)
エラートレース	日立トレース共通ライブラリ形式 (マルチプロセス)
Exception トレースログ	日立トレース共通ライブラリ形式 (マルチプロセス)
イベントトレース (DABroker for C++ をインストールしている場合)	日立トレース共通ライブラリ形式 (マルチプロセス)
XA トレース	日立トレース共通ライブラリ形式 (マルチプロセス)

表 5-46 監査ログで出力するログ

分類	内容	種別
メッセージログ	監査ログのメッセージログ	日立トレース共通ライブラリ形式 (マルチプロセス)
例外ログ	監査ログの例外情報	日立トレース共通ライブラリ形式 (マルチプロセス)

表 5-47 構築済み実行環境の切り替えで出力するログ

内容	種別
cosmienv コマンドのログ	独自形式

### (1) 日立トレース共通ライブラリ形式のログの出力形式と出力項目

日立トレース共通ライブラリ形式のログの出力形式と出力項目について説明します。

Cosminexus のログは、日立トレース共通ライブラリ形式で出力されます。

#### (a) 出力形式

日立トレース共通ライブラリ形式のログの出力形式を次に示します。

```
番号 日付 時刻 AP名 pid tid メッセージID 種別 メッセージテキスト CRLF
```

#### (b) 出力項目

日立トレース共通ライブラリ形式のログの出力項目を次に示します。

表 5-48 日立トレース共通ライブラリ形式のログの出力項目

項目名	説明
番号	トレースレコードの通番を示す 4 けたの番号が出力されます。
日付	トレースの取得日付が「yyyy/mm/dd」の形式で出力されます。

項目名	説明
時刻	トレースの取得時刻が「hh:mm:ss.sss」の形式で出力されます。
AP 名	プログラムを示す文字列が出力されます。
pid	プロセス ID が出力されます。
tid	スレッド ID が出力されます。
メッセージ ID	メッセージ ID が「XXXXnnnnn-Y」の形式で出力されます。
種別	トレース出力の契機となったイベント種別が出力されます。
メッセージテキスト	メッセージテキストが出力されます。付加情報が付くこともあります。
CRLF	レコードの終端符号 (0x0D, 0x0A) が出力されます。

#### ログファイルの編集表示について

日立トレース共通ライブラリ形式のログを Microsoft Excel で表示して、フィルタリングや並べ替えの機能を利用すると、トラブルの発生要因の調査を効率良く進められるようになります。

Microsoft Excel を利用した日立トレース共通ライブラリ形式のログの表示例を次に示します。

図 5-6 Microsoft Excel を利用した日立トレース共通ライブラリ形式のログの表示例

```

**** Windows 2000 5.0 T2=Asia /Tokyo 2003/07/08 17:59:20.570
yyyy/mm/dd hh:mm:ss.sss pid tid message-id message(LANG=ja)
0006 2004/3/2 17:59:21 HEJB 00EF8CF3 000ECD7E KDJE31000-I Start OTS mode.
0007 2004/3/2 18:00:12 HEJB 00EF8CF3 000ECD7E KDJE30028-I J2EE server
J2EEserv1 started.
0008 2004/3/2 18:01:38 HEJB 00EF8CF3 00C9630A KDJE39051-E Could not find
JSP file /bookseller/BookstoreKanda/Logout.jsp.
0009 2004/3/2 18:33:06 HEJB 00EF8CF3 001415C8 KDJE30031-I Shutting down
J2EE server J2EEserv1.
00 2004/3/2 18:33:06 HEJB 00EF8CF3 001415C8 KDJE30031-I Shutting down
J2EEserv1.

```

1	2	3	4	5	6	7	8
1	**** Windows 2000 5.0				T2=Asia	/Tokyo 2003/07/08 17:59:20.570	
2	yyyy/mm/dd	hh:mm:ss.sss	pid	tid	message-id	message(LANG=ja)	
3	0006 2004/3/2	17:59:21	HEJB	HEJB 00EF8CF3	000ECD7E	KDJE31000-I	Start OTS mode.
4	0007 2004/3/2	18:00:12	HEJB	HEJB 00EF8CF3	000ECD7E	KDJE30028-I	J2EE server J2EEserv1 started.
5	0008 2004/3/2	18:01:38	HEJB	HEJB 00EF8CF3	00C9630A	KDJE39051-E	Could not find JSP file /bookseller/BookstoreKanda/Logout.jsp.
6	0009 2004/3/2	18:33:06	HEJB	HEJB 00EF8CF3	001415C8	KDJE30031-I	Shutting down J2EE server J2EEserv1.
7	00 2004/3/2	18:33:06	HEJB	HEJB 00EF8CF3	001415C8	KDJE30031-I	J2EE server J2EEserv1 shut down.

#### 参考

- 区切るデータの形式として、「スペースによって右または左にそろえられた固定長フィールドデータ」を選択してください。
- [データのプレビュー] ボックスで不要な矢印を削除してください。
- 各列の [表示形式] に [文字列] を指定してください。

## (2) 日立トレース共通ライブラリ形式のログを参照する場合の注意

日立トレース共通ライブラリ形式のログを使用する場合の注意事項について説明します。

### (a) マルチプロセス・シングルプロセス共通の注意事項

マルチプロセス・シングルプロセス共通の注意事項について説明します。

- 出力されたログファイルを編集しないでください。
- テキストエディタの機能などを使用して、ファイルロックをしないでください。
- 出力されたログファイルにアクセス権限を手動で設定する場合は、適切なアクセス権を与えてください。
- 出力されたログファイルの更新時刻を変更しないでください。
- トレース出力中に、ログファイルを削除したり、ファイル名を変更したりしないでください。ログファイルの削除、またはファイル名の変更は、すべてのトレース出力プロセスを停止してから実行してください。

### (b) マルチプロセスの日立トレース共通ライブラリ参照時の注意事項

マルチプロセスに対応した日立トレース共通ライブラリ形式のログファイルを参照する場合の注意事項を次に示します。

- メッセージ末尾の改行コードは、使用している OS に関係なく CRLF です。
- ログファイルのファイルサイズ、面数、およびモードオプションの変更を有効にするために、ログファイルを削除する必要がある場合があります。ログファイルの削除は、すべてのトレース出力プロセスを停止してから実行してください。
- 出力したログファイル内で有効なデータは、ファイルの先頭から最初の EOF までです。ログファイルを参照するときには、最初の EOF 以降のデータは無視してください。EOF 以降のデータは、ラップアラウンド以前のログファイルの無効なデータです。有効なログファイルのデータの末尾は、次に示すデータになります。

```
EOF CRLF CRLF CRLF CRLF-----< End of Data >-----CRLF CRLF
```

EOF はトレースデータの終端を表す文字 (0x1A) です。CRLF は、改行 (0x0D, 0x0A) を表します。

- トレースが出力されても、ログファイルの更新時刻が更新されない場合があります。そのため、ファイルの更新時刻を基にトレースが出力されたかどうかを判断することはできません。

## (3) インプロセス HTTP サーバのアクセスログの出力形式と出力項目

インプロセス HTTP サーバのアクセスログの出力形式と出力項目について説明します。

アクセスログには、インプロセス HTTP サーバのリクエストの処理結果が出力されます。また、アクセスログの出力形式には、デフォルトで出力されるデフォルトフォーマットと拡張フォーマットがあります。インプロセス HTTP サーバのアクセスログのカスタマ



イズについては、マニュアル「Cosminexus システム構築ガイド」を参照してください。

出力形式は次のとおりです。

デフォルトフォーマット

```
クライアントのホスト名 - 認証ユーザ名 アクセス時刻 リクエストライン ステータスコード 送信バイト数
```

拡張フォーマット

```
クライアントのホスト名 - 認証ユーザ名 アクセス時刻 リクエストライン ステータスコード 送信バイト数 Refererヘッダの内容 User-Agentヘッダの内容
```

それぞれの項目の内容を次の表に示します

表 5-49 アクセスログの出力項目

項番	項目名	説明
1	クライアントのホスト名	クライアントのホスト名が表示されます。ホスト名が設定されていない場合は IP アドレスが表示されます。
2	認証ユーザ名	ベーシックユーザ認証名、またはフォーム認証ユーザ名が表示されます。フォームユーザ認証名および、フォームユーザ認証名が設定されていない場合はハイフン (-) が表示されます。
3	アクセス時刻	インプロセス HTTP サーバがクライアントのリクエストに回答した時間が表示されます。
4	リクエストライン	リクエストラインが表示されます。
5	ステータスコード	最終ステータスコードが表示されます。ただし内部でリダイレクトされた値は出力しません。
6	送信バイト数	HTTP ヘッダを除く送信バイト数が表示されます。0 バイトのときはハイフン (-) が表示されます。
7	Referer ヘッダの内容	遷移する前の Web ページの URL が表示されます。
8	User-Agent ヘッダの内容	Web クライアント (ブラウザ名やブラウザのバージョンなど) の情報が含まれたヘッダが表示されます。

なお、アクセスログには、クライアントの IP アドレス、J2EE サーバの IP アドレス、リクエストメソッド、クライアントからのリクエスト処理に掛かった時間などの項目を任意で追加できます。インプロセス HTTP サーバのアクセスログのカスタマイズについては、マニュアル「Cosminexus システム構築ガイド」を参照してください。

アクセスログの出力例を次に示します。

デフォルトフォーマットの場合

## 5. トラブルシューティング

```
10.20.30.40 - user1 [18/Jan/2005:13:06:10 +0900] "GET / HTTP/1.0" 200 38
10.20.30.40 - user1 [18/Jan/2005:13:06:25 +0900] "GET /demo/ HTTP/1.0" 500 684
```

### 拡張フォーマットの場合

```
10.20.30.40 - user1 [18/Jan/2005:13:06:10 +0900] "GET / HTTP/1.0" 200 38 "-"
"Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)"
10.20.30.40 - user1 [18/Jan/2005:13:06:25 +0900] "GET /demo/ HTTP/1.0" 500 684
"- "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)"
```

### (4) イベントログの出力形式と出力項目 (Windows の場合)

イベントログの出力形式と出力項目について説明します。

イベントログは、J2EE サーバの起動、停止および異常終了時に出力されるログです。出力形式は次のとおりです。

ID文字列 [pid]: メッセージテキスト

出力項目を次に示します。

表 5-50 イベントログの出力項目

項目名	説明
ID 文字列	アプリケーションを示す文字列として、「HEJB」が出力されます。
pid	プロセス ID が出力されます。
メッセージテキスト	メッセージテキストが出力されます。

### (5) syslog の出力形式と出力項目 (UNIX の場合)

syslog の出力形式と出力項目について説明します。

syslog は、J2EE サーバの起動、停止および異常終了時に出力されるログです。出力形式は次のとおりです。

日付 時刻 ホスト名 ID文字列 [pid]:メッセージテキスト

出力項目を次に示します。

表 5-51 syslog の出力項目

項目名	説明
日付	メッセージを出力した日付です。

項目名	説明
時刻	メッセージを出力した時刻です。
ホスト名	ホスト名を示す文字列が出力されます。
pid	プロセス ID が出力されます。
ID 文字列	アプリケーションを示す文字列として、「HEJB」が出力されます。
メッセージテキスト	メッセージテキストが出力されます。

### (6) 構築済み実行環境の切り替えで出力するログの出力形式と出力項目 (UNIX の場合)

構築済み実行環境を `cosmienv` コマンドで切り替えた場合に出力される、ログファイルの出力形式および出力項目について説明します。

`cosmienv` コマンド実行時に出力されるログファイルには、`cosmienv_s[2].log`、`cosmienv_d[2].log`、および `cosmienv_r[2].log` があります。これらのファイルの出力形式および出力項目はすべて同じです。

`cosmienv` コマンド実行時に出力されるログファイルの出力形式は次のとおりです。

```

YYYY/MM/DD HH:MM:SS 製品の形名 製品名 VVRRSS START
YYYY/MM/DD HH:MM:SS 構成ソフトウェアの形名 START 起動時のコマンドと引数
[YYYY/MM/DD HH:MM:SS 構成ソフトウェアの保守情報]
YYYY/MM/DD HH:MM:SS 構成ソフトウェアの形名 END 構成ソフトウェアの戻り値
:
YYYY/MM/DD HH:MM:SS 製品の形名 製品名 VVRRSS END cosmienvコマンドの戻り値

```

`cosmienv` コマンドのログの出力項目を次に示します。

表 5-52 `cosmienv` コマンドのログの出力項目

項目名	説明
YYYY/MM/DD	<code>cosmienv</code> コマンドを実行した日付が出力されます。
HH:MM:SS	<code>cosmienv</code> コマンドを実行した時刻が出力されます。
製品の形名	退避、回復、または削除の対象とした実行環境の形名が出力されま す。
製品名	退避、回復、または削除の対象とした実行環境の製品名が出力され ます。
VVRRSS	退避、回復、または削除の対象とした実行環境の製品のバージョン 情報が出力されます。
構成ソフトウェアの形名	退避、回復、または削除の対象とした実行環境の製品に含まれる構 成ソフトウェアの形名が出力されます。 構成ソフトウェアの情報は、製品に含まれる構成ソフトウェアの数 分出力されます。

## 5. トラブルシューティング

項目名	説明
起動時のコマンドと引数	内部処理（構成ソフトウェアの処理）を起動したコマンドと、引数が出力されます。
構成ソフトウェアの保守情報	内部処理（構成ソフトウェアの処理）実行時の保守情報が出力されます。
構成ソフトウェアの戻り値	内部処理（構成ソフトウェアの処理）の実行結果が出力されます。
cosmienv コマンドの戻り値	cosmienv コマンドの実行結果が出力されます。

注 構成ソフトウェアおよび cosmienv コマンドの戻り値については、マニュアル「Cosminexus リファレンス コマンド編」を参照してください。

退避機能を実行した場合の出力例を次に示します。この例では、退避機能実行時に、構成ソフトウェアの処理でディスク容量不足を検出しています。

```

2007/05/20 14:37:22 P-1M43-7K71 uCosminexus Application Server Enterprise 07-50
START
2007/05/20 14:37:22 P-1M3Z-7151 START 2007/05/20 14:37:22 P-1M43-7K71
uCosminexus Application Server Enterprise 07-50 START
2007/05/20 14:37:22 P-1M3Z-7151 START cosmippenv -s -P/home/CosmiBackup/
UCAPSE001 -ZCosmiInside
2007/05/20 14:37:24 P-1M3Z-7151 END 0
2007/05/20 14:37:47 P-1M41-E171 START cosmippenv -s -P/home/CosmiBackup/
UCAPSE001 -ZCosmiInside
2007/05/20 14:37:52 httpsd tar: tape write error: No space left on device
2007/05/20 14:37:53 P-1M41-E171 END 128
2007/05/20 14:37:54 P-1M43-7K71 uCosminexus Application Server Enterprise 07-50
END 143

```

### 5.5.2 EJB クライアントアプリケーションのログの調査

ここでは、EJB クライアントアプリケーションのログの種別について説明します。ログの種別の詳細、それぞれのログの出力形式と出力項目、および参照する場合の注意事項については、「5.5.1 Cosminexus のログの調査」を参照してください。

EJB クライアントアプリケーションのログの種別を次に示します。

表 5-53 EJB クライアントアプリケーションのログの種別

種類	内容	種別
メッセージログ	稼働ログ	日立トレース共通ライブラリ形式 (マルチプロセス)
	cjclstartap コマンドの稼働ログ	日立トレース共通ライブラリ形式 (マルチプロセス)
	cjcldellog コマンドの稼働ログ	日立トレース共通ライブラリ形式 (マルチプロセス)
ユーザログ	ユーザ出力ログ	日立トレース共通ライブラリ形式 (マルチプロセス)

種類	内容	種別
	ユーザエラーログ	日立トレース共通ライブラリ形式 (マルチプロセス)
Java ログ	JavaVM の保守情報, ガーベージコレクションのログ	独自形式
例外ログ	障害発生時の例外情報	日立トレース共通ライブラリ形式 (マルチプロセス)

### 5.5.3 Cosminexus DABroker Library のスプール情報の調査

Cosminexus DABroker Library のスプール情報の調査方法について説明します。

Cosminexus DABroker Library のスプール情報には、次の種類があります。これらの情報は、<Cosminexus DABroker Library の運用ディレクトリ>\\$spool ( Windows の場合 ), または <Cosminexus DABroker Library の運用ディレクトリ>/spool ( UNIX の場合 ) に出力されています。

表 5-54 Cosminexus DABroker Library のスプール情報

スプール情報の種類	対応するファイル
メッセージログ	dabsv.log dabsv.logold
拡張データベースアクセス トレース	db_access ディレクトリ下のファイル
エラートレース	dabevttcerr1.txt ( ログ出力用ファイル ) dabevttcerr2.txt ( バックアップ用ファイル ) <sup>1</sup>
Exception トレースログ	J2EE サーバ名称 +MTrc.log および J2EE サーバ名称 +MTrcOLD.log
イベントトレース <sup>2</sup>	dabevttcXXX1 および dabevttcXXX2
XA トレース	db_xainfo.log

注 1 これらのほか、ワークファイルが、dabevttcerr1.txt.lock という名称で作成されます。

注 2 DABroker for C++ をインストールしている場合に出力されます。

それぞれの情報について説明します。

また、このほか、Cosminexus DABroker Library のログとしては、次の JDBC ドライバのトレースが取得できます。これらのトレースは、アプリケーション開発時に設定されている場合だけ出力されます。

JDBC ドライバのトレース ( JDBC インタフェースメソッドトレース )

JDBC ドライバのトレース ( Cosminexus DABroker Library とのイベントトレース )

## 5. トラブルシューティング

### (1) メッセージログ

Cosminexus DABroker Library のメッセージが出力されたファイルです。

### (2) 拡張データベースアクセストレース

データベースへの接続から切断までのアクセス情報を出力したトレースです。データベースへの接続ごとに出力されます。

拡張データベースアクセストレースは、次のように利用できます。

データベースのアクセス中に発生したトラブルの迅速な切り分け

24 時間稼働に対応する J2EE アプリケーションの稼働状態でのトラブル調査

マルチスレッド環境で J2EE アプリケーションが動作している場合のスレッドごとのデータベースアクセス履歴取得

コネクションプーリングなどの機能を使用している場合のデータベースへの接続ごとのデータベースアクセス履歴取得

トラブルが再現調査できない場合の常時トレース取得

なお、出力するイベントの種類や、トレースファイルの最大容量などは、Cosminexus DABroker Library の [ 環境設定ユティリティ ], または Cosminexus DABroker Library 動作環境定義ファイルで定義しておきます。Cosminexus DABroker Library の動作環境の設定については、マニュアル「Cosminexus 簡易構築・運用ガイド」を参照してください。Cosminexus DABroker Library 動作環境定義ファイルの詳細については、マニュアル「Cosminexus リファレンス 定義編」を参照してください。

#### (a) 出力形式

拡張データベースアクセストレースのファイル出力形式を次に示します。斜体は可変の情報です。拡張データベースアクセストレースは、次の 2 種類の情報で構成されます。

ヘッダ情報

個々のトレース情報

個々のトレース情報は、出力項目を示すタイトル行の下に出力されます。

- Windows の場合

```

*-----*
*- DataBase Access Information (DRV)          -*
*- Date   YYYY/MM/DD hh:mm:ss.nnnnnn        -*
*-----*
DABroker Connect ID   : AAAAAA(BBBBBB)
Process ID            : CCCCCC
UserID               : DDDDD
Client Name          : EEEEE
(DB別出力情報)
Lang Mode            : GGGGG

THREAD-ID  CID  EVT  START-TIME  END-TIME  RETCODE  BLOCKCNT  (Windows
QueryPerformance Counter) HiRDB_CONNECTION_ID
(トレース情報)
(SQL)SS...SS
(トレース情報)

```

• UNIX の場合

```

*-----*
*- DataBase Access Information (DRV)          -*
*- Date   YYYY/MM/DD hh:mm:ss.nnnnnn        -*
*-----*
DABroker Connect ID   : AAAAAA(BBBBBB)
Process ID            : CCCCCC
UserID               : DDDDD
Client Name          : EEEEE
(DB別出力情報)
Lang Mode            : GGGGG

THREAD-ID  CID  EVT  START-TIME  END-TIME  RETCODE
BLOCKCNT  HiRDB_CONNECTION_ID
(トレース情報)
(SQL)SS・S
(トレース情報)

```

(b) 出力項目

拡張データベースアクセストレースの出力項目について、次の表に示します。斜体は可変の情報です。

表 5-55 拡張データベースアクセストレースの出力項目

項目	内容
ヘッダ情報	
<i>DRV</i>	Cosminexus DABroker Library のデータベースアクセスドライバの種別に従い、次の識別子が出力されます。 <ul style="list-style-type: none"> <li>• HiRDB 接続の場合：HiRDB Driver</li> <li>• Oracle8i 接続の場合：ORACLE8i Driver (Oracle Call Interface 8i を使用)</li> <li>• Oracle7 接続の場合：ORACLE7 Driver (Oracle Call Interface 7 を使用)</li> </ul>

## 5. トラブルシューティング

項目	内容
Date: <i>YYYY/MM/DD hh:mm:ss.nnnnnn</i>	ファイル作成日時 (YYYY/MM/DD hh:mm:ss.nnnnnn) が出力されます。 <ul style="list-style-type: none"> <li>• YYYY: 西暦年, MM: 月, DD: 日</li> <li>• hh: 時, mm: 分, ss: 秒, nnnnnn: マイクロ秒</li> </ul>
DABroker Connect ID: <i>AAAAA(BBBBB)</i>	データベース接続識別子が出力されます。
Process ID: <i>CCCCC</i>	実行プロセス ID が出力されます。
UserID: <i>DDDDD</i>	接続ユーザ ID が出力されます。 Cosminexus DABroker Library で ORACLE の OS 認証機能を使用した場合は、「OS authority」となります。
Client Name: <i>EEEEE</i>	J2EE アプリケーション名称が出力されます。名称が設定されていない場合は空白になります。
<i>(DB 別出力情報)</i>	接続データベースによって異なる情報が出力されます。
HiRDB の場合	<ul style="list-style-type: none"> <li>• PDHOST: HiRDB 接続時の PDHOST の設定内容です。未設定の場合は、「Not Indicator」が出力されます。</li> <li>• PDNAMEPORT: HiRDB 接続時の PDNAMEPORT の設定内容です。未設定の場合は、「Not Indicator」が出力されます。</li> </ul>
Oracle の場合	<ul style="list-style-type: none"> <li>• SQLNET: Oracle リスナー名が出力されます。未設定の場合は、「Not Indicator」が出力されます。</li> </ul>
Lang Mode: <i>GGGGG</i>	実行言語モードが出力されます。
個々のトレース情報	
THREAD-ID	実行スレッド ID が出力されます。
CID	カーソル番号が出力されます。
EVT	Cosminexus DABroker Library がデータベースに要求した処理内容を示す文字列が出力されます。データベースアクセストレースおよび拡張データベースアクセストレースの内容が含まれます。 Cosminexus DABroker Library の [ 環境設定ユティリティ ] または Cosminexus DABroker Library 動作環境定義ファイルで設定された内容のイベントが出力されます。なお、すべてのイベントを出力しない設定をしている場合は、ヘッダだけが出力されます。なお、何か一つでもイベントを出力する設定をしている場合は、GET_DIAG イベントは必ず出力されます。



項目	内容
	<p>出力されるイベントを次に示します。</p> <ul style="list-style-type: none"> <li>• ATTACH：サーバとのアクセスパス作成要求を実行します。</li> <li>• ATTRGET：ハンドル属性の取得要求を実行します。</li> <li>• ATTRSET：ハンドル属性の設定要求を実行します。</li> <li>• AUTOCOMOFF：AUTO COMMIT モードの解除要求を実行します。</li> <li>• AUTOCOMON：AUTO COMMIT モードの設定要求を実行します。</li> <li>• BIND：パラメタのバインド要求を実行します。</li> <li>• CHARFORM：キャラクタセットフォームの取得要求を実行します。</li> <li>• CHARID：キャラクタセット ID の取得要求を実行します。</li> <li>• DEFINE：パラメタの定義要求を実行します。</li> <li>• DESALLOC：記述子の領域確保を実行します。</li> <li>• DESC_OUT：前処理した SQL の出力情報を返します。</li> <li>• DESC_IN：前処理した SQL の入力情報を返します。</li> <li>• DESCANY：オブジェクトの記述要求を実行します。</li> <li>• DESFREE：記述子の領域解放を実行します。</li> <li>• DETACH：データベースアクセスパス削除要求を実行します。</li> <li>• ENVCREATE：環境作成要求を実行します。</li> <li>• ERRGET：エラー情報取得要求を実行します。</li> <li>• EXECDIRECT：SQL の前処理後の実行処理です。</li> <li>• GET_DIAG：エラー詳細情報の取得処理を実行します。</li> <li>• GETINFO：接続後の動作情報の取得要求を実行します。</li> <li>• GETPIECE：ピース情報の取得要求を実行します。</li> <li>• HDLALLOC：ハンドルの領域確保を実行します。</li> <li>• HDLFREE：ハンドルの領域解放を実行します。</li> <li>• LOBCLOSE：BFILE データのクローズ要求を実行します。</li> <li>• LOBLEN：LOB データの長さ取得要求を実行します。</li> <li>• LOBOPEN：BFILE データのオープン要求を実行します。</li> <li>• LOBREAD：LOB データの読み込み要求を実行します。</li> <li>• LOBWRITE：LOB データの書き込み要求を実行します。</li> <li>• PARAMGET：パラメタ情報の取得要求を実行します。</li> <li>• SESEBEGIN：セッション開始要求を実行します。</li> <li>• SESEND：セッション終了要求を実行します。</li> <li>• SETINFO：接続後の動作情報の設定要求を実行します。</li> <li>• SETPIECE：ピース情報の設定要求を実行します。</li> </ul>
START-TIME	<p>アクセス開始日時 (YYYY/MM/DD hh:mm:ss.nnnnnn) が出力されます。</p> <ul style="list-style-type: none"> <li>• YYYY：西暦年，MM：月，DD：日</li> <li>• hh：時，mm：分，ss：秒，nnnnnn：マイクロ秒</li> </ul>
END-TIME	<p>アクセス終了日時 (YYYY/MM/DD hh:mm:ss.nnnnnn) が出力されます。</p> <ul style="list-style-type: none"> <li>• YYYY：西暦年，MM：月，DD：日</li> <li>• hh：時，mm：分，ss：秒，nnnnnn：マイクロ秒</li> </ul>
RETCODE	データベースから返されるリターンコードが出力されます。
BLOCKCNT	FETCH 時の行取得件数が出力されます。
(Windows QueryPerformance Counter)	Windows の場合に、高分解パフォーマンス情報が表示されます。

5. トラブルシューティング

項目	内容
HiRDB_CONNECTION_ID	<p>コネクション ID (aaaa:bbbb:cccc) が出力されます。</p> <ul style="list-style-type: none"> <li>• aaaa : フロントエンドサーバ名 (HiRDB/Parallel Server 使用時) またはシングルサーバ名 (HiRDB/Single Server 使用時)</li> <li>• bbbb : aaaa で示すサーバの接続通番</li> <li>• cccc : aaaa で示すサーバのプロセス ID。</li> </ul> <p>コネクション ID は、HiRDB クライアントのバージョンが 07-01 以降で、かつサーバが HiRDB である場合だけ出力されます。サーバが HiRDB で、かつ HiRDB クライアントのバージョンが 07-01 よりも前の場合は、ヘッダ (HiRDB_CONNECTION_ID) だけが出力され、コネクション ID は出力されません。また、サーバが XDM/RD E2 の場合は、ヘッダ (HiRDB_CONNECTION_ID) およびコネクション ID は出力されません。なお、この情報は、CONNECT、COMMIT、ROLLBACK、PREPARE、EXECUTE、DIRECT を実行したとき、最新の情報に更新されます。</p>
(SQL)SS...SS	<p>クライアントから要求のあった SQL の内容が出力されます。この項目は、必要に応じて出力されます。</p>

注

Cosminexus DABroker Library から Oracle8i、Oracle9i および Oracle10g の機能を使用する場合に出力されます。

(c) ファイル名称規則

拡張データベースアクセストレースのファイルは、次のディレクトリに作成されます。

- Windows の場合  
`<Cosminexus DABroker Library 運用ディレクトリ >%spool%db_access`
- UNIX の場合  
`<Cosminexus DABroker Library 運用ディレクトリ >/spool/db_access`

拡張データベースアクセストレースのファイル名称は、次の規則に従います。*斜体は可変の情報です。*

```
spdb_XXXXXXXX_YYYYYYY_HHHHHHHHHHHHHHHH_<接続データベース名称>ZZZZZ.IIIIII
```

ファイル名称中の文字列について次に示します。

表 5-56 拡張データベーストレースのファイル名称の規則

項目	意味
XXXXXXXX	実行プロセス ID が設定されます。
YYYYYYY	データベース接続識別子が設定されます。
HHHHHHHHH HHHHHHH_	実行プロセスごとにユニークな接続識別子が設定されます。

項目	意味
<接続データベース名称>	接続データベース名称です。指定したデータベース種別によって、次のどれかが設定されます。 <ul style="list-style-type: none"> <li>• HiRDB の場合：HiRDB</li> <li>• Oracle8i 接続の場合：ORACLE8i</li> <li>• Oracle7 の場合：ORACLE7</li> </ul>
ZZZZZ	識別順序番号が設定されます。 同じ実行プロセス ID で、かつ同じデータベース接続識別子のファイルがある場合に、00000 から 32767 までの番号を付加します。
IIIII	ファイル拡張子が設定されます。 <ul style="list-style-type: none"> <li>• log：最新のトレースファイル</li> <li>• logold：1 世代前のトレースファイル</li> </ul>

### (3) エラートレース

エラートレースは、ログファイルに出力される Cosminexus DABroker Library のエラー情報です。

エラートレースとして出力される情報を次に示します。

エラー事象を示す情報

エラーメッセージなどが出力されます。

データベース接続情報

データベース接続に関連する情報が出力されます。

履歴情報

エラーが発生する前に要求を受けた情報、前回コミットまたはロールバックが実行された時間、SQL 文の実行要求処理中の実行中カーソルでの要求ごとの履歴情報などが出力されます。

#### (a) エラー事象を示す情報

エラー事象を示す情報の出力形式および出力項目について説明します。

出力形式

エラー事象を示す情報のファイル出力形式を次に示します。斜体は可変の情報です。

```
YYYY/MM/DD hh:mm:ss.sss TID=AAAA SS..SS
```

エラー事象を示す情報の出力例を次に示します。

```
2004/05/07 13:30:09.555 TID=4040 KFPA11561-E Specified authorization
identifier has no connect privilege
```

出力項目

エラー事象を示す情報の出力項目について、次の表に示します。

5. トラブルシューティング

表 5-57 エラートレースのエラー事象を示す情報の出力項目

項目	意味
<i>YYYY/MM/DD hh:mm:ss.sss</i>	エラー発生日時が出力されます。 <i>YYYY</i> : 西暦年, <i>MM</i> : 月, <i>DD</i> : 日 <i>hh</i> : 時, <i>mm</i> : 分, <i>ss</i> : 秒, <i>sss</i> : ミリ秒
<i>AAAA</i>	エラー発生スレッド ID (16 進数値) が出力されます。
<i>SS...SS</i>	エラーメッセージが出力されます。

(b) データベース接続情報

データベース接続情報の出力形式および出力項目について説明します。

出力形式

エラートレースのデータベース接続情報のファイル出力形式を次に示します。斜体は可変の情報です。

```
YYYY/MM/DD hh:mm:ss.sss PID=BBBB InitID=CCCC ConID=DDDD
CName=EEEE DBID=FFFF CID=GGGG(HHHH,I III)
```

エラートレースのデータベース接続情報の出力例を次に示します。

```
2004/05/07 13:30:21.232 PID=1752 InitID=701 ConID=10701 CName=DABLIBAP DBID=1
CID=1(0,0)
```

出力項目

エラートレースのデータベース接続情報のファイル出力項目について、次の表に示します。

表 5-58 エラートレースのデータベース接続情報の出力項目

項目	意味
<i>YYYY/MM/DD hh:mm:ss.sss</i>	データベース接続日時が出力されます。 <i>YYYY</i> : 西暦年, <i>MM</i> : 月, <i>DD</i> : 日 <i>hh</i> : 時, <i>mm</i> : 分, <i>ss</i> : 秒, <i>sss</i> : ミリ秒
<i>BBBB</i>	プロセス ID (10 進数値) が出力されます。
<i>CCCC</i>	DABroker Library 接続 ID (16 進数値) が出力されます。
<i>DDDD</i>	データベース接続識別子 (16 進数値) が出力されます。
<i>EEEE</i>	J2EE アプリケーションの名称が出力されます。ただし、名称なしの場合は空白となります。
<i>FFFF</i>	データベース ID (10 進数値) が出力されます。 1: HiRDB 7: Oracle8i 8: Oracle9i
<i>GGGG</i>	実行中カーソル ID (10 進数値) が出力されます。

項目	意味
<i>HHHH</i>	保守用情報が出力されます。
<i>III</i>	保守用情報が出力されます。

## (c) 履歴情報

履歴情報の出力形式および出力項目について説明します。

## 出力形式

エラートレースの履歴情報のファイル出力形式を次に示します。

ヘッダ「START-TIME EXEC-TIME TID COUNT REQ-CODE DATA」の下に、次に示す形式でトレース情報が出力されます。*斜体*は可変の情報です。

```
YYYY/MM/DD hh:mm:ss.sss mmmm:ss.sss JJJJJJJJ KKKKKKKK LLLL MMMM
NNNNNNNN OOOOOOOO PP..PP
```

エラートレースの履歴情報の出力例を以下に示します。

START-TIME	EXEC-TIME	TID	COUNT	REQ-CODE	DATA
2004/05/07 15:37:16.733	0:00.030	1712	7	1022 NOW	00000000 00000001
2004/05/07 15:36:18.369	0:00.000	1712	6	1022 OLD	00000001 00000002
2004/05/07 15:35:59.191	0:00.281	1712	2	1021 PRE	00000000 00000002
INSERT INTO "HS					
2004/05/07 15:36:18.369	0:00.000	1712	6	1022 EXE	00000001 00000002

## 出力項目

エラートレースの履歴情報のファイル出力項目について、次の表に示します。

表 5-59 エラートレースの履歴情報の出力項目

項目	意味
<i>YYYY/MM/DD hh:mm:ss.sss</i>	要求開始日時が出力されます。 <i>YYYY</i> : 西暦年, <i>MM</i> : 月, <i>DD</i> : 日 <i>hh</i> : 時, <i>mm</i> : 分, <i>ss</i> : 秒, <i>sss</i> : ミリ秒
<i>mmm:ss.sss</i>	要求実行時間が出力されます。 <i>mmm</i> : 分, <i>ss</i> : 秒, <i>sss</i> : ミリ秒
<i>JJJJJJJ</i>	要求元スレッド ID (16 進数値) が出力されます。
<i>KKKKKKKK</i>	コネクト要求からの累計要求カウント数 (16 進数値) が出力されます。FET の場合は、SQL 文実行時の FET の要求回数 0xFFFFFFFF を超えた場合は、0 から循環します。
<i>LLLL</i>	要求コード (10 進数値) 要求コードの詳細は、「表 5-60 要求コード・応答コード一覧」を参照してください。

5. トラブルシューティング

項目	意味
<i>MMMM</i>	大きくわけて次の 3 種類に分類されます。 <ul style="list-style-type: none"> <li>• 共通履歴情報            NOW：実行中の情報            OLD：一つ前の要求の情報</li> <li>• SQL 文の実行要求時に指定されたカーソルの履歴情報            SPRE：SQL PREPARE 処理            EXE：SQL EXECUTE 処理            FET1：SQL FETCH 処理初回            FET：SQL FETCH 処理            CLS：SQL CLOSE 処理</li> <li>• コミット/ロールバック情報            CMT：コミット処理            RBK：ロールバック処理</li> </ul> 詳細は、「表 5-61 各履歴種別による DATA 内容」を参照してください。
<i>NNNNNNNN</i>	情報 1 (16 進数値) 詳細は、「表 5-61 各履歴種別による DATA 内容」を参照してください。
<i>OOOOOOOO</i>	情報 2 (16 進数値) 詳細は、「表 5-61 各履歴種別による DATA 内容」を参照してください。
<i>PP..PP</i>	詳細情報 (任意文字列) 詳細は、「表 5-61 各履歴種別による DATA 内容」を参照してください。

データベース接続情報に出力される、要求コードおよび応答コードの一覧を次の表に示します。

表 5-60 要求コード・応答コード一覧

要求コード・応答コード (LLLL)	種別
1001	コネクション解放
1002	データベース接続
1003	データベース切り離し
1004	Commit
1005	Rollback
1006	表一覧取得準備
1007	表一覧取得
1008	表一覧取得後処理
1009	列一覧取得準備
1010	列一覧取得
1011	列一覧取得後処理

要求コード・応答コード (LLLL)	種別
1012	索引一覧取得準備
1013	索引一覧取得
1014	索引一覧取得後処理
1015	表権限一覧取得準備
1016	表権限一覧取得
1017	表権限一覧取得後処理
1018	列権限一覧取得準備
1019	列権限一覧取得
1020	列権限一覧取得後処理
1021	SQL 実行準備
1022	SQL 実行
1023	SQL 実行結果取得
1024	SQL 実行後処理
1035	SQL 実行結果取得オプション
1036	BLOB ファイル削除
1037	プライマリキー一覧取得準備
1038	プライマリキー一覧取得
1039	プライマリキー一覧取得後処理
1040	BLOB,CLOB および BFILE 分割取得
1041	自動 Commit 設定
1042	自動 Commit 解除
1050	XA トランザクション制御
1501	ストアドプロシジャー一覧取得準備
1502	ストアドプロシジャー一覧取得
1503	ストアドプロシジャー一覧取得後処理
1504	ストアドプロシジャー列一覧取得準備
1505	ストアドプロシジャー列一覧取得
1506	ストアドプロシジャー列一覧取得後処理
1507	ストアドプロシジャーパラメタ定義情報取得
1508	ストアドプロシジャー実行準備
1509	ストアドプロシジャー実行
1510	ストアドプロシジャー拡張実行
2001	正常応答
2002	異常応答

## 5. トラブルシューティング

データベース接続情報に出力される，各履歴種別による DATA 内容の一覧を次の表に示します。

表 5-61 各履歴種別による DATA 内容

履歴種別 (MMMM)	情報 1 (NNNNNNN)	情報 2 (OOOOOOO)	詳細情報 (PP..PP)
PRE	検索カラム数	? パラメタ定義数	SQL の先頭 16 バイト (マルチバイト文字コードの途中の場合はその文字は出力しません)
EXE	SQL 実行件数	? パラメタ指定数	-
FET1	要求時検索件数	最大検索件数	-
FET	累計検索件数	要求時検索件数	-
CLS	0 固定	0 固定	-
CMT	-	-	-
RBK	-	-	-
NOW	今回の要求コードで取得した情報	-	-
OLD	前回の要求コードで取得した情報	-	-

(凡例)

- : 該当なし

注

INSERT, UPDATE, DELETE 実行時の件数

### (4) Exception トレースログ

Exception トレースログは，メソッド実行時に障害が発生した場合に障害要因を分析するためのトレース情報です。障害発生時の問題個所の切り分けをするときに有効です。メソッドの呼び出し情報，メソッドの戻り値の情報，および例外 (Exception) の情報が，製品名称などが出力されたヘッダと合わせて出力されます。

Exception トレースログとして出力される情報を次に示します。

表 5-62 Exception トレースログとして出力される情報

情報の種類	内容
ヘッダ	出力情報の通番
	製品名称
メソッドの呼び出し情報	ログの取得時間
	スレッド識別情報
	メソッドの呼び出し識別情報



情報の種類	内容
	クラス名, オブジェクト ID, メソッド名
	コネクションハンドル
	セクション ID (カーソル ID と同じことを表現しています)
	メソッドの引数
メソッドの戻り情報	ログの取得時間
	スレッド識別情報
	メソッドの戻り識別情報
	クラス名, オブジェクト ID, メソッド名
	コネクションハンドル
	セクション ID (カーソル ID と同じことを表現しています)
	メソッドの引数
Exception の情報	ログの取得時間
	スレッド識別情報
	Exception のスタックトレース

Exception トレースログは、SQLException、BatchUpdateException、または XAException が発生したときに、環境変数に設定した設定情報を基に、Exception が throw される前にファイルに出力されます。Exception トレースを出力するメソッドと、ログ出力に関連する環境変数については、「5.4.1(4) Cosminexus DABroker Library のログの取得」を参照してください。

#### (a) 出力形式と出力項目

Exception トレースログの出力形式を、トレース取得レベル 5 の場合の出力例を基に説明します。

## 5. トラブルシューティング

```
1 2
[1] Cosminexus DABroker for Java , DABroker for Java Version 2 02-11-/D
3 4 5
2004/07/06 23:07:09.129 Thread[main, 5, main]@1259414: [E] [JdbcDbpsvConnection@82c01f. createStatement(1)]
6 7
2004/07/06 23:07:09.160 HANDLE(406397096) : SID(0)
Thread[main, 5, main]@1259414: [R] [JdbcDbpsvConnection@82c01f. createStatement(1)]
HANDLE(406397096) : SID(0)
Return=JP. co. Hitachi. soft. DBPSV_Driver. JdbcDbpsvStatement@1e4cbc4
2004/07/06 23:07:09.160 Thread[main, 5, main]@1259414: [E] [JdbcDbpsvStatement@1e4cbc4. execute]
HANDLE(406397096) : SID(0)
sql=DELETE FROM SEINO_TABLE
2004/07/06 23:07:14.285 Thread[main, 5, main]@1259414: [E] [JdbcDbpsvConnection@82c01f. commit]
HANDLE(406397096) : SID(0)
2004/07/06 23:07:14.301 Thread[main, 5, main]@1259414: [R] [JdbcDbpsvConnection@82c01f. commit]
HANDLE(406397096) : SID(0)
2004/07/06 23:07:14.301 Thread[main, 5, main]@1259414: [R] [JdbcDbpsvStatement@1e4cbc4. execute]
HANDLE(406397096) : SID(1)
Return=false
2004/07/06 23:07:14.301 Thread[main, 5, main]@1259414: [E] [JdbcDbpsvConnection@82c01f. prepareStatement(1)]
HANDLE(406397096) : SID(0)
9
sql=INSERT INTO SEINO_TABLE VALUES(?, ?)
10
2004/07/06 23:07:14.348 Thread[main, 5, main]@1259414: [R] [JdbcDbpsvConnection@82c01f. prepareStatement(1)]
HANDLE(406397096) : SID(1)
11
Return=JP. co. Hitachi. soft. DBPSV_Driver. JdbcDbpsvPreparedStatement@15d56d5
2004/07/06 23:07:26.567 Thread[main, 5, main]@1259414: [E] [JdbcDbpsvConnection@82c01f. commit]
HANDLE(406397096) : SID(0)
2004/07/06 23:07:26.567 Thread[main, 5, main]@1259414: [R] [JdbcDbpsvConnection@82c01f. commit]
HANDLE(406397096) : SID(0)
2004/07/06 23:07:26.567 Thread[main, 5, main]@1259414: [E] [JdbcDbpsvStatement@1e4cbc4. executeQuery]
HANDLE(406397096) : SID(0)
sql=SELECT * FROM SEINO_TABLE
2004/07/06 23:07:26.567 Thread[main, 5, main]@1259414: [E] [JdbcDbpsvStatement@1e4cbc4. execute]
HANDLE(406397096) : SID(0)
sql=SELECT * FROM SEINO_TABLE

(続く)
```

(続き)

```

2004/07/06 23:07:26.676 Thread[main, 5, main]@1259414: [R] [JdbcDbpsvStatement@1e4cbc4.execute]
HANDLE(406397096) : SID(2)
Return=true
2004/07/06 23:07:26.676 Thread[main, 5, main]@1259414: [R] [JdbcDbpsvStatement@1e4cbc4.executeQuery]
HANDLE(406397096) : SID(2)
Return=JP.co.Hitachi.soft.DBPSV_Driver.JdbcDbpsvResultSet@3eca90
2004/07/06 23:07:28.332 Thread[main, 5, main]@1259414: [E] [JdbcDbpsvResultSet@3eca90.close]
HANDLE(406397096) : SID(2)
2004/07/06 23:07:28.332 Thread[main, 5, main]@1259414: [E] [JdbcDbpsvConnection@82c01f.commit]
HANDLE(406397096) : SID(0)
2004/07/06 23:07:28.332 Thread[main, 5, main]@1259414: [R] [JdbcDbpsvConnection@82c01f.commit]
HANDLE(406397096) : SID(0)
2004/07/06 23:07:28.332 Thread[main, 5, main]@1259414: [R] [JdbcDbpsvResultSet@3eca90.close]
HANDLE(406397096) : SID(2)
2004/07/06 23:07:28.332 Thread[Thread-
0, 5, main]@30090737: [E] [JdbcDbpsvConnection@82c01f.prepareStatement(1)]
HANDLE(406397096) : SID(0)
sql=SELECT * FROM SEINO_TABLE
2004/07/06 23:07:28.332 Thread[Thread-
0, 5, main]@30090737: [R] [JdbcDbpsvConnection@82c01f.prepareStatement(1)]
HANDLE(406397096) : SID(3)
Return=JP.co.Hitachi.soft.DBPSV_Driver.JdbcDbpsvPreparedStatement@2808b3
2004/07/06 23:07:28.348 Thread[Thread-
1, 5, main]@5462872: [E] [JdbcDbpsvConnection@82c01f.prepareStatement(1)]
HANDLE(406397096) : SID(0)
sql=DELETE FROM SEINO_TABLE WHERE I1 = ?
2004/07/06 23:07:30.098 Thread[Thread-
1, 5, main]@5462872: [R] [JdbcDbpsvConnection@82c01f.prepareStatement(1)]
HANDLE(406397096) : SID(4)
Return=JP.co.Hitachi.soft.DBPSV_Driver.JdbcDbpsvPreparedStatement@922804
2004/07/06 23:07:30.332 Thread[Thread-2, 5, main]@25253977: [E] [JdbcDbpsvConnection@82c01f.rollback]
HANDLE(406397096) : SID(0)
2004/07/06 23:07:42.098 Thread[Thread-2, 5, main]@25253977: [R] [JdbcDbpsvConnection@82c01f.rollback]
HANDLE(406397096) : SID(0)
2004/07/06 23:07:42.098 Thread[Thread-2, 5, main]@25253977: [E] [JdbcDbpsvConnection@82c01f.close]
HANDLE(406397096) : SID(0)
2004/07/06 23:07:42.098 Thread[Thread-2, 5, main]@25253977: [R] [JdbcDbpsvConnection@82c01f.close]
HANDLE(406397096) : SID(0)

```

(続く)

## 5. トラブルシューティング

```

(続き)
2004/07/06 23:07:42.535 Thread[Thread-1,5.main]@5462872:JP.co.Hitachi.soft.DBPSV_Driver.SQLException:
KFDJ00002-E Statement or Connection is closed. [JdbcDbpsvPreparedStatement.setInt]
at
JP.co.Hitachi.soft.DBPSV_Driver.JdbcDbpsvPreparedStatement.setInt(JdbcDbpsvPreparedStatement.java:1363)
at Main3.run(Main3.java:35)
[2] Cosminexus DABroker for Java , DABroker for Java Version 2 02-11-/D
2004/07/06 23:07:25.723 Thread[Thread-
0,5.main]@1259414: [E] [JdbcDbpsvConnection@82c01f.prepareStatement(1)]
HANDLE(406488568) : SID(0)
sql=SELECT * FROM SEINO_TABLE
2004/07/06 23:07:25.770 Thread[Thread-1,5.main]@33409388: [E] [JdbcDbpsvConnection@82c01f.rollback]
HANDLE(406488568) : SID(0)
2004/07/06 23:07:25.770 Thread[Thread-1,5.main]@33409388: [R] [JdbcDbpsvConnection@82c01f.rollback]
HANDLE(406488568) : SID(0)
2004/07/06 23:07:25.770 Thread[Thread-
2,5.main]@11730319: [E] [JdbcDbpsvConnection@82c01f.prepareStatement(1)]
HANDLE(406488568) : SID(0)
sql=SELECT ** FROM SEINO_TABLE
2004/07/06 23:07:25.863 Thread[Thread-2,5.main]@11730319:JP.co.Hitachi.soft.DBPSV_Driver.SQLException:
KFDJ00001-E Error occurred at server. [JdbcDbpsvResultSQLExecute.SQLPrepare]
OperationType : 2002
ReturnCode : -100
ErrorCode : -105
WarningInfo : 0
ErrorMsg : KFP11105-E Invalid token "*" after token "*"
PreparedSQL : SELECT
SectionID : 1
at
JP.co.Hitachi.soft.DBPSV_Driver.JdbcDbpsvResultSQLExecute.SQLPrepare(JdbcDbpsvResultSQLExecute.java:1413)
at JP.co.Hitachi.soft.DBPSV_Driver.JdbcDbpsvStatement.prepare(JdbcDbpsvStatement.java:2536)
at
JP.co.Hitachi.soft.DBPSV_Driver.JdbcDbpsvPreparedStatement.<init>(JdbcDbpsvPreparedStatement.java:158)
at
JP.co.Hitachi.soft.DBPSV_Driver.JdbcDbpsvConnection.prepareStatement(JdbcDbpsvConnection.java:590)
at Main4.run(Main4.java:53)

```

出力される内容について次の表で示します。項番は出力例中の番号と対応しています。

表 5-63 Exception トレースログの出力項目

項番	区分	表示	意味
1	ヘッダ	[XXXX]	出力情報の通番を示します。
2		Cosminexus DABroker for Java , DABroker for Java Version 2 XX-XX-XX	Cosminexus DABroker Library の ログであることを示します。
3	共通	XXXX/XX/XX XX:XX:XX.XXX	ログの取得時間を示します。
4		Thread[XXXX,XXXXX,XXXX]@XXXX	スレッド識別情報を示します。
5		[JdbcDbpsvXXXX@XXXX.XXXX]	クラス名、オブジェクト ID、メ ソッド名を示します。
6		HANDLE(XXXX)	ConnectionHandle を示します。
7		SID(XXX)	セクション ID (カーソル ID と同じ ことを表現しています) を示しま す。

項番	区分	表示	意味
8	メソッドの呼び出し情報	[E]	メソッドの呼び出し識別情報を示します。
9		XXXX=XXXX	メソッドの引数を示します。
10	メソッドの戻り情報	[R]	メソッドからの戻り識別情報を示します。
11		Return=XXXX	メソッドの戻り値を示します。
12	Exception 情報	JP.co.Hitachi.soft.DBPSV_Driver.XXXX Exception:XXXX	Exception のスタックトレースを表示します。

## (b) 解析方法

ここでは、(a) で示した出力例を使用して、Exception トレースログの解析方法について説明します。

Exception トレースログは、次の手順で解析します。

1. 解析するログの通番から、ログ情報を抜き出します。
2. ログ情報を、Thread 識別情報で分類して、ログごとに分割します。
3. ログの取得時間によって、ログ情報を時系列に並べます。  
手順 1. ~ 3. を実施すると、次のような表を作成できます。

時間	スレッド 1	スレッド 2	スレッド 3	スレッド 4
		Thread[main,5,main] @1259414	Thread[Thread-0,5,main] @30090737	Thread[Thread-1,5,main] @5462872
2004/07/06 23:07:09.129	JdbcDbpsvConnection @82c01f.createStatement(1)	-	-	-
2004/07/06 23:07:09.160	JdbcDbpsvStatement @1e4cbc4.execute	-	-	-
2004/07/06 23:07:14.285	JdbcDbpsvConnection @82c01f.commit	-	-	-
2004/07/06 23:07:14.301	JdbcDbpsvConnection @82c01f.prepareStatement(1)	-	-	-
2004/07/06 23:07:26.567	JdbcDbpsvConnection @82c01f.commit	-	-	-

## 5. トラブルシューティング

時間	スレッド 1	スレッド 2	スレッド 3	スレッド 4
	Thread[main,5,main] @1259414	Thread[Thread-0,5,main] @30090737	Thread[Thread-1,5,main] @5462872	Thread[Thread-2,5,main] @25253977
2004/07/06 23:07:26.567	JdbcDbpsvStatement @1e4cbc4.executeQuery	-	-	-
2004/07/06 23:07:26.567	JdbcDbpsvStatement @1e4cbc4.execute	-	-	-
2004/07/06 23:07:28.332	JdbcDbpsvResultSet @3eca90.close	JdbcDbpsvConnection @82c01f.prepareStatement(1)	-	-
2004/07/06 23:07:28.332	JdbcDbpsvConnection @82c01f.commit	-	-	-
2004/07/06 23:07:28.348	-	-	JdbcDbpsvConnection @82c01f.prepareStatement(1)	-
2004/07/06 23:07:30.332	-	-	-	JdbcDbpsvConnection @82c01f.rollback
2004/07/06 23:07:42.098	-	-	-	JdbcDbpsvConnection @82c01f.close
2004/07/06 23:07:42.535	-	-	<b>SQLException 発生</b> KFDJ00002-E Statement or Connection is closed.	-

(凡例)

- : 該当しません。

### 4. Exception エラーの内容を確認します。

2004/07/06 23:07:42.535 のスレッド 3 で SQLException が発生しています。メッセージの内容から、この時点で Statement オブジェクト、または Connection オブジェクトがすでにクローズされていることによるエラーだとわかります。

### 5. 時系列にオブジェクトの流れを確認します。

2004/07/06 23:07:09.129 のスレッド 1 , 2004/07/06 23:07:28.332 のスレッド 2 , 2004/07/06 23:07:28.348 のスレッド 3 , 2004/07/06 23:07:30.332 のスレッド 4 の Connection オブジェクトのオブジェクト ID が同じであることから、四つのスレッドが同一コネクションで処理されていることがわかります。

## 6. エラーの原因となっている個所を探します。

四つのスレッドが同一コネクションであることがわかったため、Statement.close メソッド、または Connection.close メソッドを実行している個所を探します。この例の場合は、スレッド 4 が 2004/07/06 23:07:42.098 で Connection.close メソッドを実行しています。

このことから、2004/07/06 23:07:42.535 のスレッド 3 で発生した SQLException の原因は、スレッド 4 が 2004/07/06 23:07:42.098 で Connection.close メソッドを実行していたためであることが、わかります。

## (c) トレース取得レベルごとの出力例

トレース取得レベルごとの出力例を次に示します。

トレース取得レベル 1 の場合の出力例

```
[1] Cosminexus DABroker for Java , DABroker for Java Version 2 02-11-/D
2004/07/21 17:59:41.001 Thread[main,5,main]@2113601537
[E][JdbcDbpsvXADataSource@59920001.getXAConnection(1)]
      HANDLE(0) : SID(0)
2004/07/21 17:59:41.020 Thread[main,5,main]@2113601537
[E][JdbcDbpsvXAConnection@8bf8002.getXAResource]
      HANDLE(0) : SID(0)
2004/07/21 17:59:41.554 Thread[main,5,main]@2113601537
[R][JdbcDbpsvXAConnection@8bf8002.getXAResource]
      HANDLE(0) : SID(0)

Return=JP.co.Hitachi.soft.DBPSV_Driver.JdbcDbpsvXAResource@33690002
2004/07/21 17:59:41.554 Thread[main,5,main]@2113601537
[R][JdbcDbpsvXADataSource@59920001.getXAConnection(1)]
      HANDLE(0) : SID(0)

Return=JP.co.Hitachi.soft.DBPSV_Driver.JdbcDbpsvXAConnection@8bf8002
2004/07/21 17:59:41.557 Thread[main,5,main]@2113601537
[E][JdbcDbpsvXAConnection@8bf8002.getXAResource]
      HANDLE(0) : SID(0)
2004/07/21 17:59:41.557 Thread[main,5,main]@2113601537
[R][JdbcDbpsvXAConnection@8bf8002.getXAResource]
      HANDLE(0) : SID(0)

Return=JP.co.Hitachi.soft.DBPSV_Driver.JdbcDbpsvXAResource@33690002
```

## 5. トラブルシューティング

```
2004/07/21 17:59:41.561 Thread[main,5,main]@2113601537
[E][JdbcDbpsvXAConnection@8bf8002.getConnection]
    HANDLE(0) : SID(0)
2004/07/21 17:59:41.649 Thread[main,5,main]@2113601537
[R][JdbcDbpsvXAConnection@8bf8002.getConnection]
    HANDLE(883163352) : SID(0)

Return=JP.co.Hitachi.soft.DBPSV_Driver.JdbcDbpsvConnection@78688002
2004/07/21 17:59:41.649 Thread[main,5,main]@2113601537
[E][JdbcDbpsvConnection@78688002.prepareStatement(1)]
    HANDLE(883163352) : SID(0)
    sql=INSERT INTO TRC_ex VALUES(?)
2004/07/21 17:59:41.819 Thread[main,5,main]@2113601537
[R][JdbcDbpsvConnection@78688002.prepareStatement(1)]
    HANDLE(883163352) : SID(1)

Return=JP.co.Hitachi.soft.DBPSV_Driver.JdbcDbpsvPreparedStatement@d334002
2004/07/21 17:59:41.977 Thread[main,5,main]@2113601537
[E][JdbcDbpsvConnection@78688002.close]
    HANDLE(883163352) : SID(0)
2004/07/21 17:59:42.886 Thread[main,5,main]@2113601537
[R][JdbcDbpsvConnection@78688002.close]
    HANDLE(883163352) : SID(0)
2004/07/21 17:59:42.945 Thread[main,5,main]@2113601537
[E][JdbcDbpsvConnection@78688002.createStatement(1)]
    HANDLE(883163352) : SID(0)

2004/07/21 17:59:42.946
Thread[main,5,main]@2113601537:JP.co.Hitachi.soft.DBPSV_Driver.SQLException:
KFDJ00006-E Connection is closed.[JdbcDbpsvConnection.createStatement(1)]
    at
JP.co.Hitachi.soft.DBPSV_Driver.JdbcDbpsvConnection.createStatement(JdbcDbpsvC
onnection.java:529)
    at TRC_ex.main(TRC_ex.java:81)
```

### トレース取得レベル2の場合の出力例

```
[1] Cosminexus DABroker for Java , DABroker for Java Version 2 02-11-/D
2004/07/21 19:00:56.992 Thread[main,5,main]@2113866091
[E][JdbcDbpsvXADataSource@5996096b.getXAConnection(1)]
    HANDLE(0) : SID(0)
2004/07/21 19:00:57.012 Thread[main,5,main]@2113866091
[E][JdbcDbpsvXAConnection@8bb8968.getXAResource]
    HANDLE(0) : SID(0)
2004/07/21 19:00:57.561 Thread[main,5,main]@2113866091
[R][JdbcDbpsvXAConnection@8bb8968.getXAResource]
    HANDLE(0) : SID(0)

Return=JP.co.Hitachi.soft.DBPSV_Driver.JdbcDbpsvXAResource@336d0968
2004/07/21 19:00:57.561 Thread[main,5,main]@2113866091
[R][JdbcDbpsvXADataSource@5996096b.getXAConnection(1)]
    HANDLE(0) : SID(0)

Return=JP.co.Hitachi.soft.DBPSV_Driver.JdbcDbpsvXAConnection@8bb8968
```



```

2004/07/21 19:00:57.565 Thread[main,5,main]@2113866091
[E][JdbcDbpsvXAConnection@8bb8968.getXAResource]
      HANDLE(0) : SID(0)
2004/07/21 19:00:57.565 Thread[main,5,main]@2113866091
[R][JdbcDbpsvXAConnection@8bb8968.getXAResource]
      HANDLE(0) : SID(0)

Return=JP.co.Hitachi.soft.DBPSV_Driver.JdbcDbpsvXAResource@336d0968
2004/07/21 19:00:57.569 Thread[main,5,main]@2113866091
[E][JdbcDbpsvXAConnection@8bb8968.getConnection]
      HANDLE(0) : SID(0)
2004/07/21 19:00:57.660 Thread[main,5,main]@2113866091
[R][JdbcDbpsvXAConnection@8bb8968.getConnection]
      HANDLE(883163368) : SID(0)

Return=JP.co.Hitachi.soft.DBPSV_Driver.JdbcDbpsvConnection@7862c968

2004/07/21 19:00:57.660 Thread[main,5,main]@2113866091
[E][JdbcDbpsvConnection@7862c968.prepareStatement(1)]
      HANDLE(883163368) : SID(0)
      sql=INSERT INTO TRC_ex VALUES(?)
2004/07/21 19:00:57.779 Thread[main,5,main]@2113866091
[R][JdbcDbpsvConnection@7862c968.prepareStatement(1)]
      HANDLE(883163368) : SID(1)

Return=JP.co.Hitachi.soft.DBPSV_Driver.JdbcDbpsvPreparedStatement@d4dc968
2004/07/21 19:00:57.840 Thread[main,5,main]@2113866091
[E][JdbcDbpsvPreparedStatement@d4dc968.executeUpdate(1)]
      HANDLE(883163368) : SID(1)
      sql=INSERT INTO TRC_ex VALUES(?)
2004/07/21 19:00:57.854 Thread[main,5,main]@2113866091
[R][JdbcDbpsvPreparedStatement@d4dc968.executeUpdate(1)]
      HANDLE(883163368) : SID(1)
      Return=1

2004/07/21 19:00:57.854 Thread[main,5,main]@2113866091
[E][JdbcDbpsvConnection@7862c968.close]
      HANDLE(883163368) : SID(0)
2004/07/21 19:00:57.886 Thread[main,5,main]@2113866091
[R][JdbcDbpsvConnection@7862c968.close]
      HANDLE(883163368) : SID(0)
2004/07/21 19:00:57.916 Thread[main,5,main]@2113866091
[E][JdbcDbpsvConnection@7862c968.createStatement(1)]
      HANDLE(883163368) : SID(0)
2004/07/21 19:00:57.917
Thread[main,5,main]@2113866091:JP.co.Hitachi.soft.DBPSV_Driver.SQLException:
KFDJ00006-E Connection is closed.[JdbcDbpsvConnection.createStatement(1)]
    at
JP.co.Hitachi.soft.DBPSV_Driver.JdbcDbpsvConnection.createStatement(JdbcDbpsvC
onnection.java:529)
    at TRC_ex.main(TRC_ex.java:81)

```

(凡例)

下線は、トレース取得レベル1ではトレースが出力されないメソッドの情報です。

トレースレベル3の場合の出力例

## 5. トラブルシューティング

```
[1] Cosminexus DABroker for Java , DABroker for Java Version 2 02-11-/D
2004/07/21 19:01:28.755 Thread[main,5,main]@2113866075
[E][JdbcDbpsvXADataSource@5996095b.getXAConnection(1)]
    HANDLE(0) : SID(0)
2004/07/21 19:01:28.774 Thread[main,5,main]@2113866075
[E][JdbcDbpsvXAConnection@8bb8958.getXAResource]
    HANDLE(0) : SID(0)
2004/07/21 19:01:29.329 Thread[main,5,main]@2113866075
[R][JdbcDbpsvXAConnection@8bb8958.getXAResource]
    HANDLE(0) : SID(0)

Return=JP.co.Hitachi.soft.DBPSV_Driver.JdbcDbpsvXAResource@336d0958
2004/07/21 19:01:29.329 Thread[main,5,main]@2113866075
[R][JdbcDbpsvXADataSource@5996095b.getXAConnection(1)]
    HANDLE(0) : SID(0)

Return=JP.co.Hitachi.soft.DBPSV_Driver.JdbcDbpsvXAConnection@8bb8958
2004/07/21 19:01:29.332 Thread[main,5,main]@2113866075
[E][JdbcDbpsvXAConnection@8bb8958.getXAResource]
    HANDLE(0) : SID(0)

2004/07/21 19:01:29.332 Thread[main,5,main]@2113866075
[R][JdbcDbpsvXAConnection@8bb8958.getXAResource]
    HANDLE(0) : SID(0)

Return=JP.co.Hitachi.soft.DBPSV_Driver.JdbcDbpsvXAResource@336d0958
2004/07/21 19:01:29.333 Thread[main,5,main]@2113866075
[E][JdbcDbpsvXAResource@336d0958.start]
    HANDLE(0) : SID(0)
    xid=MyXid@5ffe4958
    flags=0
2004/07/21 19:01:29.372 Thread[main,5,main]@2113866075
[R][JdbcDbpsvXAResource@336d0958.start]
    HANDLE(0) : SID(0)
2004/07/21 19:01:29.389 Thread[main,5,main]@2113866075
[E][JdbcDbpsvXAConnection@8bb8958.getConnection]
    HANDLE(0) : SID(0)
2004/07/21 19:01:29.477 Thread[main,5,main]@2113866075
[R][JdbcDbpsvXAConnection@8bb8958.getConnection]
    HANDLE(883163368) : SID(0)

Return=JP.co.Hitachi.soft.DBPSV_Driver.JdbcDbpsvConnection@78b60958
```

```

2004/07/21 19:01:29.478 Thread[main,5,main]@2113866075
[E][JdbcDbpsvConnection@78b60958.prepareStatement(1)]
    HANDLE(883163368) : SID(0)
    sql=INSERT INTO TRC_ex VALUES(?)
2004/07/21 19:01:29.571 Thread[main,5,main]@2113866075
[R][JdbcDbpsvConnection@78b60958.prepareStatement(1)]
    HANDLE(883163368) : SID(1)

Return=JP.co.Hitachi.soft.DBPSV_Driver.JdbcDbpsvPreparedStatement@13934958
2004/07/21 19:01:29.614 Thread[main,5,main]@2113866075
[E][JdbcDbpsvPreparedStatement@13934958.executeUpdate(1)]
    HANDLE(883163368) : SID(1)
    sql=INSERT INTO TRC_ex VALUES(?)
2004/07/21 19:01:29.624 Thread[main,5,main]@2113866075
[R][JdbcDbpsvPreparedStatement@13934958.executeUpdate(1)]
    HANDLE(883163368) : SID(1)
    Return=1
2004/07/21 19:01:29.624 Thread[main,5,main]@2113866075
[E][JdbcDbpsvConnection@78b60958.close]
    HANDLE(883163368) : SID(0)
2004/07/21 19:01:29.656 Thread[main,5,main]@2113866075
[R][JdbcDbpsvConnection@78b60958.close]
    HANDLE(883163368) : SID(0)
2004/07/21 19:01:29.656 Thread[main,5,main]@2113866075
[E][JdbcDbpsvXAResource@336d0958.end]
    HANDLE(0) : SID(0)
    xid=MyXid@5ffe4958
    flags=67108864

2004/07/21 19:01:29.658 Thread[main,5,main]@2113866075
[R][JdbcDbpsvXAResource@336d0958.end]
    HANDLE(0) : SID(0)
2004/07/21 19:01:29.659 Thread[main,5,main]@2113866075
[E][JdbcDbpsvXAResource@336d0958.prepare]
    HANDLE(0) : SID(0)
    xid=MyXid@5ffe4958
2004/07/21 19:01:29.675 Thread[main,5,main]@2113866075
[R][JdbcDbpsvXAResource@336d0958.prepare]
    HANDLE(0) : SID(0)
    Return=0
2004/07/21 19:01:29.675 Thread[main,5,main]@2113866075
[E][JdbcDbpsvXAResource@336d0958.commit]
    HANDLE(0) : SID(0)
    xid=MyXid@5ffe4958
    onePhase=false
2004/07/21 19:01:29.681 Thread[main,5,main]@2113866075
[R][JdbcDbpsvXAResource@336d0958.commit]
    HANDLE(0) : SID(0)
2004/07/21 19:01:29.681 Thread[main,5,main]@2113866075
[E][JdbcDbpsvConnection@78b60958.createStatement(1)]
    HANDLE(883163368) : SID(0)
2004/07/21 19:01:29.682
Thread[main,5,main]@2113866075:JP.co.Hitachi.soft.DBPSV_Driver.SQLException:
KFDJ00006-E Connection is closed.[JdbcDbpsvConnection.createStatement(1)]
    at
JP.co.Hitachi.soft.DBPSV_Driver.JdbcDbpsvConnection.createStatement(JdbcDbpsvC
onnection.java:529)
    at TRC_ex.main(TRC_ex.java:81)

```

( 凡例 )

**太字**は、トレース取得レベル 1 ではトレースが出力されないメソッドの情報です。

下線は、トレース取得レベル 2 ではトレースが出力されないメソッドの情報です。

トレース取得レベル 5 の場合の出力例

## 5. トラブルシューティング

```
[I] Cosminexus DABroker for Java , DABroker for Java Version 2 02-11- /D
2004/07/21 18:58:47.122 Thread[main,5,main]@2113866026
[E][JdbcDbpsvXADataSource@5996092a.getXAConnection(1)]
    HANDLE(0) : SID(0)
2004/07/21 18:58:47.141 Thread[main,5,main]@2113866026
[E][JdbcDbpsvXAConnection@8bb8929.getXAResource]
    HANDLE(0) : SID(0)
2004/07/21 18:58:47.705 Thread[main,5,main]@2113866026
[R][JdbcDbpsvXAConnection@8bb8929.getXAResource]
    HANDLE(0) : SID(0)

Return=JP.co.Hitachi.soft.DBPSV_Driver.JdbcDbpsvXAResource@336d0929
2004/07/21 18:58:47.706 Thread[main,5,main]@2113866026
[R][JdbcDbpsvXADataSource@5996092a.getXAConnection(1)]
    HANDLE(0) : SID(0)

Return=JP.co.Hitachi.soft.DBPSV_Driver.JdbcDbpsvXAConnection@8bb8929
2004/07/21 18:58:47.709 Thread[main,5,main]@2113866026
[E][JdbcDbpsvXAConnection@8bb8929.getXAResource]
    HANDLE(0) : SID(0)

2004/07/21 18:58:47.709 Thread[main,5,main]@2113866026
[R][JdbcDbpsvXAConnection@8bb8929.getXAResource]
    HANDLE(0) : SID(0)

Return=JP.co.Hitachi.soft.DBPSV_Driver.JdbcDbpsvXAResource@336d0929
2004/07/21 18:58:47.710 Thread[main,5,main]@2113866026
[E][JdbcDbpsvXAResource@336d0929.start]
    HANDLE(0) : SID(0)
    xid=MyXid@5ffe4929
    flags=0
2004/07/21 18:58:47.750 Thread[main,5,main]@2113866026
[R][JdbcDbpsvXAResource@336d0929.start]
    HANDLE(0) : SID(0)
2004/07/21 18:58:47.770 Thread[main,5,main]@2113866026
[E][JdbcDbpsvXAConnection@8bb8929.getConnection]
    HANDLE(0) : SID(0)
2004/07/21 18:58:47.858 Thread[main,5,main]@2113866026
[R][JdbcDbpsvXAConnection@8bb8929.getConnection]
    HANDLE(883163368) : SID(0)

Return=JP.co.Hitachi.soft.DBPSV_Driver.JdbcDbpsvConnection@78454929
2004/07/21 18:58:47.858 Thread[main,5,main]@2113866026
[E][JdbcDbpsvConnection@78454929.prepareStatement(1)]
    HANDLE(883163368) : SID(0)
    sql=INSERT INTO TRC_ex VALUES(?)
2004/07/21 18:58:47.952 Thread[main,5,main]@2113866026
[R][JdbcDbpsvConnection@78454929.prepareStatement(1)]
    HANDLE(883163368) : SID(1)

Return=JP.co.Hitachi.soft.DBPSV_Driver.JdbcDbpsvPreparedStatement@d2ec929
2004/07/21 18:58:47.994 Thread[main,5,main]@2113866026
[E][JdbcDbpsvPreparedStatement@d2ec929.executeUpdate(1)]
    HANDLE(883163368) : SID(1)
    sql=INSERT INTO TRC_ex VALUES(?)
2004/07/21 18:58:47.994 Thread[main,5,main]@2113866026
[E][JdbcDbpsvPreparedStatement@d2ec929.execute(1)]
    HANDLE(883163368) : SID(1)
    sql=INSERT INTO TRC_ex VALUES(?)
2004/07/21 18:58:48.004 Thread[main,5,main]@2113866026
[R][JdbcDbpsvPreparedStatement@d2ec929.execute(1)]
    HANDLE(883163368) : SID(1)
Return=false
```

```

2004/07/21 18:58:48.004 Thread[main,5,main]@2113866026
[R][JdbcDbpsvPreparedStatement@d2ec929.executeUpdate(1)]
      HANDLE(883163368) : SID(1)
      Return=1
2004/07/21 18:58:48.004 Thread[main,5,main]@2113866026
[E][JdbcDbpsvConnection@78454929.close]
      HANDLE(883163368) : SID(0)
2004/07/21 18:58:48.037 Thread[main,5,main]@2113866026
[R][JdbcDbpsvConnection@78454929.close]
      HANDLE(883163368) : SID(0)
2004/07/21 18:58:48.037 Thread[main,5,main]@2113866026
[E][JdbcDbpsvXAResource@336d0929.end]
      HANDLE(0) : SID(0)
      xid=MyXid@5ffe4929
      flags=67108864
2004/07/21 18:58:48.039 Thread[main,5,main]@2113866026
[R][JdbcDbpsvXAResource@336d0929.end]
      HANDLE(0) : SID(0)

2004/07/21 18:58:48.039 Thread[main,5,main]@2113866026
[E][JdbcDbpsvXAResource@336d0929.prepare]
      HANDLE(0) : SID(0)
      xid=MyXid@5ffe4929
2004/07/21 18:58:48.053 Thread[main,5,main]@2113866026
[R][JdbcDbpsvXAResource@336d0929.prepare]
      HANDLE(0) : SID(0)
      Return=0
2004/07/21 18:58:48.053 Thread[main,5,main]@2113866026
[E][JdbcDbpsvXAResource@336d0929.commit]
      HANDLE(0) : SID(0)
      xid=MyXid@5ffe4929
      onePhase=false
2004/07/21 18:58:48.059 Thread[main,5,main]@2113866026
[R][JdbcDbpsvXAResource@336d0929.commit]
      HANDLE(0) : SID(0)
2004/07/21 18:58:48.059 Thread[main,5,main]@2113866026
[E][JdbcDbpsvConnection@78454929.createStatement(1)]
      HANDLE(883163368) : SID(0)
2004/07/21 18:58:48.060
Thread[main,5,main]@2113866026:JP.co.Hitachi.soft.DBPSV_Driver.SQLException:
KFDJ00006-E Connection is closed.[JdbcDbpsvConnection.createStatement(1)]
      at
JP.co.Hitachi.soft.DBPSV_Driver.JdbcDbpsvConnection.createStatement(JdbcDbpsvC
onnection.java:529)
      at TRC_ex.main(TRC_ex.java:81)

```

(凡例)

**太字**は、トレース取得レベル1ではトレースが出力されないメソッドの情報です。

下線は、トレース取得レベル2ではトレースが出力されないメソッドの情報です。

## (5) イベントトレース

イベントトレースは、DABroker for C++ をインストールしている場合に出力されます。

DABroker for C++ の機能を使ってデータベースにアクセスする場合のイベントトレースが出力されます。イベントトレースは、次のファイルに取得されます。

- Windows の場合  
<Cosminexus DABroker Library 運用ディレクトリ >%spool%\dabevttccpp1
- UNIX の場合  
<Cosminexus DABroker Library 運用ディレクトリ >/spool/dabevttccpp1

## 5. トラブルシューティング

なお、イベントトレースファイルが満杯になった場合、ラップアラウンドでトレースが出力されます。この場合は、バックアップファイル（dabevtttrccpp2）が作成され、1世代だけ（dabevtttrccpp1）が管理されます。

取得するイベントトレースの設定については、マニュアル「Cosminexus システム構築ガイド」のCosminexus DABroker Libraryの動作環境の設定に関する説明を参照してください。

### (a) 出力形式

イベントトレースの出力形式について説明します。

ヘッダ「LEVEL VPID PID TID DATE TIME EVENT」の下に、次に示す形式でトレース情報が出力されます。

レベル 仮想プロセスID プロセスID スレッドID 日付 時刻 動作結果

イベントトレースの出力例を次に示します。

LEVEL	VPID	PID	TID	DATE	TIME	EVENT
40000010	00120000	711a	0001	2004/06/01	11:38:29.300000	イベント1
40000010	00120000	711a	0002	2004/06/01	11:38:30.100000	イベント2
40000010	00120000	711b	0001	2004/06/01	11:38:30.150000	イベント3
40000010	00120000	711b	0002	2004/06/01	11:38:30.250000	イベント4

### (b) 出力項目

イベントトレースの出力項目について、次の表に示します。

表 5-64 Cosminexus DABroker Library のイベントトレースの出力項目

ヘッダ	項目	説明
LEVEL	レベル	トレースに出力したレベルが出力されます。
VPID	仮想プロセスID	仮想プロセスIDが出力されます。
PID	プロセスID	プロセスIDが出力されます。
TID	スレッドID	スレッドIDが出力されます。
DATE	日付	取得した日付が出力されます。
TIME	時刻	取得した時刻が出力されます。
EVENT	動作結果	実行時の動作結果のログが出力されます。

### (6) XA トレースの取得形式

XA インタフェースでのトラブルシュート情報は、XA トレースファイルとして次のファイルに出力されます。



## 5. トラブルシューティング

項目名	説明
<code>rmid=rrrr</code>	RMID が出力されます。
<code>flags=ffff</code>	フラグパラメタが出力されます。
<code>RMName="nnnn"</code>	RM 名称が出力されます。
<code>xa_info="iii "</code>	XA 文字列が出力されます。 この項目は、XA 関数が、 <code>xa_open()</code> 、または <code>xa_close()</code> の場合だけ、出力されます。
<code>xid=[ffff:gggg:bbbb:dddd]</code>	XID が出力されます。 この項目は、XA 関数が、 <code>xa_open()</code> 、または <code>xa_close()</code> 以外の場合だけ、出力されます。 <i>ffff</i> : フォーマット ID が出力されます。 <i>gggg</i> : グローバルトランザクション ID 長が出力されます。 <i>bbbb</i> : ブランチ修飾子長が出力されます。 <i>dddd</i> : XID データが出力されます。
<code>count=cccc</code>	リカバリ情報取得カウントが出力されます。 この項目は、XA 関数が <code>xa_recover()</code> の場合だけ、出力されます。
<code>return code = rrrr</code>	XA 関数の戻り値が出力されます。 この項目は、イベント種別が "end" の場合だけ出力されます。

### (7) JDBC ドライバのトレース (JDBC インタフェースメソッドトレース)

J2EE アプリケーションまたはバッチアプリケーション開発時に、Cosminexus DABroker Library の API を使用して、JDBC ドライバのトレースを出力するように設定できます。

ここでは、JDBC ドライバのトレースのうち、JDBC インタフェースメソッドのトレースについて説明します。

#### (a) 出力形式

JDBC インタフェースメソッドのトレースの出力形式を次に示します。斜体は可変の情報です。

```
[DABroker for Java Version 2] [JDBC Interface メソッド状態][JdbcDbpsv
クラス名.メソッド名] 引数または戻り値=値
```

JDBC インタフェースメソッドのトレースの出力例を次の図に示します。



```

[DABroker for Java Version 2][JDBC Interface Entry ][JdbcDbpsvDriver.connect]
[DABroker for Java Version 2]          url=jdbc:hitachi:dbpsv://ikn0205/DB=hirdb
[DABroker for Java Version 2]          info={user=jdbc, password=jdbc,
UAPNAME=sample, JDBC_IF=ON, SV_EVENT=ON, TRC_NO=1000}
[DABroker for Java Version 2][JDBC Interface Return][JdbcDbpsvDriver.connect]
[DABroker for Java Version 2]          Return=JP.co.Hitachi.soft.DBPSV_Driver.
JdbcDbpsvConnection@1cee91
:

```

## (b) 出力項目

JDBC インタフェースメソッドのトレースの出力項目について、次の表に示します。

表 5-66 JDBC インタフェースメソッドのトレースの出力項目

項目	説明
[DABroker for Java Version 2]	Cosminexus DABroker Library のトレースであることを示します。
[JDBC Interface メソッド状態]	使用した JDBC インタフェースメソッドの状態を示します。 Entry：メソッドの呼び出し Return：メソッドの戻り
[JdbcDbpsv クラス名.メソッド名]	使用した JDBC インタフェースメソッドの「クラス名.メソッド名」を示します。
引数または戻り値= 値	該当する「クラス名.メソッド名」の引数の値、または戻り値が出力されます。 引数の場合、 <i>引数名= 値</i> の形式で出力されます。 戻り値の場合、 <i>Return= 値</i> の形式で出力されます。

## (8) JDBC ドライバのトレース (Cosminexus DABroker Library とのイベントトレース)

J2EE アプリケーションまたはバッチアプリケーション開発時に、Cosminexus DABroker Library の API を使用して、JDBC ドライバのトレースを出力するように設定できます。

ここでは、JDBC ドライバと Cosminexus DABroker Library とのイベントトレースについて説明します。

## (a) 出力形式

Cosminexus DABroker Library とのイベントトレースの出力形式を次に示します。

```
[DABroker for Java Version 2] [Server Event イベント] 送信データ種別
```

Cosminexus DABroker Library とのイベントトレースの出力例を次に示します。

5. トラブルシューティング

```
[DABroker for Java Version 2][Server Event Get]Result(0,0) SIZE:91
[DABroker for Java Version 2][Server Event Put]Connect(Database) SIZE:112
:
```

(b) 出力項目

Cosminexus DABroker Library とのイベントトレースの出力項目について、次の表に示します。

表 5-67 DABroker とのイベントトレースの出力項目

項目	説明
[DABroker for Java Version 2]	Cosminexus DABroker Library のトレースであることを示します。
[Server Event イベント]	Cosminexus DABroker Library でのイベントを示します。
送信データ種別	送受信データ種別が、 <i>aa...aa &lt;bb...bb&gt; (cc...cc,dd...dd) SIZE : XXX</i> の形式で出力されます
<i>aa...aa</i>	オペレーション種別が出力されます。表示されるオペレーション種別とその意味は次のとおりです。 <ul style="list-style-type: none"> <li>• Disconnect(Connection) : Cosminexus DABroker Library とのコネクション解放要求</li> <li>• Connect(Database) : データベースとの接続要求</li> <li>• Disconnect(Database) : データベースとの接続解放要求</li> <li>• Commit : トランザクションのコミット要求</li> <li>• Rollback : トランザクションのロールバック要求</li> <li>• LIST(Table) : 表一覧取得要求</li> <li>• LIST(Column) : 列一覧取得要求</li> <li>• LIST(Index) : 索引一覧取得要求</li> <li>• LIST(TablePrivileges) : 表権限一覧取得要求</li> <li>• LIST(ColumnPrivileges) : 列権限一覧取得要求</li> <li>• LIST(Procedure) : プロシジャ一覧取得要求</li> <li>• LIST(ProcedureColumn) : プロシジャ列一覧取得要求</li> <li>• EXT-CMD : SQL の実行要求</li> <li>• EXE(Procedure) : プロシジャの実行要求</li> <li>• Cancel : 非同期キャンセル要求</li> <li>• Result : 正常応答</li> <li>• Error : エラー応答</li> <li>• Result(Cancel) : 正常応答 (非同期キャンセル)</li> <li>• Error(Cancel) : エラー応答 (非同期キャンセル)</li> <li>• XA : CALL : XA オペレーションの実行</li> <li>• XA : AUTOCOMMIT&lt;ON&gt; : XALocalCommitMode ( true ) の実行</li> <li>• XA : AUTOCOMMIT&lt;OFF&gt; : XALocalCommitMode ( false ) の実行</li> </ul>
<i>bb...bb</i>	付加情報が出力されます。オペレーション種別が、LIST ( xxxxx ) および EXT-CMD の場合に表示されます。

項目	説明	
	<i>cc...cc</i>	リターンコードが出力されます。
	<i>dd...dd</i>	詳細コードが出力されます。
	<i>XXX</i>	Cosminexus DABroker Library でのイベントが Put の場合は、送信したデータのサイズが出力されます。 Cosminexus DABroker Library でのイベントが Get の場合は、受け取ったデータのサイズが出力されます。

### 5.5.4 性能解析トレースの調査

性能解析トレースを調査することで、リクエストごとに処理進ちょくやボトルネックになっている処理を解析できます。また、セッショントレースを調査すると、セッションのライフサイクルが確認できます。

これらを基にして、トラブルが発生した個所を特定したり、ボトルネックになっている個所に適切に対処したりできます。必要に応じて確認してください。

性能解析トレースの出力内容については、「3. システムの処理性能の解析」を参照してください。

### 5.5.5 JavaVM のスレッドダンプの調査

JavaVM のスレッドダンプを調査すると、システムのデッドロックなどの java プログラムレベルでのトラブル要因の調査が容易になります。

出力される情報の種類は、J2EE サーバの起動時に指定しているオプションによって異なります。JavaVM の資料取得の設定については、マニュアル「Cosminexus システム構築ガイド」を参照してください。

スレッドダンプ情報の構成を、次に示します。

表 5-68 スレッドダンプ情報の構成

出力情報	内容
ヘッダ	日付, JavaVM バージョン情報, 起動コマンドラインを出力します。
システム設定	次の情報を出力します。 <ul style="list-style-type: none"> <li>• JDK の実行環境のインストール場所を表す Java ホームパス</li> <li>• JDK を構成するライブラリのインストールディレクトリを表す Java DLL パス</li> <li>• システムクラスパス</li> <li>• Java コマンドオプション</li> </ul>

## 5. トラブルシューティング

出力情報	内容
動作環境	次の情報を出力します。 <ul style="list-style-type: none"> <li>• ホスト名</li> <li>• OS バージョン</li> <li>• CPU 情報</li> <li>• リソース情報 (UNIX の場合)</li> </ul>
メモリ情報 (Windows の場合)	現在のメモリ使用量および各未使用サイズ情報を出力します。
Java ヒープ情報	Java ヒープの各世代のメモリ使用状況を出力します。
JavaVM 内部メモリマップ情報	JavaVM 自身の確保しているメモリの領域情報を出力します。
JavaVM 内部メモリサイズ情報	JavaVM 自身の確保しているメモリのサイズ情報を出力します。
アプリケーション環境	次の情報を出力します。 <ul style="list-style-type: none"> <li>• シグナルハンドラ</li> <li>• 環境変数</li> </ul>
ライブラリ情報	ローディングされているライブラリの情報を出力します。
スレッド情報 <スレッド 1> : <スレッド n>	スレッドごとにスレッド情報を出力します。
Java モニタダンプ	Java モニタオブジェクトの一覧を表示します。
Raw モニタダンプ	Java 内部で使われるモニタ情報の一覧を出力します。Row monitor が使用された場合だけ出力されます。
フッタ	スレッドダンプが終了した時刻を表示します。

それぞれの出力内容の詳細については、マニュアル「Cosminexus リファレンス 定義編」を参照してください。

また、J2EE アプリケーションやバッチアプリケーションでスローダウンやハングアップが発生したときに、スレッドダンプと性能解析トレースを対応づけることで、問題が発生した個所を調査できます。

スレッドダンプと性能解析トレースファイルの対応づけには、スレッドダンプに出力されたスレッド情報の nativeID (OS レベルのスレッド ID) と性能解析トレースファイルに出力されたスレッド ID を使用します。スレッドダンプから対応する性能解析トレースファイルを特定する手順について説明します。

1. スレッドダンプ、および性能解析トレースファイルを収集します。  
スレッドダンプの収集方法については、「5.4.5 JavaVM のスレッドダンプの取得」を参照してください。性能解析トレースファイルの収集方法については、「3.2.1 性能解析トレースファイルの収集方法」を参照してください。
2. 使用するスレッドダンプ、および性能解析トレースファイルを選びます。  
スレッドダンプと性能解析トレースファイルが出力された時刻を基に、調査に使用するスレッドダンプ、および性能解析トレースファイルを選びます。出力された時刻は、次の情報を参考にしてください。

## スレッドダンプ

ファイル名, およびファイルの末尾に出力される時刻  
ファイルの末尾に出力される時刻の例を次に示します。

```

:
:
Full thread dump completed.   Fri Jul 21 19:22:47 2006

```

## 性能解析トレースファイル

「Time」および「Time(msec/usec/nsec)」

性能解析トレースファイルの「Time」および「Time(msec/usec/nsec)」を次の図  
に示します。

	C	D	E	F	G	H	I
1	Thread(hashcode)	Trace	ProcessName	Event	Date	Time	Time(msec/usec/nsec)
2999	4736(7719486)	132	MyServer	0x8e04	2006/7/21	19:22:37	168/000/000
3000	4388(348051)	27	MyServer	0x8e05	2006/7/21	19:22:37	184/000/000
3001	4388(348051)	28	MyServer	0x8e06	2006/7/21	19:22:37	184/000/000
3002	4388(348051)	29	MyServer	0x8e05	2006/7/21	19:22:37	184/000/000

「Time」および「Time(msec/usec/nsec)」

## 3. スレッドダンプの「nid」(16進数)を10進数に変換します。

```

:
:
"VBJ ThreadPool Worker" daemon prio=5 jid=0x00054f93 tid=0x04cef380 nid=0x1124
in Object.wait() [0x0632f000..0x0632fd18]
stack=[0x06330000..0x062f5000..0x062f1000..0x062f0000]
[user cpu time=0ms, kernel cpu time=15ms] [blocked count=1, waited count=29]
at java.lang.Object.wait(Native Method)
:
:

```

1124 (16進数) = 4388 (10進数)

## 4. 性能解析トレースファイルの「Thread (hashcode)」のスレッドID (10進数)が手順3.で10進数に変換した「nid」と一致する行を探して、トレース情報を特定します。

	C	D	E	F	G	H	I
1	Thread(hashcode)	Trace	ProcessName	Event	Date	Time	Time(msec/usec/nsec)
2999	4736(7719486)	132	MyServer	0x8e04	2006/7/21	19:22:37	168/000/000
3000	4388(348051)	27	MyServer	0x8e05	2006/7/21	19:22:37	184/000/000
3001	4388(348051)	28	MyServer	0x8e06	2006/7/21	19:22:37	184/000/000
3002	4388(348051)	29	MyServer	0x8e05	2006/7/21	19:22:37	184/000/000

## 5.5.6 JavaVMのガーベージコレクションログの調査

ガーベージコレクションのログは, 日立JavaVMログファイルに出力されます。

## 5. トラブルシューティング

詳細は、「5.5.7 日立 JavaVM 固有のログの調査 (日立 JavaVM ログファイル)」を参照してください。

### 5.5.7 日立 JavaVM 固有のログの調査 (日立 JavaVM ログファイル)

日立 JavaVM ログファイルは、日立が標準の JavaVM に追加した拡張オプションを使用して出力されます。このログを取得するためには、対象の J2EE サーバを起動するときに必要なオプションを指定しておく必要があります。

#### (1) 日立 JavaVM ログファイルを出力するオプション

日立 JavaVM ログファイルを出力するオプションを次に示します。

`-XX:+HitachiOutOfMemoryStackTrace`

OutOfMemoryError 発生時にスタックトレースを出力するオプションです。なお、このオプションを指定した場合に同時に指定される、`-XX:+HitachiOutOfMemorySize` および `-XX:+HitachiOutOfMemoryCause` が指定された場合も、日立 JavaVM ログファイルが出力されます。

`-XX:+HitachiVerboseGC`

ガーベージコレクションが発生した時の拡張 `verbosegc` 情報を出力するオプションです。拡張 `verbosegc` 情報の取得については、「(2) 拡張 `verbosegc` 情報の取得」を参照してください。

`-XX:+HitachiJavaClassLibTrace`

`System.gc()`、`System.exit()`、`Runtime.exit()`、または `Runtime.halt()` のどれかの API を実行したときに、これらの API の呼び出しトレースを出力するオプションです。

なお、`-XX:+HitachiJavaClassLibTraceLineSize` オプションを指定している場合、出力されるトレースは、指定した文字数 (バイト数) 以内で出力されます。1 行の文字数が指定した値を超える場合は、「at」以降の文字列の前半部分が削除されて、指定した文字数分出力されます。

それぞれのオプションを指定した場合の出力内容の詳細については、マニュアル「Cosminexus リファレンス 定義編」を参照してください。

#### (2) 拡張 `verbosegc` 情報の取得

J2EE サーバの `usrconf.cfg` ファイルに、次の表に示すオプションを指定すると、拡張 `verbosegc` 情報を取得できます。拡張 `verbosegc` 情報からは、そのサーバで必要とする Java ヒープ領域サイズ、Perm 領域サイズなどを見積もるための情報が取得できます。なお、Java ログには、OutOfMemoryError 発生時のスタックトレースも出力されます。

表 5-69 拡張 verbosegc 情報の取得を指定するオプション

オプション	意味
-XX:+HitachiVerboseGC	拡張 verbosegc 情報を出力するかどうかを指定します。ガーベージコレクションの内部領域である Eden, Survivor, Tenured, Perm 種別ごとに情報を出力します。デフォルトでは出力されません。-XX:+HitachiVerboseGC と指定すると拡張 verbosegc 情報が出力され、-XX:-HitachiVerboseGC と指定すると拡張 verbosegc 情報は出力されません。
-XX:+HitachiVerboseGCPrintDate	拡張 verbosegc 情報を出力するログの各行の先頭に、ログを出力した日付を表示するかどうかを指定します。
-XX:+HitachiVerboseGCCpuTime	ガーベージコレクションの開始から終了までの間で、ガーベージコレクションの実行スレッドのユーザモードおよびカーネルモードに費やされた時間だけを表示するか、ガーベージコレクションの開始から終了までの実時間を表示するかを指定します。 ただし、HP-UX を使用している場合、この機能は使用できません。オプションを指定しても無効になります。
-XX:HitachiVerboseGCIntervalTime= < 時間間隔 >	-XX:+HitachiVerboseGC に対する出力時間の間隔を数値（単位：秒）で指定します。時間間隔のデフォルト値は 0（ガーベージコレクション発生のたびに出力）です。なお、時間間隔を指定すると、その時間間隔の間に発生したガーベージコレクション回数も表示されます。
-XX:+HitachiVerboseGCPrintCause	拡張 verbosegc 情報を出力するログに、ガーベージコレクションが発生した原因を表示するかどうかを指定します。
-XX:+HitachiCommaVerboseGC	拡張 verbosegc 情報を出力するログを CSV 形式で出力するかどうかを指定します。CSV 形式で出力する場合、拡張 verbosegc 情報の括弧「()」「[]」「<>」や、区切り「:」はすべて省略され、数値または文字列が「,」で区切られて出力されます。

ログファイルの出力形式と出力例を次に示します。

#### 出力形式

```
[id]<date>(Skip Full:full_count, Copy:copy_count)[gc_kind gc_info, gc_time secs][DefNew::Eden: eden_info][DefNew::Survivor: survivor_info][Tenured: tenured_info][Perm: perm_info][cause:cause_info][User: user_cpu secs][Sys: system_cpu secs]
```

#### 説明

- id : 日立 JavaVM ログファイル識別子
- date : 日時
- full\_count : Full GC をスキップした回数 (-XX:HitachiVerboseGCIntervalTime 指定した場合だけ)
- copy\_count : Copy GC をスキップした回数 (-XX:HitachiVerboseGCIntervalTime 指定した場合だけ)
- gc\_kind : ガーベージコレクション種別 (Full GC または GC)

## 5. トラブルシューティング

- gc\_info : ガーベージコレクション情報 (ガーベージコレクション前の領域長 -> ガーベージコレクション後の領域長 (領域サイズ))  
(例) 264K->0K(512K)
- gc\_time : ガーベージコレクション経過時間 (単位: 秒)
- eden\_info : Eden 情報
- survivor\_info : Survivor 情報
- tenured\_info : Tenured 情報
- perm\_info : Perm 情報
- cause\_info : ガーベージコレクションの原因
- user\_cpu secs : ガーベージコレクションスレッドがユーザモードに費やした CPU 時間 (単位: 秒)
- system\_cpu secs : ガーベージコレクションスレッドがカーネルモードに費やした CPU 時間 (単位: 秒)

### 出力例

```
[VGC]<Thu Apr 24 19:24:48 2003>(Skip Full:3, Copy:8)[GC 20641K->4505K(260224K), 0.0108370 secs][DefNew::Eden: 16255K->0K(16256K)][DefNew::Survivor: 351K->471K(1984K)][Tenured: 4033K->4033K(241984K)][Perm: 12071K->12071K(12288K)]
[VGC]<Thu Apr 24 19:35:43 2003>(Skip Full:0, Copy:10)[GC 20550K->4414K(260224K), 0.0086975 secs][DefNew::Eden: 16255K->0K(16256K)][DefNew::Survivor: 260K->381K(1984K)][Tenured: 4033K->4033K(241984K)][Perm: 12071K->12071K(12288K)][cause: System.gc][User: 0.0156250 secs][Sys: 0.0312500 secs]
```

## 5.5.8 JavaVM が出力するメッセージログの調査 (標準出力およびエラーレポートファイル)

JavaVM でクラッシュが発生した場合、JavaVM によってデバッグ情報が標準出力とエラーレポートファイルに出力されます。

エラーレポートファイルに出力されるのは、次の場合です。

JNI の中でシグナルが発生したとき

JavaVM で C ヒープ不足が発生したとき

JavaVM で予期しないシグナルが発生したとき

JavaVM で Internal Error (内部論理エラー) が発生したとき

ここでは、次の場合のメッセージログの出力内容について説明します。

表 5-70 JavaVM が出力するメッセージログ

メッセージの種類	出力先
JNI の中でシグナルが発生した場合、または JavaVM でのメッセージ	標準出力 エラーレポートファイル



メッセージの種類	出力先
C ヒープが不足した場合のメッセージ	標準出力 エラーレポートファイル
Internal Error が発生した場合のメッセージ	標準出力 エラーレポートファイル
スレッド作成に失敗した場合のメッセージ	標準出力

注 日立の JavaVM 独自の出力先または出力内容があります。

なお、スレッド作成に失敗した場合のメッセージは、標準出力だけに出力されます。

### (1) シグナルが発生した場合

シグナルが発生した場合、次に示す項目がログ出力されます。出力内容には、日立の JavaVM として拡張された内容が含まれます。

異常終了位置とシグナル種別

カレントスレッド情報

シグナル情報の格納先アドレス

シグナル情報

siginfo 情報 (UNIX の場合)

レジスタ情報

スタックの先頭から格納されている情報

命令コード情報

スタックトレース

スレッド情報

VM の状態

メモリ情報

Java ヒープの使用状況

ライブラリ

コマンドおよび VM パラメタ

環境変数

登録済みシグナルハンドラ

マシン情報

システム名, CPU, 実メモリ, および VM 情報

## 5. トラブルシューティング

### 時間情報

javatrace 起動コマンドのコマンドライン (UNIX の場合)

注 日立で拡張された出力内容です。

それぞれの出力内容について説明します。

#### (a) 異常終了とシグナル種別

異常終了時の状態に応じて、次のどちらかの内容が出力されます。この内容は、日立の JavaVM で拡張された出力内容です。

#### シグナルを検出した場合

```
#  
# An unexpected error has been detected by HotSpot Virtual Machine:  
#  
# <発生したシグナル名> (<シグナル番号>) at pc=<PCアドレス>, pid=<プロセスID>, tid=<ス  
# レッドID>  
#  
# Java VM: Java HotSpot(TM) Server VM (1.5.0_05-b05-CDK0700-20051129 mixed mode)  
# Problematic frame:  
# <種別コード> [<シグナルが発生したライブラリ名> + <オフセット>]  
#
```

注 シグナルが発生した関数名が取り出せた場合、「<シグナルが発生したライブラリ名> + <オフセット>」に続いて、その関数名とオフセットが表示されることがあります。

#### 内部論理エラーが発生した場合

```
#  
# An unexpected error has been detected by HotSpot Virtual Machine:  
#  
# Internal Error (5448524541440E4350500D8F), pid=<プロセスID>, tid=<スレッドID>  
#  
# Java VM: Java HotSpot(TM) Server VM (1.5.0_05-b05-CDK0700-20051129 mixed mode)  
#  
# Error: 内部論理エラーメッセージ
```

#### (b) カレントスレッド情報

次のどちらかの内容が出力されます。

```
Current thread (<アドレス>): <スレッド名> "<スレッド名称>" [_<状態>, id=<スレッドID>]
```

```
Current thread (<アドレス>): <スレッド名> [_id=<スレッドID>]
```

## (c) シグナル情報の格納先アドレス

次の内容が出力されます。この内容は、日立の JavaVM で拡張された出力内容です。

```
siginfo address: <アドレス>, context address: <アドレス>
```

## (d) シグナル情報

次の内容が出力されます。

Windows の場合

EXCEPTION\_ACCESS\_VIOLATION(読み込み違反)

```
siginfo: ExceptionCode=<シグナル番号>, reading address <アドレス>
```

EXCEPTION\_ACCESS\_VIOLATION(書き込み違反)

```
siginfo: ExceptionCode=<シグナル番号>, writing address <アドレス>
```

EXCEPTION\_ACCESS\_VIOLATION(その他)

```
siginfo: ExceptionCode=<シグナル番号>, ExceptionInformation=<付加情報>
```

EXCEPTION\_ACCESS\_VIOLATION 以外

```
siginfo: ExceptionCode=<シグナル番号>, ExceptionInformation=<付加情報1> <付加情報2> . . .
```

UNIX の場合

```
siginfo: si_signo=<シグナル番号>, si_errno=<番号>, si_code=<番号>, si_addr=<アドレス>
```

## (e) siginfo 情報 (UNIX の場合)

次の内容が出力されます。この内容は、日立の JavaVM で拡張された出力内容です。

## 5. トラブルシューティング

```
siginfo structure dump (location: <siginfoのアドレス>)  
<siginfoのアドレス> <siginfoのアドレス> <siginfoのアドレス> <siginfoのアドレス>  
...  
<siginfoのアドレス> <siginfoのアドレス> <siginfoのアドレス> <siginfoのアドレス>
```

注 <siginfo のアドレス> は 16 進数で出力されます。

### (f) レジスタ情報

次の内容が出力されます。ただし、内部論理エラーの場合は出力されません。

```
Registers:<レジスタ情報>  
:
```

注 UNIX の場合、BSP レジスタの値がデバッガ (gdb) の値と異なって出力されます。これは、デバッガでは BSP の指すバッキングストア領域の内容を出力しており、BSP の指す位置を修正しているためです。

### (g) スタックの先頭から格納されている情報

次の内容が出力されます。ただし、内部論理エラーの場合は出力されません。

```
Top of Stack: (sp=<スタックポインタのアドレス>)  
<アドレス>: <格納されている内容>  
:
```

注 <格納されている内容> は 16 進数で出力されます。

### (h) 命令コード情報

次の内容が出力されます。ただし、内部論理エラーの場合は出力されません。

```
Instructions: (pc=<プログラムカウンタのアドレス>)  
<アドレス>: <命令コード>  
:
```

注 <命令コード> は 16 進数で出力されます。

### (i) スタックトレース

次の内容が出力されます。ただし、Current thread が JavaThread 以外の場合は出力されません。

```
Java frames: (J=compiled Java code, j=interpreted, Vv=VM code)
<スタックトレース>
:
```

## (j) スレッド情報

次の内容が出力されます。

```
Java Threads: ( => current thread )
<アドレス>  JavaThread  "<スレッド名称>"  [<状態>, id=<スレッドID>]
:
=><アドレス>  JavaThread  "<スレッド名称>"
Other Threads:
<アドレス>  <スレッド名> [id=<スレッドID>]
:
```

## (k) VM の状態

次の内容が出力されます。

```
VM state:<現在の状態>
VM Mutex/Monitor currently owned by a thread: <mutexs/moniter>
```

注 この情報に続いて、ロック情報が出力される場合があります。

## (l) メモリ情報

次の内容が出力されます。この内容は、日立の JavaVM で拡張された出力内容です。

```
Memory :
<メモリ確保関数>:address<開始アドレス> - <終了アドレス>(size:<サイズ>)
:
    Heap Size:<確保しているメモリサイズ>
    Alloc Size:<使用中のメモリサイズ>
    Free Size:<未使用のメモリサイズ>
```

<メモリ確保関数> は、mmap() または malloc() のどちらかです。アドレスは 16 進数で表示されます。

各種メモリサイズの単位はバイトです。

## (m) Java ヒープの使用状況

次の内容が出力されます。この内容は、日立の JavaVM で拡張された出力内容です。

## 5. トラブルシューティング

```
Heap
<Javaヒープ情報>
```

### (n) ライブラリ

次の内容に続いて、ローディングされているライブラリの一覧が出力されます。

```
Dynamic libraries:
<ライブラリ>
:
```

### (o) コマンドおよびVM パラメタ

次の内容が出力されます。この内容は、日立のJavaVMで拡張された出力内容です。

```
Command : <コマンドライン>

Java Home Dir   : <JDK実行環境インストールディレクトリ>
Java DLL Dir    : <JDKのライブラリインストールディレクトリ>
Sys Classpath  : <システムクラスパス>
User Args      :
<コマンドオプション1>
<コマンドオプション2>
:
```

### (p) 環境変数

次の内容が出力されます。

```
Environment Variables:
<環境変数=値>
:
```

### (q) 登録済みシグナルハンドラ

次の内容が出力されます。

```
Signal Handlers:
<シグナル種別> :
  [<シグナルハンドラアドレス>], sa_mask[0]=<シグナルマスク>, sa_flags=<特殊フラグ>
:
Changed Signal Handlers -
<シグナル種別> : [<シグナルハンドラアドレス>], sa_mask[0]=<シグナルマスク>, sa_flags=<
特殊フラグ>
:
```

出力内容の意味は次のとおりです。

- <シグナル種別> : /usr/include/sys/signal.h に定義されているシグナル名です。
- <シグナルハンドラアドレス> : シグナルハンドラのアドレスが 16 進で出力された値です。「ライブラリ名+オフセット」という形式で表示されることもあります。
- <シグナルマスク> : sigaction() で取り出せる構造の sa\_mask フィールド値が 16 進で出力された値です。
- <特殊フラグ> : sigaction() で取り出せる構造の sa\_flags フィールド値が 16 進で出力された値です。

#### (r) マシン情報

次の内容が出力されます。この内容は、日立の JavaVM で拡張された出力内容です。

```
Host: <ホスト名>:<IPアドレス>
```

注 <IP アドレス> には複数の IP アドレスが表示されることがあります。

#### (s) システム名, CPU, 実メモリ, および VM 情報

次の内容が出力されます。

Windows の場合

```
OS: <OSバージョン>
CPU: <利用可能CPU数>, <CPU種別>
Memory: <実メモリ情報>
vm_info: <VM情報>
```

UNIX の場合

```
OS: <OSバージョン>
uname: <uname出力>
libc: <libcのバージョン番号> (出力されない場合があります)
rlimit: <リミット値>
load average: <ロードアベレージ>
CPU: <利用可能CPU数>, <CPU種別>
Memory: <実メモリ情報>
vm_info: <VM情報>
```

#### (t) 時間情報

次の内容が出力されます。この内容は、日立の JavaVM で拡張された出力内容です。

## 5. トラブルシューティング

```
Local Time = <実行日付>
Elapsed Time = <実行時間>
```

実行日付には出力内容の例を次に示します。

例 : 「Wed Aug 25 14:55:04 2004」

(u) javatrace 起動コマンドのコマンドライン (UNIX の場合)

次の内容が出力されます。この内容は、日立の JavaVM で拡張された出力内容です。

```
# You can get further information from javatrace.log file generated
# by using javatrace command.
# usage: javatrace core-file-name loadmodule-name [out-file-name]
# [-l(library-name)...]
# Please use javatrace command as follows and submit a bug report
# to Hitachi with javatrace.log file:
#[<インストールディレクトリ>/jre/bin/javatrace <coreファイル> <ロードモジュール>]
```

### (2) C ヒープが不足した場合

C ヒープが不足した場合、次の順序でメッセージ出力およびダンプ出力、または core ダンプの生成が実行されます。

1. C ヒープ不足を示すメッセージログが、エラーレポートファイルおよび標準出力に出力されます。
2. 1. の実行中にメモリ不足が発生した場合、簡易メッセージが標準出力に出力されず。
3. Windows の場合、ワトソンログが起動され、ダンプ出力が実行されます。UNIX の場合、簡易メッセージの出力処理中にさらにメモリ不足が発生したときに、メッセージおよびエラーログファイルの出力処理を中止して、core ダンプを生成します。

それぞれの出力形式を次に示します。

(a) C ヒープ不足を示すメッセージログの出力内容

C ヒープ不足を示すメッセージログの出力形式を次に示します。この形式は、エラーレポートファイルと標準出力で共通です。

- Windows の場合



```

Exception in thread <ThreadName> java.lang.OutOfMemoryError:requested <n> bytes
[ for <message>].

Memory Status
-----
Memory in use      :使用率%
Physical memory   :空きメモリサイズ /総メモリサイズ free
Virtual memory    :空きメモリサイズ/総メモリサイズ free
Paging file       :空き容量/総容量 free

Heap Status
-----
<Java ヒープ情報>
-----
Stack Trace
-----
<スタックトレース>
Insufficient memory for malloc. JVM generates core file.
-----

```

- AIX , HP-UX , または Linux の場合

```

Exception in thread <ThreadName> java.lang.OutOfMemoryError: requested <n>
bytes [for <message>].

Memory Status
-----
maximum size of data segment
  soft(current) limit :getrlimit(RLIMIT_DATA)で取得したソフトリミット値 kbytes
(16進数)
  hard limit          :getrlimit(RLIMIT_DATA)で取得したハードリミット値 kbytes
(16進数)
current end of the heap :sbrk(0)によって取得した値
JVM allocation size by malloc :JavaVMが割り当てたメモリサイズ kbytes (16進数)
malloc information
  total space in arena           :mallinfo.arenaの値
  number of ordinary blocks     :mallinfo.ordblksの値
  number of small blocks        :mallinfo.smlblksの値
  number of holding blocks      :mallinfo.hblksの値
  space in holding block headers :mallinfo.hblkhdの値
  space in small blocks in use  :mallinfo.usmlblksの値
  space in free small blocks    :mallinfo.fsmlblksの値
  space in ordinary blocks in use :mallinfo.uordblksの値
  space in free ordinary blocks  :mallinfo.fordblksの値
  cost of enabling keep option  :mallinfo.keepcostの値

Heap Status
-----
<Java ヒープ情報>
-----
Stack Trace
-----
<スタックトレース>
-----

```

- Solaris の場合

## 5. トラブルシューティング

```
Exception in thread <ThreadName> java.lang.OutOfMemoryError: requested <n>
bytes [for <message>].
```

```
Heap Status
-----
<Java ヒープ情報>
-----
Stack Trace
-----
<スタックトレース>
-----
```

このようなメッセージが出力された場合は、C ヒープを減らすなど、適切な対策をしてください。

出力項目について次に示します。

表 5-71 C ヒープが不足した場合のメッセージログの出力項目

出力項目	説明
ThreadName	Thread#getName() メソッドで取り出せるスレッド名称が出力されます。
n	メモリ確保要求サイズが出力されます。
message	保守員による調査に必要な内部メッセージが出力されます。出力されない場合もあります。
Java ヒープ情報	Java ヒープの使用状況が出力されます。
スタックトレース	メモリ不足の発生したスレッドが Java コードを実行しているスレッドである場合に、スタックトレースが出力されます。コンパイル処理のような JavaVM の内部処理を実行するスレッドでメモリ不足が発生した場合には出力されません。

### (b) メモリ不足を示すメッセージの出力内容

C ヒープ不足を示すメッセージログが出力されている間にさらにメモリ不足が発生した場合は、処理の続行ができません。この場合は、次の形式の簡易メッセージが標準出力に出力されます。

```
java.lang.OutOfMemoryError:requested <n> bytes for <message>
```

出力項目について次に示します。

表 5-72 メモリ不足が発生した場合の簡易メッセージの出力項目

出力項目	説明
n	メモリ確保要求サイズが出力されます。

出力項目	説明
message	保守員による調査に必要な内部メッセージが出力されます。

(c) ワトソンログが正しく出力されたかどうかを示すメッセージの出力内容 (Windows の場合)

簡易メッセージの出力後は、ワトソンログが起動され、ダンプ出力が実行されます。ただし、ワトソンログが正常に出力できない場合がありますので、標準出力に出力される処理結果のメッセージを確認してください。

次のメッセージは、ワトソンログの起動に失敗した場合に出力されます。

```
WatosonLog process generated.But couldn't generate dump file,because of resource shortage.
```

次のメッセージは、ワトソンログの起動後、ダンプ出力に失敗した場合に出力されます。

```
Couldn't generate dump file,because of resource shortage.
```

これらの処理に成功した場合は、次に示すメッセージが出力されます。ただし、この場合もワトソンログが正常にダンプ出力しない場合があります。このメッセージが出力された場合も、実際にダンプが出力されているかどうかを確認してください。

```
Although Dr.Watson process finished successfully,dump file isn't always generated.
Check your directory where dump file is stored.
```

(d) core ダンプの生成を示すメッセージの出力内容

簡易メッセージの出力処理中にさらにメモリ不足が発生した場合、メッセージおよびエラーログファイルの出力処理を中止して、core ダンプを生成します。core ダンプが生成されると、次の形式のメッセージが標準出力に出力されます。

```
Can't create logs because of memory shortage.
Insufficient memory for malloc. JVM generates core file
```

### (3) Internal Error が発生した場合

JavaVM 内部の論理エラーである Internal Error が発生した場合、次の情報が出力されます。

- 異常終了位置とシグナル種別

## 5. トラブルシューティング

- カレントスレッド情報
- シグナル情報の格納先アドレス
- スレッド情報
- VM の状態
- メモリ情報
- ヒープ情報
- ライブラリ
- コマンド・VM パラメタ
- 環境変数
- 登録済みシグナルハンドラ
- マシン情報
- システム名, CPU, 実メモリ, および VM 情報
- 時間情報
- javatrace 起動コマンドのコマンドライン (UNIX の場合)

注 日立で拡張された出力内容です。

それぞれの情報の出力形式については、「(1) シグナルが発生した場合」を参照してください。

### (4) スレッド作成に失敗した場合

メモリ不足 (OutOfMemoryError) などが発生して新しくスレッドを作成できなかった場合, その時点でのスレッド数が, 標準出力に出力されます。ここには, 作成に失敗したスレッド数も含まれます。

メモリ不足などによってスレッドの作成に失敗した場合の出力例を次に示します。

```
java.lang.OutOfMemoryError:unable to create new native thread.1200 threads  
exist.  
...
```

JavaVM 起動時にスレッド作成に失敗した場合の出力例を次に示します。

```
Error occurred during initialization of VM  
Could not create thread for VM:VM Thread.5 threads exist.
```

## 5.5.9 OS の状態情報および OS のログの調査

取得した OS の状態・ログ情報で異常な傾向の有無を確認します。具体的な資料の見方については OS 付属のマニュアルなどを参照してください。

## 5.5.10 スタックトレースの調査

ここでは、スタックトレースに出力される情報のうち、日立の JavaVM で拡張された内容について説明します。

サーバやアプリケーションでトラブルが発生した場合、トラブル発生までのスタックトレースの内容を確認することで、トラブル発生の要因を調査できます。

スタックトレースは、次のどれかのタイミングで出力されます。

J2EE サーバまたは J2EE アプリケーションで例外が発生した場合

バッチサーバまたはバッチアプリケーションで例外が発生した場合

JavaVM のスレッドダンプが出力された場合

日立の JavaVM では、サーバを起動するときのオプションの指定によって、スタックトレースに Java メソッドのローカル変数についての情報を出力できます。例外発生時に Java メソッドに定義されていたローカル変数の情報は、トラブルの要因を分析するために有効です。

なお、ここで対象にするローカル変数とは、メソッドに渡される引数およびインスタンスメソッドでの呼び出しの対象になるオブジェクト (this) のことです。ローカル変数情報には、これらのローカル変数の変数名、型名、およびローカル変数の値が出力されます。なお、型名は、基本形名、クラス名 (インタフェース名を含みます) または配列形名のことです。

スタックトレースにローカル変数情報を出力するためのオプションを、次の表に示します。JavaVM の資料取得の設定については、マニュアル「Cosminexus システム構築ガイド」を参照してください。

表 5-73 スタックトレースにローカル変数情報を出力するためのオプション

起動オプション	スタックトレースを出力する タイミング	同時に指定できるオプション
-XX:+HitachiLocalsInThrowable	サーバまたはアプリケーション内で例外が発生したとき	<ul style="list-style-type: none"> <li>• -XX:MaxJavaStackTraceDepth=&lt;最大の深さ&gt;</li> <li>• -XX:+HitachiLocalsSimpleFormat</li> <li>• -XX:+HitachiTrueTypeInLocals</li> <li>• -XX:HitachiCallToString=&lt;適用範囲指定&gt;</li> </ul>
-XX:+HitachiLocalsInStackTrace	JavaVM のスレッドダンプを出力したとき	<ul style="list-style-type: none"> <li>• -XX:MaxJavaStackTraceDepth=&lt;最大の深さ&gt;</li> <li>• -XX:+HitachiLocalsSimpleFormat</li> <li>• -XX:+HitachiTrueTypeInLocals</li> </ul>

注 ただし、発生した例外が「java.lang.StackOverflowError」または「java.lang.OutOfMemoryError」の場合、スタックトレースにローカル変数は出力されません。

## 5. トラブルシューティング

次に、それぞれのオプションを指定した場合に出力される内容について、例を基に説明します。それぞれのオプションを指定した時に出力される項目の詳細については、マニュアル「Cosminexus リファレンス 定義編」を参照してください。

### (1) -XX:+HitachiLocalsInThrowable オプションが指定されている場合

java.lang.Throwable.printStackTrace メソッドで出力されるスタックトレース情報の 1 スタックフレーム情報ごとに、そのスタックフレームに対応するメソッド内のローカル変数情報が挿入されて出力されます。

#### (a) 標準の形式および簡易出力フォーマットでの出力例

ローカル変数を出力するための機能として、-XX:+HitachiLocalsInThrowable オプションだけを指定している場合の出力例です。

Java プログラムの例と、それに対するスタックトレース内のローカル変数情報の出力例を示します。

#### Java プログラムの例 1

```
class Example1 {
    public static void main(String[] args) {
        Example1 e1 = new Example1();
        Object obj = new Object();
        e1.method(1, 'Q', obj); // e1.methodメソッドを実行します (5行目)。
    }

    void method(int l1, char l2, Object l3) {
        float l4 = 4.0f;
        boolean l5 = true;
        double l6 = Double.MAX_VALUE;
        Object[] l7 = new Object[10];

        try {
            <例外発生!> // methodメソッドの処理内で例外が発生した場合の処理です (15行目)。
        } catch (Exception e) {
            e.printStackTrace(); // スタックトレース情報を出力します (17行目)。
        }
    }
}
```

出力例を次に示します。

この例は、Java プログラムの例 1 の 5 行目で実行した e1.method メソッドの処理内で例外が発生したときに、17 行目の e.printStackTrace メソッドによって出力されるスタックトレース情報です。

図 5-7 Java プログラムの例 1 に対するローカル変数情報の出力例（-g オプションまたは -g:vars オプションを指定して作成された class ファイルの場合）

at Example1.method(Example1.java:15)	1.
locals:	
name: this	
type: Example1	2.
value: <0x922f42d0>	
name: l1 [arg1]	
type: int	
value: 1	
name: l2 [arg2]	
type: char	3.
value: 'Q'	
name: l3 [arg3]	
type: java.lang.Object	
value: <0xaf112f08>	
name: l4	
type: float	
value: 4.000000	
name: l5	
type: boolean	
value: true	
name: l6	
type: double	4.
value: 1.79769E+308	
name: l7	
type: java.lang.Object[]	
value: <0x922f42d8>	
at Example1.main(Example1.java:5)	
locals:	
...	

出力内容について説明します。

1. スタックトレースを出力する処理を実行したメソッドの情報が出力されます。この例では、Java プログラム 1 の 15 行目で例外が発生したときのスタックトレース情報が出力されることを示しています。
2. ローカル変数情報として、インスタンスメソッドの呼び出しの対象になるオブジェクトについての情報が出力されます。この例の場合は、Java プログラムの例 1 の 3 行目で作成した Example1 クラスのオブジェクトのクラス名とアドレスが出力されます。
3. ローカル変数情報として、method メソッドの引数として指定されたローカル変数の値についての情報が出力されます。変数名の後ろの [arg\*] は、何番目のメソッド引数かを示す情報です。Java プログラムの例 1 の 5 行目で e1.method メソッドを実行したときに指定された値が出力されます。  
なお、l1 および l2 は基本型（int 型および char 型）の変数なので実際の値が出力されません。l3 は java.lang.Object クラス型の変数なので、アドレスで出力されます。
4. ローカル変数情報として、method メソッド内のローカル変数のうち、メソッド引数

## 5. トラブルシューティング

として指定した以外のローカル変数の値についての情報が出力されません。

なお、14 ~ 16 は基本型 (float 型, boolean 型および double 型) の変数なので実際の値が出力されます。17 は java.lang.Object クラス型の変数なので, アドレスで出力されます。

次に, `-XX:+HitachiLocalsSimpleFormat` オプションを指定した場合の, 簡易出力フォーマットでの出力例を次に示します。なお, 出力内容の説明は, 標準の形式と同じです。

図 5-8 `-XX:+HitachiLocalsSimpleFormat` オプションを指定した場合の出力例

```
at Example1.method(Example1.java:15)
  locals:
    (Example1) this = <0x922f42d0>
    (int) i1 [arg1] = 1
    (char) i2 [arg2] = 'Q'
    (java.lang.Object) i3 [arg3] = <0xaf112f08>
    (float) i4 = 4.000000
    (boolean) i5 = true
    (double) i6 = 1.79769E+308
    (java.lang.Object[]) i7 = <0x922f42d8>
at Example1.main(Example1.java:5)
  locals:
  ...
```

1. (Example1) this = <0x922f42d0>  
2. (int) i1 [arg1] = 1  
3. (char) i2 [arg2] = 'Q'  
4. (java.lang.Object) i3 [arg3] = <0xaf112f08>  
(float) i4 = 4.000000  
(boolean) i5 = true  
(double) i6 = 1.79769E+308  
(java.lang.Object[]) i7 = <0x922f42d8>

また, javac コマンド実行時に, `-g` オプションまたは `-g:vars` オプションを指定しなかった場合, ローカル変数情報がないため, 出力内容が次のように制限されます。これらは, native メソッドを実行した場合も同様です。

出力できるローカル変数が, メソッドに渡される引数とインスタンスメソッド呼び出し対象オブジェクト (this) だけになります。

メソッドに渡される引数は, 変数名が出力されません。引数番号だけが出力されます。

native メソッドの場合, ローカル変数の値として native メソッドを呼び出した時点での値が出力されます。native メソッドの実行結果を反映した出力結果にはなりません。

ローカル変数情報がない場合の, Java プログラムの例 1 に対する出力例を次に示します。ここでは, 簡易出力フォーマットでの出力例を示します。

図 5-9 ローカル変数情報がない場合の出力例 (簡易出力フォーマット)

```
at Example1.method(Example1.java:15)
  locals:
    (Example1) this = <0x922f42d0>
    (int) [arg1] = 1
    (char) [arg2] = 'Q'
    (java.lang.Object) [arg3] = <0xaf112f08>
at Example1.main(Example1.java:5)
  locals:
  ...
```

11~13のローカル変数情報。



図 5-8 に比べて、次の違いがあります。

メソッドに渡される引数 (11 ~ 13) の変数名が出力されません。

メソッド呼び出し時点の値が出力されるため、メソッド内で設定されたローカル変数についての情報 (14 ~ 17) が出力されません。

(b) クラスまたは配列型の変数を文字列として出力する場合の出力例

出力するローカル変数がクラスまたは配列型の場合、アドレスだけの値表現ではトラブルシューティングに必要な情報が取得できない場合があります。このとき、`-XX:HitachiCallToString` オプションの指定をしておく、クラスまたは配列型の変数の値を文字列で取得できます。オプションには、適用範囲として、`minimal` または `full` が指定できます。

`-XX:HitachiCallToString=minimal` オプションが指定されている場合は、`java.lang` パッケージ内のクラスのうち、`String`、`StringBuffer`、`Boolean`、`Byte`、`Character`、`Short`、`Integer`、`Long`、`Float`、または `Double` が対象になります。

`-XX:HitachiCallToString=full` オプションが指定されている場合は、すべてのクラスが対象になります。

次に、Java プログラムの例と、`-XX:HitachiCallToString` オプションが指定されている場合の出力例を示します。なお、ここでは簡易出力フォーマットで示します。

Java プログラムの例 2

```
class Example2 {
    public static void main(String[] args) {
        Example2 e2 = new Example2();
        e2.method();// e2.methodメソッドを実行します (4行目)。
    }

    void method() {
        String l1 = "local 1";
        StringBuffer l2 = new StringBuffer(l1);
        l2.append(" + local 2");
        Boolean l3 = new Boolean(false);
        Character l4 = new Character('X');
        Long l5 = new Long(Long.MIN_VALUE);
        Object l6 = new Thread();
        Object[] l7 = new Thread[10];

        try {
            <例外発生!> // methodメソッドの処理内で例外が発生した場合の処理です (18行目)。
        } catch (Exception e) {
            e.printStackTrace(); // スタックトレース情報を出力します (20行目)。
        }
    }

    public String toString() {
        return "I am an Example2 instance.";
    }
}
```

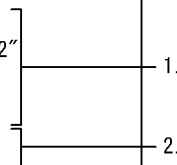
`-XX:HitachiCallToString=minimal` オプションを指定した場合の出力例を次に示します。

## 5. トラブルシューティング

この例は、Java プログラムの例 1 の 4 行目で実行した e2.method メソッドの処理内で例外が発生したときに、20 行目の e.printStackTrace メソッドによって出力されるスタックトレース情報です。

図 5-10 -XX:HitachiCallToString=minimal オプションを指定した場合の出力例（簡易出力フォーマット）

```
at Example2.method(Example2.java:18)
  locals:
    (Example2) this = <0xaa07db58>
    (java.lang.String) l1 = <0xae173a28> "local 1"
    (java.lang.StringBuffer) l2 = <0xaa07dca0> "local 1 + local 2"
    (java.lang.Boolean) l3 = <0xaa07de18> "false"
    (java.lang.Character) l4 = <0xaa07df68> "X"
    (java.lang.Long) l5 = <0xaa07e078> "-9223372036854775808"
    (java.lang.Object) l6 = <0xaa07e1a8>
    (java.lang.Object[]) l7 = <0xaa07e298>
at Example2.main(Example2.java:4)
  locals:
  ...
```



出力内容について説明します。

1. クラス型のローカル変数のうち、文字列を出力する適用対象のクラスの型のローカル変数情報です。アドレスに続いて、文字列に変換した値が出力されます。
2. クラス型のローカル変数のうち、文字列を出力する適用対象外のクラスの型のローカル変数情報です。アドレスだけが出力されます。

次に、-XX:HitachiCallToString=full オプションを指定した場合の出力例を次に示します。

図 5-11 -XX:HitachiCallToString=full オプションを指定した場合の出力例（簡易出力フォーマット）

```
at Example2.method(Example2.java:18)
  locals:
    (Example2) this = <0xaa07db58> "I am an Example2 instance."
    (java.lang.String) l1 = <0xae173a28> "local 1"
    (java.lang.StringBuffer) l2 = <0xaa07dca0> "local 1 + local 2"
    (java.lang.Boolean) l3 = <0xaa07de18> "false"
    (java.lang.Character) l4 = <0xaa07df68> "X"
    (java.lang.Long) l5 = <0xaa07e078> "-9223372036854775808"
    (java.lang.Object) l6 = <0xaa07e1a8> "Thread[Thread-0.5.main]"
    (java.lang.Object[]) l7 = <0xaa07e298> "[Ljava.lang.Thread;@26e431"
at Example2.main(Example2.java:4)
  locals:
  ...
```

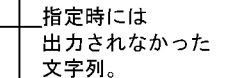


図 5-10 に比べて、次の違いがあります。

minimal を指定した場合に適用対象外のクラスだった java.lang.Object クラスおよび java.lang.Object クラスの配列のローカル変数 (l6 および l7) に対しても、文字列が

出力されません。

ただし、次の場合は、文字列は出力されないで、アドレスだけが出力されます。

ローカル変数の値が null だった場合

文字列で出力するときに再度例外が発生して、正常に値が得られなかった場合

### ! 注意事項

ローカル変数情報で出力するオブジェクトは、ほかのスレッドで並行して操作していることがあります。このため、このオプションを指定して出力した文字列は、実際に例外が発生したときのオブジェクトに対応する情報とは異なっているおそれがあります。

#### (c) クラスまたは配列型の変数の実際の型名を出力する場合の出力例

出力するローカル変数がクラスまたは配列型の場合、変数の型名と実際に代入されている値の型名が異なる場合があります。例えば、クラスの継承関係やインタフェースの実装関係によっては、異なる型名の値が変数に代入されます。

この場合に、-XX:+HitachiTrueTypeInLocals オプションの指定によって、クラスまたは配列型の変数に実際に代入されている値の型名を追記して取得できます。

取得した文字列は、値表現の末尾に半角スペース 1 文字分が追記され、() で囲まれて出力されます。このとき、出力文字列長に制限はありません。

次に、-XX:+HitachiTrueTypeInLocals オプションが指定されている場合の出力例を示します。ここでは簡易出力フォーマットで示します。実行するプログラムは、Java プログラムの例 2 です。また、この例は、-XX:HitachiCallToString=minimal オプションが指定されている場合の例です。

図 5-12 -XX:+HitachiTrueTypeInLocals オプションを指定した場合の出力例 (簡易出力フォーマット)

```

at Example2.method(Example2.java:18)
  locals:
    (Example2) this = <0xaa07db58> (Example2)
    (java.lang.String) l1 = <0xae173a28> "local 1" (java.lang.String)
    (java.lang.StringBuffer) l2 = <0xaa07dca0> "local 1 + local 2" (java.lang.StringBuffer)
    (java.lang.Boolean) l3 = <0xaa07de18> "false" (java.lang.Boolean)
    (java.lang.Character) l4 = <0xaa07df68> "X" (java.lang.Character)
    (java.lang.Long) l5 = <0xaa07e078> "-9223372036854775808" (java.lang.Long)
    (java.lang.Object) l6 = <0xaa07e1a8> (java.lang.Thread)
    (java.lang.Object[]) l7 = <0xaa07e298> (java.lang.Thread[]) ] 1.
at Example2.main(Example2.java:4)
  locals:
  ...

```

出力内容について説明します。

## 5. トラブルシューティング

1. 変数の型名は `java.lang.Object` クラスまたは `java.lang.Object` クラスの配列ですが、実際に代入されている値の型名は `java.lang.Thread` クラスおよび `java.lang.Thread` クラスの配列であることが出力されています。

なお、`-XX:+HitachiTrueTypeInLocals` オプションは、`-XX:HitachiCallToString=full` オプションとも同時に指定できます。

### (2) `-XX:+HitachiLocalsInStackTrace` オプションが指定されている場合

スレッドダンプ内のスタックトレース情報の1スタックフレーム情報ごとに、そのスタックフレームに対応するメソッド内のローカル変数情報が挿入されて、出力されます。

出力形式および出力内容は、`-XX:+HitachiLocalsInThrowable` が指定されている場合に標準出力に出力される内容と同じです。

ただし、`-XX:+HitachiLocalsInStackTrace` オプションでは、次のオプションの指定は無効になります。

`-XX:HitachiCallToString` オプション

Java プログラムの例と、それに対するスタックトレース内のローカル変数情報の出力例を示します。

#### Java プログラムの例 3

```
class Example3 {
    public static void main(String[] args) {
        Example3 e3 = new Example3();
        e3.method();
    }

    synchronized void method() {
        int l1 = 1;
        float l2 = 2.0f;
        String l3 = "local 3";
        Character l4 = new Character('X');
        Object l5 = new Thread();
        Object[] l6 = new Thread[10];

        <ここでスレッドダンプ出力!>
    }
}
```

出力例を次に示します。この例は、次の場合の例です。

- `-g` オプションまたは `-g:vars` オプションを指定して作成された class ファイルである
- `-XX:+HitachiLocalSimpleFormat` オプションを指定している
- `-XX:+HitachiTrueTypeInLocals` オプションを指定している

図 5-13 Java プログラムの例 3 に対するローカル変数情報の出力例

```
...
"main" prio=1 tid=0xb6e88d20 nid=0xb7492080 runnable [bffff000..bffff474]
  at Example3.method(Example3.java:15)
    - locked <0xab040550> (a Example3)
      locals:
        (Example3) this = <0xab040550> (Example3)
        (int) i1 = 1
        (float) f2 = 2.000000
        (java.lang.String) s3 = <0xaf112cc0> (java.lang.String)
        (java.lang.Character) c4 = <0xab040698> (java.lang.Character)
        (java.lang.Object) o5 = <0xab0407c8> (java.lang.Thread)
        (java.lang.Object[]) a6 = <0xab0408b8> (java.lang.Thread[])
  at Example3.main(Example3.java:4)
    locals:
      (java.lang.String[]) args [arg1] = <0xab040540> (java.lang.String[])
      (Example3) e3 = <0xab040550> (Example3)
...
```

## 5.6 トラブルへの対処と回復

---

この節では、次のようなトラブルが発生した場合の対処方法および回復方法について説明します。

構成ソフトウェアのプロセス（論理サーバ）が異常終了した場合

J2EE アプリケーションの強制停止に失敗した場合

セッションフェイルオーバ機能を使用時にトラブルが発生した場合

データベースと接続中にトラブルが発生した場合

JavaVM が異常終了した場合

JP1 と連携したシステムでトラブルが発生した場合

1:1 系切り替えシステムでトラブルが発生した場合

N:1 リカバリシステムでトラブルが発生した場合

EJB クライアントでトラブルが発生した場合

### 5.6.1 構成ソフトウェアのプロセス（論理サーバ）が異常終了した場合

ここでは、Cosminexus の構成ソフトウェアのプロセスが異常終了したときに、システムを再起動する方法について説明します。

Cosminexus システムでは、Cosminexus の構成ソフトウェアのプロセス（論理サーバ）が異常終了したとき、起動順序が設定されていれば Management Server によって自動的に再起動されます。Management Server では、構成ソフトウェアのプロセスを論理サーバとして管理しています。

手動で再起動を実行する場合、異常終了した構成ソフトウェアの起動順序に依存関係があるときは、前提となる構成ソフトウェアが起動していることを確認してから再起動する必要があります。前提となる構成ソフトウェアは、システムの運用形態によって異なります。以降で、構成ソフトウェアの起動順序の依存関係、および構成ソフトウェアの再起動方法について説明します。

---

#### ポイント

CTM を利用したシステムでは、CTM によって、すぐに再起動すれば、クライアントにエラーが返却される前にシステムを回復するように設定できます。ただし、リクエストキューの最大登録数を超えた場合は、クライアントにエラーが返却されます。

---

## (1) 構成ソフトウェアのプロセスの起動順序の依存関係

Cosminexus で構築したシステムを構成する構成ソフトウェアのプロセスの起動順序の依存関係について説明します。

以降は、J2EE サーバの動作モード別に、構成ソフトウェアのプロセスの起動順序の依存関係について説明します。

## (a) プロセスの依存関係

プロセスの依存関係を次に示します。

表 5-74 プロセスの依存関係

プロセスの種類	前提プロセス
パフォーマンストレーサ	-
スマートエージェント <sup>1, 2</sup>	-
CTM ドメインマネージャ <sup>1</sup>	スマートエージェント <sup>1, 2</sup>
CORBA ネーミングサービス	-
CTM デモン <sup>1</sup>	<ul style="list-style-type: none"> <li>• パフォーマンストレーサ</li> <li>• スマートエージェント</li> <li>• CTM ドメインマネージャ</li> <li>• CORBA ネーミングサービス</li> </ul>
SFO サーバ	<ul style="list-style-type: none"> <li>• パフォーマンストレーサ</li> </ul>
J2EE サーバ	<ul style="list-style-type: none"> <li>• パフォーマンストレーサ</li> <li>• スマートエージェント</li> <li>• CTM ドメインマネージャ<sup>3</sup></li> <li>• CORBA ネーミングサービス<sup>4</sup></li> <li>• CTM デモン<sup>3</sup></li> </ul>
Web サーバ	パフォーマンストレーサ

## (凡例)

- : 前提プロセスがない。

## 注 1

CTM を使用する場合に起動するプロセスです。

## 注 2

トランザクションサービスを使用する場合に起動するプロセスです。

## 注 3

CTM を使用する場合に前提となるプロセスです。

## 注 4

CORBA ネーミングサービスをアウトプロセスで使用する場合に起動するプロセスです。  
CORBA ネーミングサービスをインプロセスで使用する場合は必要ありません。

## 5. トラブルシューティング

### (2) プロセスの再起動方法

システムでプロセスが異常終了した場合のプロセスの再起動方法について説明します。

なお、CTM に関連するプロセスが異常終了して再起動できない場合は、次の手順で対処してください。

1. プロセスが再起動できない場合、出力されたエラーメッセージからトラブルの要因を調査する。
2. ctmraset コマンドを実行して CTM の実行環境のバックアップを取得する。

#### (a) 再起動の手順

プロセスの再起動（回復）の手順を次に示します。

表 5-75 プロセスの再起動方法

異常プロセス	開始コマンド	再起動（回復）手順
データベースサーバ	-	DB サーバを再起動してください。
OpenTP1	-	OpenTP1 を再起動してください。
パフォーマンスストレサ	cprfstart	パフォーマンスストレサを再起動してください。
スマートエージェント <sup>1, 2</sup>	osagent	スマートエージェントを再起動してください。
CTM ドメインマネージャ <sup>1</sup>	ctmdmstart	CTM ドメインマネージャを再起動してください。
CORBA ネーミングサービス	nameserv	次に示す手順で対処してください。なお、CTM を使用しない場合、手順 1 および手順 4 は不要です。 1. CTM デーモンの強制停止 2. J2EE サーバの強制停止 3. CORBA ネーミングサービスの再起動 4. CTM デーモンの再起動 5. J2EE サーバ再起動
CTM デーモン <sup>1</sup>	ctmstart	次に示す手順で対処してください。 1. CORBA ネーミングサービスの強制停止 2. J2EE サーバの強制停止 3. CORBA ネーミングサービスの再起動 4. CTM デーモンの再起動 5. J2EE サーバの再起動
CTM レギュレータ	-	CTM デーモンによって自動再起動されるため、再起動は不要です。
SFO サーバ <sup>3</sup>	cjstartsv	SFO サーバを再起動してください。
J2EE サーバ	cjstartsv	J2EE サーバを再起動してください。
Web サーバ	-	Web サーバを再起動してください。

(凡例)



- : 使用する製品によって開始コマンドが異なる, または該当する開始コマンドがない。

注 1

CTM を使用する場合に起動するプロセスです。

注 2

トランザクションサービスを使用する場合に起動するプロセスです。

注 3

セッションフェイルオーバ機能を使用する場合に起動するプロセスです。

## 5.6.2 J2EE アプリケーションの強制停止に失敗した場合

J2EE アプリケーションの強制停止に失敗した場合, J2EE サーバによって, 次の情報が出力されます。

### メッセージ

メッセージログに出力されます。メッセージログの出力先については、「5.4.1 Cosminexus のログの取得 (J2EE アプリケーションを実行するシステム)」を参照してください。

### スタックトレース

例外ログとスレッドダンプに出力されます。例外ログの出力先については、「5.4.1 Cosminexus のログの取得 (J2EE アプリケーションを実行するシステム)」を参照してください。スレッドダンプについては、「5.5.5 JavaVM のスレッドダンプの調査」を参照してください。

なお, J2EE アプリケーションの強制停止では, 内部でメソッドキャンセルが実行されています。メソッドキャンセル実行時に出力される障害情報については、「付録 D 機能ごとに出力されるログ情報」を参照してください。

## 5.6.3 セッションフェイルオーバ機能でトラブルが発生した場合

ここでは, セッションフェイルオーバ機能にトラブルが発生した場合の対処方法として, 次の 2 種類のトラブルへの対処について説明します。

J2EE サーバまたは Web サーバでトラブルが発生した場合

SFO サーバでトラブルが発生した場合

### (1) J2EE サーバまたは Web サーバでトラブルが発生した場合

J2EE サーバまたは Web サーバでトラブルが発生した場合は、「5.1 対処の手順」に示した手順に従って対処してから, J2EE サーバまたは Web サーバを起動してください。J2EE サーバまたは Web サーバの起動方法については, マニュアル「Cosminexus 簡易構築・運用ガイド」の日常運用の操作に関する説明を参照してください。

## 5. トラブルシューティング

なお、トラブルによって J2EE サーバが停止した場合は、トラブルが発生した J2EE サーバに対してリクエストを送信していた Web サーバも再起動してください。再起動しないと、リダイレクタは、すでに無効になっている常設コネクションを使用し続けます。この結果、トラブルが発生した J2EE サーバにリクエストを転送して通信タイムアウトが発生するまで待ち状態になり、通信タイムアウト後、リクエストはエラーとしてクライアントに返却されることとなります。

通信タイムアウトの設定については、マニュアル「Cosminexus システム構築ガイド」を参照してください。

### (2) SFO サーバにトラブルが発生した場合

SFO サーバにトラブルが発生した場合は、J2EE サーバにトラブルが発生した場合と同様に、必要な資料を取得してトラブルの要因を取り除いたあとで、SFO サーバを再起動します。

なお、SFO サーバを再起動すると、J2EE サーバから SFO サーバに再接続する処理が実行されます。このとき、J2EE サーバと SFO サーバ間では、初期化時と同様のネゴシエーション処理が実行されます。

ネゴシエーション処理では、冗長化された J2EE サーバ上の J2EE アプリケーション間で次の情報が共通になっているかが確認されます。

Web アプリケーションのコンテキストルート名

HttpSession 数の上限値

セッション情報の最大サイズ

グローバルセッション ID が使用する HTTP Cookie の名称

SFO サーバ内のロック待ちでのタイムアウト時間

冗長化された J2EE サーバ上の J2EE アプリケーション間でこれらの情報が異なっている場合、その Web アプリケーションはセッションフェイルオーバー機能が無効のまま処理を続けますが、SFO サーバへの再接続を行いません。

この場合は、該当する J2EE アプリケーションを一度停止して、J2EE アプリケーションのセッションフェイルオーバーについての設定を正しく設定し直してから、J2EE アプリケーションを再起動してください。

### (3) セッションフェイルオーバー機能で出力されるログの確認

セッションフェイルオーバー機能を使用している場合、セッションフェイルオーバー機能を使用する J2EE サーバ、および SFO サーバでログが出力されます。トラブルが発生した場合に必要な応じて確認してください。

J2EE サーバでのログの出力レベルと出力される内容の対応を、次の表に示します。

表 5-76 J2EE サーバでのログの出力レベルと出力される内容の対応

ログ出力レベル	出力される内容
Error	<ul style="list-style-type: none"> <li>セッションフェイルオーバ機能開始、終了についての情報</li> <li>SFO サーバとの接続についての情報</li> <li>SFO サーバでの障害発生についての情報</li> <li>グローバルセッション情報のサイズ見積もり機能が有効であることを通知する情報</li> <li>グローバルセッション情報のサイズ見積もり機能によるサイズ情報</li> </ul>
Warning	<ul style="list-style-type: none"> <li>グローバルセッション情報の引き継ぎ処理についての情報</li> <li>無効なグローバルセッション（有効期限切れ）のリクエストによってグローバルセッション ID が見つからなかったことの通知</li> </ul>
Information	<ul style="list-style-type: none"> <li>ヘルスチェックスレッドのチェック結果</li> </ul>

SFO サーバでのログの出力レベルと出力される内容の対応を、次の表に示します。

表 5-77 SFO サーバでのログの出力レベルと出力される内容の対応

ログ出力レベル	出力される内容
Error	<ul style="list-style-type: none"> <li>SFO サーバの起動・停止に関する情報</li> <li>SFO サーバで発生した障害に関する情報</li> <li>グローバルセッション数が最大値に対して「0%」、「20%」、「40%」、「60%」、「80%」または「100%」になったことの通知</li> <li>SFO サーバで有効期限切れによって破棄されたグローバルセッションの数</li> </ul>
Warning	<ul style="list-style-type: none"> <li>SFO サーバで有効期限切れによって破棄されたグローバルセッションのグローバルセッション ID</li> <li>グローバルセッション数の最大値を超えて作成要求があったことの通知</li> <li>無効なグローバルセッション（有効期限切れ）の取得要求があったことの通知</li> </ul>
Information	<ul style="list-style-type: none"> <li>SFO サーバで有効期限切れによって破棄されたグローバルセッションの詳細情報</li> </ul>

## 5.6.4 データベースと接続中にトラブルが発生した場合

データベースとの接続中に、Cosminexus の構成ソフトウェアである Cosminexus DABroker Library でトラブルが発生した場合、Cosminexus DABroker Library によってデータベースとの接続は切り離されます。また、Cosminexus DABroker Library を強制終了した場合も同様です。

このとき、次の処理が実行されます。

Cosminexus DABroker Library が異常終了した場合  
データベースで実行中の処理はロールバックされます。

DBMS が異常終了した場合  
データベースで実行中の処理は、DBMS の制御に従って処理されます。

トラブルへの対応後、必要に応じてデータベースに再接続してください。

また、データベースアクセス、Cosminexus DABroker Library または JDBC ドライバで

## 5. トラブルシューティング

エラーが発生した場合、例外として `SQLException` がスローされます。この `SQLException` オブジェクトの `reason` には、次の項目が設定されています。トラブルへの対処に利用してください。

JDBC ドライバのメッセージ

エラーを検知したクラス名称とメソッド名称

Cosminexus DABroker Library からのメッセージ (メッセージがある場合)

なお、次のエラーが発生した場合は、`SQLException` はスローされません。

`Driver.connection` メソッドでデータベースに接続できない場合

URL 指定で `null` が返却される場合

Cosminexus DABroker Library で対象にしていない URL が指定された場合などが該当します。

この場合は、JDBC インタフェースの仕様で決まっている処理が実行されます。

なお、データベースとして `HiRDB`、または `Oracle` を利用している場合、性能解析トレースファイルに出力されたコネクション ID と、Cosminexus DABroker Library や `HiRDB` クライアント、または `Oracle` クライアントのログファイルやトレースファイルに出力されたコネクション ID を突き合わせて確認することで、トラブルが発生したコネクションを確認できます。確認方法については、「付録 C データベースと接続中にトラブルが発生したコネクションの特定」を参照してください。

### 5.6.5 JavaVM が異常終了した場合

JavaVM で異常終了が発生した場合の対処方法と、異常終了が発生した場合に出力される情報について説明します。

#### (1) 異常終了時の対処方法

UNIX の場合、異常終了時には、次の手順で対処してください。なお、Windows の場合には、この対処は不要です。

1. 異常終了が発生したマシン上で `jvstrace` コマンドを実行します。  
`jvstrace.log` ファイルが出力されます。`jvstrace` コマンドの実行方法については、「5.4.16 JavaVM のスタクトレース情報の取得」を参照してください。
2. 取得した `jvstrace.log` ファイルを、エラーリポートファイル (`hs_err_pid< プロセス ID>.log`) とあわせて保守員に送付します。  
エラーリポートファイルの取得方法については、「5.4.9 JavaVM 出力メッセージログの取得 (標準出力またはエラーリポートファイル)」を参照してください。
3. `snapcore` コマンド (AIX の場合) または `car_tar_Z` コマンド (HP-UX, Linux, Solaris の場合) を実行します。

JavaVM の実行ファイルやトラブル発生時にロードされていたライブラリ，および core ダンプが，pax 形式（AIX の場合）または tar + compress 形式のデータアーカイブファイルとして作成されます。

4. 作成されたアーカイブファイルを，保守員に送付します。

## （2）異常終了時に出力される情報

JavaVM で次の異常終了が発生した場合に出力される情報について説明します。

- JavaVM 処理中に C ヒープ不足が発生した場合
- JavaVM 処理中の C ヒープ不足以外の OutOfMemoryError が発生した場合
- 内部論理エラーが発生した場合

（a）JavaVM 処理中に C ヒープ不足が発生した場合

C ヒープが不足すると，次のようなメッセージに続いて，メモリの状態，Java ヒープ情報，およびスタックトレース情報が標準出力およびエラーリポートファイル（hs\_err\_pid<プロセス ID>.log）に出力されます。そのあとで，JavaVM が強制終了されます。

```
Exception in thread <スレッド名称>
java.lang.OutOfMemoryError:requested <メモリ確保要求サイズ> bytes [
for <内部調査用メッセージ>] .
```

出力された情報を確認して対処してください。

（b）JavaVM 処理中の C ヒープ不足以外の OutOfMemoryError が発生した場合

J2EE サーバを起動するときのオプションとして -XX:+HitachiOutOfMemoryAbort を設定している場合，次の要因によって OutOfMemoryError が発生したときに，メッセージとメモリダンプを出力して JavaVM が強制終了します。出力された情報を確認して対処してください。

強制終了する要因

- Java ヒープ不足
- Perm ヒープ不足
- J2SE クラスライブラリでの C ヒープ不足

なお，JavaVM 処理中に C ヒープ不足になった場合は，このオプションの指定に関係なく強制終了します。

終了時には，次のメッセージと，メモリダンプが出力されます。

```
java.lang.OutOfMemoryError occurred.
JavaVM aborted because of specified -XX:+HitachiOutOfMemoryAbort options.
```

## 5. トラブルシューティング

オプションとして `-XX:+HitachiOutOfMemoryStackTrace` が設定されている場合は、スタックトレースが出力されたあとで終了します。ただし、あらかじめ `java.io.File.deleteOnExit` メソッドや `java.lang.Runtime.addShutdownHook` メソッドによって JavaVM 終了時に実行する処理を登録していた場合も、それらは実行されないで強制終了します。

### (c) 内部論理エラーが発生した場合

内部論理エラーが発生すると、エラーが発生した JavaVM の情報、調査用のエラー ID および問題が発生したスレッドを示すメッセージが、標準出力およびエラーレポートファイル (`hs_err_pid<プロセス ID>.log`) に出力されます。出力された情報を保守員に送付してください。

## 5.6.6 JP1 と連携したシステムでトラブルが発生した場合

JP1 と連携したシステムでトラブルが発生した場合、次に示す対処を実施する必要があります。

### (1) JP1/IM と連携したシステムでのトラブルへの対処

JP1/IM と連携したシステムで予想されるトラブルとその対処方法を次に示します。

表 5-78 JP1/IM と連携したシステムで予想されるトラブルとその対処方法

トラブルの種類	対処方法
監視ツリーの自動生成時にトラブルが発生した	トラブルが発生したら、JP1/Base のプラグインサービログファイルに出力された Cosminexus アダプタコマンドのメッセージを基にトラブルの発生要因を調査してください。JP1/Base のプラグインサービログファイルの詳細については、マニュアル「JP1/Base」を参照してください。

トラブルの種類	対処方法
JP1 イベントが JP1/IM に通知されない	<p>Cosminexus で構築したシステムから JP1/IM に JP1 イベントが発行されているかどうか、運用管理エージェント・運用監視エージェント・Management Server のログを確認してください。運用管理エージェント・運用監視エージェント・Management Server のログの格納場所については、「5.4.1(1) Cosminexus Component Container のログの取得」、または「5.4.2(1) Cosminexus Component Container のログの取得」を参照してください。</p> <p>運用管理エージェント・運用監視エージェント・Management Server のログの内容に応じて、次の対処を実施してください。</p> <ul style="list-style-type: none"> <li>• JP1 イベント発行のログが出力されていない場合 Cosminexus の運用管理サーバでの JP1 イベント発行の設定内容を確認してください。</li> <li>• JP1 イベント発行のログが出力されている場合、JP1 統合運用管理サーバでの JP1/Base の構成定義の作成内容を確認してください。また、Cosminexus の運用管理サーバおよび J2EE サーバでの JP1/Base のイベントサービスの動作環境設定内容を確認してください。</li> </ul> <p>Cosminexus の運用管理サーバおよび JP1 統合運用管理サーバでの障害監視の設定については、マニュアル「Cosminexus システム構築ガイド」を参照してください。</p>
JP1/IM-View のモニタ起動操作を実行したあと Web ブラウザが起動しない	<p>モニタ起動コマンドをコピーしたディレクトリに格納されている mngsvrmonitor.log に出力されたメッセージを基にトラブルの発生要因を調査してください。</p>

## (2) JP1/AJS2 と連携したシステムでのトラブルへの対処

JP1/AJS2 連携による運用管理コマンド (mngsvrutil) の自動実行でエラーが発生した場合、JP1/AJS2-View のジョブネットモニタウィンドウから、トラブルが発生したジョブの詳細情報を表示し、実行結果を確認してください。実行結果詳細に表示された mngsvrutil コマンドのエラーメッセージを基にトラブルの発生要因を調査してください。JP1/AJS2-View での操作の詳細については、マニュアル「JP1/Automatic Job Management System 2 操作ガイド」を参照してください。

### 5.6.7 1:1 系切り替えシステムでトラブルが発生した場合

1:1 系切り替えシステムで、待機系のホストへの系切り替え処理がデータベースサーバの障害 (サーバダウンやデッドロックなど) によってタイムアウトした場合の対処について、OS ごとに説明します。

#### (1) Windows の場合

ログを取得してから、手動で待機系ホストをオンラインにしてください。

##### (a) 1:1 系切り替えシステムのログの取得

1:1 系切り替えシステムでトラブルが発生した場合、クラスタログを取得する必要があります。

## 5. トラブルシューティング

まず、Windows を標準パスにインストールした場合のクラスタログの出力先を次に示します。

```
C:\¥WINDOWS¥cluster¥cluster.log
```

クラスタログには次の情報が出力されます。

- クラスタサービスの稼働ログ
- VBScript の構文にエラーがあった場合のエラーメッセージ
- 汎用スクリプトの Resource.LogInformation メソッド
- その他のメッセージ

クラスタログの詳細については、Windows のドキュメントを参照してください。

また、J2EE サーバが動作しない場合は、Cosminexus Component Container のログも参照してください。

### (b) 1:1 系切り替えシステムの手動リカバリ

次の手順で、1:1 系切り替えシステムを手動でリカバリしてください。

1. データベースを再起動するなどして、タイムアウトした原因を解消します。
2. 待機系のホストの対象リソースをオンラインにします。

### (2) UNIX の場合

データベースを再起動するなどして、タイムアウトした原因を解消してください。

## 5.6.8 N:1 リカバリシステムでトラブルが発生した場合

N:1 リカバリシステムでトラブルが発生した場合のリカバリ手順について、OS ごとに説明します。

### (1) Windows の場合

N:1 リカバリシステムで、待機系のホスト（リカバリ専用サーバ）でのリカバリ処理がデータベースサーバの障害（サーバダウンやデッドロックなど）によってタイムアウトした場合、ログを取得して、手動でリカバリを実行してください。リカバリ手順を次に示します。

1. N:1 系切り替えシステムのログを取得します。  
N:1 リカバリシステムでトラブルが発生した場合、クラスタログを取得する必要があります。取得するログは、1:1 系切り替えシステムでトラブルが発生した場合に取得するログと同じです。取得するログの詳細については、「5.6.7 1:1 系切り替えシステムでトラブルが発生した場合」を参照してください。
2. N:1 系切り替えシステムを手動でリカバリします。



次のどちらかの方法で、N:1 リカバリシステムを手動でリカバリしてください。

- 待機系のホストの対象リソースをオンラインにする
- J2EE サーバのトランザクション回復コマンド (cjstartrecover) を実行する

(a) 待機系のホストの対象リソースをオンラインにしてリカバリを実行する

待機系のホストの対象リソースをオンラインにしてリカバリを実行する手順について説明します。

1. データベースを再起動するなどして、タイムアウトした原因を解消します。
2. 待機系のホストの対象リソースをオンラインにします。

(b) J2EE サーバのトランザクション回復コマンド (cjstartrecover) を実行してリカバリを実行する

J2EE サーバのトランザクション回復コマンド (cjstartrecover) を実行してリカバリを実行する手順について説明します。

1. データベースを再起動するなどして、タイムアウトした原因を解消します。
2. 待機系のホストの汎用スクリプトの「Dir\_Name」に指定したパスにフォルダを作成します。  
すでにフォルダが存在する場合は、削除してからフォルダを作成してください。
3. 待機系のホストの汎用スクリプトがオンラインのときに、クラスタのログを参考に、cjstartrecover を実行します。
4. リカバリに成功したら、「Dir\_Name」に指定したパスに作成したフォルダを削除します。

cjstartrecover の詳細については、マニュアル「Cosminexus リファレンス コマンド編」を参照してください。

## (2) UNIX の場合

N:1 リカバリシステムで、待機系のホスト (リカバリ専用サーバ) でのリカバリ処理がデータベースサーバの障害 (サーバダウンやデッドロックなど) によりタイムアウトした場合、次の手順に従って手動でリカバリを実行します。

1. データベースを再起動するなどして、タイムアウトした原因を解消します。
2. 待機系のホストで、ダウンした実行系のホストに対応する monbegin を実行します。

```
# monbegin サーバの識別名
```

下線部分には、servers ファイルのオペランド「alias」に指定されている実行系のサーバの識別名を指定します。

## 5. トラブルシューティング

3. 待機系のホストで、ダウンした実行系のホストに対応する `monact` を実行します。

```
# monact サーバの識別名
```

下線部分には、`servers` ファイルのオペランド「`alias`」に指定されている実行系のサーバの識別名を指定します。

これによって、待機系のホスト（リカバリ専用サーバ）で、ダウンした実行系のホストで未決着だったトランザクションのリカバリ処理が実行されます。

### 参考

`servers` ファイルの定義など、HA モニタでのサーバ対応の環境設定については、マニュアル「`Cosminexus システム構築ガイド`」の HA モニタの環境設定に関する説明を参照してください。

## 5.6.9 EJB クライアントでトラブルが発生した場合

グローバルトランザクションを使用している EJB クライアントがダウンした場合、グローバルトランザクションをリカバリする必要があります。グローバルトランザクションのリカバリは、EJB クライアントを再起動することで実行できます。

グローバルトランザクションのリカバリの処理が完了したかどうかを確認する方法を次に示します。

- アプリケーションサーバで `cjlisttrn` コマンドを実行して、トランザクションがアクティブな状態になっているかを確認します。`cjlisttrn` コマンドの詳細については、マニュアル「`Cosminexus リファレンス コマンド編`」を参照してください。
- リソースアダプタを停止できるかどうかを確認してください。停止できる場合はグローバルトランザクションのリカバリ処理が完了しています。
- アプリケーションサーバが正常停止できるかどうかを確認してください。正常停止できる場合はグローバルトランザクションのリカバリ処理が完了しています。
- 各リソースが提供するツール、コマンドなどを使用して確認してください。

また、複数の EJB クライアントが存在する場合に、どの EJB クライアントがグローバルトランザクションのリカバリに必要なかを調査する手順を次に示します。

1. `cjlisttrn` コマンドを `-pending` オプションを指定して実行して未決着のトランザクションを確認します。

実行形式

```
cjlisttrn [<サーバ名称>] -pending -bqual
```

## 実行例

```
cjlisttrn MyServer -pending -bqual
```

2. 手順 1. で確認した未決着のトランザクションから、次のすべての条件に当てはまるトランザクションがあるかどうかを確認します。
  - cjlisttrn コマンドを発行するたびに経過時間が増加している
  - ブランチの種類が「Sub」または「Sub(recovered)」である
  - グローバルトランザクション ID に、停止した EJB クライアントの起動時に出力される KFCB40051-I メッセージの TmHash の値が含まれる

これらのすべての条件に当てはまるトランザクションが、グローバルトランザクションのリカバリに必要です。

## 5.7 トラブルシューティングに関連する留意事項

ここでは、トラブルシューティングに関連する留意事項について説明します。

### 5.7.1 EJB クライアントアプリケーションのシステムログに関する留意事項

ここでは、サブディレクトリ共有モードの EJB クライアントアプリケーションで出力されるシステムログを参照する場合、およびサブディレクトリ共有モードの EJB クライアントアプリケーションを運用する場合の留意事項について説明します。

ログファイル内で有効なデータは、ファイルの先頭から EOF までです。

EJB クライアントアプリケーションのシステムログでは、ログファイルがラップア라운드したときにラップア라운드以前のログのデータが削除されないで、先頭から順番に上書きされていきます。このため、ログファイルを参照するときには、EOF 以降のデータは無視してください。EOF 以降のデータは、ラップア라운드以前のログファイルの無効なデータです。

有効なログファイルのデータの末尾は、次に示すデータになります。

```
EOF CRLF CRLF CRLF CRLF-----< End of Data >-----CRLF CRLF
```

EOF はトレースデータの終端を表す文字 (0x1A) です。CRLF は、改行 (0x0D, 0x0A) を表します。

出力例を次に示します。なお、トレースの終端を表す文字は、[ EOF ] と表記します。

- Windows の場合

```
**** Windows XP 5.1                TZ=GMT+09:00                xxxxx/xx/xx
xx:xx:xx.xxx
   yyyy/mm/dd hh:mm:ss.sss                pid      tid      message-id
message (LANG=ja)
0000 xxxx/xx/xx xx:xx:xx.xxx    HEJB     BE3F6FE9 015EE671 KDJEXXXX-W
xxxxxxxxxx
0001 xxxx/xx/xx xx:xx:xx.xxx    HEJB     BE3F6FE9 015EE671 KDJEYYYYY-I
YYYYYYYYY
0002 xxxx/xx/xx xx:xx:xx.xxx    HEJB     BE3F6FE9 015EE671 KDJEZZZZZ-I
ZZZZZZZZZ
[ EOF ]
```

```
-----< End of Data >-----
```

```
<<ラップア라운드前の無効なデータ>>...
```

```
...
```

```
...
```

- UNIX の場合

```

**** "OS名 (OSバージョンなど含む)"          TZ=Asia/Tokyo          xxxx/
xx/xx  xx:xx:xx.xxx                          pid      tid      message-id
      yyyymm/dd hh:mm:ss.sss
message(LANG=ja)
0000  xxxx/xx/xx  xx:xx:xx.xxx  HEJB      BE3F6FE9  015EE671  KDJEXXXXXX-W
XXXXXXXXXX
0001  xxxx/xx/xx  xx:xx:xx.xxx  HEJB      BE3F6FE9  015EE671  KDJEYYYYY-I
YYYYYYYYYY
0002  xxxx/xx/xx  xx:xx:xx.xxx  HEJB      BE3F6FE9  015EE671  KDJEZZZZZ-I
ZZZZZZZZZ
[EOF]

-----< End of Data >-----

<<ラップアラウンド前の無効なデータ>>...
...
...

```

EJB クライアントアプリケーションのプロセス起動時に、システムプロパティの `ejbserver.logger.channels.define.<チャンネル名>.filenum` キーで指定した面数のトレースファイルはすべて作成されます。このとき、トレースファイルは空白 (0x20) で初期化されます。

ユーザログファイルの容量は、システムプロパティの `ejbserver.logger.channels.define.<チャンネル名>.filesize` キーで指定した容量で固定です。指定したサイズのトレースファイルがプロセス起動時に作成されます。このため、ログが出力されるのに伴って、容量が増減することはありません。

ログファイルの容量または面数を変更する場合、該当するログファイルに出力しているプロセスをすべて停止して、ログファイルと `mmap` ディレクトリ以下のログ管理ファイルを別のディレクトリへ移動、または削除する必要があります。

ログファイルの容量または面数を変更する場合以外では、ログファイル、およびログ管理ファイルを変更または削除しないでください。変更または削除した場合、以降のログが正しく出力されなくなるおそれがあります。

サブディレクトリ共有モードで動作する EJB クライアントアプリケーションがログを出力しているサブディレクトリを、`cjcdellog` コマンドで削除しないでください。削除した場合、以降のログが正しく出力されなくなるおそれがあります。

すでにサブディレクトリ専有モードで動作している EJB クライアントアプリケーションがある環境でサブディレクトリ共有モードで EJB クライアントアプリケーションを開始する場合、システムプロパティの `ejbserver.client.ejb.log` キーに、サブディレクトリ専有モードで動作している EJB クライアントアプリケーションとは異なる値を指定してください。同じ値を指定した場合、サブディレクトリ専有モードで動作する EJB クライアントアプリケーションのサブディレクトリ数を正しく管理できません。また、サブディレクトリ共有モードで動作している EJB クライアントアプリケーション

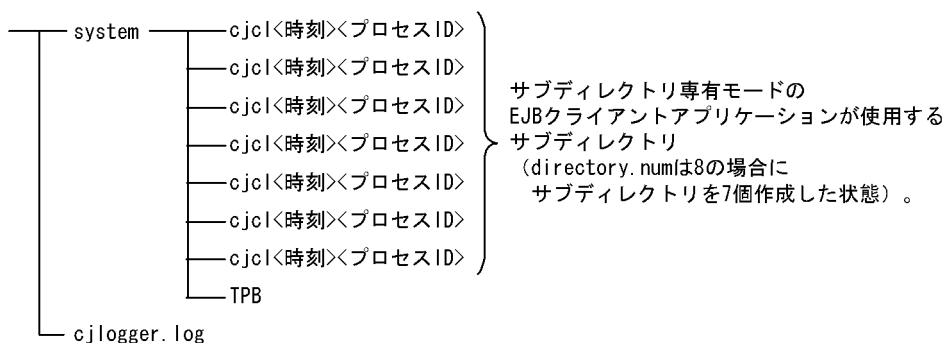
## 5. トラブルシューティング

ンがある環境では、サブディレクトリ専有モードで開始する EJB クライアントアプリケーションで、システムプロパティの `ejbserver.client.ejb.log` キーに "system" を指定しないでください。サブディレクトリ共有モードでは `ejbserver.client.ejb.log` キーのデフォルト値として "system" を使用するため、サブディレクトリ数が正しく管理できなくなります。

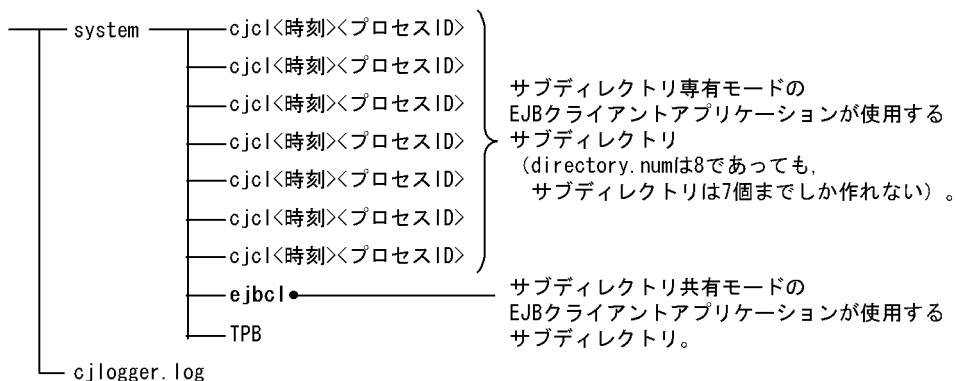
すでに Cosminexus 06-50 よりも前のバージョンで構築した環境で、サブディレクトリ専有モードで動作させている EJB クライアントアプリケーションを、システムプロパティ `ejbserver.client.ejb.log` キーに "system" を指定した状態で 06-50 以降に移行した場合、サブディレクトリ専有モードの EJB クライアントアプリケーションとサブディレクトリ共有モードの EJB クライアントアプリケーションが混在することになります。この場合、次の図のような状態になります。

図 5-14 サブディレクトリ専有モードで動作させている EJB クライアントアプリケーションがある環境にサブディレクトリ共有モードで動作する EJB クライアントアプリケーションを混在させた場合のディレクトリ構成

### ●06-50よりも前の環境



### ●06-50以降の環境



このように、混在した環境では、サブディレクトリ専有モードの EJB クライアントアプ

リケーションが作成できるサブディレクトリの数が、サブディレクトリ共有モードの EJB クライアントアプリケーションが使用するディレクトリの数（1 個）分、少なくともなります。なお、すでに `directorynum` で指定した数分のサブディレクトリが作成された状態で移行した場合は、サブディレクトリ専有モードで使用するサブディレクトリの数が減ることはありません。

`cjclstartap` コマンドはサブディレクトリ共有モードだけで使用してください。サブディレクトリ専有モードで `cjclstartap` コマンドを実行して EJB クライアントを起動した場合、EJB クライアント起動コマンドの稼働時の情報（`cjclstartap[n].log`）が正しく表示されません。

## 5.7.2 Cosminexus DABroker Library のトレースファイルの運用時の留意事項

ここでは、Cosminexus DABroker Library のトレースファイル運用時の留意事項について説明します。

### （1）拡張データベースアクセストレースファイル

拡張データベースアクセストレースファイル運用時の注意事項を次に示します。

拡張データベースアクセストレースでは、Cosminexus DABroker Library 動作環境定義ファイルで設定した行数を超える出力があった場合、現在のトレースファイルがバックアップファイルとして別名で保存されます。トレース内容は、トレースファイルの先頭から再び出力されます。

バックアップファイルは 1 世代前までしか残りません。このため、すでにバックアップファイルがある場合に、現在のトレースファイルが設定した行数を超えると、既存のバックアップファイルが上書きされます。複数世代のバックアップファイルを保存する必要がある場合は、あらかじめ別のディレクトリに退避してください。

トレースファイルは実行しているコネクションの数だけ作成されます。

トレースファイルは自動的に削除されません。ユーザが不要になった時点で削除してください。

ハードディスクの空き容量が十分でない場合、トレースの出力が停止されることがあります。

### （2）エラートレースファイル

エラートレースファイル運用時の注意事項を次に示します。

`dabevttcerr1.txt` ファイルのサイズが 512 キロバイトを超えている場合にログを出力すると、`dabevttcerr1.txt` ファイルをバックアップ用ファイル（`dabevttcerr2.txt`）に名称を変更したあと、新規作成した `dabevttcerr1.txt` ファイルにログを出力します。

## 5. トラブルシューティング

バックアップ用ファイルは1世代前までだけ残せます。バックアップファイルを保存する必要がある場合は、あらかじめ別のディスクに退避してください。

ハードディスクの空き容量が十分でない場合、ログの出力を停止します。

### (3) Exception トレースログのファイル

Exception トレースログのファイル運用時の注意事項を次に示します。

Exception トレースログのファイルは1世代前まで保持されます。ファイル出力時にすでにカレントファイルと1世代前のファイルが存在する場合は、1世代前のファイルが削除され、カレントファイルが1世代前のファイルに置き換えられます。

ファイルの出力先は、環境変数 `DAB_FILEDIR` で設定できます。

デフォルトは、`<Cosminexus DABroker Library 運用ディレクトリ>%spool` (Windows の場合)、または `<Cosminexus DABroker Library 運用ディレクトリ>/spool` (UNIX の場合) です。

`<ファイルの出力先ディレクトリ>%J2EE サーバ名称 +MTrc.log1` (Windows の場合)、または `<ファイルの出力先ディレクトリ>/J2EE サーバ名称 +MTrc.log1` (UNIX の場合) のバイト数が259バイトを超えないように注意してください。また、`<ファイルの出力先ディレクトリ>%J2EE サーバ名称 +MTrcOLD.log` (Windows の場合)、または `<ファイルの出力先ディレクトリ>/J2EE サーバ名称 +MTrcOLD.log` (UNIX の場合) のバイト数が259バイトを超える場合も注意が必要です。

259バイトを超えた場合の動作は、次のようになります。

- `<ファイルの出力先ディレクトリ>%J2EE サーバ名称 +MTrc.log` のバイト数が259バイトを超えた場合、ログファイル、および1世代前のファイルが作成されません。
- `<ファイルの出力先ディレクトリ>%<J2EE サーバ名称 +MTrc.log>` のバイト数が259バイトを超えない場合に、`<ファイルの出力先ディレクトリ>%<J2EE サーバ名称 +MTrcOLD.log>` のバイト数が259バイトを超えたときは、ログファイルは作成されますが、1世代前のファイルが作成されません。この場合、ファイル出力情報量を超えても1世代前のファイルが作成されないで、カレントファイルへ出力され続けます。

ファイルの作成や出力に失敗した場合、ログファイルに情報が出力されません。このとき、J2EE アプリケーションへのエラーメッセージの返却や、リトライは実行されません。エラーになった要因が解決されると、ログファイルにファイルが出力されるようになります。

ログファイル初回接続時にエラーが発生した場合に考えられる要因を次に示します。

- ディレクトリが存在しない、またはディレクトリに対する権限がない
- カレントファイルを1世代前のファイルに移動できない
- 1世代前のファイルを削除できない

ログファイル出力時にエラーが発生した場合に考えられる要因を次に示します。

- ディレクトリが存在しない、またはディレクトリに対する権限がない



- カレントファイルを作成または出力できない
- カレントファイルを 1 世代前のファイルに移動できない
- 1 世代前のファイルを削除できない

#### (4) JDBC ドライバのトレースファイル

JDBC ドライバのトレースファイル運用時の注意事項を次に示します。

JDBC ドライバ内でトレース情報の数がエントリ数を超えた場合は、トレースの古い順にトレース情報が上書きされます。

JDBC インタフェースメソッドトレース (キーワード: JDBC\_IF) は、Entry および Return でそれぞれ 1 エントリのトレース領域を使用します。

DABroker とのイベントトレース (キーワード: SV\_EVENT) は、送信また受信イベントでそれぞれ 1 エントリのトレース領域を使用します。

### 5.7.3 CTM 使用時の留意事項

ここでは、CTM 使用時の留意事項について説明します。CTM では次のことに注意してください。

CTM の障害情報は、CTM ドメイン単位の CTMSPOOL 環境変数の下に取得されています。CTM デーモンおよび CTM ドメインマネージャのダウンの場合には再開時にも障害情報を取得するため、障害発生後に障害情報を退避してください。

### 5.7.4 PRF 使用時の留意事項

ここでは、PRF 使用時の留意事項について説明します。PRF では次のことに注意してください。

PRF の障害情報は、PRF デーモン識別子単位の PRFSPOOL 環境変数の下に取得されています。PRF デーモンのダウンの場合には再開時にも障害情報を取得するため、障害発生後に障害情報を退避してください。



# 6

## 監査ログを使用したシステムの監査

この章では、監査ログを使用したシステムの監査について説明します。

---

6.1 監査ログを使用したシステム監査の概要

---

6.2 監査ログの収集と調査

---

6.3 監査ログを出力するコマンド・操作一覧

---

6.4 監査ログの出力ポイント

---

## 6.1 監査ログを使用したシステム監査の概要

---

この節では、監査ログを使用したシステムの監査の概要について説明します。

監査ログとは、システム構築者や運用者が Cosminexus のプログラムに対して実行した操作、およびその操作に伴うプログラムの動作の履歴が出力されるファイルです。監査者が監査ログを調査することで、「いつ」「だれが」「何をしたか」を知ることができ、システムの運用が法規制やセキュリティ評価基準、または業界ごとの各種の基準に準拠していることを証明できます。

## 6.2 監査ログの収集と調査

この節では、システムが出力する監査ログの出力先、出力形式、および出力項目の詳細について説明します。

### 6.2.1 監査ログの出力先

デフォルトの場合の監査ログの出力先を次に示します。

- Windows の場合  
`<Cosminexus のインストールディレクトリ>%auditlog%audit[n] .log`
- UNIX の場合  
`/opt/Cosminexus/auditlog/audit[n] .log`

注 [n] の部分には、面の番号が付きます。

デフォルトでは、監査ログの最大ファイルサイズは「32MB」、ファイル面数は「4」に設定されています。最大ファイルサイズ、ファイル面数、および出力先は、監査ログの定義ファイル (auditlog.properties) で変更できます。監査ログの設定については、マニュアル「Cosminexus システム構築ガイド」の監査ログ取得の設定についての説明を参照してください。

### 6.2.2 監査ログの出力形式

監査ログは、メッセージとして出力されます。監査ログのメッセージは、次の形式で出力されます。

```
CALFHM 1.0,出力項目1=値1,出力項目2=値2,出力項目3=値3,・・・出力項目n=値n
```

先頭の「CALFHM 1.0」は、ヘッダ情報です。監査ログに、共通で出力されます。

システムの監査で使用するメッセージの出力例を次に示します。

出力例

```
CALFHM 1.0,seqnum=1,msgid=KDJE54400-I,date= 2007-01-22T16:09:59.884+09:00,
progid=Cosminexus,compid=CCC,pid=00EB7859,ocp:host=host01,
ctgry=ConfigurationAccess,result=Occurrence,subj:euid=account01,
obj="Server01",op="Add",to:host=host02,to:port=28080,
msg="account01 executes the request (cjimportapp Server01 -f Appl.ear)."
```

注 実際に出力されるメッセージは、1行で表示されます。この例のように改行されません。

このメッセージは、「2007年1月22日16時9分59.884秒」に、「ホスト01」というホストで、cjimportapp コマンドが実行されたことを示しています。また、コマンドを実行したアカウントは「account01」、「Server01」というサーバに対してコマンド実行された

## 6. 監査ログを使用したシステムの監査

ことを示しています。メッセージ内容の詳細については、「6.2.3 メッセージの出力項目」を参照してください。

### 6.2.3 メッセージの出力項目

監査で使用するメッセージの出力項目には、すべてのメッセージで共通の項目、およびメッセージごとに固有の項目があります。それぞれの項目の意味を次に示します。

#### すべてのメッセージで共通の項目

すべてのメッセージで同じ意味の変数値が出力されたり、共通の文字列が出力されたりする項目が該当します。

#### メッセージごとに固有の項目

メッセージごとに固有の意味を持つ変数値が出力されたり、メッセージごとに出力される文字列が決まっていたりする項目が該当します。

監査で使用するメッセージの出力項目の詳細について、次の表に示します。

表 6-1 監査で使用するメッセージの出力項目

出力項目名	出力項目の意味	詳細	共通 / 固有
seqnum	通番	監査ログの通番が出力されます。 出力される値は、1 から 9999999999 までの整数です。	共通
msgid	メッセージ ID	メッセージ ID が出力されます。	固有
date	日付・時刻	メッセージが出力された日時が、次の形式で出力されます。 <i>YYYY-MM-DDT hh:mm:ss.sssTZD</i> <ul style="list-style-type: none"><li>• <i>YYYY</i>: 西暦年</li><li>• <i>MM</i>: 月</li><li>• <i>DD</i>: 日</li><li>• <i>T</i>: 日付と時間の区切り</li><li>• <i>hh</i>: 時</li><li>• <i>mm</i>: 分</li><li>• <i>ss</i>: 秒</li><li>• <i>sss</i>: マイクロ秒</li><li>• <i>TZD</i>: タイムゾーン <sup>1</sup></li></ul>	共通
progid	発生プログラム名	監査事象が発生したプログラムの名称が出力されます。 Cosminexus の場合、「Cosminexus」という文字列が出力されます。	共通

出力項目名	出力項目の意味	詳細	共通 / 固有
compid	発生構成ソフトウェア名	<p>監査事象が発生した構成ソフトウェアの略称が、次の形式で出力されます。</p> <ul style="list-style-type: none"> <li>• CCC : Cosminexus Component Container</li> <li>• CTM : Cosminexus Component Transaction Monitor</li> <li>• HWS : Hitachi Web Server</li> <li>• MNG : Management Server</li> <li>• PRF : Cosminexus Performance Tracer</li> <li>• UAP_&lt;任意の名称&gt;<sup>2</sup> : ユーザが J2EE アプリケーション、またはバッチアプリケーション実装時に指定した任意の名称</li> </ul>	固有
pid	発生プロセス ID	監査事象が発生したプロセスのプロセス ID が出力されます。	共通
ocp:host	発生場所	監査事象が発生したホストの名称が出力されます。ただし、発生場所の情報が取得できなかった場合、「(null)」が出力されます。	共通

6. 監査ログを使用したシステムの監査

出力項目名	出力項目の意味	詳細	共通 / 固有
ctgry	監査事象の種別	<p>監査事象の種別が、次のとおり分類されて出力されます。</p> <p>StartStop サーバ、プロセス、サービスなどの起動・終了を示す事象です。</p> <p>Authentication ログイン、ログアウトなど、Management Server への管理者 ID と管理者パスワードによる認証が実行されたことを示す事象です。</p> <p>ConfigurationAccess 設定および構成の変更、構成情報の参照を示す事象です。</p> <p>AccessControl <sup>2</sup> 管理者またはエンドユーザが管理リソース、およびセキュリティリソースへのアクセスを試みて、成功または失敗したことを示す事象です。</p> <p>Failure ソフトウェアの異常を示す事象です。</p> <p>LinkStatus <sup>2</sup> 機器間のリング状態を示す事象です。</p> <p>ExternalService <sup>2</sup> 外部サービスとの通信結果を示す事象です。</p> <p>ContentAccess <sup>2</sup> ユーザの重要なデータへのアクセスを試みて、成功または失敗したことを示す事象です。</p> <p>Maintenance <sup>2</sup> 保守作業を実行して、操作が正常終了または失敗したことを示す事象です。</p> <p>AnomalyEvent <sup>2</sup> しきい値のオーバなどの異常が発生したこと、または異常な通信の発生を示す事象です。</p> <p>ManagementAction <sup>2</sup> プログラムの重要なアクションが実行されたことを示す事象、またはほかの監査事象を契機として実行されるアクションを示す事象です。</p>	固有
result	監査事象の結果	<p>監査事象の結果（成功・失敗・発生）が、次の形式で出力されます。</p> <p>Success 監査事象の成功を示します。</p> <p>Failure 監査事象の失敗を示します。</p> <p>Occurrence 成功および失敗の区別がない事象の発生を示します。</p>	固有
subj:uid	サブジェクト識別情報（アカウント識別子の場合）	<p>監査事象を発生させたものがアカウント情報に割り付けられている場合、アカウント識別子（ユーザ ID など）が出力されます。</p>	固有



出力項目名	出力項目の意味	詳細	共通 / 固有
subj:uuid	サブジェクト識別情報 (OSのアカウントの場合)	監査事象を発生させたものがOSが提供するアカウント情報に割り付けられている場合、OSのアカウント <sup>3</sup> が出力されます。ただし、サブジェクト識別情報が取得できなかった場合、「null」が出力されます。	固有
obj	オブジェクト情報	監査事象となった操作の対象の情報が、「"」で囲まれた形式で出力されます。	固有
op	動作情報	監査事象となった操作の種別が、「"」で囲まれた次の形式で出力されます。 Start 起動を示します。 Stop 停止を示します。 Login ログインを示します。 Logout ログアウトを示します。 Logon ログオンを示します。 Logoff ログオフを示します。 Refer 設定情報の参照を示します。 Add 設定情報の追加を示します。 Update 設定情報の更新を示します。 Delete 設定情報の削除を示します。 Occur 障害などの発生を示します。	固有

6. 監査ログを使用したシステムの監査

出力項目名	出力項目の意味	詳細	共通 / 固有
		Enforce <sup>2</sup> 処理の実施を示します。 Up <sup>2</sup> リンクの活性を示します。 Down <sup>2</sup> リンクの非活性を示します。 Request <sup>2</sup> 要求を示します。 Response <sup>2</sup> 応答を示します。 Send <sup>2</sup> 発信を示します。 Receive <sup>2</sup> 受信を示します。 Install <sup>2</sup> インストールを示します。 Uninstall <sup>2</sup> アンインストールを示します。 Backup <sup>2</sup> バックアップを示します。 Maintain <sup>2</sup> 保守作業を示します。 Invoke <sup>2</sup> 管理者などの呼び出しを示します。 Notify <sup>2</sup> 管理者などへの通知を示します。	
objloc	オブジェクトロケーション情報	オブジェクトロケーション情報が出力されます。	固有
to:host	リクエスト送信先ホスト	監査事象が複数のプログラム間で連携して動作するリクエストに関連する場合に、リクエストの送信先のホストの情報が出力されます。	固有
to:port	リクエスト送信先ポート番号	監査事象が複数のプログラム間で連携して動作するリクエストに関連する場合に、リクエストの送信先のポート番号が出力されます。	固有
msg	自由記述	監査事象の内容を示す文章が出力されます。	固有
before <sup>2</sup>	変更前情報	変更前の情報が出力されます。	固有
after <sup>2</sup>	変更後情報	変更後の情報が出力されます。	固有
auth <sup>2</sup>	権限情報	監査事象を発生させたときのサブジェクトの権限が出力されます。	固有

出力項目名	出力項目の意味	詳細	共通 / 固有
sins <sup>2</sup>	サービスインスタンス名	サービスの利用者の識別子が出力されます。	固有
haid <sup>2</sup>	冗長化識別情報	監査事象の発生場所が冗長化構成の場合、発生場所の系を表す情報が出力されます。	固有
from:host <sup>2</sup>	リクエスト送信元ホスト	監査事象が複数のプログラム間で連携して動作するリクエストに関連する場合に、リクエストの送信元の場所情報が出力されます。	固有
from:port <sup>2</sup>	リクエスト送信元ポート番号	監査事象が複数のプログラム間で連携して動作するリクエストに関連する場合に、リクエストの送信元のポート番号が出力されます。	固有
outp:host <sup>2</sup> (ホスト名で出力する場合)	出力元の場所	出力元が動作している場所情報が出力されます。	固有
subjp:host <sup>2</sup> (ホスト名で出力する場合)	指示元の場所	サブジェクトが指示を出した場所情報が出力されます。	固有
dtp:host <sup>2</sup> (ホスト名で出力する場合)	検出場所	検出エンティティが動作している場所情報が出力されます。	固有
loc <sup>2</sup>	ロケーション情報	ユーザが設定したロケーション識別情報が出力されず。ユーザが設定しなかった場合は、ルートアプリケーション情報が、「」で囲まれた形式で出力されます。ただし、ルートアプリケーション情報が取得できなかった場合、「null」が「」で囲まれた形式で出力されます。	固有

## (凡例)

共通：すべてのメッセージで共通の項目であることを示します。

固有：メッセージごとに固有の項目であることを示します。メッセージごとに固有の項目の詳細については、マニュアル「Cosminexus メッセージ 1 KAWS / KDAL / KDJE 編」、マニュアル「Cosminexus メッセージ 2 KEOS / KEUC / KFCB 編」、およびマニュアル「Cosminexus メッセージ 3 KFCT / KFDB / KFDJ 編」のそれぞれのメッセージの説明を参照してください。

注 1 タイムゾーンは、UTC からの時差で表示されます。表示形式を次に示します。

- $+hh:mm$  : UTC から  $hh$  時間  $mm$  分進んでいる
- $-hh:mm$  : UTC から  $hh$  時間  $mm$  分遅れている
- $Z$  : UTC と同じ

(例) 日本の場合 : +09:00

注 2 監査ログ出力用の API を使用して、J2EE アプリケーション、またはバッチアプリケーションから出力するように実装した場合にだけ出力される情報です。

## 6. 監査ログを使用したシステムの監査

注 3 OS のアカウントには、Windows の場合はユーザ名が、UNIX の場合は、実効ユーザ ID が出力されます。

## 6.3 監査ログを出力するコマンド・操作一覧

この節では、監査ログを出力するコマンドおよび操作について説明します。監査ログは、ここで説明するコマンドと操作を実行したタイミング、およびそのコマンドと操作による処理が実行されたタイミングで出力されます。

ここでは、コマンドと操作を次の分類に分けて、それぞれの分類で監査ログを出力するコマンドおよび操作の一覧を示します。

- J2EE サーバで使用するコマンド
- 性能解析トレース・CTM で使用するコマンド
- Management Server で使用するコマンド
- EJB クライアントアプリケーションで使用するコマンド
- Hitachi Web Server で使用するコマンド・操作

### 6.3.1 J2EE サーバで使用するコマンド一覧

J2EE サーバで使用するコマンドのうち、監査ログを出力するコマンドの一覧を次の表に示します。

表 6-2 監査ログを出力するコマンドの一覧 (J2EE サーバで使用するコマンド)

コマンド名	監査ログ出力ポイントの参照先
cjaddapp	6.4.1(3)
cjaddsec	6.4.1(3)
cjchmodapp	6.4.1(3)
cjclearpool	6.4.1(3)
cjclosecn	6.4.1(3)
cjcopyres	6.4.1(3)
cjdeleteapp	6.4.1(3)
cjdeletejb	6.4.1(3)
cjdeletelibjar	6.4.1(3)
cjdeleteres	6.4.1(3)
cjdeletesec	6.4.1(3)
cjdeployrar	6.4.1(3)
cjdumpsv	6.4.1(2)
cjenvsetup	6.4.1(2)
cjenvupdate	6.4.1(2)
cjexportapp	6.4.1(3)
cjexportrar	6.4.1(3)

## 6. 監査ログを使用したシステムの監査

コマンド名	監査ログ出力ポイントの参照先
cjgencmpsql	6.4.1(3)
cjgetappprop	6.4.1(3)
cjgetjbprop	6.4.1(3)
cjgetrarprop	6.4.1(3)
cjgetresprop	6.4.1(3)
cjgetstubsjar	6.4.1(3)
cjimportapp	6.4.1(3)
cjimportjb	6.4.1(3)
cjimportlibjar	6.4.1(3)
cjimportres	6.4.1(3)
cjlistapp	6.4.1(3)
cjlistjb	6.4.1(3)
cjlistlibjar	6.4.1(3)
cjlistpool	6.4.1(3)
cjlistrar	6.4.1(3)
cjlistres	6.4.1(3)
cjlistsec	6.4.1(3)
cjlistthread	6.4.1(3)
cjlisttrn	6.4.1(3)
cjlisttrnfile	6.4.1(3)
cjmapsec	6.4.1(3)
cjrarupdate	6.4.1(2)
cjreloadapp	6.4.1(3)
cjrenameapp	6.4.1(3)
cjreplaceapp	6.4.1(3)
cjresetsv	6.4.1(4)
cjresumepool	6.4.1(3)
cjsetappprop	6.4.1(3)
cjsetjbprop	6.4.1(3)
cjsetrarprop	6.4.1(3)
cjsetresprop	6.4.1(3)
cjsetup	6.4.1(2)
cjstartapp	6.4.1(3)
cjstartjb	6.4.1(3)
cjstartrar	6.4.1(3)

コマンド名	監査ログ出力ポイントの参照先
cjstartrecover	6.4.1(2)
cjstartsv	6.4.1(1)
cjstopapp	6.4.1(3)
cjstopjb	6.4.1(3)
cjstoprar	6.4.1(3)
cjstopsv	6.4.1(1)
cjstopthread	6.4.1(3)
cjsuspendpool	6.4.1(3)
cjtestres	6.4.1(3)
cjundeployrar	6.4.1(3)
cjunmapsec	6.4.1(3)
cjupdateapp	6.4.1(3)

注 監査ログが出力されるタイミングについての情報の参照先を示します。

### 6.3.2 バッチサーバで使用するコマンド一覧

バッチサーバで使用するコマンドのうち、監査ログを出力するコマンドの一覧を次の表に示します。

表 6-3 監査ログを出力するコマンドの一覧（バッチサーバで使用するコマンド）

コマンド名	監査ログ出力ポイントの参照先
cjclearpool	6.4.1(3)
cjcopyres	6.4.1(3)
cjdeleteres	6.4.1(3)
cjdeployrar	6.4.1(3)
cjdumpsv	6.4.1(2)
cjenvsetup	6.4.1(2)
cjenvupdate	6.4.1(2)
cjexecjob	6.4.2
cjexportrar	6.4.1(3)
cjgetrarprop	6.4.1(3)
cjgetresprop	6.4.1(3)
cjimportres	6.4.1(3)
cjkilljob	6.4.2
cjlistpool	6.4.1(3)

## 6. 監査ログを使用したシステムの監査

コマンド名	監査ログ出力ポイントの参照先
cjlistrar	6.4.1(3)
cjlistres	6.4.1(3)
cjlisthread	6.4.1(3)
cjrarupdate	6.4.1(2)
cjresetsv	6.4.1(4)
cjresumepool	6.4.1(3)
cjsetrarprop	6.4.1(3)
cjsetresprop	6.4.1(3)
cjsetup	6.4.1(2)
cjstartrar	6.4.1(3)
cjstartrecover	6.4.1(2)
cjstartsv	6.4.1(1)
cjstoprar	6.4.1(3)
cjstopsv	6.4.1(1)
cjstophread	6.4.1(3)
cjsuspendpool	6.4.1(3)
cjtestres	6.4.1(3)
cjundeployrar	6.4.1(3)

注 監査ログが出力されるタイミングについての情報の参照先を示します。

### 6.3.3 性能解析トレース・CTM で使用するコマンド一覧

性能解析トレース，および CTM で使用するコマンドのうち，監査ログを出力するコマンドの一覧を次の表に示します。

表 6-4 監査ログを出力するコマンドの一覧（性能解析トレース・CTM で使用するコマンド）

コマンド名	監査ログ出力ポイントの参照先
cprfgetpid	6.4.3(3)
cprfstart	6.4.3(1)
cprfstop	6.4.3(1)
ctmdmstart	6.4.3(1)
ctmdmstop	6.4.3(1)
ctmgetpid	6.4.3(3)
ctmstart	6.4.3(1)



コマンド名	監査ログ出力ポイントの参照先
ctmstop	6.4.3(1)
ctmstsstart	6.4.3(2)
ctmstsstop	6.4.3(2)

注 監査ログが出力されるタイミングについての情報の参照先を示します。

### 6.3.4 Management Server で使用するコマンド一覧

Management Server で使用するコマンドうち、監査ログを出力するコマンドの一覧を次の表に示します。

表 6-5 監査ログを出力するコマンドの一覧 (Management Server で使用するコマンド)

コマンド名 (サブコマンド)	監査ログ出力ポイントの参照先 <sup>1</sup>
adminagentctl ( start )	6.4.4(3)
adminagentctl ( stop )	6.4.4(3)
cmx_add_serverref	6.4.4(5)
cmx_admin_passwd	6.4.4(5)
cmx_build_model	6.4.4(5)
cmx_build_system <sup>2</sup>	6.4.4(6)
cmx_change_model	6.4.4(5)
cmx_define_application	6.4.4(5)
cmx_define_resource	6.4.4(5)
cmx_delete_serverref	6.4.4(5)
cmx_delete_system <sup>2</sup>	6.4.4(5)
cmx_deploy_application <sup>2</sup>	6.4.4(5)
cmx_deploy_resource <sup>2</sup>	6.4.4(5)
cmx_export_model	6.4.4(4)
cmx_list_model	6.4.4(4)
cmx_list_status	6.4.4(4)
cmx_register_application	6.4.4(5)
cmx_register_resource	6.4.4(5)
cmx_resume_lb	6.4.4(5)
cmx_scaleout_host	6.4.4(5)
cmx_start_application <sup>2</sup>	6.4.4(4)
cmx_start_resource <sup>2</sup>	6.4.4(4)

## 6. 監査ログを使用したシステムの監査

コマンド名 (サブコマンド)	監査ログ出力ポイントの参照先 <sup>1</sup>
cmx_start_target <sup>2</sup>	6.4.4(5)
cmx_stop_application <sup>2</sup>	6.4.4(4)
cmx_stop_resource <sup>2</sup>	6.4.4(4)
cmx_stop_target <sup>2</sup>	6.4.4(5)
cmx_test_lb <sup>2</sup>	6.4.4(4)
cmx_undefine_resource	6.4.4(5)
cmx_undefined_application	6.4.4(5)
cmx_undeploy_application <sup>2</sup>	6.4.4(5)
cmx_undeploy_resource <sup>2</sup>	6.4.4(5)
cmx_unregister_application	6.4.4(5)
cmx_unregister_resource	6.4.4(5)
mngsvrctl ( setup )	6.4.4(1) , 6.4.4(5)
mngsvrctl ( start ) <sup>2</sup>	6.4.4(2)
mngsvrctl ( stop )	6.4.4(2)
Hitachi Web Server を操作するコマンド	6.4.4(7)

注 1 監査ログが出力されるタイミングについての情報の参照先を示します。

注 2 これらのコマンドを実行すると、内部処理のために複数のコマンドが自動で実行される場合があります。監査ログを調査して監査を実施するには、内部処理のために実行されるそれぞれのコマンドが出力する監査ログも調査してください。内部処理のために複数のコマンドが自動で実行されるコマンドと、実行されるコマンドごとの監査ログの出力の可否を次の表に示します。

表 6-6 内部処理のために複数のコマンドが自動で実行されるコマンドと実行されるコマンドごとの監査ログの出力の可否

実行するコマンド (サブコマンドまたは引数)	内部処理で実行されるコマンド (引数)	監査ログの出力有無
cmx_build_system	mstartupwebsetup	×
	mngsvrssh	×
	mngsvrsetsid	×
	cjsetup	
	hwsserveredit	
	hwsconfigedit	
cmx_delete_system	mngsvrssh	×
	mngsvrsetsid	×
cmx_deploy_application	cjimportapp	

実行するコマンド (サブコマンドまたは引数)	内部処理で実行されるコマンド (引数)	監査ログの出力有無
cmx_deploy_resource	mappsetl	×
cmx_deploy_resource ( -resname <リソースアダプタ表示名> )	cjimportres	
	cjsetresprop	
	cjdeployrar	
cmx_deploy_resource ( -resname <JavaBeansリソース表示名> )	cjimportjtb	
cmx_start_application	mappsetl	×
	cjstartapp	
cmx_start_resource	mappsetl	×
cmx_start_resource ( -resname <リソースアダプタ表示名> )	cjstartrar	
cmx_start_resource ( -resname <JavaBeansリソース表示名> )	cjstartjtb	
cmx_start_target	prfinit	×
	mstartupwebinit	×
	mstartupwebstart	×
	ctdminit	×
	mngsvrssh	×
	mngsvrsetsid	×
	cprfstart	
	cprfgetpid	
	cjstartsv	
	httpspd	
	osagent	×
	nameserv	×
	ctmdmstart	
	ctmgetpid	
	ctmstart	
cmx_stop_application	mappsetl	×
	cjstopapp	
cmx_stop_resource	mappsetl	×
cmx_stop_resource ( -resname <リソースアダプタ表示名> )	cjstoprar	
cmx_stop_resource ( -resname <JavaBeansリソース表示名> )	cjstopjtb	

## 6. 監査ログを使用したシステムの監査

実行するコマンド (サブコマンドまたは引数)	内部処理で実行されるコマンド (引数)	監査ログの出力有無
cmx_stop_target	mstartupwebstop	×
	mngsvrssh	×
	mngsvrsetsid	×
	cprfstop	
	cjstopsv	
	httpsd	
	ctmdmstop	
	ctmstop	
	ctmstsstop	
cmx_test_lb	mngsvrssh	×
	mngsvrsetsid	×
cmx_undeploy_application	mappsctl	×
	cjdeleteapp	
cmx_undeploy_resource	mappsctl	×
cmx_undeploy_resource (-resname <リソースアダプタ表示名>)	cjundeployrar	
	cjdeleteres	
cmx_undeploy_resource (-resname <JavaBeans リソース表示名>)	cjdeletejb	
mngsvrctl (start)	mngsvrctl (setup)	

注 UNIX の場合にだけ内部処理のために実行されるコマンドです。

### 6.3.5 EJB クライアントアプリケーションで使用するコマンド一覧

EJB クライアントアプリケーションで使用するコマンドのうち、監査ログを出力するコマンドの一覧を次の表に示します。

表 6-7 監査ログを出力するコマンドの一覧 (EJB クライアントアプリケーションで使用するコマンド)

コマンド名	監査ログ出力ポイントの参照先
cjcdelloog	6.4.5
cjcdumpap	6.4.5
cjclstartap	6.4.5

注 監査ログが出力されるタイミングについての情報の参照先を示します。

### 6.3.6 Hitachi Web Server で使用するコマンド・操作一覧 ( Windows の場合 )

Windows の場合の Hitachi Web Server で使用するコマンドおよび操作のうち、監査ログを出力するコマンドおよび操作の一覧を次の表に示します。

表 6-8 監査ログを出力するコマンド・操作の一覧 ( Windows の場合の Hitachi Web Server で使用するコマンド・操作 )

コマンド名 ( オプション ), 操作	監査ログ出力ポイントの参照先
httpsd ( -k gracefulstop )	6.4.6(1)
httpsd ( -k restart )	6.4.6(1)
httpsd ( -k start )	6.4.6(1)
httpsd ( -k stop )	6.4.6(1)
コントロールパネルによる操作	6.4.6(2)

注 監査ログが出力されるタイミングについての情報の参照先を示します。

### 6.3.7 Hitachi Web Server で使用するコマンド・操作一覧 ( UNIX の場合 )

UNIX の場合の Hitachi Web Server で使用するコマンドおよび操作のうち、監査ログを出力するコマンドおよび操作の一覧を次の表に示します。

表 6-9 監査ログを出力するコマンド・操作の一覧 ( UNIX の場合の Hitachi Web Server で使用するコマンド・操作 )

コマンド名, 操作	監査ログ出力ポイントの参照先 <sup>1</sup>
httpsd <sup>2</sup>	6.4.7(2)
httpsdctl による操作	6.4.7(1)
シグナル送信による操作	6.4.7(3)

注 1 監査ログが出力されるタイミングについての情報の参照先を示します。

注 2 コマンドを直接実行した場合です。

## 6.4 監査ログの出力ポイント

---

ここでは、監査ログを出力するコマンド・操作を実行したとき、およびそのコマンド・操作による処理が実行されたときの、監査ログが出力されるポイントについて説明します。

ここでは、コマンドおよび操作を次の分類に分けて、それぞれのコマンドで監査ログを出力するポイントについて説明します。

- J2EE サーバで使用するコマンド
- 性能解析トレース・CTM で使用するコマンド
- Management Server で使用するコマンド
- EJB クライアントアプリケーションで使用するコマンド
- Hitachi Web Server で使用するコマンド

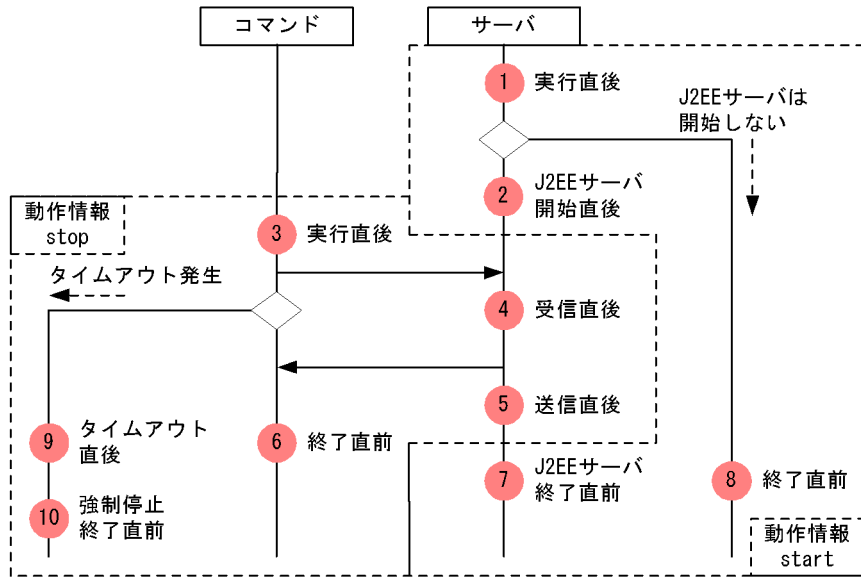
### 6.4.1 J2EE サーバで使用するコマンドの監査ログの出力ポイント

J2EE サーバで使用するコマンドの監査ログの出力ポイントについて説明します。

- (1) サーバプロセスを起動・停止するコマンド（サーバ管理コマンド以外）

サーバプロセスを起動・停止する、サーバ管理コマンド以外のコマンドの監査ログの出力ポイントを次の図に示します。

図 6-1 監査ログの出力ポイント（サーバプロセスを起動・停止するサーバ管理コマンド以外のコマンド）



(凡例) ● : 監査ログの出力ポイントを示します。  
 → : 同期メッセージを示します。

この図に示した監査ログの出力ポイントと出力される監査ログのメッセージ ID の対応を次の表に示します。

表 6-10 監査ログの出力ポイントとメッセージ ID の対応（サーバプロセスを起動・停止するサーバ管理コマンド以外のコマンド）

コマンド（オプション）	図中の番号	出力される監査ログのメッセージ ID
cjstartsv	1	KDJE54112-I
	2	KDJE54113-I
	7	KDJE54126-I
		KDJE54127-E
	8	KDJE54114-I
		KDJE54115-E
cjstopsv (-wait)	3	KDJE54116-I
	6	KDJE54117-I
		KDJE54118-E
9	KDJE54119-W	

## 6. 監査ログを使用したシステムの監査

コマンド (オプション)	図中の番号	出力される監査ログのメッセージ ID
cjstopsv (-f, または -fd)	3	KDJE54116-I
	6	KDJE54117-I
		KDJE54118-E
cjstopsv (-wait -f, または -wait -fd)	3	KDJE54116-I
	6	KDJE54117-I
		KDJE54118-E
	9	KDJE54119-W
	10	KDJE54117-I
KDJE54118-E		
cjstopsv (-wait -f, -fd, -wait -f, および -wait -fd 以外のオプション)	3	KDJE54116-I
	6	KDJE54117-I
		KDJE54118-E
- 1	4	KDJE54138-I
	5	KDJE54139-I
		KDJE54140-E
- 2	KDJE54155-E	

注 1 この表で示しているすべてのコマンドに共通で出力されるメッセージです。ただし、-f、および -fd 以外のオプションを指定してコマンドを実行した場合に限ります。

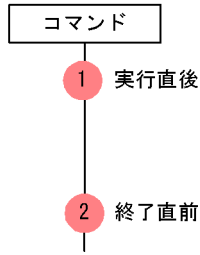
注 2 この表で示しているコマンドを実行して、メモリ不足が発生した場合に出力されることがあります。

### (2) サーバプロセスを起動・停止するコマンド以外のコマンド (サーバ管理コマンド以外)

サーバプロセスを起動・停止するサーバ管理コマンド以外のコマンドの監査ログの出力ポイントを次の図に示します。



図 6-2 監査ログの出力ポイント（サーバプロセスを起動・停止するサーバ管理コマンド以外のコマンド）



(凡例) ● : 監査ログの出力ポイントを示します。

この図に示した監査ログの出力ポイントと出力される監査ログのメッセージ ID の対応を次の表に示します。

表 6-11 監査ログの出力ポイントとメッセージ ID の対応（サーバプロセスを起動・停止するサーバ管理コマンド以外のコマンド）

コマンド	図中の番号	出力される監査ログのメッセージ ID
cjsetup	1	KDJE54100-I
	2	KDJE54101-I KDJE54102-E
cjenvsetup	1	KDJE54103-I
	2	KDJE54104-I KDJE54105-E
cjenvupdate	1	KDJE54106-I
	2	KDJE54107-I KDJE54108-E
cjrupdate	1	KDJE54109-I
	2	KDJE54110-I KDJE54111-E
cjstartrecover <sup>1</sup>	1	KDJE54120-I
	2	KDJE54121-I KDJE54122-E
cjdumpsv	1	KDJE54123-I
	2	KDJE54124-I KDJE54125-E
-	- 2	KDJE54155-E

## 6. 監査ログを使用したシステムの監査

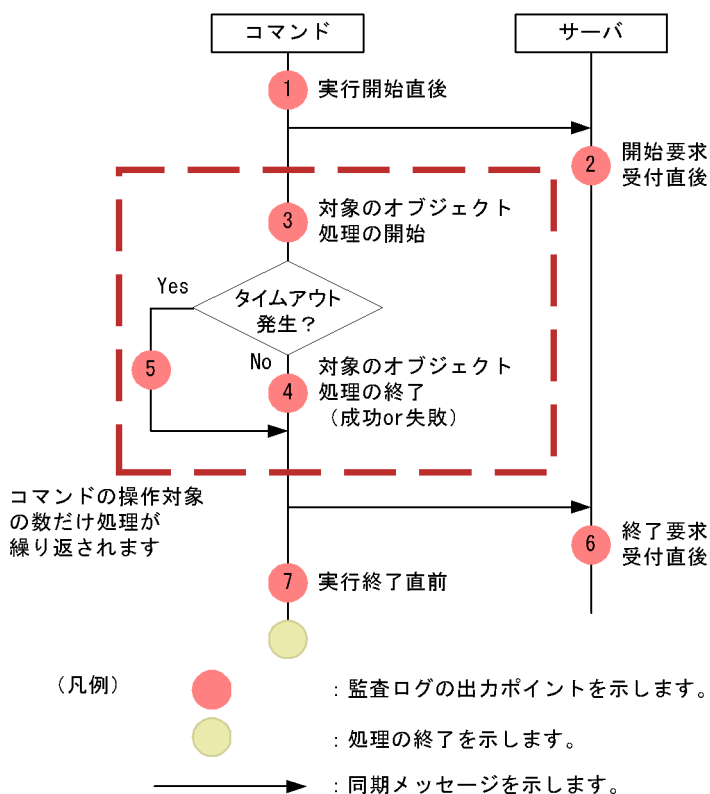
注 1 cjstartrecover コマンドは、内部処理で cjstartsv コマンドを実行します。そのため、cjstartrecover コマンド実行時には cjstartsv コマンドの監査ログも出力されます。

注 2 この表で示しているコマンドを実行して、メモリ不足が発生した場合に出力されることがあります。

### (3) サーバ管理コマンド (cjresetsv 以外のサーバ管理コマンド)

サーバ管理コマンド (cjresetsv 以外のサーバ管理コマンド) の監査ログの出力ポイントを次の図に示します。

図 6-3 監査ログの出力ポイント (cjresetsv 以外のサーバ管理コマンド)



この図に示した監査ログの出力ポイントと出力される監査ログのメッセージ ID の対応を次の表に示します。

表 6-12 監査ログの出力ポイントとメッセージ ID の対応 (cjresetsv 以外のサーバ管理コマンド)

コマンド	図中の番号	出力される監査ログのメッセージ ID	動作情報
cjaddapp	3	KDJE54424-I	Add
	4	KDJE54425-I	

コマンド	図中の番号	出力される監査ログ のメッセージ ID	動作情報
cjaddsec		KDJE54426-E	
	5	KDJE54427-E	
	3	KDJE54576-I	
	4	KDJE54577-I	
		KDJE54578-E	
5	KDJE54579-E		
cjchmodapp	3	KDJE54608-I	Update
	4	KDJE54609-I	
		KDJE54610-E	
	5	KDJE54611-E	
cjcLEARpool	3	KDJE54600-I	Delete
	4	KDJE54601-I	
		KDJE54602-E	
	5	KDJE54603-E	
cjclosecn	3	KDJE54604-I	
	4	KDJE54605-I	
		KDJE54606-E	
	5	KDJE54607-E	
cjcopyres	3	KDJE54476-I	Add
	4	KDJE54477-I	
		KDJE54478-E	
	5	KDJE54479-E	
cjdeleteapp	3	KDJE54428-I	Delete
	4	KDJE54429-I	
		KDJE54430-E	
	5	KDJE54431-E	
cjdeletejb	3	KDJE54440-I	
	4	KDJE54441-I	
		KDJE54442-E	
	5	KDJE54443-E	
cjdelelibjar	3	KDJE54436-I	
	4	KDJE54437-I	
		KDJE54438-E	
	5	KDJE54439-E	

## 6. 監査ログを使用したシステムの監査

コマンド	図中の番号	出力される監査ログ のメッセージ ID	動作情報	
cjdeleteres	3	KDJE54432-I		
	4	KDJE54433-I		
		KDJE54434-E		
5	KDJE54435-E			
cjdeletesec	3	KDJE54580-I		
	4	KDJE54581-I		
		KDJE54582-E		
5	KDJE54583-E			
cjdeployrar	3	KDJE54568-I	Update	
	4	KDJE54569-I		
		KDJE54570-E		
5	KDJE54571-E			
cjexportapp	3	KDJE54416-I	Refer	
	4	KDJE54417-I		
		KDJE54418-E		
5	KDJE54419-E			
cjexportrar	3	KDJE54420-I		
	4	KDJE54421-I		
		KDJE54422-E		
5	KDJE54423-E			
cjgencmpsql	3	KDJE54620-I		Update
	4	KDJE54621-I		
		KDJE54622-E		
5	KDJE54623-E			
cjgetappprop	3	KDJE54444-I	Refer	
	4	KDJE54445-I		
		KDJE54446-E		
5	KDJE54447-E			
cjgetjbprop	3	KDJE54456-I		
	4	KDJE54457-I		
		KDJE54458-E		
5	KDJE54459-E			
cjgetrarprop	3	KDJE54452-I		
	4	KDJE54453-I		

コマンド	図中の番号	出力される監査ログ のメッセージ ID	動作情報
cjgetresprop		KDJE54454-E	
	5	KDJE54455-E	
	3	KDJE54448-I	
	4	KDJE54449-I KDJE54450-E	
	5	KDJE54451-E	
cjgetstubsjar	3	KDJE54612-I	
	4	KDJE54613-I KDJE54614-E	
	5	KDJE54615-E	
cjimportapp	3	KDJE54400-I	Add
	4	KDJE54401-I KDJE54402-E	
	5	KDJE54403-E	
cjimportjb	3	KDJE54412-I	
	4	KDJE54413-I KDJE54414-E	
	5	KDJE54415-E	
cjimportlibjar	3	KDJE54408-I	
	4	KDJE54409-I KDJE54410-E	
	5	KDJE54411-E	
cjimportres	3	KDJE54404-I	
	4	KDJE54405-I KDJE54406-E	
	5	KDJE54407-E	
cjlistapp	3	KDJE54480-I	Refer
	4	KDJE54481-I KDJE54482-E	
	5	KDJE54483-E	
cjlistjb	3	KDJE54512-I	
	4	KDJE54513-I KDJE54514-E	
	5	KDJE54515-E	

6. 監査ログを使用したシステムの監査

コマンド	図中の番号	出力される監査ログ のメッセージ ID	動作情報
cjlstlibjar	3	KDJE54492-I	
	4	KDJE54493-I	
		KDJE54494-E	
5	KDJE54495-E		
cjlstpool	3	KDJE54508-I	
	4	KDJE54509-I	
		KDJE54510-E	
5	KDJE54511-E		
cjlstrar	3	KDJE54488-I	
	4	KDJE54489-I	
		KDJE54490-E	
5	KDJE54491-E		
cjlstres	3	KDJE54484-I	
	4	KDJE54485-I	
		KDJE54486-E	
5	KDJE54487-E		
cjlstsec	3	KDJE54584-I	
	4	KDJE54585-I	
		KDJE54586-E	
5	KDJE54587-E		
cjlstthread	3	KDJE54496-I	
	4	KDJE54497-I	
		KDJE54498-E	
5	KDJE54499-E		
cjlstrn	3	KDJE54500-I	
	4	KDJE54501-I	
		KDJE54502-E	
5	KDJE54503-E		
cjlstrnfile	3	KDJE54504-I	
	4	KDJE54505-I	
		KDJE54506-E	
5	KDJE54507-E		
cjmapsec	3	KDJE54588-I	Update
	4	KDJE54589-I	

コマンド	図中の番号	出力される監査ログ のメッセージ ID	動作情報
cjreloadapp		KDJE54590-E	
	5	KDJE54591-E	
	3	KDJE54556-I	
	4	KDJE54557-I KDJE54558-E	
	5	KDJE54559-E	
cjrenameapp	3	KDJE54616-I	
	4	KDJE54617-I KDJE54618-E	
	5	KDJE54619-E	
cjreplaceapp	3	KDJE54536-I	
	4	KDJE54537-I KDJE54538-E	
	5	KDJE54539-E	
cjresumepool	3	KDJE54564-I	Start
	4	KDJE54565-I KDJE54566-E	
	5	KDJE54567-E	
cjsetappprop	3	KDJE54460-I	Update
	4	KDJE54461-I KDJE54462-E	
	5	KDJE54463-E	
cjsetjbprop	3	KDJE54472-I	
	4	KDJE54473-I KDJE54474-E	
	5	KDJE54475-E	
cjsetrarprop	3	KDJE54468-I	
	4	KDJE54469-I KDJE54470-E	
	5	KDJE54471-E	
cjsetresprop	3	KDJE54464-I	
	4	KDJE54465-I KDJE54466-E	
	5	KDJE54467-E	

6. 監査ログを使用したシステムの監査

コマンド	図中の番号	出力される監査ログ のメッセージ ID	動作情報
cjstartapp	3	KDJE54520-I	Start
	4	KDJE54521-I	
		KDJE54522-E	
5	KDJE54523-E		
cjstartjb	3	KDJE54548-I	
	4	KDJE54549-I	
		KDJE54550-E	
5	KDJE54551-E		
cjstartrar	3	KDJE54528-I	
	4	KDJE54529-I	
		KDJE54530-E	
5	KDJE54531-E		
cjstopapp	3	KDJE54524-I	Stop
	4	KDJE54525-I	
		KDJE54526-E	
5	KDJE54527-E		
cjstopjb	3	KDJE54552-I	
	4	KDJE54553-I	
		KDJE54554-E	
5	KDJE54555-E		
cjstoprar	3	KDJE54532-I	
	4	KDJE54533-I	
		KDJE54534-E	
5	KDJE54535-E		
cjstopthread	3	KDJE54544-I	
	4	KDJE54545-I	
		KDJE54546-E	
5	KDJE54547-E		
cjsuspendpool	3	KDJE54560-I	
	4	KDJE54561-I	
		KDJE54562-E	
5	KDJE54563-E		
cjtestres	3	KDJE54516-I	Refer
	4	KDJE54517-I	



コマンド	図中の番号	出力される監査ログ のメッセージ ID	動作情報
cjundeployrar		KDJE54518-E	Update
	5	KDJE54519-E	
	3	KDJE54572-I	
	4	KDJE54573-I	
		KDJE54574-E	
5	KDJE54575-E		
cjunmapsec	3	KDJE54592-I	
	4	KDJE54593-I	
		KDJE54594-E	
	5	KDJE54595-E	
	cjunupdateapp	3	
4		KDJE54541-I	
		KDJE54542-E	
5		KDJE54543-E	

なお、それぞれのコマンドが出力する監査ログの動作情報ごとに、図中の「1」、「2」、「6」および「7」で出力されるメッセージが異なります。動作情報と図中の「1」、「2」、「6」および「7」で出力されるメッセージ ID の対応を次の表に示します。

表 6-13 動作情報とメッセージ ID の対応 (cjresetsv 以外のサーバ管理コマンド)

動作情報	図中の番号	メッセージ ID
Start	1	KDJE54624-I
	2	KDJE54141-I
	6	KDJE54142-I
	7	KDJE54625-I
		KDJE54626-E
Stop	1	KDJE54627-I
	2	KDJE54143-I
	6	KDJE54144-I
	7	KDJE54628-I
		KDJE54629-E
Refer	1	KDJE54630-I
	2	KDJE54145-I
	6	KDJE54146-I

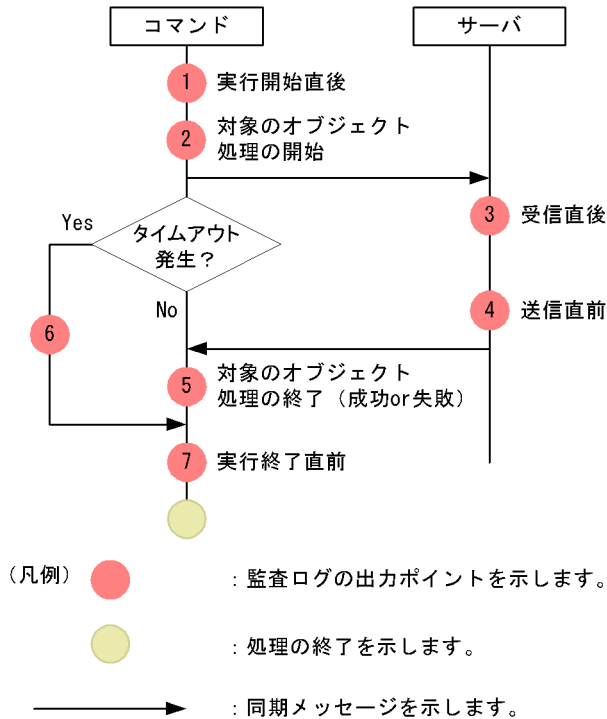
## 6. 監査ログを使用したシステムの監査

動作情報	図中の番号	メッセージ ID
	7	KDJE54631-I
		KDJE54632-E
Add	1	KDJE54633-I
	2	KDJE54147-I
	6	KDJE54148-I
	7	KDJE54634-I
		KDJE54635-E
Update	1	KDJE54636-I
	2	KDJE54149-I
	6	KDJE54150-I
	7	KDJE54637-I
		KDJE54638-E
Delete	1	KDJE54639-I
	2	KDJE54151-I
	6	KDJE54152-I
	7	KDJE54640-I
		KDJE54641-E

### (4) サーバ管理コマンド (cjresetsv)

サーバ管理コマンド (cjresetsv) の監査ログの出力ポイントを次の図に示します。

図 6-4 監査ログの出力ポイント (cjresetsv)



この図に示した監査ログの出力ポイントと出力される監査ログのメッセージ ID の対応を次の表に示します。

表 6-14 監査ログの出力ポイントとメッセージ ID の対応 (cjresetsv)

コマンド	図中の番号	出力される監査ログのメッセージ ID	
cjresetsv	1	KDJE54642-I	
	2	KDJE54596-I	
	3	KDJE54153-I	
	4	KDJE54154-I	
	5		KDJE54597-I
			KDJE54598-E
	6	KDJE54599-E	
	7		KDJE54643-I
KDJE54644-E			

## 6.4.2 バッチサーバで使用するコマンドの監査ログの出力ポイント

バッチサーバで使用するコマンドの監査ログの出力ポイントについて説明します。

ここでは、バッチサーバで使用するコマンドのうち、cjexecjob コマンド、および ckilljob コマンドの監査ログの出力ポイントについて説明します。

バッチサーバで使用するそのほかのコマンドについては、「6.4.1 J2EE サーバで使用するコマンドの監査ログの出力ポイント」の次の表に示した個所を参照してください。

表 6-15 バッチサーバで使用するそのほかのコマンドの参照先

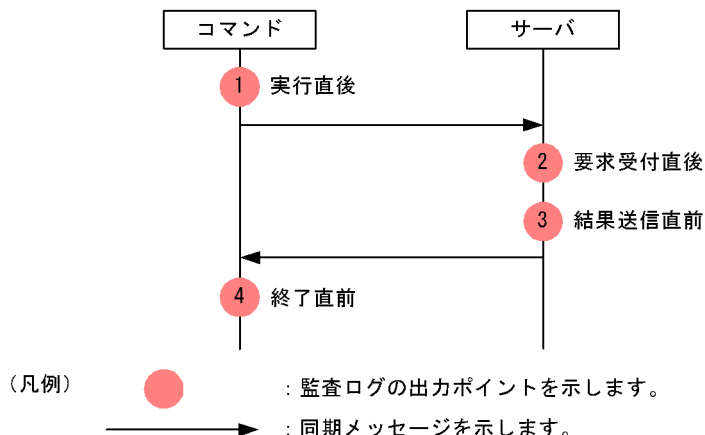
バッチサーバで使用するコマンド	参照先
サーバプロセスを起動・停止するコマンド（サーバ管理コマンド以外）	6.4.1(1)
サーバプロセスを起動・停止するコマンド以外のコマンド（サーバ管理コマンド以外）	6.4.1(2)
サーバ管理コマンド（cjresetsv 以外のサーバ管理コマンド）	6.4.1(3)
サーバ管理コマンド（cjresetsv）	6.4.1(4)

注 「6.4.1(3) サーバ管理コマンド（cjresetsv 以外のサーバ管理コマンド）」で説明しているコマンドのうち、バッチサーバで利用できるのは次のコマンドだけです。

cclearpool, cccopyres, cjdeleteres, cjdeployrar, cjexportapp, cjexportrar, cjgetrarprop, cjgetresprop, cjimportres, cjlistpool, cjlistrar, cjlistres, cjlistthread, cjresumepool, cjsetrarprop, cjsetresprop, cjstartrar, cjstoprar, cjstopthread, cjsuspendpool, cjtestres, cjundeployrar, cjunmapsec

cjexecjob コマンド、および ckilljob コマンドの監査ログの出力ポイントを次の図に示します。

図 6-5 監査ログの出力ポイント（cjexecjob コマンド、および ckilljob コマンド）



この図に示した監査ログの出力ポイントと出力される監査ログのメッセージ ID の対応を次の表に示します。

表 6-16 監査ログの出力ポイントとメッセージ ID の対応 (cjexecjob コマンド, および ckilljob コマンド)

コマンド	図中の番号	出力される監査ログのメッセージ ID
cjexecjob	1	KDJE54156-I
	4	KDJE54157-I
		KDJE54158-E <sup>1</sup>
ckilljob	1	KDJE54162-I
	4	KDJE54163-I
		KDJE54164-W
cjstartsv (cjexecjob コマンド実行時)	2	KDJE54159-I
	3	KDJE54160-I
		KDJE54161-E <sup>2</sup>
cjstartsv (ckilljob コマンド実行時)	2	KDJE54165-I
	3	KDJE54166-I
		KDJE54167-E

注 1 KDJE54158-E は次の場合に出力されます。

- 指定したサーバが見つからないなどの原因でバッチの実行に失敗した場合
- バッチの実行時, 0 以外の終了コードが出力された場合

注 2 KDJE54161-E は次の場合に出力されます。

- main メソッドが見つからないなどの原因でバッチの実行に失敗した場合
- バッチの実行時, 0 以外の終了コードが出力された場合

### 6.4.3 性能解析トレース・CTM で使用するコマンドの監査ログの出力ポイント

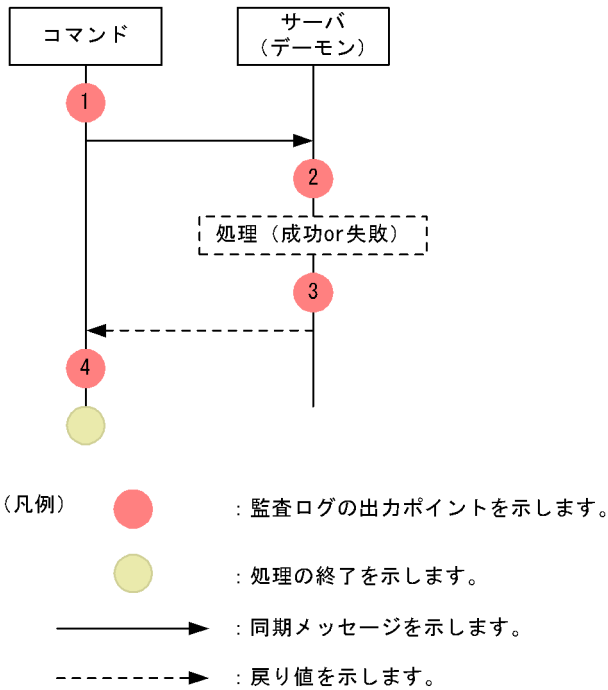
性能解析トレース・CTM で使用するコマンドの監査ログの出力ポイントについて説明します。

#### (1) サーバプロセスを起動・停止するコマンド

サーバプロセス(デーモン)を起動・停止するコマンドの監査ログの出力ポイントを次の図に示します。

## 6. 監査ログを使用したシステムの監査

図 6-6 監査ログの出力ポイント（サーバプロセスを起動・停止するコマンド）



この図に示した監査ログの出力ポイントと出力される監査ログのメッセージ ID の対応を次の表に示します。

表 6-17 監査ログの出力ポイントとメッセージ ID の対応（サーバプロセスを起動・停止するコマンド）

コマンド	図中の番号	出力される監査ログのメッセージ ID
cprfstart	1	KFCT81000-I
	4	KFCT81001-I
		KFCT81002-E
cprfstop	1	KFCT81020-I
	4	KFCT81021-I
		KFCT81022-E
ctmdmstart	1	KFCT91000-I
	4	KFCT91001-I
		KFCT91002-E
ctmdmstop	1	KFCT91020-I
	4	KFCT91021-I
		KFCT91022-E

コマンド	図中の番号	出力される監査ログのメッセージ ID
ctmstart	1	KFCT91040-I
	4	KFCT91041-I
		KFCT91042-E
ctmstop	1	KFCT91060-I
	4	KFCT91061-I
		KFCT91062-E

なお、それぞれのコマンドがリクエストを送信するサーバプロセス（デーモン）ごとに、図中の「2」および「3」で出力されるメッセージが異なります。サーバプロセスの種類と図中の「2」および「3」で出力されるメッセージ ID の対応を次の表に示します。

表 6-18 サーバプロセス（デーモン）の種類とメッセージ ID の対応（サーバプロセスを起動・停止するコマンド）

サーバプロセス（デーモン）	図中の番号	メッセージ ID
PRF デーモン	2	KFCT81010-I
		KFCT81030-I
	3	KFCT81011-I
		KFCT81012-E
		KFCT81031-I
		KFCT81032-E
CTM ドメインマネージャ	2	KFCT91010-I
		KFCT91030-I
	3	KFCT91011-I
		KFCT91012-E
		KFCT91031-I
		KFCT91032-E
CTM デーモン	2	KFCT91050-I
		KFCT91070-I
	3	KFCT91051-I
		KFCT91052-E
		KFCT91071-I
		KFCT91072-E

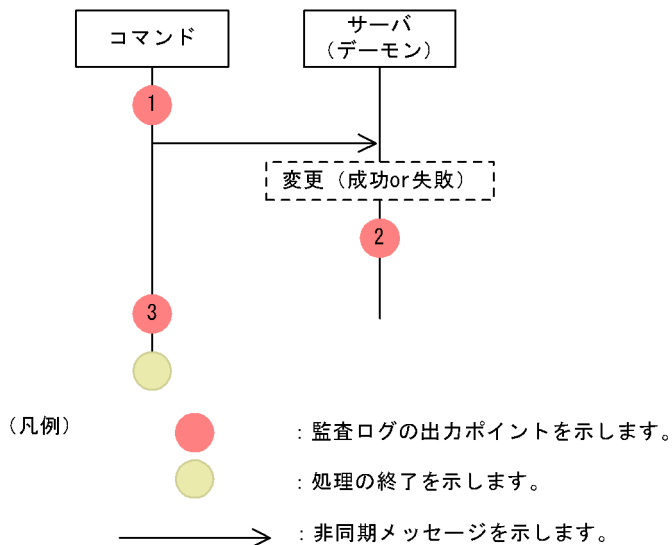
## (2) プロセスへの通信・共有メモリへのアクセスを実行するコマンド

プロセスへの通信・共有メモリへのアクセスを実行するコマンドの監査ログの出力ポイ

## 6. 監査ログを使用したシステムの監査

ントを次の図に示します。

図 6-7 監査ログの出力ポイント（プロセスへの通信・共有メモリへのアクセスを実行するコマンド）



この図に示した監査ログの出力ポイントと出力される監査ログのメッセージ ID の対応を次の表に示します。

表 6-19 監査ログの出力ポイントとメッセージ ID の対応（プロセスへの通信・共有メモリへのアクセスを実行するコマンド）

コマンド	図中の番号	出力される監査ログのメッセージ ID
ctmstsstart	1	KFCT91080-I
	3	KFCT91081-I
		KFCT91082-E
ctmstsstop	1	KFCT91100-I
	3	KFCT91101-I
		KFCT91102-E
-	2	KFCT91091-I
		KFCT91111-I

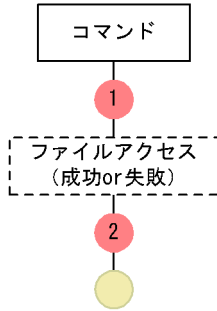
注 この表で示しているすべてのコマンドに共通で出力されるメッセージです。

### (3) 情報を参照するコマンド

情報を参照するコマンドの監査ログの出力ポイントを次の図に示します。



図 6-8 監査ログの出力ポイント（情報を参照するコマンド）



(凡例) ● : 監査ログの出力ポイントを示します。

● : 処理の終了を示します。

この図に示した監査ログの出力ポイントと出力される監査ログのメッセージ ID の対応を次の表に示します。

表 6-20 監査ログの出力ポイントとメッセージ ID の対応（情報を参照するコマンド）

コマンド	図中の番号	出力される監査ログのメッセージ ID
cprfgetpid	1	KFCT84000-I
	2	KFCT84001-I
		KFCT84002-E
ctmgetpid	1	KFCT94000-I
	2	KFCT94001-I
		KFCT94002-E

#### 6.4.4 Management Server で使用するコマンドの監査ログの出力ポイント

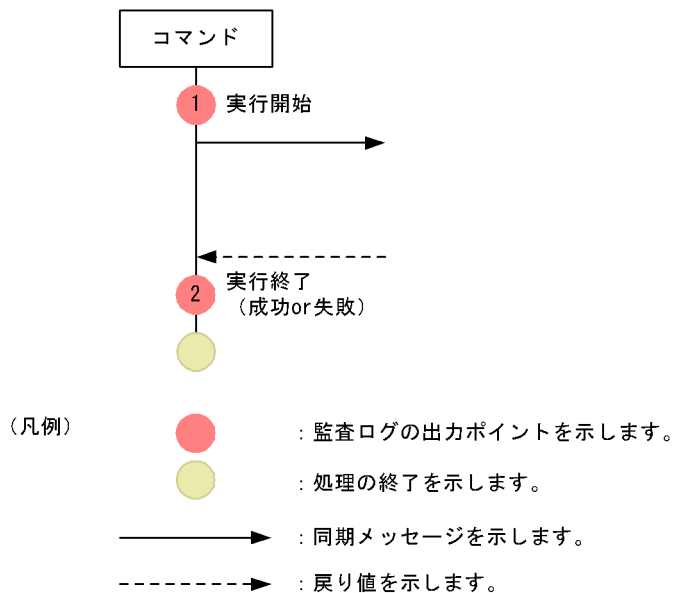
Management Server で使用するコマンドの監査ログの出力ポイントについて説明します。

##### (1) Management Server のセットアップコマンド

Management Server のセットアップコマンドの監査ログの出力ポイントを次の図に示します。

## 6. 監査ログを使用したシステムの監査

図 6-9 監査ログの出力ポイント (Management Server のセットアップコマンド)



この図に示した監査ログの出力ポイントと出力される監査ログのメッセージ ID の対応を次の表に示します。

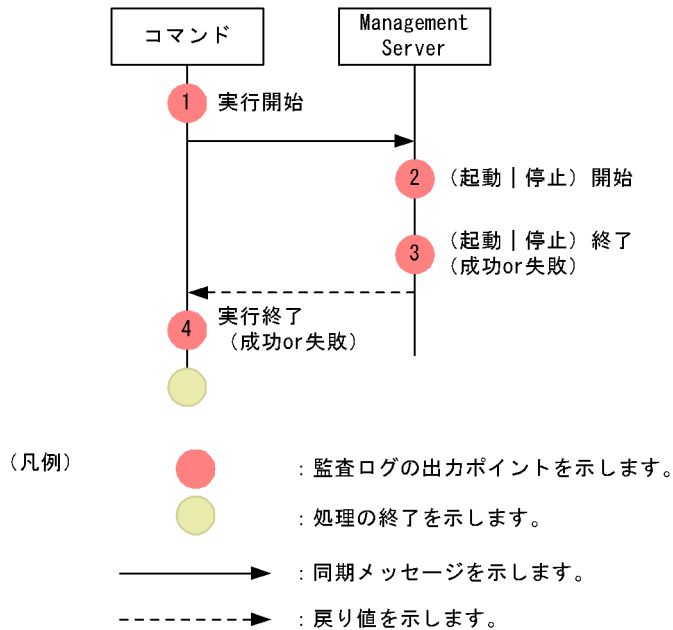
表 6-21 監査ログの出力ポイントとメッセージ ID の対応 (Management Server のセットアップコマンド)

コマンド (オプション)	図中の番号	出力される監査ログのメッセージ ID
mngsvrctl ( setup )	1	KEOS80006-I
	2	KEOS80007-I
		KEOS80008-E

### (2) Management Server を起動・停止するコマンド

Management Server を起動・停止するコマンドの監査ログの出力ポイントを次の図に示します。

図 6-10 監査ログの出力ポイント（Management Server を起動・停止するコマンドコマンド）



この図に示した監査ログの出力ポイントと出力される監査ログのメッセージ ID の対応を次の表に示します。

表 6-22 監査ログの出力ポイントとメッセージ ID の対応（Management Server を起動・停止するコマンド）

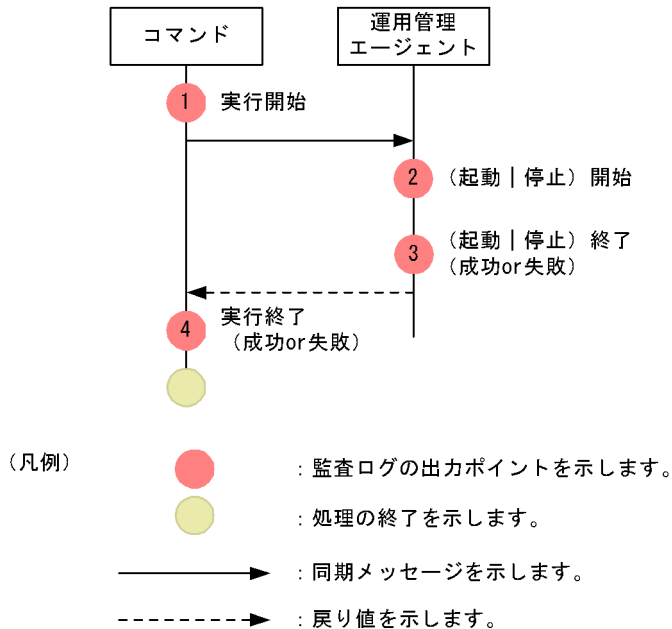
コマンド（サブコマンド）	図中の番号	出力される監査ログのメッセージ ID
mngsvrctl ( start )	1	KEOS80000-I
	2	KEOS80108-I
	3	KEOS80109-I
		KEOS80110-E
	4	KEOS80001-I
		KEOS80002-E
	mngsvrctl ( stop )	1
	2	KEOS80111-I
	3	KEOS80112-I
		KEOS80113-E
	4	KEOS80004-I
		KEOS80005-E

6. 監査ログを使用したシステムの監査

(3) 運用管理エージェントの起動・停止を実行するコマンド

運用管理エージェントの起動・停止の監査ログの出力ポイントを次の図に示します。

図 6-11 監査ログの出力ポイント（運用管理エージェントの起動・停止）



この図に示した監査ログの出力ポイントと出力される監査ログのメッセージ ID の対応を次の表に示します。

表 6-23 監査ログの出力ポイントとメッセージ ID の対応（運用管理エージェントの起動・停止）

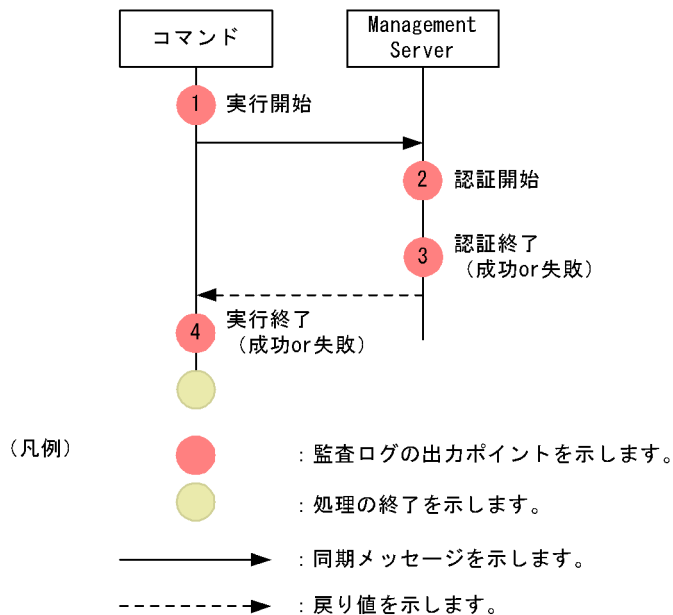
コマンド（サブコマンド）	図中の番号	出力される監査ログのメッセージ ID
adminagentctl ( start )	1	KEOS80009-I
	2	KEOS80114-I
	3	KEOS80115-I
		KEOS80116-E
	4	KEOS80010-I
KEOS80011-E		
adminagentctl ( stop )	1	KEOS80012-I
	2	KEOS80117-I
	3	KEOS80118-I
		KEOS80119-E

コマンド (サブコマンド)	図中の番号	出力される監査ログのメッセージ ID
	4	KEOS80013-I
		KEOS80014-E

#### (4) Management Server による承認処理を実行するコマンド

Management Server による承認処理を実行するコマンドの監査ログの出力ポイントを次の図に示します。

図 6-12 監査ログの出力ポイント (Management Server による承認処理を実行するコマンド)



この図に示した監査ログの出力ポイントと出力される監査ログのメッセージ ID の対応を次の表に示します。

表 6-24 監査ログの出力ポイントとメッセージ ID の対応 (Management Server による承認処理を実行するコマンド)

コマンド	図中の番号	出力される監査ログのメッセージ ID
cmx_test_lb	1	KEOS80036-I
	4	KEOS80037-I
		KEOS80038-E
cmx_list_status	1	KEOS80045-I

## 6. 監査ログを使用したシステムの監査

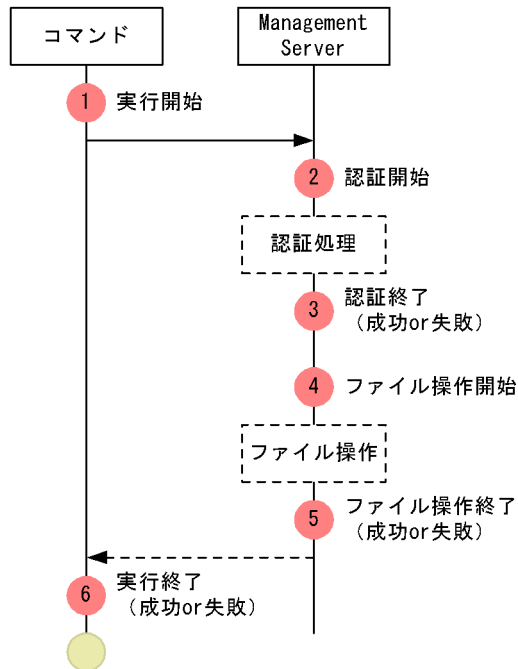
コマンド	図中の番号	出力される監査ログのメッセージ ID
	4	KEOS80046-I
		KEOS80047-E
cmx_start_resource	1	KEOS80063-I
	4	KEOS80064-I
		KEOS80065-E
cmx_stop_resource	1	KEOS80066-I
	4	KEOS80067-I
		KEOS80068-E
cmx_start_application	1	KEOS80087-I
	4	KEOS80088-I
		KEOS80089-E
cmx_stop_application	1	KEOS80090-I
	4	KEOS80091-I
		KEOS80092-E
cmx_list_model	1	KEOS80102-I
	4	KEOS80103-I
		KEOS80104-E
cmx_export_model	1	KEOS80105-I
	4	KEOS80106-I
		KEOS80107-E
-	2	KEOS80120-I
	3	KEOS80121-I
		KEOS80122-E

注 この表で示しているすべてのコマンドに共通で出力されるメッセージです。

### (5) Management Server によるファイル操作を実行するコマンド

Management Server によるファイル操作を実行するコマンドの監査ログの出力ポイントを次の図に示します。

図 6-13 監査ログの出力ポイント（Management Server によるファイル操作を実行するコマンド）



- (凡例)
- : 監査ログの出力ポイントを示します。
  - : 処理の終了を示します。
  - ▶ : 同期メッセージを示します。
  - - - - -▶ : 戻り値を示します。

この図に示した監査ログの出力ポイントと出力される監査ログのメッセージ ID の対応を次の表に示します。

表 6-25 監査ログの出力ポイントとメッセージ ID の対応（Management Server によるファイル操作を実行するコマンド）

コマンド	図中の番号	出力される監査ログのメッセージ ID
mngsvrctl setup	1	KEOS80006-I
	6	KEOS80007-I
		KEOS80008-E
cmx_admin_passwd	1	KEOS80015-I
	6	KEOS80016-I
		KEOS80017-E

## 6. 監査ログを使用したシステムの監査

コマンド	図中の番号	出力される監査ログのメッセージ ID
cmx_build_model	1	KEOS80018-I
	6	KEOS80019-I
		KEOS80020-E
cmx_change_model	1	KEOS80024-I
	6	KEOS80025-I
		KEOS80026-E
cmx_delete_system	1	KEOS80027-I
	6	KEOS80028-I
		KEOS80029-E
cmx_resume_lb	1	KEOS80030-I
	6	KEOS80031-I
		KEOS80032-E
cmx_scaleout_host	1	KEOS80033-I
	6	KEOS80034-I
		KEOS80035-E
cmx_add_serverref	1	KEOS80039-I
	6	KEOS80040-I
		KEOS80041-E
cmx_delete_serverref	1	KEOS80042-I
	6	KEOS80043-I
		KEOS80044-E
cmx_start_target	1	KEOS80048-I
	6	KEOS80049-I
		KEOS80050-E
cmx_stop_target	1	KEOS80051-I
	6	KEOS80052-I
		KEOS80053-E
cmx_define_resource	1	KEOS80054-I
	6	KEOS80055-I
		KEOS80056-E
cmx_deploy_resource	1	KEOS80057-I
	6	KEOS80058-I
		KEOS80059-E
cmx_register_resource	1	KEOS80060-I



コマンド	図中の番号	出力される監査ログのメッセージ ID
	6	KEOS80061-I
		KEOS80062-E
cmx_undefine_resource	1	KEOS80069-I
	6	KEOS80070-I KEOS80071-E
cmx_undeploy_resource	1	KEOS80072-I
	6	KEOS80073-I KEOS80074-E
cmx_unregister_resource	1	KEOS80075-I
	6	KEOS80076-I KEOS80077-E
cmx_define_application	1	KEOS80078-I
	6	KEOS80079-I KEOS80080-E
cmx_deploy_application	1	KEOS80081-I
	6	KEOS80082-I KEOS80083-E
cmx_register_application	1	KEOS80084-I
	6	KEOS80085-I KEOS80086-E
cmx_undefined_application	1	KEOS80093-I
	6	KEOS80094-I KEOS80095-E
cmx_undeploy_application	1	KEOS80096-I
	6	KEOS80097-I KEOS80098-E
cmx_unregister_application	1	KEOS80099-I
	6	KEOS80100-I KEOS80101-E
-	2	KEOS80120-I
	3	KEOS80121-I KEOS80122-E
	4	KEOS80123-I

6. 監査ログを使用したシステムの監査

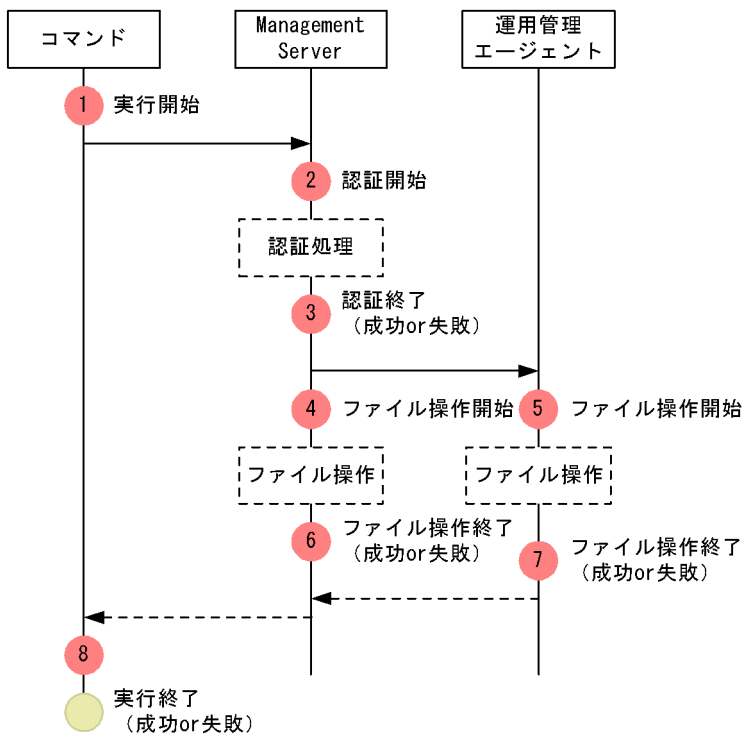
コマンド	図中の番号	出力される監査ログのメッセージ ID
	5	KEOS80124-I
		KEOS80125-E

注 この表で示しているすべてのコマンドに共通で出力されるメッセージです。

(6) Management Server・運用管理エージェントによるファイル操作を実行するコマンド

Management Server・運用管理エージェントによるファイル操作を実行するコマンドの監査ログの出力ポイントを次の図に示します。

図 6-14 監査ログの出力ポイント (Management Server・運用管理エージェントによるファイル操作を実行するコマンド)



- (凡例)
- : 監査ログの出力ポイントを示します。
  - : 処理の終了を示します。
  - > : 同期メッセージを示します。
  - - - - -> : 戻り値を示します。

この図に示した監査ログの出力ポイントと出力される監査ログのメッセージ ID の対応を次の表に示します。

表 6-26 監査ログの出力ポイントとメッセージ ID の対応 ( Management Server ・ 運用管理エージェントによるファイル操作を実行するコマンド )

コマンド	図中の番号	出力される監査ログのメッセージ ID
cmx_build_system	1	KEOS80021-I
	2	KEOS80120-I
	3	KEOS80121-I
		KEOS80122-E
	4	KEOS80123-I
	6	KEOS80124-I
		KEOS80125-E
	8	KEOS80022-I
KEOS80023-E		

なお、それぞれのコマンドが操作を実行するファイルごとに、図中の「5」および「7」で出力されるメッセージが異なります。操作を実行するファイルと図中の「5」および「7」で出力されるメッセージ ID の対応を次の表に示します。

表 6-27 操作を実行するファイルと出力されるメッセージ ID の対応

操作を実行するファイル	図中の番号	出力される監査ログのメッセージ ID
運用監視エージェントの設定ファイル	5	KEOS80126-I
	7	KEOS80127-I
		KEOS80128-E
J2EE サーバ用 JP1 イベント発行用プロパティファイル	5	KEOS80129-I
	7	KEOS80130-I
		KEOS80131-E
Management イベント発行用プロパティファイル	5	KEOS80132-I
	7	KEOS80133-I
		KEOS80134-E
J2EE サーバ用ユーザプロパティファイル	5	KEOS80135-I

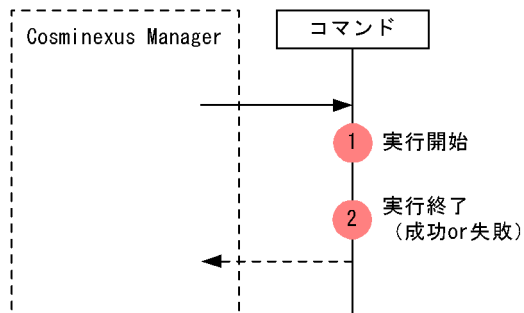
## 6. 監査ログを使用したシステムの監査

操作を実行するファイル	図中の番号	出力される監査ログのメッセージ ID	
	7	KEOS80136-I	
		KEOS80137-E	
J2EE サーバ用オプション定義ファイル	5	KEOS80138-I	
	7	KEOS80139-I	
Hitachi Web Server 用リダイレクタ動作定義ファイル	5	KEOS80144-I	
		7	KEOS80145-I
ワーカ定義ファイル	5	KEOS80147-I	
		7	KEOS80148-I
			KEOS80149-E

### (7) Hitachi Web Server を操作するコマンド

Hitachi Web Server を操作するコマンドの監査ログの出力ポイントを次の図に示します。

図 6-15 監査ログの出力ポイント (Hitachi Web Server を操作するコマンド)



- (凡例)
- : 監査ログの出力ポイントを示します。
  - : 同期メッセージを示します。
  - : 戻り値を示します。

この図に示した監査ログの出力ポイントと出力される監査ログのメッセージ ID の対応を次の表に示します。

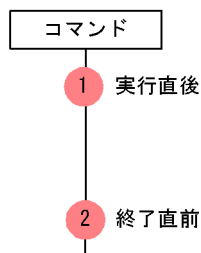
表 6-28 監査ログの出力ポイントとメッセージ ID の対応 (Hitachi Web Server を操作するコマンド)

コマンド	図中の番号	出力される監査ログのメッセージ ID
設定ファイル編集コマンド	1	KAWS10501-I
	2	KAWS10502-I
		KAWS10503-E
	1	KAWS10504-I
	2	KAWS10505-I
		KAWS10506-E
hwsserveredit	1	KAWS10601-I
	2	KAWS10602-I
		KAWS10603-E
	1	KAWS10604-I
	2	KAWS10605-I
		KAWS10606-E
	1	KAWS10607-I
	2	KAWS10608-I
		KAWS10609-E

#### 6.4.5 EJB クライアントアプリケーションで使用するコマンドの監査ログの出力ポイント

EJB クライアントアプリケーションで使用するコマンドの監査ログの出力ポイントについて説明します。

図 6-16 監査ログの出力ポイント (EJB クライアントアプリケーションで使用するコマンド)



(凡例) ● : 監査ログの出力ポイントを示します。

この図に示した監査ログの出力ポイントと出力される監査ログのメッセージ ID の対応を次の表に示します。

## 6. 監査ログを使用したシステムの監査

表 6-29 監査ログの出力ポイントとメッセージ ID の対応 (EJB クライアントアプリケーションで使用するコマンド)

コマンド	図中の番号	出力される監査ログのメッセージ ID
cjcstartap	1	KDJE54129-I
	2	KDJE54130-I
		KDJE54131-E
cjcldumpap	1	KDJE54132-I
	2	KDJE54133-I
		KDJE54134-E
cjcldellog	1	KDJE54135-I
	2	KDJE54136-I
		KDJE54137-E

注 KDJE54131-E は次の場合に出力されます。

- JavaVM の初期化など、コマンドの内部処理が失敗した場合
- Java アプリケーションの実行後に終了コードが 0 以外の場合

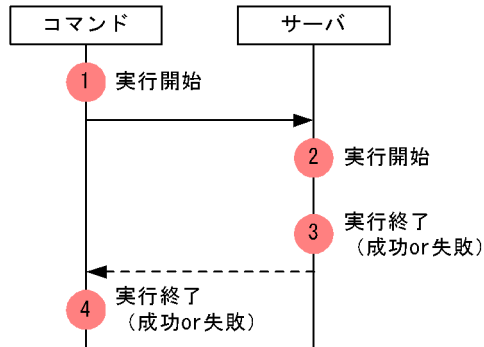
### 6.4.6 Hitachi Web Server で使用するコマンド、および Hitachi Web Server に対する操作の監査ログの出力ポイント (Windows の場合)

Windows の場合の Hitachi Web Server で使用するコマンドおよび Hitachi Web Server に対する操作の監査ログの出力ポイントについて説明します。

#### (1) Hitachi Web Server の起動・停止を実行するコマンド

Hitachi Web Server の起動・停止を実行するコマンドの監査ログの出力ポイントを次の図に示します。

図 6-17 監査ログの出力ポイント（Hitachi Web Server の起動・停止を実行するコマンド）



- (凡例)
- : 監査ログの出力ポイントを示します。
  - : 同期メッセージを示します。
  - : 戻り値を示します。

この図に示した監査ログの出力ポイントと出力される監査ログのメッセージ ID の対応を次の表に示します。

表 6-30 監査ログの出力ポイントとメッセージ ID の対応（Hitachi Web Server の起動・停止を実行するコマンド）

コマンド（オプション）	図中の番号	出力される監査ログのメッセージ ID
httpsd (-k start)	1	KAWS10001-I
	2	KAWS10004-I
	3	KAWS10005-I
		KAWS10006-E
	4	KAWS10002-I
		KAWS10003-E
httpsd (-k stop)	1	KAWS10010-I
	2	KAWS10013-I
	3	KAWS10014-I
		KAWS10015-E
	4	KAWS10011-I
		KAWS10012-E
httpsd (-k restart)	1	KAWS10019-I
	2	KAWS10022-I

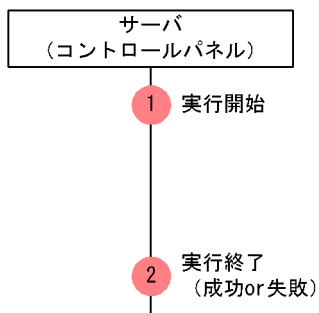
## 6. 監査ログを使用したシステムの監査

コマンド (オプション)	図中の番号	出力される監査ログのメッセージ ID
	3	KAWS10023-I
		KAWS10024-E
	4	KAWS10020-I
		KAWS10021-E
httpsd (-k gracefulstop)	1	KAWS10025-I
	2	KAWS10028-I
	3	KAWS10029-I
		KAWS10030-E
	4	KAWS10026-I
		KAWS10027-E

### (2) Hitachi Web Server の起動・停止を実行する操作 (コントロールパネルから)

コントロールパネルから Hitachi Web Server の起動・停止を実行する操作の監査ログの出力ポイントを次の図に示します。

図 6-18 監査ログの出力ポイント (コントロールパネルから Hitachi Web Server の起動・停止を実行する操作)



(凡例) ● : 監査ログの出力ポイントを示します。

この図に示した監査ログの出力ポイントと出力される監査ログのメッセージ ID の対応を次の表に示します。



表 6-31 監査ログの出力ポイントとメッセージ ID の対応 (コントロールパネルから Hitachi Web Server の起動・停止を実行する操作)

図中の番号	出力される監査ログのメッセージ ID
1	KAWS10004-I
	KAWS10007-I
	KAWS10013-I
	KAWS10016-I
2	KAWS10005-I
	KAWS10006-E
	KAWS10008-I
	KAWS10009-E
	KAWS10014-I
	KAWS10015-E
	KAWS10017-I
	KAWS10018-E

#### 6.4.7 Hitachi Web Server で使用するコマンド，および Hitachi Web Server に対する操作の監査ログの出力ポイント (UNIX の場合)

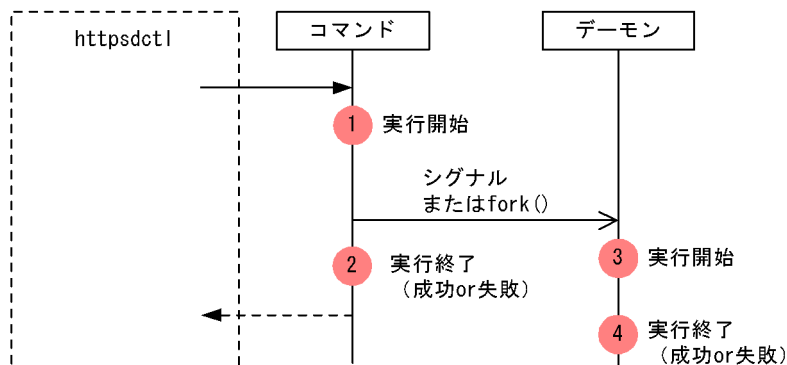
UNIX の場合の Hitachi Web Server で使用するコマンドおよび Hitachi Web Server に対する操作の監査ログの出力ポイントについて説明します。

##### (1) Hitachi Web Server の起動・停止を実行するコマンド (httpsdctl ユティリティから)

httpsdctl ユティリティから Hitachi Web Server の起動・停止を実行するコマンドの監査ログの出力ポイントを次の図に示します。

## 6. 監査ログを使用したシステムの監査

図 6-19 監査ログの出力ポイント（httpsdctl ユティリティから Hitachi Web Server の起動・停止を実行するコマンド）



- (凡例)
- : 監査ログの出力ポイントを示します。
  - : 同期メッセージを示します。
  - >> : 非同期メッセージを示します。
  - >> : 戻り値を示します。

この図に示した監査ログの出力ポイントと出力される監査ログのメッセージ ID の対応を次の表に示します。

表 6-32 監査ログの出力ポイントとメッセージ ID の対応（httpsdctl ユティリティから Hitachi Web Server の起動・停止を実行するコマンド）

コマンド（httpsdctl ユティリティの操作）	図中の番号	出力される監査ログのメッセージ ID
httpsd ( httpsdctl start )	1	KAWS10101-I
	2	KAWS10102-I
		KAWS10103-E
	3	KAWS10151-I
4	KAWS10152-I	
	KAWS10153-E	
httpsd ( httpsdctl restart )	1	KAWS10104-I
	2	KAWS10105-I
		KAWS10106-E
		KAWS10116-I
KAWS10117-E		
3	KAWS10154-I	
4	KAWS10155-I	

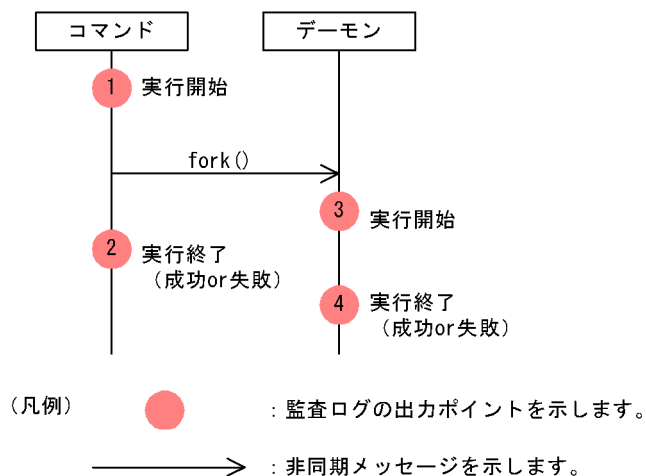
コマンド ( httpsdctl ユティリティの操作 )	図中の番号	出力される監査ログのメッセージ ID	
		KAWS10156-E	
httpsd ( httpsdctl graceful )	1	KAWS10107-I	
	2	KAWS10108-I	
		KAWS10109-E	
		KAWS10118-I	
		KAWS10119-E	
	3	KAWS10157-I	
	4	KAWS10158-I	
		KAWS10159-E	
httpsd ( httpsdctl stop )	1	KAWS10110-I	
	2	KAWS10111-I	
		KAWS10112-E	
	3	KAWS10160-I	
	4	KAWS10161-I	
		KAWS10162-E	
	httpsd ( httpsdctl gracefulstop )	1	KAWS10113-I
		2	KAWS10114-I
KAWS10115-E			
3		KAWS10163-I	
4		KAWS10164-I	
		KAWS10165-E	

## (2) httpsd コマンド ( コマンドの直接実行 )

httpsd コマンドを直接実行する場合の監査ログの出力ポイントを次の図に示します。

## 6. 監査ログを使用したシステムの監査

図 6-20 監査ログの出力ポイント（直接実行する httpsd コマンド）



この図に示した監査ログの出力ポイントと出力される監査ログのメッセージ ID の対応を次の表に示します。

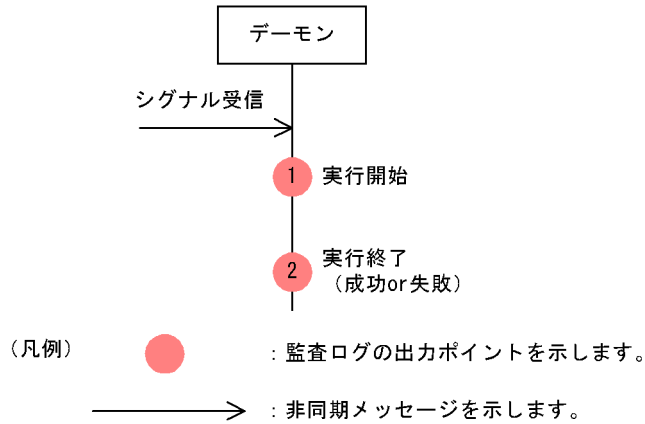
表 6-33 監査ログの出力ポイントとメッセージ ID の対応（直接実行する httpsd コマンド）

コマンド	図中の番号	出力される監査ログのメッセージ ID
httpsd	1	KAWS10101-I
	2	KAWS10102-I
		KAWS10103-E
	3	KAWS10151-I
	4	KAWS10152-I
		KAWS10153-E

### (3) シグナル送信によるデーモンプロセスの操作

シグナル送信によってデーモンプロセスを操作するときの監査ログの出力ポイントを次の図に示します。

図 6-21 監査ログの出力ポイント（シグナル送信によるデーモンプロセスの操作）



この図に示した監査ログの出力ポイントと出力される監査ログのメッセージ ID の対応を次の表に示します。

表 6-34 監査ログの出力ポイントとメッセージ ID の対応（シグナル送信によるデーモンプロセスの操作）

シグナル送信による操作	図中の番号	出力される監査ログのメッセージ ID
再起動 (SIGHUP)	1	KAWS10154-I
	2	KAWS10155-I
		KAWS10156-E
再起動 (SIGUSR1)	1	KAWS10157-I
	2	KAWS10158-I
		KAWS10159-E
停止 (SIGTERM)	1	KAWS10160-I
	2	KAWS10161-I
		KAWS10162-E
計画停止 (SIGUSR2)	1	KAWS10163-I
	2	KAWS10164-I
		KAWS10165-E



# 7

## データベース監査証跡と連携した運用

この章では、データベースが提供する監査証跡の機能と連携して、データベース監査証跡に Cosminexus システムの情報を出力した場合の運用方法について説明します。

---

7.1 データベース監査証跡と連携した運用の概要

---

7.2 データベース監査証跡と連携した運用例

---

7.3 注意事項

---

## 7.1 データベース監査証跡と連携した運用の概要

---

この節では、データベースが提供する監査証跡の機能と連携し、Cosminexus システムの情報をデータベース監査証跡として出力した場合の運用の概要について説明します。

データベース監査証跡とは、データベースの監査に利用するためのデータベースの操作記録です。監査者がデータベース監査証跡を調査することによって、監査対象のテーブルに決められた操作権限どおりに運用が行われているか、不正なアクセスがないかをチェックできます。データベース監査証跡の出力、取得、参照などをするための機能は、データベースが提供しています。データベース監査証跡の詳細については、データベースのマニュアルを参照してください。また、Cosminexus システムと連携できるデータベースについては、マニュアル「Cosminexus 機能解説」のデータベース監査証跡との連携に関する説明を参照してください。

Cosminexus システムでは、データベース監査証跡連携機能を使用して、Cosminexus システムの情報（ルートアプリケーション情報）をデータベース監査証跡に出力できます。データベース監査証跡に出力されたルートアプリケーション情報を利用して、データベースの監査対象のテーブルにアクセスしたリクエストを特定できます。

データベース監査証跡連携機能の詳細については、マニュアル「Cosminexus 機能解説」のデータベース監査証跡との連携に関する説明を、データベース監査証跡連携機能を使用するための設定については、マニュアル「Cosminexus システム構築ガイド」のデータベース監査証跡との連携の設定に関する説明を参照してください。



## 7.2 データベース監査証跡と連携した運用例

この節では、データベース監査証跡と連携した運用例について説明します。この運用例では、アプリケーションのユーザログ、データベース監査証跡、および性能解析トレースに出力されたルートアプリケーション情報を利用して、データベースの監査対象のテーブルにアクセスしたアプリケーションのログイン名やセッション ID を特定します。この場合、次の作業を実行します。

1. アプリケーションのユーザログを取得する
2. データベースの監査証跡情報を取得する
3. 性能解析トレースを取得する
4. 1. ~ 3. で取得した情報を突き合わせて、ログイン名、セッション ID を特定する

それぞれの出力例や作業について説明します。

### 7.2.1 アプリケーションのユーザログの取得

あらかじめ J2EE アプリケーションやバッチアプリケーションでユーザログを取得するための実装をしておく、次の情報がユーザログに出力されます。

- アプリケーションのログイン名
- ルートアプリケーション情報
- セッション ID

ここでは、J2EE アプリケーションのユーザログの実装例および出力例を示します。

#### 実装例

ログイン名、ルートアプリケーション情報、およびセッション ID を J2EE アプリケーションのユーザログに出力する場合の実装例を示します。なお、ログイン名は Web コンテナの Basic 認証または Form 認証でログインしたユーザ ID を取得して出力します。

```
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpSession;
import com.hitachi.software.ejb.application.prf.CprfTrace;

(中略)
public void doGet (HttpServletRequest req, HttpServletResponse res)
    throws ServletException, IOException {

    HttpSession ses = req.getSession();
    String sessionID = (ses != null) ? ses.getId() : null; // セッションIDを取得
    String rootAPInfo = CprfTrace.getRootApInfo(); // ルートアプリケーション情報を取得
    String userID = req.getRemoteUser(); // J2EEアプリケーションのログイン名を取得

    ユーザログ出力メソッド(sessionID,rootAPInfo,userID);
    :
```

#### 出力例

## 7. データベース監査証跡と連携した運用

J2EE アプリケーションのユーザログの出力例を示します。この例では、「セッション ID」、「ルートアプリケーション情報」、「ログイン名」の順で出力されています。

```
010D7AE670AEB55C022B6A4DA1C1E8B1CtEPmR9H, 10. 209. 15. 130/1234/0x0000000000000001, user1 :
```

### 7.2.2 データベースの監査証跡情報の取得

データベースが保持する監査証跡の情報を取得します。データベースの監査証跡表にデータ登録（監査証跡表へのデータロード）を行い、監査証跡表の「PRODUCT\_INFO\_1」列で監査対象のテーブルのルートアプリケーション情報を取得します。データベースの監査証跡の情報の取得方法については、マニュアル「HiRDB システム運用ガイド」を参照してください。監査証跡表の出力例を次に示します。

出力例

HiRDB の監査証跡表の出力例を示します。

列	EVENT_TYPE	EVENT_SUBTYPE	OBJECT_NAME	SQL_SOURCE	PRODUCT_INFO_1
行	ACS	INS	Table1	INSERT...	10. 209. 15. 130/1234/0x0000000000000001
:					

この例で出力されている「SQL\_SOURCE」列は、オプションを指定することで出力できます。オプションを指定して出力する列については、使用するシステムの要件に合わせて設定してください。オプションの設定方法、および監査証跡表の詳細については、マニュアル「HiRDB システム運用ガイド」を参照してください。

### 7.2.3 性能解析トレースの取得

性能解析トレースファイルを取得します。性能解析トレースの収集方法、および出力内容の詳細については、「3.2 性能解析トレースファイルの収集」を参照してください。性能解析トレースの出力例を次に示します。

出力例

```
Event , Date, Time , Time(msec/usec/nsec)··, RootAP IP , RootAP PID, RootAP CommonNo .  
0x8405, 2007/xx/yy, 12:00:02, 000/000/000··, 10. 209. 15. 130, 1234 , 0x0000000000000001 .  
0x8406, 2007/xx/yy, 12:00:02, 000/000/000··, 10. 209. 15. 130, 1234 , 0x0000000000000001 .  
:
```

この例では、性能解析トレースのトレース取得レベル（PRFトレース取得レベル）を「詳細レベル」にしてセッション ID を出力しています。PRFトレース取得レベルについては、使用するシステムの要件に合わせて設定してください。PRFトレース取得レベルの設定方法については、マニュアル「Cosminexus システム構築ガイド」のトラブルシューティングの資料取得の設定に関する説明を参照してください。また、PRFトレースレベルの変更方法については、マニュアル「Cosminexus リ

ファレンス コマンド編」の cprflvel コマンドの説明を参照してください。

## 7.2.4 情報の特定

アプリケーションのユーザログ、データベースの監査証跡表、および性能解析トレースの情報を突き合わせて、監査対象のテーブルにアクセスしたアプリケーションのログイン名、およびセッション ID を特定します。

1. データベースの監査証跡表に出力されている実行日時  
「EXEC\_DATE,EXEC\_TIME,EXEC\_TIME\_MICRO」の列の時間と、性能解析トレースの実行時間とを突き合わせて、監査対象のテーブルにアクセスしたリクエストのルートアプリケーション情報を特定します。
2. 1. で特定したルートアプリケーション情報と、アプリケーションのユーザログに出力されたルートアプリケーション情報とを突き合わせて、リクエストを実行したアプリケーションのログイン名およびセッション ID を特定します。

7.2.1, 7.2.2, 7.2.3 の例では、セッション ID

「010D7AE670AEB55C022B6A4DA1C1E8B1CtEPmR9H」のリクエスト

「10.209.15.130/1234/0x0000000000000001」(ルートアプリケーション情報) から、

ユーザ ID 「USER1」(ログイン名) で Bean の実行中 (0x8405 ~ 0x8406) に、データベースのテーブル「Table1」に INSERT 文を実行したことが確認できます。

## 7.3 注意事項

---

ルートアプリケーション情報をデータベースに設定するときに、データベースからエラーが返ってきた場合、KDJE54051-E のメッセージが出力されます。この場合、リクエストの処理は続行されますが、データベース監査証跡にルートアプリケーション情報は出力されません。

KDJE54051-E のメッセージが出力された場合は、Cosminexus Component Container およびデータベースの保守情報を取得して、保守員に連絡してください。また、KDJE54051-E のメッセージはデータベースから初めてエラーが返ってきた時だけ出力され、それ以降はエラーが発生しても出力されません。そのため、ログを調査する場合などには注意してください。

# 8

## クラスタソフトウェアを使用したシステムの起動と停止（Windows の場合）

この章では、クラスタソフトウェアを使用して、系切り替えシステムを運用する場合の、システムの起動と停止方法について説明します。Windows の場合、クラスタソフトウェアには、Microsoft Cluster Service を使用します。Microsoft Cluster Service を使用すると、1:1 系切り替えシステム、相互系切り替えシステム、および N:1 リカバリシステムを構築、運用することができます。

- 
- 8.1 アプリケーションサーバを対象にした 1:1 系切り替えシステムの起動と停止

---

  - 8.2 運用管理サーバを対象にした 1:1 系切り替えシステムの起動と停止

---

  - 8.3 相互系切り替えシステムの起動と停止

---

  - 8.4 N:1 リカバリシステムの起動と停止
-

## 8.1 アプリケーションサーバを対象にした 1:1 系切り替えシステムの起動と停止

---

この節では、アプリケーションサーバを対象にした 1:1 系切り替えシステムを利用する場合の、システムの起動と停止の手順について説明します。稼働開始後に J2EE サーバ、またはバッチサーバをメンテナンスするときの、起動と停止の手順についても説明します。

アプリケーションサーバを対象にした 1:1 系切り替えシステムでは、クラスタサービスを使用します。クラスタサービスの詳細については、OS のマニュアルを参照してください。

1:1 系切り替えシステムを利用して運用する場合は、あらかじめ現用系と予備系の 2 種類のホストの用意や、クラスタサービスの対象となる運用管理エージェントを監視・起動・停止するスクリプトの登録など、必要な環境設定をしておく必要があります。設定方法については、マニュアル「Cosminexus システム構築ガイド」の 1:1 系切り替えシステムの設定に関する説明を参照してください。

---

### ポイント

1:1 系切り替えシステムを利用する場合、Management Server は運用管理サーバで起動します。なお、個々のアプリケーションサーバのホストでは、運用管理エージェントを OS 起動と同時に起動する設定にしないでください。

---

---

### 参考

動作中の系を区別する場合、「実行系」と「待機系」の 2 種類の系に分けられます。

- 実行系  
現在実行中のサーバがある系のことです。系切り替えが発生すると、待機系になります。
- 待機系  
現在待機中のサーバがある系のことです。系切り替えが発生すると、実行系になります。

また、起動前の系を区別する場合、「現用系」と「予備系」の 2 種類の系に分けられます。

- 現用系  
起動したときに、最初に実行系として起動される系のことです。
- 予備系  
起動したときに、最初に待機系として起動される系のことです。

なお、この節では、手順内で計画的な系切り替えを実行する前には、現用系ホストが実行系ホストとして起動されていることを前提にして説明します。

---

## 8.1.1 アプリケーションサーバを対象にした 1:1 系切り替えシステムの起動

ここでは、アプリケーションサーバを対象にした 1:1 系切り替えシステムの起動方法について説明します。

アプリケーションサーバを対象にした 1:1 系切り替えシステムを起動するには、あらかじめ運用管理サーバを起動しておく必要があります。運用管理サーバを起動していない場合には、運用管理サーバを先に起動しておいてください。運用管理サーバの起動方法については、マニュアル「Cosminexus 運用管理操作ガイド」の運用管理ポータルと運用管理コマンドによるシステムの起動と停止に関する説明を参照してください。

アプリケーションサーバを対象にした 1:1 系切り替えシステムの起動手順を次に示します。

1. [スタート]メニューの[コントロールパネル] - [パフォーマンスとメンテナンス] - [管理ツール]から,[クラスタ アドミニストレータ]を選択します。  
クラスタアドミニストレータが開始されます。
2. コンソールツリー(左ペイン)で現用系ノードを選択し,[ファイル]メニューの[クラスタサービスの開始]を選択します。  
現用系ノードのクラスタサービスが開始されます。
3. コンソールツリー(左ペイン)で予備系ノードを選択し,[ファイル]メニューの[クラスタサービスの開始]を選択します。  
予備系ノードのクラスタサービスが開始されます。
4. コンソールツリー(左ペイン)で実行系ノードおよび待機系ノードが含まれているリソースグループを選択して,[ファイル]メニューの[オンラインにする]を選択します。  
1:1 系切り替えシステムが起動します。

## 8.1.2 アプリケーションサーバを対象にした 1:1 系切り替えシステムの停止

ここでは、アプリケーションサーバを対象にした 1:1 系切り替えシステムの停止方法について説明します。

1. [スタート]メニューの[コントロールパネル] - [パフォーマンスとメンテナンス] - [管理ツール]から,[クラスタ アドミニストレータ]を選択します。  
クラスタアドミニストレータが開始されます。
2. コンソールツリー(左ペイン)で実行系ノードおよび待機系ノードが含まれているリソースグループを選択して,[ファイル]メニューの[オフラインにする]を選択します。  
実行系ノードおよび待機系ノードが含まれているリソースグループがオフラインにな

## 8. クラスタソフトウェアを使用したシステムの起動と停止 (Windows の場合)

ります。

3. コンソールツリー (左ペイン) で予備系ノードを選択し、[ファイル]メニューの [クラスタサービスの停止] を選択します。  
予備系ノードのクラスタサービスが停止します。
4. コンソールツリー (左ペイン) で現用系ノードを選択し、[ファイル]メニューの [クラスタサービスの停止] を選択します。  
現用系ノードのクラスタサービスが停止します。

なお、運用管理サーバは最後に手動で停止します。運用管理サーバの停止方法については、マニュアル「Cosminexus 運用管理操作ガイド」の運用管理ポータルと運用管理コマンドによるシステムの起動と停止に関する説明を参照してください。

### 8.1.3 アプリケーションサーバを対象にした 1:1 系切り替えシステムで計画的に系を切り替えるときの起動と停止

トラブル発生時以外で、アプリケーションサーバを対象にした 1:1 系切り替えシステムの実行系と待機系を切り替える手順を次に示します。系を切り替えるときは、待機系のホストで Cosminexus が待機状態になっている必要があります。

1. [スタート]メニューの [コントロールパネル] - [パフォーマンスとメンテナンス] - [管理ツール] から、[クラスタ アドミニストレータ] を選択します。  
クラスタアドミニストレータが開始されます。
2. コンソールツリー (左ペイン) で実行系ノードおよび待機系ノードが含まれているリソースグループを選択して、[ファイル]メニューの [グループの移動] を選択します。  
系が切り替わります。

### 8.1.4 アプリケーションサーバを対象にした 1:1 系切り替えシステムをメンテナンスする場合の起動と停止

アプリケーションサーバを対象にした 1:1 系切り替えシステムをメンテナンスする場合の、起動と停止の手順を次に示します。

なお、これらの手順は、すでに実行系のホストで Cosminexus が起動している場合の手順です。

#### (1) 再起動が不要なメンテナンスの場合の手順

1. 実行系のホストで、動作中の J2EE アプリケーションとリソースアダプタを停止します。  
バッチサーバの場合は、バッチアプリケーションが実行中でないことを確認してからリソースアダプタを停止してください。  
停止方法については、マニュアル「Cosminexus リファレンス コマンド編」の



マンドによるシステムの起動と停止に関する説明を参照してください。

2. サーバ管理コマンドを使用してメンテナンス処理を実行します。  
サーバ管理コマンドを使用した J2EE アプリケーションやリソースの設定方法については、マニュアル「Cosminexus システム構築ガイド」の J2EE アプリケーションの設定に関する説明、およびリソースの設定に関する説明を参照してください。
3. 手順 1. で停止した J2EE アプリケーションとリソースアダプタを開始します。  
J2EE サーバの場合は J2EE アプリケーションとリソースアダプタを開始します。また、バッチサーバの場合はリソースアダプタを開始します。  
開始方法については、マニュアル「Cosminexus リファレンス コマンド編」のコマンドによるシステムの起動と停止に関する説明を参照してください。
4. 必要に応じて J2EE アプリケーションとリソースアダプタの動作確認をします。
5. [スタート]メニューの[コントロールパネル] - [パフォーマンスとメンテナンス] - [管理ツール] から、[クラスタ アドミニストレータ]を選択します。  
クラスタアドミニストレータが開始されます。
6. コンソールツリー (左ペイン) で実行系ノードおよび待機系ノードが含まれているリソースグループを選択して、[ファイル]メニューの[グループの移動]を選択します。  
系が切り替わります。
7. 切り替え後の実行系のホストで手順 1. ~ 手順 6. の手順を実行します。
8. 切り替え後の実行系のホストを実行系に戻す場合、コンソールツリー (左ペイン) で実行系ノードおよび待機系ノードが含まれているリソースグループを選択して、[ファイル]メニューの[グループの移動]を選択します。  
メンテナンスが完了します。

## (2) 再起動が必要なメンテナンスの場合 (両方の系を同時に停止する方法)

1. 論理サーバを一括停止する設定にしていない場合は、実行系のホストで各論理サーバを停止します。  
Management Server によって一括停止する設定の場合は、不要な手順です。停止方法については、マニュアル「Cosminexus 運用管理操作ガイド」の運用管理ポータルと運用管理コマンドによるシステムの起動と停止に関する説明を参照してください。
2. [スタート]メニューの[コントロールパネル] - [パフォーマンスとメンテナンス] - [管理ツール] から、[クラスタ アドミニストレータ]を選択します。  
クラスタアドミニストレータが開始されます。
3. コンソールツリー (左ペイン) で実行系ノードおよび待機系ノードが含まれているリソースグループを選択して、[ファイル]メニューの[オフラインにする]を選択します。

## 8. クラスタソフトウェアを使用したシステムの起動と停止 (Windows の場合)

実行系ノードおよび待機系ノードが含まれているリソースグループがオフラインになります。

4. コンソールツリー (左ペイン) で予備系ノードを選択し, [ファイル] メニューの [クラスタサービスの停止] を選択します。  
予備系ノードのクラスタサービスが停止します。
5. コンソールツリー (左ペイン) で現用系ノードを選択し, [ファイル] メニューの [クラスタサービスの停止] を選択します。  
現用系ノードのクラスタサービスが停止します。
6. 現用系と予備系の両方のホストで, 定義ファイルの変更など, メンテナンス作業を実行します。
7. コンソールツリー (左ペイン) で現用系ノードを選択し, [ファイル] メニューの [クラスタサービスの開始] を選択します。  
現用系ノードのクラスタサービスが開始されます。
8. コンソールツリー (左ペイン) で予備系ノードを選択し, [ファイル] メニューの [クラスタサービスの開始] を選択します。  
予備系ノードのクラスタサービスが開始されます。
9. コンソールツリー (左ペイン) で現用系ノードおよび予備系ノードが含まれているリソースグループを選択して, [ファイル] メニューの [オンラインにする] を選択します。  
1:1 系切り替えシステムが再起動します。
10. 論理サーバを一括起動する設定にしていない場合は, 実行系のホストで各論理サーバを起動します。  
Management Server によって一括起動する設定の場合は, 不要な手順です。起動方法については, マニュアル「Cosminexus 運用管理操作ガイド」の運用管理ポータルと運用管理コマンドによるシステムの起動と停止に関する説明を参照してください。
11. 実行系のホストで, 必要に応じて, 定義の変更に関連した動作確認を実行します。  
メンテナンスが完了します。

### (3) 再起動が必要なメンテナンスの場合 (両方の系を同時に停止しない方法)

1. 待機系のホストで, 定義ファイルの変更など, メンテナンス作業を実行します。
2. [スタート] メニューの [コントロールパネル] - [パフォーマンスとメンテナンス] - [管理ツール] から, [クラスタ アドミニストレータ] を選択します。  
クラスタアドミニストレータが開始されます。
3. コンソールツリー (左ペイン) で実行系ノードおよび待機系ノードが含まれているリソースグループを選択して, [ファイル] メニューの [グループの移動] を選択します。  
実行系のホストの Cosminexus が停止してから, 今まで待機していた系で

## 8. クラスタソフトウェアを使用したシステムの起動と停止 (Windows の場合)

Cosminexus が起動します。これによって、メンテナンス済みのホストが実行系ホストになります。

4. 切り替え後の実行系のホストで、必要に応じて、定義の変更に関連した動作確認を実行します。
5. 切り替え後の待機系のホストで定義ファイルの編集など、メンテナンス作業を実行します。
6. 切り替え後の実行系のホストを実行系に戻す場合、コンソールツリー (左ペイン) で実行系ノードおよび待機系ノードが含まれているリソースグループを選択して、[ファイル]メニューの [グループの移動] を選択します。
7. 切り替え後の実行系のホストで、必要に応じて、定義の変更に関連した動作確認を実行します。  
メンテナンスが完了します。

## 8.2 運用管理サーバを対象にした 1:1 系切り替えシステムの起動と停止

---

ここでは、運用管理サーバを対象にした 1:1 系切り替えシステムを利用する場合の、システムの起動と停止の方法について説明します。

運用管理サーバを対象にした 1:1 系切り替えシステムでは、クラスタサービスを使用します。クラスタサービスの詳細については、OS のマニュアルを参照してください。

運用管理サーバを対象にした 1:1 系切り替えシステムを利用するには、あらかじめ現用系と予備系の 2 種類のホストの用意や、クラスタサービスの対象となる Management Server を監視・起動・停止するスクリプトの登録など、必要な環境設定をしておく必要があります。設定方法については、マニュアル「Cosminexus システム構築ガイド」の 1:1 系切り替えシステムの設定に関する説明を参照してください。現用系と予備系の詳細については、「8.1 アプリケーションサーバを対象にした 1:1 系切り替えシステムの起動と停止」を参照してください。

### 8.2.1 運用管理サーバを対象にした 1:1 系切り替えシステムの起動

ここでは、運用管理サーバを対象にした 1:1 系切り替えシステムの起動方法について説明します。

運用管理サーバを対象にした 1:1 系切り替えシステムを起動するには、あらかじめ起動するアプリケーションサーバの運用管理エージェントを起動しておく必要があります。運用管理エージェントを起動していない場合には、運用管理エージェントを先に起動しておいてください。運用管理エージェントの起動方法については、マニュアル「Cosminexus 運用管理操作ガイド」の運用管理ポータルと運用管理コマンドによるシステムの起動と停止に関する説明を参照してください。

運用管理サーバを対象にした 1:1 系切り替えシステムの起動手順を次に示します。

1. [スタート]メニューの[コントロールパネル] - [パフォーマンスとメンテナンス] - [管理ツール] から、[クラスタ アドミニストレータ] を選択します。  
クラスタアドミニストレータが開始されます。
2. コンソールツリー (左ペイン) で運用管理サーバの現用系ノードを選択し、[ファイル]メニューの[クラスタサービスの開始]を選択します。  
現用系ノードのクラスタサービスが開始されます。
3. コンソールツリー (左ペイン) で運用管理サーバの予備系ノードを選択し、[ファイル]メニューの[クラスタサービスの開始]を選択します。  
予備系ノードのクラスタサービスが開始されます。

4. コンソールツリー (左ペイン) で運用管理サーバの現用系ノードおよび予備系ノードが含まれているリソースグループを選択して、[ファイル]メニューの[オンラインにする]を選択します。  
運用管理サーバを対象にした 1:1 系切り替えシステムが起動します。

## 8.2.2 運用管理サーバを対象にした 1:1 系切り替えシステムの停止

ここでは、運用管理サーバを対象にした 1:1 系切り替えシステムの停止方法について説明します。

1. [スタート]メニューの[コントロールパネル] - [パフォーマンスとメンテナンス] - [管理ツール] から、[クラスタ アドミニストレータ] を選択します。  
クラスタアドミニストレータが開始されます。
2. コンソールツリー (左ペイン) で運用管理サーバの実行系ノードおよび待機系ノードが含まれているリソースグループを選択して、[ファイル]メニューの[オフラインにする]を選択します。  
運用管理サーバの実行系ノードおよび待機系ノードが含まれているリソースグループがオフラインになります。
3. コンソールツリー (左ペイン) で運用管理サーバの予備系ノードを選択し、[ファイル]メニューの[クラスタサービスの停止]を選択します。  
運用管理サーバの予備系ノードのクラスタサービスが停止します。
4. コンソールツリー (左ペイン) で運用管理サーバの現用系ノードを選択し、[ファイル]メニューの[クラスタサービスの停止]を選択します。  
運用管理サーバの現用系ノードのクラスタサービスが停止します。
5. アプリケーションサーバの運用管理エージェントを停止します。  
運用管理エージェントの停止方法については、マニュアル「Cosminexus 運用管理操作ガイド」の運用管理ポータルと運用管理コマンドによるシステムの起動と停止に関する説明を参照してください。

## 8.2.3 運用管理サーバを対象にした 1:1 系切り替えシステムで計画的に系を切り替えるときの起動と停止

トラブル発生時以外で、運用管理サーバを対象にした 1:1 系切り替えシステムの実行系と待機系を切り替える手順を次に示します。系を切り替えるときは、待機系のホストで Cosminexus が待機状態になっている必要があります。

1. システムの運用中に Management Server の設定情報を変更した場合、実行系の Management Server の設定情報を収集して、実行系から待機系へコピーします。また、mstrimport コマンドを実行して、コピーした Management Server の設定情報を待機系に適用します。

## 8. クラスタソフトウェアを使用したシステムの起動と停止 (Windows の場合)

実行系の Management Server と待機系の Management Server が同じ設定になります。

Management Server の設定情報を実行系から待機系へコピーして、待機系に適用する方法については、マニュアル「Cosminexus システム構築ガイド」の 1:1 系切り替えシステムの設定に関する説明を参照してください。

2. [スタート]メニューの[コントロールパネル] - [パフォーマンスとメンテナンス] - [管理ツール] から,[クラスタ アドミニストレータ]を選択します。  
クラスタアドミニストレータが開始されます。
3. コンソールツリー(左ペイン)で実行系ノードおよび待機系ノードが含まれているリソースグループを選択して,[ファイル]メニューの[グループの移動]を選択します。  
系が切り替わります。

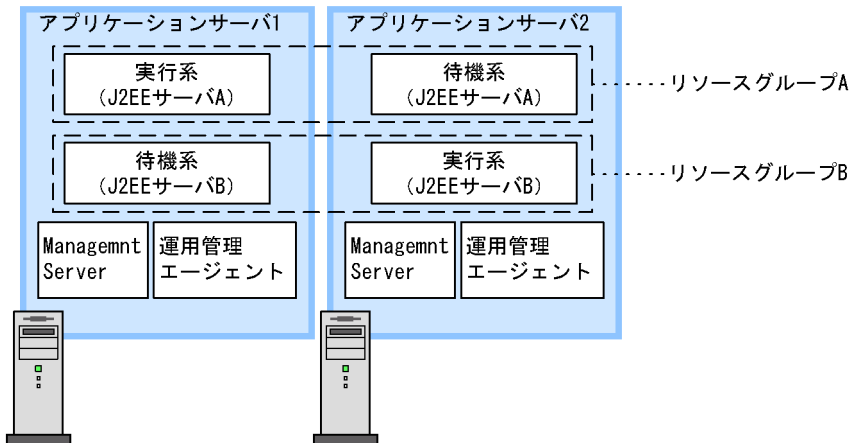
## 8.3 相互系切り替えシステムの起動と停止

ここでは、相互系切り替えシステムを利用する場合の、システムの起動と停止方法について説明します。

相互系切り替えシステムは、アプリケーションサーバの実行系と待機系を 1:1 にする構成の一つです。それぞれのアプリケーションサーバを実行系として稼働させながら、同時にお互いの待機系として構成します。現用系と待機系の詳細については、「8.1 アプリケーションサーバを対象にした 1:1 系切り替えシステムの起動と停止」を参照してください。

ここでは、それぞれのアプリケーションサーバを、アプリケーションサーバ 1、アプリケーションサーバ 2 として、アプリケーションサーバ 1 には、J2EE サーバ A の実行系と J2EE サーバ B の待機系を、アプリケーションサーバ 2 には J2EE サーバ A の待機系と J2EE サーバ B の実行系を配置する構成を例にして説明します。例にする構成を次の図に示します。この例では、J2EE サーバ A の実行系と J2EE サーバ A の待機系をリソースグループ A、J2EE サーバ B の実行系と J2EE サーバ B の待機系をリソースグループ B と設定しています。

図 8-1 相互スタンバイシステムの構成例



### 8.3.1 相互系切り替えシステムの起動

ここでは、相互系切り替えシステムを利用する場合のシステムの起動手順について説明します。

相互系切り替えシステムを起動するには、あらかじめ Management Server、および運用管理エージェントを起動しておくことが必要です。Management Server、および運用管理エージェントを起動していない場合には、これらを先に起動しておいてください。

## 8. クラスタソフトウェアを使用したシステムの起動と停止 (Windows の場合)

相互系切り替えシステムの起動手順を次に示します。

1. [スタート]メニューの[コントロールパネル] - [パフォーマンスとメンテナンス] - [管理ツール] から,[クラスタ アドミニストレータ]を選択します。  
クラスタアドミニストレータが開始されます。
2. コンソールツリー (左ペイン) でアプリケーションサーバ 1 のノードを選択し,  
[ファイル]メニューの[クラスタサービスの開始]を選択します。  
アプリケーションサーバ 1 のノードのクラスタサービスが開始されます。
3. コンソールツリー (左ペイン) でアプリケーションサーバ 2 のノードを選択し,  
[ファイル]メニューの[クラスタサービスの開始]を選択します。  
アプリケーションサーバ 2 のノードのクラスタサービスが開始されます。
4. コンソールツリー (左ペイン) でリソースグループ A を選択して,[ファイル]メニューの[オンラインにする]を選択します。  
実行系 (アプリケーションサーバ A) がオンラインになります。
5. コンソールツリー (左ペイン) でリソースグループ B を選択して,[ファイル]メニューの[オンラインにする]を選択します。  
実行系 (アプリケーションサーバ B) がオンラインになります。

### 8.3.2 相互系切り替えシステムの停止

ここでは,相互系切り替えシステムの停止方法について説明します。

1. [スタート]メニューの[コントロールパネル] - [パフォーマンスとメンテナンス] - [管理ツール] から,[クラスタ アドミニストレータ]を選択します。  
クラスタアドミニストレータが開始されます。
2. コンソールツリー (左ペイン) でリソースグループ B を選択して,[ファイル]メニューの[オフラインにする]を選択します。  
リソースグループ B がオフラインになります。
3. コンソールツリー (左ペイン) でリソースグループ A を選択して,[ファイル]メニューの[オフラインにする]を選択します。  
リソースグループ A がオフラインになります。
4. コンソールツリー (左ペイン) でアプリケーションサーバ 2 のノードを選択し,  
[ファイル]メニューの[クラスタサービスの停止]を選択します。  
アプリケーションサーバ 2 のノードのクラスタサービスが停止します。
5. コンソールツリー (左ペイン) でアプリケーションサーバ 1 のノードを選択し,  
[ファイル]メニューの[クラスタサービスの停止]を選択します。  
アプリケーションサーバ 1 のノードのクラスタサービスが停止します。

なお, Management Server, および運用管理エージェントは最後に手動で停止します。  
これらの停止方法については, マニュアル「Cosminexus 運用管理操作ガイド」の運用



管理ポータルと運用管理コマンドによるシステムの起動と停止に関する説明を参照してください。

### 8.3.3 相互系切り替えシステムで計画的に系を切り替える場合の起動と停止

トラブル発生時以外で、相互系切り替えシステムの実行系と待機系を切り替える手順を次に示します。系を切り替えるときは、待機系のホストで Cosminexus が待機状態になっている必要があります。

ここでは、J2EE サーバ A の系を切り替える場合について説明します。J2EE サーバ B の場合も同じ手順で系を切り替えることができます。

1. [スタート]メニューの [コントロールパネル] - [パフォーマンスとメンテナンス] - [管理ツール] から、[クラスタ アドミニストレータ] を選択します。  
クラスタアドミニストレータが開始されます。
2. コンソールツリー (左ペイン) でリソースグループ A を選択して、[ファイル]メニューの [グループの移動] を選択します。  
系が切り替わります。

## 8.4 N:1 リカバリシステムの起動と停止

---

この節では、N:1 リカバリシステムを利用する場合の、システムの起動と停止の手順について説明します。

N:1 リカバリシステムを利用して運用する場合は、あらかじめ現用系と予備系の2種類のホストの用意や、クラスタサービスの対象となる運用管理エージェントを監視・起動・停止するスクリプトの登録など、必要な環境設定をしておく必要があります。設定方法については、マニュアル「Cosminexus システム構築ガイド」のN:1 リカバリシステムの設定に関する説明を参照してください。

---

### ポイント

N:1 リカバリシステムを利用する場合、Management Server は運用管理サーバで起動します。なお、個々のアプリケーションサーバのホストでは、運用管理エージェントを OS 起動と同時に起動する設定にしないでください。

---

---

### 参考

動作中の系を区別する場合、「実行系」と「待機系」の2種類の系に分けられます。

- 実行系  
業務処理を実行する N 台のアプリケーションサーバがある系のことです。
  - 待機系  
1 台のリカバリ専用サーバがある系のことです。
- 

### 8.4.1 N:1 リカバリシステムの起動

ここでは N:1 リカバリシステムを利用する場合のシステムの起動手順について説明します。

N:1 リカバリシステムを利用するには、あらかじめ運用管理サーバを起動しておくことが必要です。運用管理サーバを起動していない場合には、運用管理サーバを先に起動しておいてください。運用管理サーバの起動方法については、マニュアル「Cosminexus 運用管理操作ガイド」の運用管理ポータルと運用管理コマンドによるシステムの起動と停止に関する説明を参照してください。

また、N:1 リカバリシステムでは、先に起動したノードが実行系ノードとして、あとに起動したノードが待機系ノードとして動作します。必ず現用系ノードを予備系ノードよりも先に起動してください。

N:1 リカバリシステムの起動手順を次に示します。

1. [スタート]メニューの[コントロールパネル] - [パフォーマンスとメンテナンス]

- [管理ツール] から,[クラスタ アドミニストレータ] を選択します。  
クラスタアドミニストレータが開始されます。
- 2. コンソールツリー (左ペイン) で現用系ノードを選択し,[ファイル] メニューの [クラスタサービスの開始] を選択します。  
選択した現用系ノードのクラスタサービスが開始されます。すべての現用行系ノードに対して実行してください。
- 3. コンソールツリー (左ペイン) で予備系ノードを選択し,[ファイル] メニューの [クラスタサービスの開始] を選択します。  
予備系ノードのクラスタサービスが開始されます。
- 4. コンソールツリー (左ペイン) で現用系ノードおよび予備系ノードが含まれているリソースグループを選択して,[ファイル] メニューの [オンラインにする] を選択します。  
選択したリソースグループがオンラインになります。すべてのリソースグループに対して実行してください。
- 5. 実行系のホストの各論理サーバを起動します。  
起動方法については, マニュアル「Cosminexus 運用管理操作ガイド」の運用管理ポータルと運用管理コマンドによるシステムの起動と停止に関する説明を参照してください。

## 8.4.2 N:1 リカバリシステムの停止

ここでは, N:1 リカバリシステムの停止方法について説明します。

1. [スタート] メニューの [コントロールパネル] - [パフォーマンスとメンテナンス] - [管理ツール] から,[クラスタ アドミニストレータ] を選択します。  
クラスタアドミニストレータが開始されます。
2. コンソールツリー (左ペイン) で実行系ノードおよび待機系ノードが含まれているリソースグループを選択して,[ファイル] メニューの [オフラインにする] を選択します。  
選択したリソースグループがオフラインになります。すべてのリソースグループに対して実行してください。
3. コンソールツリー (左ペイン) で予備系ノードを選択し,[ファイル] メニューの [クラスタサービスの停止] を選択します。  
予備系ノードのクラスタサービスが停止します。
4. コンソールツリー (左ペイン) で現用系ノードを選択し,[ファイル] メニューの [クラスタサービスの停止] を選択します。  
選択した現用系ノードのクラスタサービスが停止します。すべての現用系ノードに対して実行してください。

なお, 運用管理サーバは最後に手動で停止します。運用管理サーバの停止方法については, マニュアル「Cosminexus 運用管理操作ガイド」の運用管理ポータルと運用管理コ

## 8. クラスタソフトウェアを使用したシステムの起動と停止 (Windows の場合)

マンドによるシステムの起動と停止に関する説明を参照してください。

# 9

## クラスタソフトウェアを使用したシステムの起動と停止（UNIX の場合）

この章では、クラスタソフトウェアを使用して、系切り替えシステムを運用する場合の、システムの起動と停止方法について説明します。UNIX の場合、クラスタソフトウェアには、HA モニタを使用します。HA モニタを使用すると、1:1 系切り替えシステム、相互系切り替えシステム、および N:1 リカバリシステムを構築、運用することができます。

---

9.1 1:1 系切り替えシステムの起動と停止

---

9.2 相互系切り替えシステムの起動と停止

---

9.3 N:1 リカバリシステムの起動と停止

---

## 9.1 1:1 系切り替えシステムの起動と停止

この節では、HA モニタによる 1:1 系切り替えシステムを利用する場合の、システムの起動と停止の手順について説明します。稼働開始後に J2EE サーバ、バッチサーバ、Management Serverなどをメンテナンスする場合の、起動と停止の手順についても説明します。なお、HA モニタを利用できるのは、AIX、HP-UX または Linux の場合だけです。Solaris の場合、HA モニタは利用できません。

1:1 系切り替えシステムを利用して運用する場合は、あらかじめ現用系と予備系の 2 種類のホストの用意や、HA モニタの監視の対象となる運用管理エージェント (アプリケーションサーバの 1:1 系切り替えシステムの場合) または Management Server (運用管理サーバの 1:1 系切り替えシステムの場合) を監視・起動・停止するスクリプトの登録など、必要な環境設定をしておく必要があります。設定方法については、マニュアル「Cosminexus システム構築ガイド」のシステムの可用性を高める設定に関する説明を参照してください。

アプリケーションサーバの 1:1 系切り替えシステムと、運用管理サーバの 1:1 系切り替えシステムで実行できる運用操作およびその参照先を次の表に示します。

表 9-1 1:1 系切り替えシステムで実行できる運用操作

運用操作	アプリケーションサーバの 1:1 系切り替えシステム	運用管理サーバの 1:1 系切り替えシステム
システムの起動	9.1.1 参照	9.1.1 参照
システムの停止	9.1.2 参照	9.1.2 参照
実行系と待機系を計画的に系切り替えする場合のシステムの起動と停止	9.1.3 参照	9.1.3 参照
メンテナンスする場合のシステムの起動と停止	9.1.4 参照	9.1.5 参照

また、次のコマンドは、HA モニタが提供しているコマンドです。詳細は、マニュアル「高信頼化システム監視機能 HA モニタ」を参照してください。

- monbegin (HA モニタとのインタフェースを持たないサーバを起動)
- monend (HA モニタとのインタフェースを持たない実行サーバの停止連絡)
- monsbystp (待機サーバの停止)
- monswap (計画系切り替え)

## 参考

HA モニタの動作中の系を区別する場合、「実行系」と「待機系」の 2 種類の系に分けられます。

- 実行系  
現在実行中のサーバがある系のことです。系切り替えが発生すると、待機系になります。
- 待機系  
現在待機中のサーバがある系のことです。系切り替えが発生すると、実行系になります。

また、起動前の HA モニタを静的に区別する場合、「現用系」と「予備系」の 2 種類の系に分けられます。

- 現用系  
起動したときに、最初に実行系として起動される系のことです。なお、実行系として起動することは、HA モニタのサーバ対応環境設定用の定義ファイルで定義しておきます。
- 予備系  
起動したときに、最初に待機系として起動される系のことです。なお、待機系として起動することは、HA モニタのサーバ対応環境設定用の定義ファイルで定義しておきます。

## 9.1.1 1:1 系切り替えシステムの起動

1:1 系切り替えシステムを利用する場合のシステム起動時の留意事項、およびシステムの起動手順について説明します。

### (1) システム起動時の留意事項

1:1 系切り替えシステムを利用する場合のシステム起動時の留意事項を次に示します。

- 現用系と予備系のホストに配置された HA モニタは、OS の起動と同時に起動されています。
- アプリケーションサーバの 1:1 系切り替えシステムを起動する場合、運用管理サーバを使用するときは、運用管理サーバをあらかじめ起動しておいてください。運用管理サーバの起動方法については、マニュアル「Cosminexus 運用管理操作ガイド」の運用管理サーバの起動の流れに関する説明を参照してください。
- 運用管理サーバの 1:1 系切り替えシステムを起動する場合、アプリケーションサーバを別ホストに配置しているときは、運用管理ドメイン内のアプリケーションサーバの運用管理エージェントをあらかじめ起動しておいてください。運用管理エージェントの起動方法については、マニュアル「Cosminexus 運用管理操作ガイド」の運用管理ポータルと運用管理コマンドによるシステムの起動と停止に関する説明を参照してください。

### (2) システムの起動手順

1:1 系切り替えシステムを利用する場合のシステムの起動手順を次に示します。

1. 現用系のホストで `monbegin` コマンドを実行して、現用系のホストを実行系として起動します。

## 9. クラスタソフトウェアを使用したシステムの起動と停止 (UNIX の場合)

```
# monbegin サーバの識別名
```

下線部分には、servers ファイルのオペランド「alias」に指定されているサーバの識別名を指定します。

これによって、運用管理エージェントを起動するスクリプトファイルに定義されている処理が実行されて、現用系のホストが実行系として起動します。

2. 予備系のホストで monbegin コマンドを実行して、予備系のホストを待機系として起動します。

```
# monbegin サーバの識別名
```

下線部分には、servers ファイルのオペランド「alias」に指定されているサーバの識別名を指定します。

これによって、予備系のホストが待機系となり、実行系の障害に備えます。

### ! 注意事項

Management Server 起動時に運用管理ドメイン内の論理サーバを一括起動する設定にしない場合は、現用系のホストの各論理サーバを起動してください。なお、Management Server 起動時に運用管理ドメイン内の論理サーバを一括起動するように設定している場合、論理サーバの手動起動は不要です。起動方法については、マニュアル「Cosminexus 運用管理操作ガイド」の運用管理ポータルと運用管理コマンドによるシステムの起動と停止に関する説明を参照してください。

### 参考

servers ファイルの定義 (HA モニタでのサーバの環境設定) については、マニュアル「Cosminexus システム構築ガイド」の HA モニタの設定に関する説明を参照してください。

## 9.1.2 1:1 系切り替えシステムの停止

1:1 系切り替えシステムの停止手順について説明します。

ここでは、次の二つの場合の停止手順について説明します。

- 実行系と待機系の両方のホストを停止する場合
- 待機系のホストだけ停止する場合



## (1) 実行系と待機系の両方のホストを停止する場合

実行系と待機系の両方のホストを停止する場合の手順を次に示します。

1. 運用管理エージェントを停止するスクリプトで、運用管理ドメイン内の論理サーバを一括停止する設定にしていない場合は、実行系のホストの各論理サーバを停止します。  
運用管理エージェントを停止するスクリプトで論理サーバを停止する設定にしている場合は、不要な手順です。論理サーバの停止方法については、マニュアル「Cosminexus 運用管理操作ガイド」の運用管理ポータルと運用管理コマンドによるシステムの起動と停止に関する説明を参照してください。
2. 実行系のホストで `monend` コマンドを実行して、実行系のホストの運用管理エージェントを停止します。

```
# monend サーバの識別名
```

下線部分には、`servers` ファイルのオペランド「`alias`」に指定されているサーバの識別名を指定します。

これによって、実行系のホストの運用管理エージェントが停止します。また、HA モニタによって待機系に自動的に停止指示が出され、待機系のホストの運用管理エージェントも停止します。

なお、アプリケーションサーバの 1:1 系切り替えシステムの場合、運用管理サーバを使用しているときは、運用管理サーバは最後に手動で停止します。運用管理サーバの停止方法については、マニュアル「Cosminexus 運用管理操作ガイド」の運用管理ポータルと運用管理コマンドによるシステムの起動と停止に関する説明を参照してください。

### 参考

`servers` ファイルの定義 (HA モニタでのサーバの環境設定) については、マニュアル「Cosminexus システム構築ガイド」の HA モニタの設定に関する説明を参照してください。

## (2) 待機系のホストだけ停止する場合

実行系は停止しないで待機系で待機状態をやめる場合は、待機系のホストで次のコマンドを実行します。

```
# monsbystp サーバの識別名
```

下線部分には、`servers` ファイルのオペランド「`alias`」に指定されているサーバの識別名を指定します。

### 9.1.3 1:1 系切り替えシステムで計画的に系を切り替える場合の起動と停止

1:1 系切り替えシステムでトラブル発生時以外の場合に、実行系と待機系を計画的に切り替えるときの手順について説明します。系を切り替えるときには、待機系のホストで Cosminexus が待機状態になっている必要があります。

1:1 系切り替えシステムで実行系と待機系を計画的に切り替えるときの手順を次に示します。なお、現用系のホストが実行系として起動されていることを前提としています。

1. 運用管理エージェントを停止するスクリプトで、運用管理ドメイン内の論理サーバを一括停止する設定にしていない場合は、実行系のホストで各論理サーバを停止します。

運用管理エージェントを停止するスクリプトで論理サーバを停止する設定にしている場合は、不要な手順です。論理サーバの停止方法については、マニュアル「Cosminexus 運用管理操作ガイド」の運用管理ポータルと運用管理コマンドによるシステムの起動と停止に関する説明を参照してください。

2. 実行系のホストで monswap コマンドを実行して、系を切り替えます。

```
# monswap サーバの識別名
```

下線部分には、servers ファイルのオペランド「alias」に指定されているサーバの識別名を指定します。

運用管理エージェントを停止するスクリプトに定義されている処理が実行されて、実行系のホストの Cosminexus が停止します。そのあと、今まで待機していた系で Cosminexus が起動します。これで、計画的な系切り替えの手順は完了です。

#### 参考

- 1:1 系切り替えシステムをメンテナンスする場合の起動と停止の手順については、「9.1.4 アプリケーションサーバの 1:1 系切り替えシステムをメンテナンスする場合の起動と停止」または「9.1.5 運用管理サーバの 1:1 系切り替えシステムをメンテナンスする場合の起動と停止」を参照してください。
- servers ファイルの定義 (HA モニタでのサーバの環境設定) については、マニュアル「Cosminexus システム構築ガイド」の HA モニタの設定に関する説明を参照してください。

### 9.1.4 アプリケーションサーバの 1:1 系切り替えシステムをメンテナンスする場合の起動と停止

アプリケーションサーバの 1:1 系切り替えシステムをメンテナンスする場合の、起動と

停止の手順について説明します。

ここでは、次の三つの場合の停止手順について説明します。

- 再起動が不要なメンテナンスの場合
- 再起動が必要なメンテナンスの場合（両方の系を同時に停止する方法）
- 再起動が必要なメンテナンスの場合（両方の系を同時に停止しない方法）

なお、これらの手順は、すでに実行系のホストで Cosminexus が起動している場合の手順です。

### （1）再起動が不要なメンテナンスの場合

再起動が不要なメンテナンスをする場合の起動と停止の手順を次に示します。

1. 実行系のホストで、動作中の J2EE アプリケーションとリソースアダプタを停止します。  
パッチサーバの場合は、パッチアプリケーションが実行中でないことを確認してからリソースアダプタを停止してください。  
停止方法については、マニュアル「Cosminexus 運用管理操作ガイド」の運用管理ポータルと運用管理コマンドによるシステムの起動と停止に関する説明を参照してください。
2. サーバ管理コマンドを使用してメンテナンス処理を実行します。  
サーバ管理コマンドを使用した J2EE アプリケーションやリソースの設定方法については、マニュアル「Cosminexus システム構築ガイド」の J2EE アプリケーションに関する説明、およびリソースの設定に関する説明を参照してください。
3. 手順 1. で停止した J2EE アプリケーションとリソースアダプタを開始します。  
J2EE サーバの場合は J2EE アプリケーションとリソースアダプタを開始します。また、パッチサーバの場合はリソースアダプタを開始します。  
開始方法については、マニュアル「Cosminexus 運用管理操作ガイド」の運用管理ポータルと運用管理コマンドによるシステムの起動と停止に関する説明を参照してください。
4. 必要に応じて J2EE アプリケーションとリソースアダプタの動作確認をします。
5. 実行系のホスト（現用系のホスト）で、monswap コマンドを実行して、系を切り替えます。  
実行系のホスト（現用系ホスト）が待機系のホストに切り替わり、待機系だったホスト（予備系ホスト）が実行系ホストになります。
6. 切り替え後の実行系のホスト（予備系のホスト）で手順 1. ~ 手順 4. を実行します。
7. 切り替え後の実行系のホスト（予備系のホスト）で、必要に応じて、再度 monswap コマンドを実行します。  
現用系のホストを実行系に戻す場合に実行してください。  
これで、メンテナンス時の起動と停止の手順は完了です。

## 9. クラスタソフトウェアを使用したシステムの起動と停止 (UNIX の場合)

### (2) 再起動が必要なメンテナンスの場合 (両方の系を同時に停止する方法)

再起動が必要なメンテナンスをする場合に、両方の系を同時に停止するときの起動と停止の手順を次に示します。

1. 運用管理エージェントを停止するスクリプトで、運用管理ドメイン内の論理サーバを一括停止する設定にしていない場合は、実行系のホストで各論理サーバを停止します。  
運用管理エージェントを停止するスクリプトで論理サーバを停止する設定にしている場合は、不要な手順です。論理サーバの停止方法については、マニュアル「Cosminexus 運用管理操作ガイド」の運用管理ポータルと運用管理コマンドによるシステムの起動と停止に関する説明を参照してください。
2. 実行系のホストで `monend` コマンドを実行して、システムを停止します。

```
# monend サーバの識別名
```

下線部分には、`servers` ファイルのオペランド「`alias`」に指定されているサーバの識別名を指定します。

実行系のホストが停止します。また、HA モニタによって待機系に自動的に停止指示が出され、待機系のホストも停止します。

3. 現用系と予備系の両方のホストで、定義ファイルの変更など、メンテナンス作業を実行します。
4. 現用系のホストで `monbegin` コマンドを実行して、現用系のホストの運用管理エージェントを起動します。

```
# monbegin サーバの識別名
```

下線部分には、`servers` ファイルのオペランド「`alias`」に指定されているサーバの識別名を指定します。

これによって、現用系のホストが、実行系として起動します。

5. 予備系のホストで `monbegin` コマンドを実行します。

```
# monbegin サーバの識別名
```

下線部分には、`servers` ファイルのオペランド「`alias`」に指定されているサーバの識別名を指定します。

これによって、予備系のホストが、待機系として起動します。

6. 運用管理エージェントを起動するスクリプトで運用管理ドメイン内の論理サーバを一括起動する設定にしていない場合は、実行系のホストの各論理サーバを起動します。運用管理エージェントを起動するスクリプトで論理サーバを起動する設定にしている

場合は、不要な手順です。論理サーバの起動方法については、マニュアル「Cosminexus 運用管理操作ガイド」の運用管理ポータルと運用管理コマンドによるシステムの起動と停止に関する説明を参照してください。

7. 実行系のホストで、必要に応じて、定義の変更に関連した動作確認を実行します。

---

参考

servers ファイルの定義（HA モニタでのサーバの環境設定）については、マニュアル「Cosminexus システム構築ガイド」の HA モニタの設定に関する説明を参照してください。

---

### （3）再起動が必要なメンテナンスの場合（両方の系を同時に停止しない方法）

再起動が必要なメンテナンスをする場合に、両方の系を同時に停止しないときの起動と停止の手順を次に示します。

1. 待機系のホストで、定義ファイルの変更など、メンテナンス作業を実行します。
2. 実行系のホストで monswap コマンドを実行して、系を切り替えます。

```
# monswap サーバの識別名
```

下線部分には、servers ファイルのオペランド「alias」に指定されているサーバの識別名を指定します。

実行系のホストの Cosminexus が停止してから、今まで待機していた系で Cosminexus が起動します。これによって、メンテナンス済みのホストが実行系ホストになります。

3. 切り替え後の実行系のホスト（予備系のホスト）で、必要に応じて、定義の変更に関連した動作確認を実行します。
4. 切り替え後の待機系のホスト（現用系のホスト）で定義ファイルの編集など、メンテナンス作業を実行します。
5. 切り替え後の実行系のホスト（予備系のホスト）で、必要に応じて、再度 monswap コマンドを実行します。  
現用系のホストを実行系に戻す場合に実行してください。
6. 切り替え後の実行系のホスト（現用系のホスト）で、必要に応じて、定義の変更に関連した動作確認を実行します。  
これで、メンテナンス時の起動と停止の手順は完了です。

## 9. クラスタソフトウェアを使用したシステムの起動と停止 (UNIX の場合)

### 参考

servers ファイルの定義 (HA モニタでのサーバの環境設定) については、マニュアル「Cosminexus システム構築ガイド」の HA モニタの設定に関する説明を参照してください。

### 9.1.5 運用管理サーバの 1:1 系切り替えシステムをメンテナンスする場合の起動と停止

Management Server の設定を変更するために、運用管理サーバの 1:1 系切り替えシステムをメンテナンスする場合の起動と停止の手順について説明します。

なお、これらの手順は、すでに実行系のホストで Management Server が起動している場合の手順です。

1. 実行系のホスト (現用系のホスト) で、mstrexport コマンドを実行します。

```
# mstrexport mstruct.zip
```

現用系のホストの Management Server の設定情報が自動収集されて、収集された情報が ZIP 形式のファイルに保存されます。

下線部分には、自動収集された設定情報を保存する ZIP 形式のファイルのパスを指定します。

2. 手順 1. で保存された ZIP 形式のファイルを、実行系から待機系にコピーします。
3. 待機系のホスト (予備系のホスト) で、mstrimport コマンドを実行します。

```
# mstrimport mstruct.zip
```

下線部分には、手順 2 でコピーした ZIP 形式のファイルのパスを指定します。

これによって、手順 2. でコピーした Management Server の設定情報が待機系のホストに展開されて、実行系の Management Server と待機系の Management Server が同じ設定になります。

4. 実行系のホストで monswap コマンドを実行して、系を切り替えます。

```
# monswap サーバの識別名
```

下線部分には、servers ファイルのオペランド「alias」に指定されているサーバの識別名を指定します。

実行系のホストの Management Server が停止してから、今まで待機していた系で Management Server が起動します。Management Server の起動時に実行系の設定情報がシステムに反映されます。

5. 切り替え後の実行系のホスト (予備系のホスト) で、必要に応じて、定義の変更に関連した動作確認を実行します。
6. 切り替え後の実行系のホスト (予備系のホスト) で、必要に応じて、再度 `monswap` コマンドを実行します。  
現用系のホストを実行系に戻す場合に実行してください。
7. 切り替え後の実行系のホスト (現用系のホスト) で、必要に応じて、定義の変更に関連した動作確認を実行します。  
これで、メンテナンス時の起動と停止の手順は完了です。

---

参考

`servers` ファイルの定義 (HA モニタでのサーバの環境設定) については、マニュアル「Cosminexus システム構築ガイド」の HA モニタの設定に関する説明を参照してください。

---

## 9.2 相互系切り替えシステムの起動と停止

---

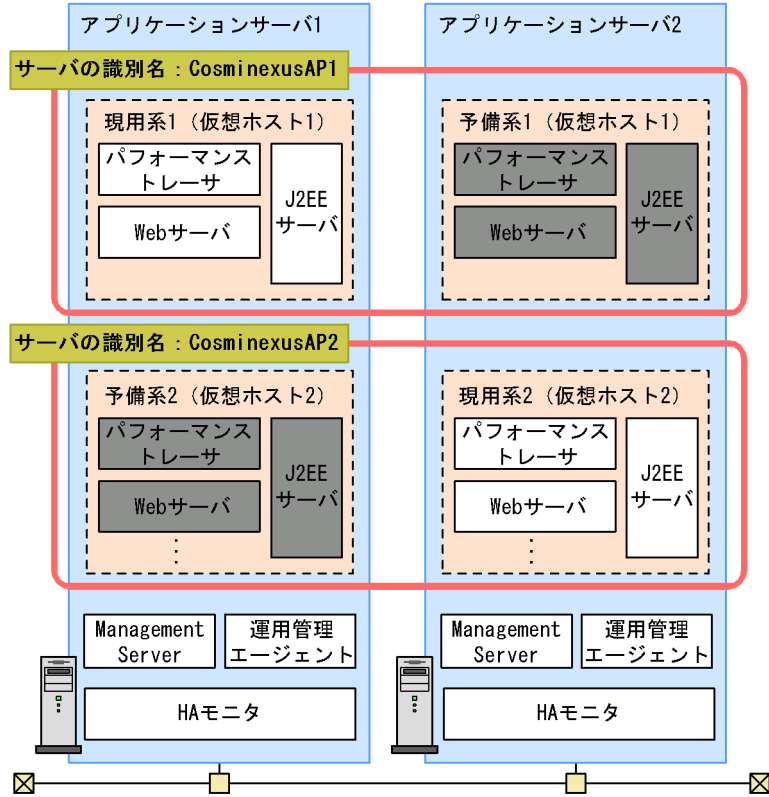
この節では、HA モニタによる相互系切り替えシステムを利用する場合の、システムの起動と停止の手順について説明します。稼働開始後に J2EE サーバ、およびバッチサーバをメンテナンスする場合の起動と停止の手順についても説明します。なお、HA モニタが利用できるのは、AIX、HP-UX または Linux の場合だけです。Solaris の場合、HA モニタは利用できません。

相互系切り替えシステムを利用して運用する場合は、あらかじめ現用系と予備系の 2 種類のホストの用意や、HA モニタの監視の対象となる運用管理エージェントを監視・起動・停止するスクリプトの登録など、必要な環境設定をしておく必要があります。設定方法については、マニュアル「Cosminexus システム構築ガイド」の HA モニタの設定に関する説明を参照してください。

相互系切り替えシステムのシステム構成例を次の図に示します。この節では、このシステム構成例を基に、相互系切り替えシステムの起動および停止の手順について説明します。



図 9-1 相互系切り替えシステムのシステム構成例



(凡例)

- : 系切り替えの単位となります。サーバの外から見た場合、一つのサーバに見えます。
- : 起動しているプロセス。
- : 起動していないプロセス。

図中の各アプリケーションサーバでの servers ファイルの定義（HA モニタでのサーバの環境設定）のうち、起動・停止に関連する定義項目を次の表に示します。

表 9-2 アプリケーションサーバ 1 の servers ファイルの定義

オペランド	説明	指定値	
		現用系 1 (仮想ホスト 1)	予備系 2 (仮想ホスト 2)
alias	サーバの識別名	CosminexusAP1	CosminexusAP2
initial	HA モニタでのサーバの起動方法	online	standby

## 9. クラスタソフトウェアを使用したシステムの起動と停止 (UNIX の場合)

表 9-3 アプリケーションサーバ 2 の servers ファイルの定義

オペランド	説明	指定値	
		予備系 1 (仮想ホスト 1)	現用系 2 (仮想ホスト 2)
alias	サーバの識別名	CosminexusAP1	CosminexusAP2
initial	HA モニタでのサーバの起動方法	standby	online

なお、servers ファイルの定義の詳細については、マニュアル「Cosminexus システム構築ガイド」の HA モニタの設定に関する説明を参照してください。

また、次のコマンドは、HA モニタが提供しているコマンドです。詳細は、マニュアル「高信頼化システム監視機能 HA モニタ」を参照してください。

- monbegin (HA モニタとのインタフェースを持たないサーバを起動)
- monend (HA モニタとのインタフェースを持たない実行サーバの停止連絡)
- monsbystp (待機サーバの停止)
- monswap (計画系切り替え)

### 参考

HA モニタの動作中の系を区別する場合、「実行系」と「待機系」の 2 種類の系に分けられます。

- 実行系  
現在実行中のサーバがある系のことです。系切り替えが発生すると、待機系になります。
- 待機系  
現在待機中のサーバがある系のことです。系切り替えが発生すると、実行系になります。

また、起動前の HA モニタを静的に区別する場合、「現用系」と「予備系」の 2 種類の系に分けられます。

- 現用系  
起動したときに、最初に実行系として起動される系のことです。なお、実行系として起動することは、HA モニタのサーバ対応環境設定用の定義ファイルで定義しておきます。
- 予備系  
起動したときに、最初に待機系として起動される系のことです。なお、待機系として起動することは、HA モニタのサーバ対応環境設定用の定義ファイルで定義しておきます。

## 9.2.1 相互系切り替えシステムの起動

相互系切り替えシステムを利用する場合のシステム起動時の留意事項、およびシステムの起動手順について説明します。

### （１）システム起動時の留意事項

相互系切り替えシステムを利用する場合のシステム起動時の留意事項を次に示します。

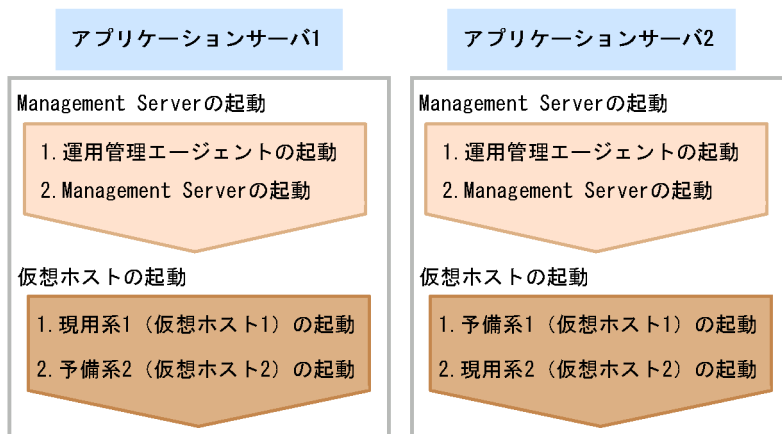
- アプリケーションサーバ 1 とアプリケーションサーバ 2 のホストに配置された HA モニタは、OS の起動と同時に起動されています。
- アプリケーションサーバ 1 およびアプリケーションサーバ 2 の各ホストで仮想ホストを起動する前に、Management Server をあらかじめ起動しておいてください。

### （２）システムの起動手順

相互系切り替えシステムの起動の流れを次の図に示します。

アプリケーションサーバ 1 およびアプリケーションサーバ 2 の各ホストで次の図に示す流れでシステムを起動してください。

図 9-2 相互系切り替えシステムの起動の流れ



#### 参考

アプリケーションサーバ 1 およびアプリケーションサーバ 2 の各ホストでの仮想ホストの起動順序は任意です。現用系仮想ホストまたは予備系仮想ホストのどちらから起動してもかまいません。なお、このマニュアルでは、仮想ホスト 1、仮想ホスト 2 の順序で起動する場合の手順について説明します。

図中の Management Server の起動の手順 1. ~ 手順 2. については、マニュアル「Cosminexus 運用管理操作ガイド」の運用管理ポータルと運用管理コマンドによるシステムの起動と停止に関する説明の各項を参照してください。なお、自動起動の設定をしている場合、ホストの起動時に、運用管理エージェント、Management Server も起動されるため、手順 1. ~ 手順 2. は省略できます。システムの起動方法の設定については、マニュアル「Cosminexus システム構築ガイド」のシステムの起動と停止の設定に関する説明を参照してください。

## 9. クラスタソフトウェアを使用したシステムの起動と停止 (UNIX の場合)

ここでは、図中の仮想ホストの起動手順について説明します。アプリケーションサーバ 1 およびアプリケーションサーバ 2 の各ホストでの起動手順を次に示します。

アプリケーションサーバ 1 での起動手順

### 1. 現用系 1 (仮想ホスト 1) の起動

monbegin コマンドを実行して、現用系 1 (仮想ホスト 1) を実行系として起動します。

```
# monbegin CosminexusAP1
```

これによって、現用系の仮想ホストを起動するスクリプトファイルに定義されている処理が実行されて、現用系の仮想ホストが実行系として起動されます。

### 2. 予備系 2 (仮想ホスト 2) の起動

monbegin コマンドを実行して、予備系 2 (仮想ホスト 2) を待機系として起動します。

```
# monbegin CosminexusAP2
```

これによって、予備系の仮想ホストが待機系となり、実行系の障害に備えます。

アプリケーションサーバ 2 での起動手順

### 1. 予備系 1 (仮想ホスト 1) の起動

monbegin コマンドを実行して、予備系 1 (仮想ホスト 1) を待機系として起動します。

```
# monbegin CosminexusAP1
```

これによって、予備系の仮想ホストが待機系となり、実行系の障害に備えます。

### 2. 現用系 2 (仮想ホスト 2) の起動

monbegin コマンドを実行して、現用系 2 (仮想ホスト 2) を実行系として起動します。

```
# monbegin CosminexusAP2
```

これによって、現用系の仮想ホストを起動するスクリプトファイルに定義されている処理が実行されて、現用系の仮想ホストが実行系として起動されます。

## 9.2.2 相互系切り替えシステムの停止

相互系切り替えシステムの停止手順について説明します。

ここでは、次の二つの場合の停止手順について説明します。

- 実行系と待機系の両方の仮想ホストを停止する場合

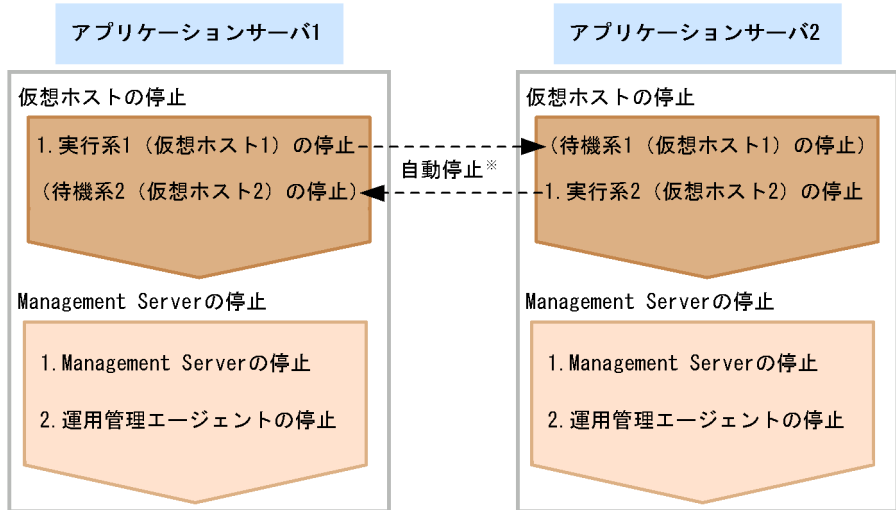
- ・ 待機系の仮想ホストだけ停止する場合

### （１）実行系と待機系の両方の仮想ホストを停止する場合

実行系と待機系の両方の仮想ホストを停止する場合の手順について説明します。

実行系と待機系の両方の仮想ホストを停止する場合の、相互系切り替えシステムの停止の流れを次の図に示します。

図 9-3 相互系切り替えシステムの停止の流れ



（凡例）

--▶ : HAモータによる自動停止の指示の流れ

注※ 実行系が停止したあと、HAモータによって待機系に停止指示が出され、待機系が自動停止します。

#### 参考

仮想ホスト 1 および仮想ホスト 2 の停止順序は任意です。このマニュアルでは、仮想ホスト 1、仮想ホスト 2 の順序で停止する場合の手順について説明します。

すべての仮想ホストを停止したあと、アプリケーションサーバ 1 およびアプリケーションサーバ 2 の各ホストの Management Server を停止します。

図中の Management Server の停止の手順 1. ~ 手順 2. については、マニュアル「Cosminexus 運用管理操作ガイド」の運用管理ポータルと運用管理コマンドによるシステムの起動と停止に関する説明の各項を参照してください。なお、自動停止の設定をしている場合、ホストの停止時に、Management Server、および運用管理エージェントも停止されるため、手順 1. ~ 手順 2. は省略できます。システムの停止方法の設定については、マニュアル「Cosminexus システム構築ガイド」のシステムの起動と停止の設定

## 9. クラスタソフトウェアを使用したシステムの起動と停止 (UNIX の場合)

に関する説明を参照してください。

ここでは、図中の仮想ホストの停止手順について説明します。アプリケーションサーバ 1 およびアプリケーションサーバ 2 の各ホストでの停止手順を次に示します。

アプリケーションサーバ 1 での停止手順

1. アプリケーションサーバ 1 で `monend` コマンドを実行して、実行系の仮想ホスト 1 を停止します。

```
# monend CosminexusAP1
```

これによって、アプリケーションサーバ 1 の実行系 1 (仮想ホスト 1) が停止します。また、HA モニタによって待機系に自動的に停止指示が出され、アプリケーションサーバ 2 の待機系 1 (仮想ホスト 1) も停止します。

アプリケーションサーバ 2 での停止手順

1. アプリケーションサーバ 2 で `monend` コマンドを実行して、実行系の仮想ホスト 2 を停止します。

```
# monend CosminexusAP2
```

これによって、アプリケーションサーバ 2 の実行系 2 (仮想ホスト 2) が停止します。また、HA モニタによって待機系に自動的に停止指示が出され、アプリケーションサーバ 1 の待機系 2 (仮想ホスト 2) も停止します。

### (2) 待機系の仮想ホストだけ停止する場合

実行系の仮想ホストを停止しないで待機系の仮想ホストで待機状態をやめる場合の手順について説明します。

アプリケーションサーバ 1 の待機系 2 (仮想ホスト 2) で待機状態をやめる場合  
アプリケーションサーバ 1 のホストで次のコマンドを実行します。

```
# monsbystp CosminexusAP2
```

アプリケーションサーバ 2 の待機系 1 (仮想ホスト 1) で待機状態をやめる場合  
アプリケーションサーバ 2 のホストで次のコマンドを実行します。

```
# monsbystp CosminexusAP1
```

## 9.2.3 相互系切り替えシステムで計画的に系を切り替える場合の起動と停止

相互系切り替えシステムでトラブル発生時以外の場合に、実行系と待機系を計画的に切り替えるときの手順について説明します。系を切り替えるときには、待機系の仮想ホストが待機状態になっている必要があります。

ここでは、仮想ホスト 1 および仮想ホスト 2 の実行系と待機系を計画的に切り替える場合の手順について説明します。なお、現用系のホストが実行系として起動されていることを前提にして説明します。

### 参考

相互系切り替えシステムをメンテナンスする場合の起動と停止の手順については、「9.2.4 相互系切り替えシステムをメンテナンスする場合の起動と停止」を参照してください。

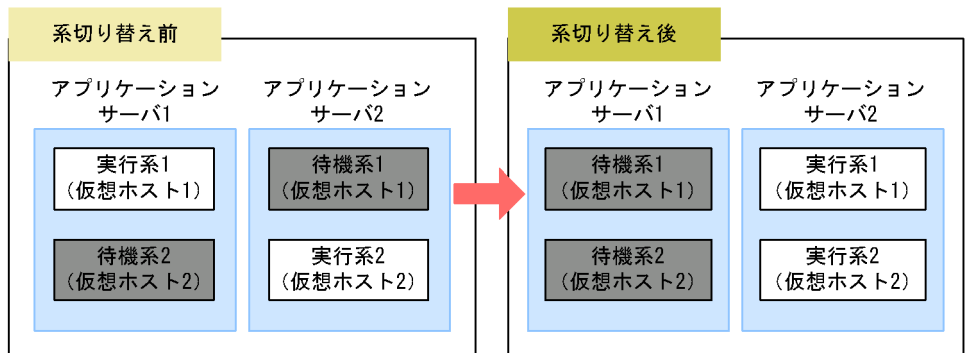
### (1) 仮想ホスト 1 を計画的に系切り替えする場合

仮想ホスト 1 を計画的に系切り替えする場合の手順を次に示します。

1. アプリケーションサーバ 1 で monswap コマンドを実行して、系を切り替えます。

```
# monswap CosminexusAP1
```

これによって、次の図に示すように、アプリケーションサーバ 1 の実行系 1（仮想ホスト 1）が待機系の仮想ホストに切り替わり、アプリケーションサーバ 2 の待機系 1（仮想ホスト 1）が実行系の仮想ホストに切り替わります。

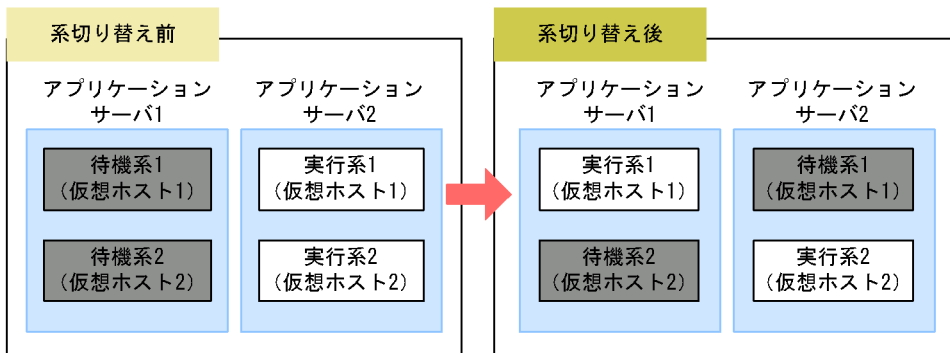


2. 通常運用の実行系と待機系の状態に戻す場合、アプリケーションサーバ 2 で monswap コマンドを実行して、系を切り替えます。

```
# monswap CosminexusAP1
```

## 9. クラスタソフトウェアを使用したシステムの起動と停止 (UNIX の場合)

これによって、次の図に示すように、アプリケーションサーバ2の実行系1(仮想ホスト1)が待機系の仮想ホストに切り替わり、アプリケーションサーバ1の待機系1(仮想ホスト1)が実行系の仮想ホストに切り替わります。これで、通常運用の実行系と待機系の状態に戻ります。



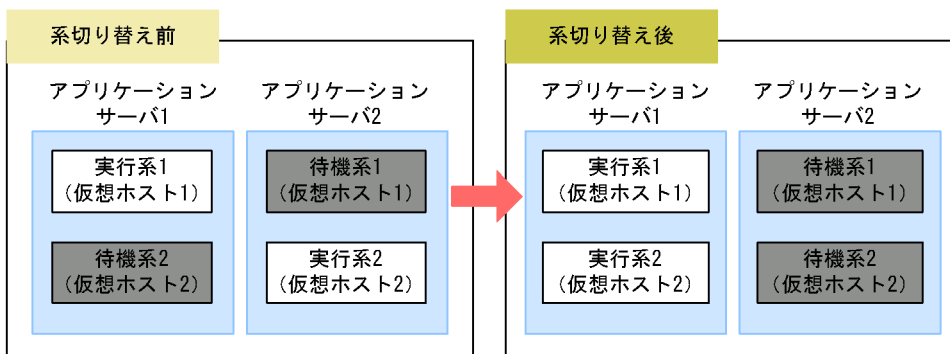
### (2) 仮想ホスト2を計画的に系切り替えする場合

仮想ホスト2を計画的に系切り替えする場合の手順を次に示します。

1. アプリケーションサーバ2で monswap コマンドを実行して、系を切り替えます。

```
# monswap CosminexusAP2
```

これによって、次の図に示すように、アプリケーションサーバ2の実行系2(仮想ホスト2)が待機系の仮想ホストに切り替わり、アプリケーションサーバ1の待機系2(仮想ホスト2)が実行系の仮想ホストに切り替わります。

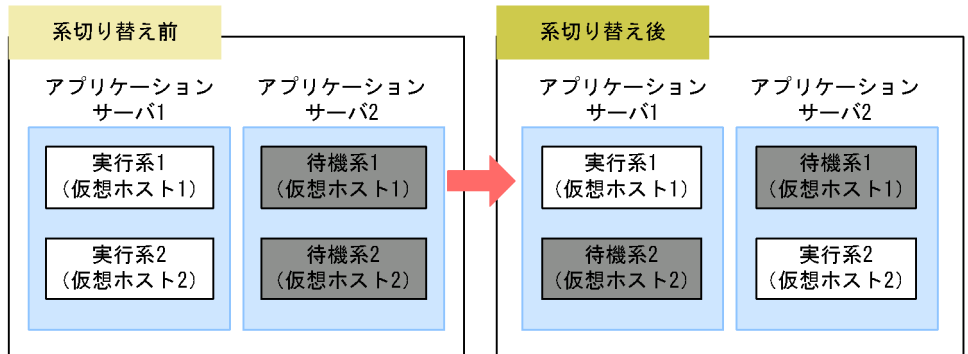


2. 通常運用の実行系と待機系の状態に戻す場合、アプリケーションサーバ1で monswap コマンドを実行して、系を切り替えます。

```
# monswap CosminexusAP2
```



これによって、次の図に示すように、アプリケーションサーバ 1 の実行系 2（仮想ホスト 2）が待機系の仮想ホストに切り替わり、アプリケーションサーバ 2 の待機系 2（仮想ホスト 2）が実行系の仮想ホストに切り替わります。これで、通常運用の実行系と待機系の状態に戻ります。



## 9.2.4 相互系切り替えシステムをメンテナンスする場合の起動と停止

相互系切り替えシステムをメンテナンスする場合の、起動と停止の手順を次に示します。

ここでは、次の三つの場合の停止手順について説明します。

- 再起動が不要なメンテナンスの場合
- 再起動が必要なメンテナンスの場合（両方の系を同時に停止する方法）
- 再起動が必要なメンテナンスの場合（両方の系を同時に停止しない方法）

なお、これらの手順は、すでに実行系の仮想ホストが両方とも起動している場合の手順です。

### （1）再起動が不要なメンテナンスの場合

再起動が不要なメンテナンスの場合の起動と停止の手順を次に示します。

1. 実行系の仮想ホスト（現用系の仮想ホスト）で、動作中の J2EE アプリケーションとリソースアダプタを停止します。  
バッチサーバの場合は、バッチアプリケーションが実行中でないことを確認してからリソースアダプタを停止してください。  
停止方法については、マニュアル「Cosminexus 運用管理操作ガイド」の運用管理ポータルと運用管理コマンドによるシステムの起動と停止に関する説明を参照してください。
2. サーバ管理コマンドを使用してメンテナンス処理を実行します。  
サーバ管理コマンドを使用した J2EE アプリケーションやリソースの設定方法については、マニュアル「Cosminexus システム構築ガイド」の J2EE アプリケーションの設定に関する説明、およびリソースの設定に関する説明を参照してください。

## 9. クラスタソフトウェアを使用したシステムの起動と停止（UNIX の場合）

- 手順 1. で停止した J2EE アプリケーションとリソースアダプタを開始します。  
J2EE サーバの場合は J2EE アプリケーションとリソースアダプタを開始します。また、バッチサーバの場合はリソースアダプタを開始します。  
開始方法については、マニュアル「Cosminexus 運用管理操作ガイド」の運用管理ポータルと運用管理コマンドによるシステムの起動と停止に関する説明を参照してください。
- 必要に応じて J2EE アプリケーションとリソースアダプタの動作確認をします。
- 実行系の仮想ホスト（現用系の仮想ホスト）に対して、monswap コマンドを実行して、系を切り替えます。

```
# monswap サーバの識別名
```

下線部分には、実行系の仮想ホストが配置されているアプリケーションサーバの、servers ファイルのオペランド「alias」に指定されている実行系の仮想ホストのサーバの識別名を指定します。

実行系の仮想ホスト（現用系の仮想ホスト）が待機系の仮想ホストに切り替わり、待機系だった仮想ホスト（予備系の仮想ホスト）が実行系の仮想ホストになります。

- 切り替え後の実行系の仮想ホスト（予備系の仮想ホスト）で手順 1. ～手順 4. を実行します。
- 切り替え後の実行系のホスト（予備系の仮想ホスト）で、必要に応じて、再度 monswap コマンドを実行します。

```
# monswap サーバの識別名
```

下線部分には、手順 5. で monswap コマンドの引数に指定したサーバの識別名と同じサーバの識別名を指定します。

現用系の仮想ホストを実行系に戻す場合に実行してください。

これで、メンテナンス時の起動と停止の手順は完了です。

### （2）再起動が必要なメンテナンスの場合（両方の系を同時に停止する方法）

再起動が必要なメンテナンスの場合の起動と停止の手順を次に示します。

- 実行系の仮想ホストに対して monend コマンドを実行して、システムを停止します。

```
# monend サーバの識別名
```

下線部分には、実行系の仮想ホストが配置されているアプリケーションサーバの、servers ファイルのオペランド「alias」に指定されている実行系の仮想ホストのサーバの識別名を指定します。

実行系の仮想ホストが停止します。また、HA モニタによって待機系に自動的に停止指示が出され、待機系の仮想ホストも停止します。

2. 現用系と予備系の両方の仮想ホストに対して、定義ファイルの変更など、メンテナンス作業を実行します。
3. 現用系の仮想ホストに対して、monbegin コマンドを実行します。

```
# monbegin サーバの識別名
```

下線部分には、手順 1. で monend コマンドの引数に指定したサーバの識別名と同じサーバの識別名を指定します。

これによって、現用系の仮想ホストが、実行系として起動します。

4. 予備系の仮想ホストに対して、monbegin コマンドを実行します。

```
# monbegin サーバの識別名
```

下線部分には、手順 1. で monend コマンドの引数に指定したサーバの識別名と同じサーバの識別名を指定します。

これによって、予備系の仮想ホストが、待機系として起動します。

5. 実行系の仮想ホストで、必要に応じて、定義の変更に関連した動作確認を実行します。

---

#### 参考

servers ファイルの定義（HA モニタでのサーバの環境設定）については、マニュアル「Cosminexus システム構築ガイド」の HA モニタの設定に関する説明を参照してください。

---

### （3）再起動が必要なメンテナンスの場合（両方の系を同時に停止しない方法）

再起動が必要なメンテナンスの場合の起動と停止の手順を次に示します。

1. 待機系の仮想ホストで、定義ファイルの変更など、メンテナンス作業を実行します。
2. 実行系の仮想ホストに対して monswap コマンドを実行して、系を切り替えます。

```
# monswap サーバの識別名
```

下線部分には、実行系の仮想ホストが配置されているアプリケーションサーバの、servers ファイルのオペラント「alias」に指定されている実行系の仮想ホストのサーバの識別名を指定します。

## 9. クラスタソフトウェアを使用したシステムの起動と停止 (UNIX の場合)

実行系の仮想ホストが停止してから、待機していた系が起動します。これによって、メンテナンス済みの仮想ホストが実行系の仮想ホストになります。

3. 切り替え後の実行系のホスト (予備系の仮想ホスト) に対して、必要に応じて、定義の変更に関連した動作確認を実行します。
4. 切り替え後の待機系のホスト (現用系の仮想ホスト) で定義ファイルの編集など、メンテナンス作業を実行します。
5. 切り替え後の実行系の仮想ホスト (予備系のホスト) に対して、必要に応じて、再度 monswap コマンドを実行します。

```
# monswap サーバの識別名
```

下線部分には、手順 2. で monswap コマンドの引数に指定したサーバの識別名と同じサーバの識別名を指定します。

現用系の仮想ホストを実行系に戻す場合に実行してください。

6. 切り替え後の実行系の仮想ホスト (現用系の仮想ホスト) で、必要に応じて、定義の変更に関連した動作確認を実行します。  
これで、メンテナンス時の起動と停止の手順は完了です。

## 9.3 N:1 リカバリシステムの起動と停止

---

この節では、HA モニタによる N:1 リカバリシステムを利用する場合の、システムの起動と停止の手順について説明します。なお、HA モニタが利用できるのは、AIX、HP-UX または Linux の場合だけです。Solaris の場合、HA モニタは利用できません。

HA モニタによる N:1 リカバリシステムを利用して運用する場合は、あらかじめ実行系と待機系の 2 種類のホストの用意や、HA モニタの監視の対象となる運用管理エージェントを監視・起動・停止するスクリプトの登録など、必要な環境設定をしておく必要があります。設定方法については、マニュアル「Cosminexus システム構築ガイド」システムの起動と停止の設定に関する説明を参照してください。

### ポイント

---

HA モニタによる N:1 リカバリシステムを利用する場合、Management Server は運用管理サーバで起動して、個々のアプリケーションサーバのホストでは、運用管理エージェントだけを起動します。なお、個々のアプリケーションサーバのホストでは、運用管理エージェントを OS 起動と同時に起動する設定にしないでください。

また、次のコマンドは、HA モニタが提供しているコマンドです。詳細は、マニュアル「高信頼化システム監視機能 HA モニタ」を参照してください。

- monbegin (HA モニタとのインタフェースを持たないサーバを起動)
- monend (HA モニタとのインタフェースを持たない実行サーバの停止連絡)

### 参考

---

HA モニタの動作中の系を区別する場合、「実行系」と「待機系」の 2 種類の系に分けられます。

- 実行系  
業務処理を実行する N 台のアプリケーションサーバがある系のことです。
- 待機系  
1 台のリカバリ専用サーバがある系のことです。

### 9.3.1 N:1 リカバリシステムの起動

HA モニタによる N:1 リカバリシステムを利用する場合のシステム起動時の留意事項、およびシステムの起動手順について説明します。

#### (1) システム起動時の留意事項

HA モニタによる N:1 リカバリシステムを利用する場合のシステム起動時の留意事項を次に示します。

## 9. クラスタソフトウェアを使用したシステムの起動と停止 (UNIX の場合)

- 実行系と待機系のホストに配置された HA モニタは、OS の起動と同時に起動されています。
- 運用管理サーバを起動していない場合は、運用管理サーバをあらかじめ起動しておいてください。運用管理サーバの起動方法については、マニュアル「Cosminexus 運用管理操作ガイド」の運用管理ポータルと運用管理コマンドによるシステムの起動と停止に関する説明を参照してください。

### (2) システムの起動手順

HA モニタによる N:1 リカバリシステムを利用する場合のシステムの起動手順を次に示します。

1. 待機系のホストで、各実行系に対応する monbegin コマンドを N 回実行します。

```
# monbegin サーバの識別名
```

下線部分には、servers ファイルのオペランド「alias」に指定されている実行系のサーバの識別名を指定します。

これによって、待機系のリカバリ専用サーバが実行系の監視を開始します。

2. 実行系の各ホスト (N 台) で monbegin コマンドを実行して、実行系の各ホストの運用管理エージェントを起動します。

```
# monbegin サーバの識別名
```

下線部分には、servers ファイルのオペランド「alias」に指定されているサーバの識別名を指定します。

これによって、実行系の各ホストの運用管理エージェントが起動します。

#### ! 注意事項

Management Server 起動時に運用管理ドメイン内の論理サーバを一括起動する設定にしている場合は、実行系のホストの論理サーバを起動してください。なお、Management Server 起動時に運用管理ドメイン内の論理サーバを一括起動するように設定にしている場合、論理サーバの手動起動は不要です。起動方法については、マニュアル「Cosminexus 運用管理操作ガイド」の運用管理ポータルと運用管理コマンドによるシステムの起動と停止に関する説明を参照してください。

## 参考

---

servers ファイルの定義 (HA モニタでのサーバの環境設定) については、マニュアル「Cosminexus システム構築ガイド」の HA モニタの設定に関する説明を参照してください。

---

### 9.3.2 N:1 リカバリシステムの停止

HA モニタによる N:1 リカバリシステムを利用する場合のシステムの停止手順を次に示します。

1. 運用管理エージェントを停止するスクリプトで、運用管理ドメイン内の論理サーバを停止する設定にしていない場合は、実行系のホストで各論理サーバを停止します。運用管理エージェントを停止するスクリプトで論理サーバを停止する設定にしている場合は、不要な手順です。論理サーバの停止方法については、マニュアル「Cosminexus 運用管理操作ガイド」の運用管理ポータルと運用管理コマンドによるシステムの起動と停止に関する説明を参照してください。
2. 実行系の各ホスト (N 台) で monend コマンドを実行して、実行系のホストの運用管理エージェントを停止します。

```
# monend サーバの識別名
```

下線部分には、servers ファイルのオペランド「alias」に指定されているサーバの識別名を指定します。

これによって、実行系の各ホストの運用管理エージェントが停止します。

## 参考

---

servers ファイルの定義 (HA モニタでのサーバの環境設定) については、マニュアル「Cosminexus システム構築ガイド」の HA モニタの設定に関する説明を参照してください。

---

なお、運用管理サーバは最後に手動で停止します。運用管理サーバの停止方法については、マニュアル「Cosminexus 運用管理操作ガイド」の運用管理ポータルと運用管理コマンドによるシステムの起動と停止に関する説明を参照してください。





# 付録

---

付録 A トレース取得ポイントと PRF トレース取得レベル

---

付録 B snapshot ログの収集対象一覧

---

付録 C データベースと接続中にトラブルが発生したコネクションの特定

---

付録 D 機能ごとに出力されるログ情報

---

付録 E 障害発生時の CMR 用の表の回復

---

付録 F ベーシックモードの利用（互換機能）

---

付録 G サブレットエンジンモードでのシステム運用

---

付録 H 代替手段で実行できる運用作業

---

付録 I 用語解説

---

## 付録 A トレース取得ポイントと PRF トレース取得レベル

ここでは、トレース取得ポイントと PRF トレース取得レベルについて説明します。

トレース取得ポイントは大きく分けて、J2EE サーバの開始・終了時でのトレース取得と、各機能レイヤでの処理中のトレース取得があります。

J2EE サーバの開始・終了時のトレース取得

J2EE サーバの開始処理の完了時、および J2EE サーバの終了処理の開始時に、トレース情報を取得できます。取得できるイベント ID と参照先を次に示します。

- イベント ID

0x8FFE ~ 0x8FFF

- 参照先

J2EE サーバの開始・終了時のトレース取得の詳細については、「付録 A.16 J2EE サーバの開始・終了時のトレース取得ポイント」を参照してください。

各機能レイヤでのトレース取得

取得できるイベント ID と機能レイヤの対応を次の表に示します。

表 A-1 取得できるイベント ID と機能レイヤ

イベント ID	機能レイヤ	図中の番号	参照先
0x1101 ~ 0x1102 0x1301 ~ 0x1302 0x1401 ~ 0x1406 0x2002 ~ 0x2003 0x2101 ~ 0x2104 0x3000 ~ 0x3008	CTM	5	付録 A.1
0x8000 ~ 0x8004 0x8100 ~ 0x8104	リダイレクタ	1	付録 A.2
0x8200 ~ 0x8203 0x8206 ~ 0x8218 0x8300 ~ 0x8303 0x8306 ~ 0x8307 0x8311 ~ 0x8318 0x8E01 ~ 0x8E06	Web コンテナ	2	付録 A.3 付録 A.4 付録 A.5 付録 A.6
0x8401 ~ 0x840A 0x8425 ~ 0x8428 0x842D ~ 0x8434 0x8453 ~ 0x8454 0x8460 ~ 0x846D 0x8470 ~ 0x8477 0x8490 ~ 0x8491 0x8C41	EJB コンテナ	6	付録 A.7

イベント ID	機能レイヤ	図中の番号	参照先
0x8603 ~ 0x861C	JNDI	4	付録 A.8
0x8811 ~ 0x8820 0x8C41	JTA	7	付録 A.9
0x8B00 ~ 0x8B01 0x8B80 ~ 0x8B89 0x8B8A ~ 0x8C03 0x8C10 ~ 0x8C13 0x8C20 ~ 0x8C41 0x8C60 ~ 0x8C65 0x8C80 ~ 0x8C93 0x8CC0 ~ 0x8CD9 0x8D00 ~ 0x8D19 0x8D60 ~ 0x8D63 0x8D80 ~ 0x8D89 0x8D8A ~ 0x8D8F 0x8D90 ~ 0x8D99	DB Connector , JCA コ ンテナ	8	付録 A.10
0x8E01 ~ 0x8E06	RMI	3	付録 A.11
0x9400 ~ 0x94FF	OTS	9	付録 A.12
0x9C00 ~ 0x9C03	標準出力 / 標準エラー 出力 / ユーザログ	-	付録 A.13
0x9D00 , 0x9D01	DI	10	付録 A.14
0xA100 , 0xA101	バッチアプリケーション 実行機能	11	付録 A.15

( 凡例 )

- : 該当しません。

注

図 A-1 ~ 図 A-3 中の番号と対応しています。

機能レイヤとトレース取得ポイントについて、次の図に示します。

図 A-1 トレース取得ポイント (Web クライアント構成の場合)

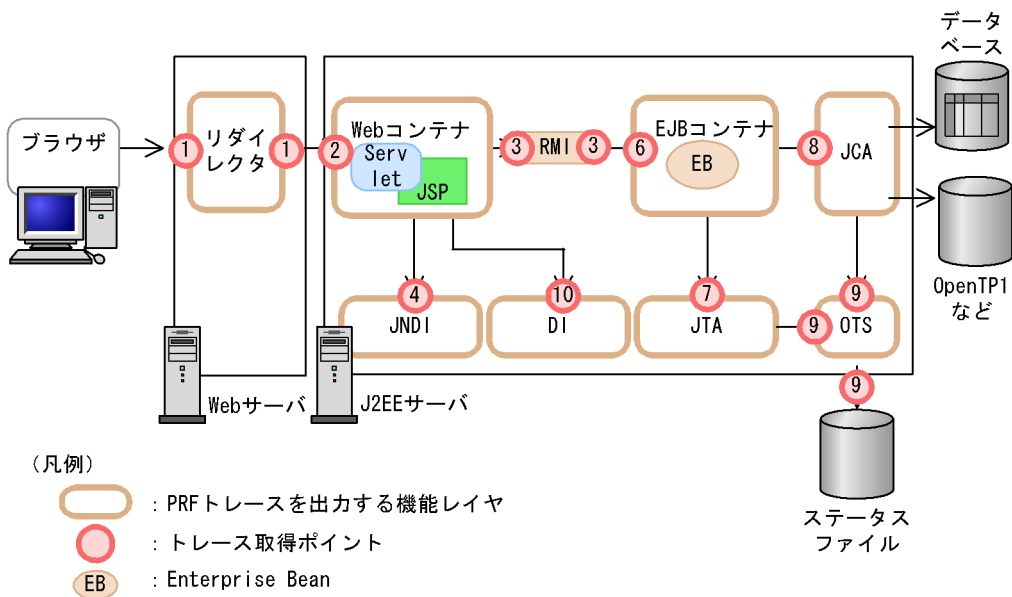


図 A-2 トレース取得ポイント (バッチアプリケーションを実行するシステムの場合)

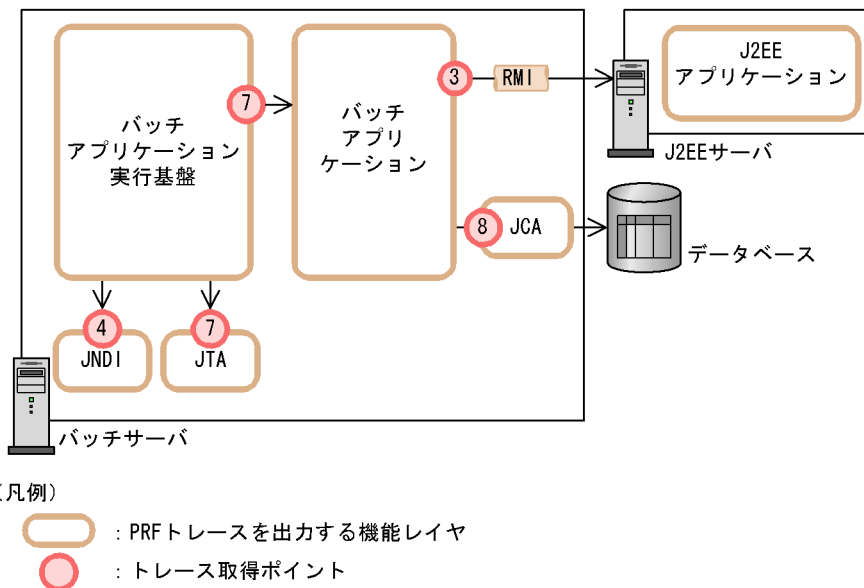
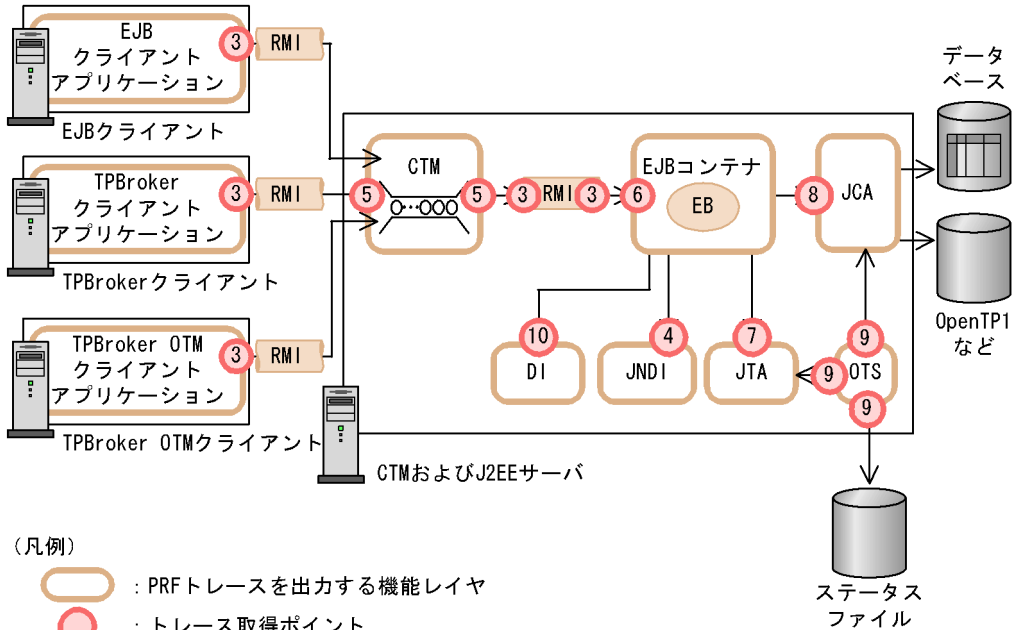


図 A-3 トレース取得ポイント (EJB クライアント /CORBA クライアント /OTM クライアント構成の場合)



(凡例)

- : PRF トレースを出力する機能レイヤ
- : トレース取得ポイント
- EB : Enterprise Bean

トレース取得ポイントは、各機能レイヤ内でさらに詳細に分かれており、トレース取得ポイントによって、PRF トレース取得レベルが異なります。各機能レイヤの詳細なトレース取得ポイント、および PRF トレース取得レベルについては、付録 A.1 ~ 付録 A.15 を参照してください。

参考

表 A-1 に示した機能レイヤのほか、Application Server のプロセス、構成ソフトウェアおよび関連プログラムでも、PRF トレースが取得できるものがあります。

表 A-1 に示した以外で PRF トレースを取得できる機能レイヤとイベント ID の対応を、次の表に示します。

表 A-2 表 A-1 に示した以外で PRF トレースを取得できる機能レイヤとイベント ID の対応

イベント ID	機能レイヤ
0x9000 ~ 0x90FF	• Cosminexus Web Services - Base
0x9100 ~ 0x91FF	• uCosminexus TP1 Connector • TP1/Client/J
0x9200 ~ 0x92FF	• TP1/MQ Access
0x9300 ~ 0x93FF	• Cosminexus RM
0x9800 ~ 0x9B6E	• HCSC サーバ
0x9F00 ~ 0x9FFF	• uCosminexus Service Adapter for Object Access
0x9E00 ~ 0x9EFF	• uCosminexus Service Coordinator Interactive Workflow
0xA000 ~ 0xA0FF	• uCosminexus Service Adapter for Flat Files

なお、ここではトレース取得レベルが標準レベルと詳細レベルのトレース取得ポイントについて説明します。保守レベルは、障害発生時などの保守情報を取得するためのレベルのため、通常は収集する必要はありません。

## 付録 A.1 CTM のトレース取得ポイント

ここでは、CTM でのトレース取得ポイントと、取得できるトレース情報について説明します。

### トレース取得ポイントと PRF トレース取得レベル

イベント ID、トレース取得ポイント、および PRF トレース取得レベルについて、次の表に示します。

表 A-3 CTM でのトレース取得ポイントの詳細

イベント ID	図中の番号	トレース取得ポイント	レベル
0x1101	21	OTM ゲートウェイからのリクエスト送信直前	A
0x1102	22	OTM ゲートウェイからのレスポンス受信直前	A
0x1301	4	EJB レギュレータからのリクエスト送信直前	A
0x1302	13	EJB レギュレータへのレスポンス受信直後	A
0x1401	1	CTM 内の Create メソッドの入り口	A
0x1402	2	CTM 内の Create メソッドの出口	A
0x1403	5	EJB レギュレータからのリクエスト受信直後	A

イベント ID	図中の番号	トレース取得ポイント	レベル
0x1404	8	CTM から J2EE サーバへのリクエスト送信直前	A
0x1405	9	CTM での、J2EE サーバからのリクエスト受信直後	A
0x1406	12	CTM から EJB レギュレータへのレスポンス送信直前	A
0x2002	23	OTM クライアントからのリクエスト受信直前	A
0x2003	24	OTM クライアントからのレスポンス受信直前	A
0x2101	3	EJB クライアントからのリクエスト受信直後	A
0x2102	14	EJB クライアントへのレスポンス送信直前	A
0x2103	15	CTM 内の Remove メソッドの入り口	A
0x2104	16	CTM 内の Remove メソッドの出口	A
0x3000	6	リクエストをキューイングする直前	A
0x3001	7	キューからリクエストを取り出した直後	A
0x3002	17	他 CTM へのリクエスト送信直前	A
0x3003	18	他 CTM からのリクエスト受信直後	A
0x3004	10	リクエスト応答をキューイングする直前	B
0x3005	11	キューからリクエスト応答を取り出した直後	B
0x3006	19	他 CTM へのリクエスト応答送信直前	A
0x3007	20	他 CTM からのリクエスト応答受信直後	A
0x3008	-	サーバ状態変更時に取得されます。 (リクエスト処理中には取得されません)	B

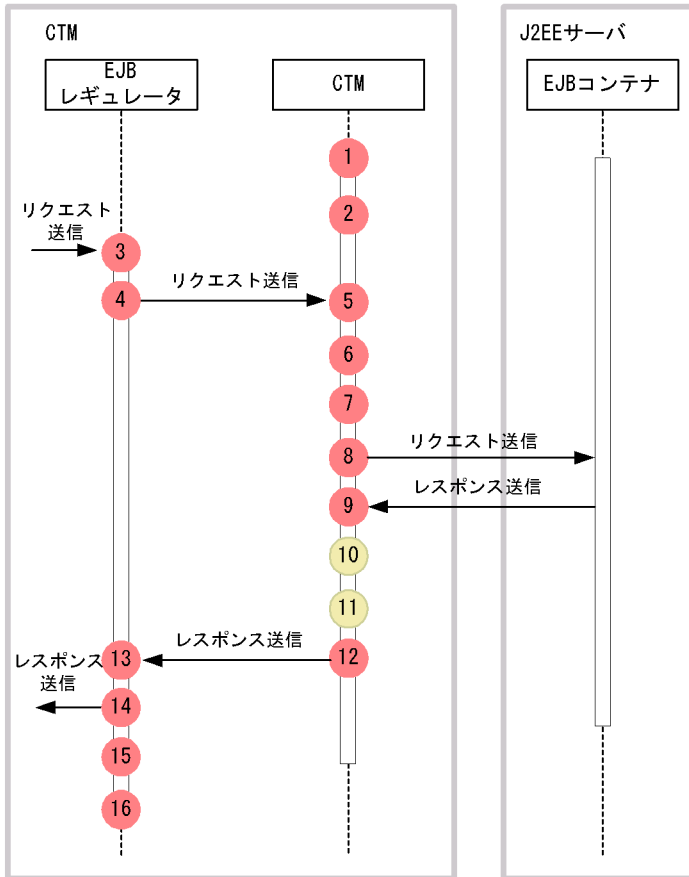
(凡例) A: 標準 B: 詳細 -: 該当なし

注

図 A-4 ~ 図 A-7 中の番号と対応しています。

CTM でのトレース取得ポイントを次の図に示します。

図 A-4 CTM のトレース取得ポイント

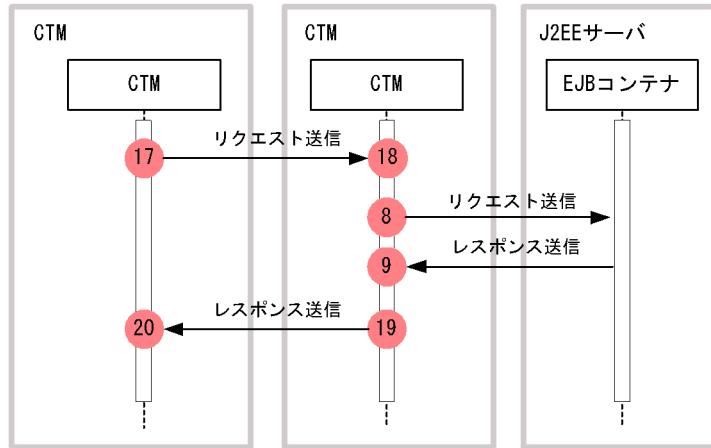


- (凡例)
- (Red circle) : トレース取得ポイントを示します。PRF トレース取得レベルは「標準」です。
  - (Yellow circle) : トレース取得ポイントを示します。PRF トレース取得レベルは「詳細」です。

ほかの CTM と連携している場合は、次の図に示す場所でもトレース情報を取得します。上記の図とあわせて参照してください。なお、この図では、ほかの CTM と連携している個所だけを取り上げています。



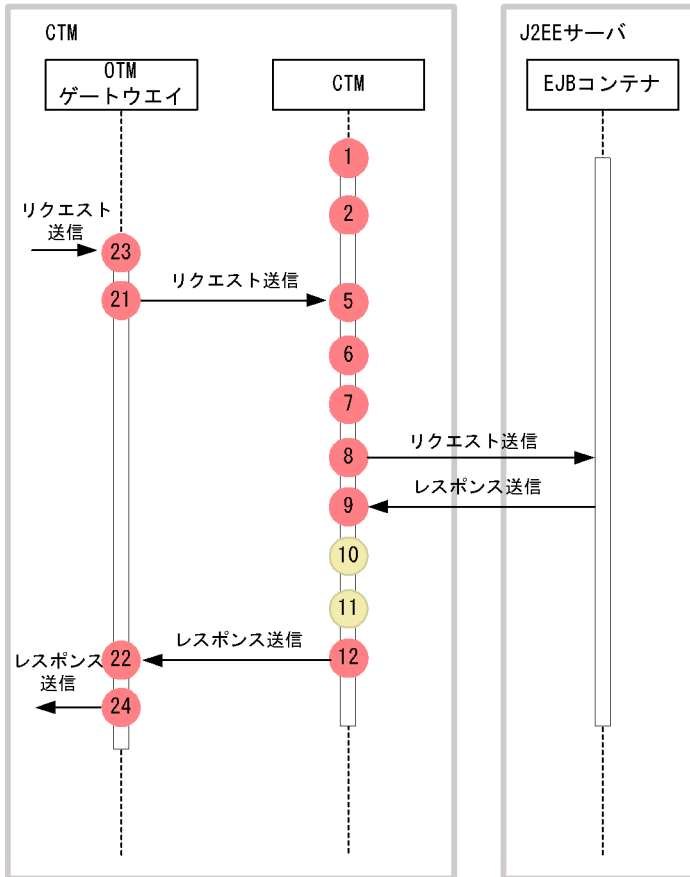
図 A-5 CTM のトレース取得ポイント (他 CTM と連携している場合)



(凡例) ● : トレース取得ポイントを示します。PRF トレース取得レベルは「標準」です。

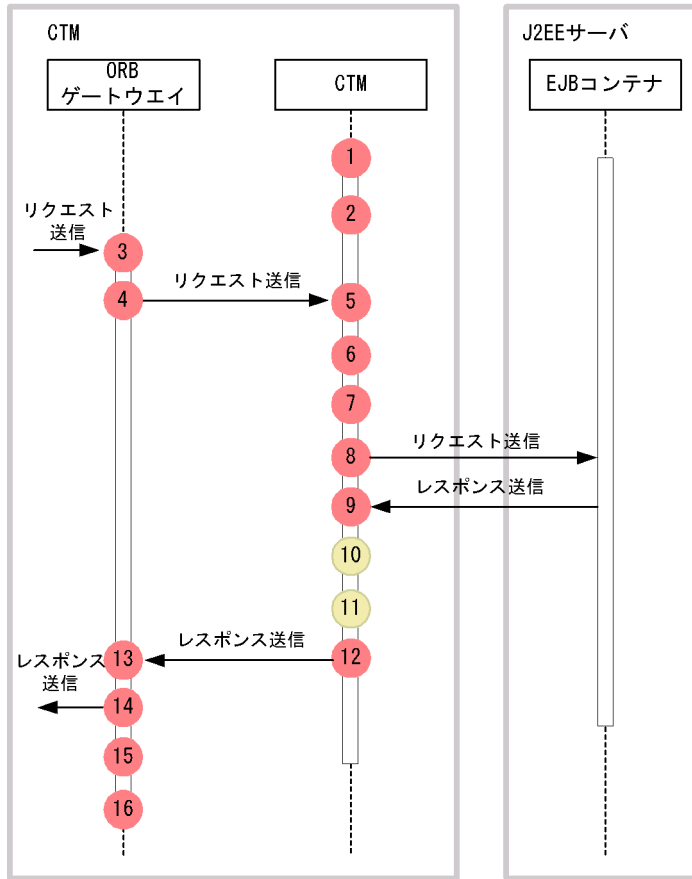
また、OTM ゲートウェイまたは ORB ゲートウェイを使用する場合は、次の図に示す場所でもトレース情報を取得します。上記の図とあわせて参照してください。なお、この図では、ほかの CTM と連携している個所だけを取り上げています。

図 A-6 CTM のトレース取得ポイント (OTM ゲートウェイを使用する場合)



- (凡例)
- (Red circle) : トレース取得ポイントを示します。PRF トレース取得レベルは「標準」です。
  - (Yellow circle) : トレース取得ポイントを示します。PRF トレース取得レベルは「詳細」です。

図 A-7 CTM のトレース取得ポイント (ORB ゲートウェイを使用する場合)



- (凡例)
- (Red circle) : トレース取得ポイントを示します。PRF トレース取得レベルは「標準」です。
  - (Yellow circle) : トレース取得ポイントを示します。PRF トレース取得レベルは「詳細」です。

取得できるトレース情報

CTM で取得できるトレース情報を次の表に示します。

表 A-4 CTM で取得できるトレース情報

図中の番号	イベント ID	レベル	取得できる情報		
			インタフェース名	オペレーション名	オプション
1	0x1401	A	リモートインタフェース名	メソッド名	-
2	0x1402	A	リモートインタフェース名	メソッド名	内部情報
3	0x2101	A	リモートインタフェース名	メソッド名	-
4	0x1301	A	リモートインタフェース名	メソッド名	-

図中の 番号	イベン ト ID	レベ ル	取得できる情報		
			インタフェース名	オペレーショ ン名	オプション
5	0x1403	A	リモートインタフェース名	メソッド名	-
6	0x3000	A	-	-	-
7	0x3001	A	-	-	内部情報
8	0x1404	A	リモートインタフェース名	メソッド名	内部情報
9	0x1405	A	リモートインタフェース名	メソッド名	内部情報
10	0x3004	B	-	-	-
11	0x3005	B	-	-	内部情報
12	0x1406	A	リモートインタフェース名	メソッド名	-
13	0x1302	A	リモートインタフェース名	メソッド名	-
14	0x2102	A	リモートインタフェース名	メソッド名	-
15	0x2103	A	リモートインタフェース名	メソッド名	-
16	0x2104	A	リモートインタフェース名	メソッド名	-
17	0x3002	A	-	-	-
18	0x3003	A	-	-	-
19	0x3006	A	-	-	-
20	0x3007	A	-	-	-
21	0x1101	A	リモートインタフェース名	メソッド名	-
22	0x1102	A	リモートインタフェース名	メソッド名	-
23	0x2002	A	リモートインタフェース名	メソッド名	-
24	0x2003	A	リモートインタフェース名	メソッド名	-
-	0x3008	B	-	-	内部情報

(凡例) A:標準 B:詳細 -:該当なし

注

図 A-4 ~ 図 A-7 中の番号と対応しています。

## 付録 A.2 リダイレクタのトレース取得ポイント

ここでは、リダイレクタのトレース取得ポイントと、取得できるトレース情報について説明します。

なお、PRF トレース取得レベルを「詳細」にしている場合、リクエスト処理のトレースとセッショントレースが出力されます。

トレース取得ポイントと PRF トレース取得レベル

イベント ID、トレース取得ポイント、および PRF トレース取得レベルについて、次の

表に示します。なお、「0x8003」のポイントで、セッショントレースは出力されません。

表 A-5 リダイレクタでのトレース取得ポイントの詳細

イベント ID	図中の番号	トレース取得ポイント	レベル
0x8000	1	リクエスト処理要求の取得後	A / B
0x8001	2	Web コンテナへのリクエストヘッダ情報の送信	A / B
0x8002	3	Web コンテナからの HTTP リクエストボディ情報の要求受信	B
0x8003	5	Web コンテナからの HTTP レスポンスヘッダ情報の受信	B
0x8004	7	Web コンテナからの HTTP レスポンスボディ情報の送信開始	B
0x8100	10	リダイレクタ処理完了	A / B
0x8101	9	Web コンテナからのレスポンス完了通知の受信	A / B
0x8102	4	Web コンテナへの HTTP リクエストボディ情報の送信完了	B
0x8103	6	Web サーバへの HTTP レスポンスヘッダ情報の設定完了	B
0x8104	8	Web サーバへの HTTP レスポンスボディ情報の送信完了	B

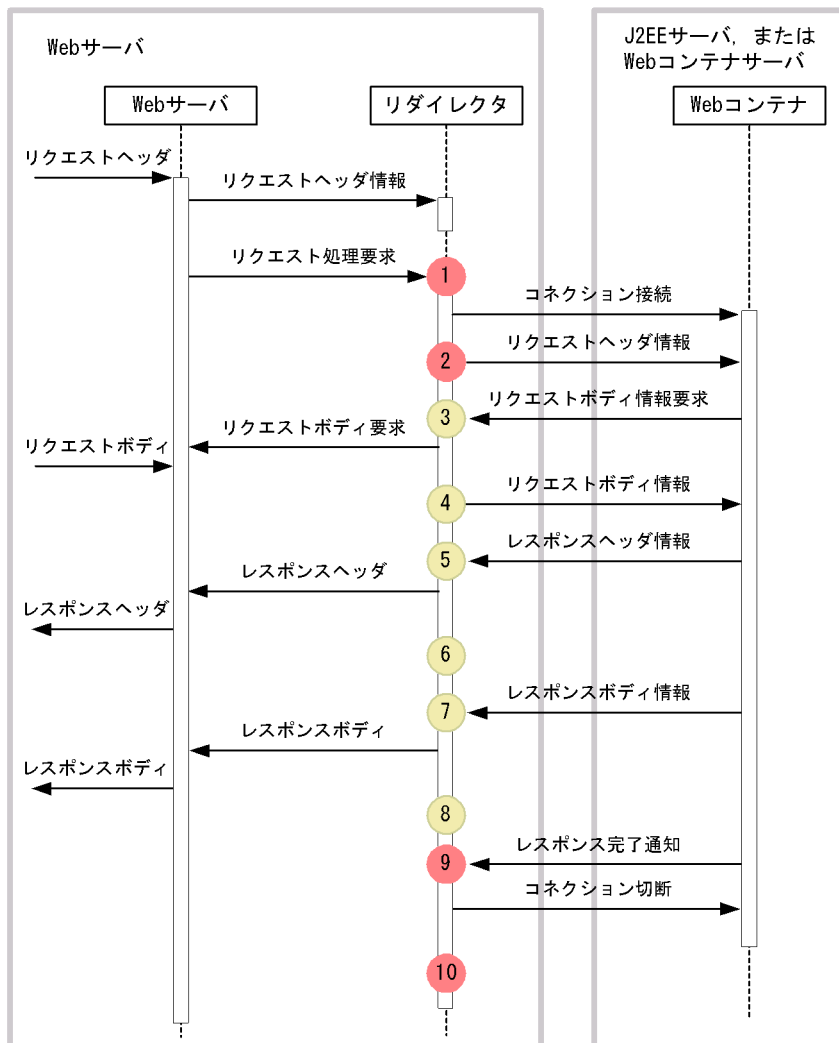
(凡例) B : 詳細 A / B : 標準と詳細で異なる情報を取得

注

図 A-8 中の番号と対応しています。

リダイレクタでのトレース取得ポイントを次の図に示します。

図 A-8 リダイレクタのトレース取得ポイント



- (凡例)
- (Red circle) : トレース取得ポイントを示します。PRF トレース取得レベルは「標準」です。
  - (Yellow circle) : トレース取得ポイントを示します。PRF トレース取得レベルは「詳細」です。

コネクションは、通常、常設コネクションのために使い回されるため、リクエストのために切断されることはありません。コネクションは、通信で例外が発生した場合、または常設コネクションの上限数に達していた場合だけ切断されます。

なお、3 と 4 のポイントでは、Web コンテナからリクエストボディ情報の要求があった場合だけトレース情報が出力されます。また、3、4、7、8 のポイントでは、リクエストボディ情報とレスポンスボディ情報が複数回クライアントに送信されることがあるため、トレース情報も複数回出力されることがあります。

また、次に示すポイントでは、無効なセッション ID を取得する場合や、セッション ID

を取得しない場合があります。

#### 1 ~ 4 のポイント

セッション ID を取得できます。ただし、リクエストヘッダの Cookie または URL からセッション ID を取得するため、無効なセッション ID (J2EE アプリケーションで破棄された HttpSession の ID または有効期限切れで破棄された HttpSession の ID) を取得する場合があります。

また、有効なセッション ID を取得した場合も、J2EE アプリケーションでセッションが破棄されることがあります。

#### 6 ~ 10 のポイント

セッションを生成した場合だけ、セッション ID を取得できます。

#### 取得できるトレース情報

リダイレクタで取得できるトレース情報を次の表に示します。

表 A-6 リダイレクタで取得できるトレース情報

図中の 番号 1	イベント ID	レベ ル	取得できる情報		
			インタフェース名	オペレー ション名	オプション
1	0x8000	A / B	クライアントのアドレス : HTTP メソッド	URI	2
2	0x8001	A / B	コンテナのアドレス : ポート番号	-	2
3	0x8002	B	要求サイズ	-	<セッション ID 文字数 : セッショ ン ID: 取得方法 >
4	0x8102	B	読み込んだサイズ	-	<入り口時刻 >< セッション ID 文 字数 : セッション ID: 取得方法 >
5	0x8003	B	-	-	-
6	0x8103	B	-	-	3
7	0x8004	B	送信サイズ	-	セッション ID
8	0x8104	B	送信できたサイズ	-	<入り口時刻 >< セッション ID 文 字数 : セッション ID >
9	0x8101	A / B	コンテナのアドレス : ポート番号	-	4
10	0x8100	A / B	クライアントのアドレス : HTTP メソッド	URI	5

(凡例) B: 詳細 A / B: 標準と詳細で異なる情報を取得

- : 該当なし

注 1

図 A-8 中の番号と対応しています。

注 2

レベルが「標準」の場合、何も出力されません。

レベルが「詳細」の場合、セッション ID 文字数: セッション ID: 取得方法が出力されます。

注 3

正常に処理された場合、入り口時刻とセッション ID が表示されます。

例外が発生した場合、入り口時刻に加え、4 バイトの保守情報、およびセッション ID 文字数:

セッション ID が表示されます。

注 4

レベルが「標準」の場合、入り口時刻が表示されます。

レベルが「詳細」の場合、入り口時刻に加え、セッション ID 文字数: セッション ID が表示されます。

注 5

レベルが A の場合、常に、入り口時刻とステータスコードが表示されます。

ただし、Microsoft IIS 使用時に例外が発生した場合は、入り口時刻とステータスコードに加え、4 バイトの保守情報が表示されます。

レベルが B の場合、レベルが A の場合の情報に加え、セッション ID 文字数: セッション ID が表示されます。

### 付録 A.3 Web コンテナのトレース取得ポイント (リクエスト処理のトレース)

ここでは、Web コンテナのトレース取得ポイントと、取得できるトレース情報について説明します。なお、Web コンテナでは、リクエスト処理のトレースとセッショントレースが出力されます。ここでは、リクエスト処理のトレースについて説明します。また、インプロセス HTTP サーバを使用したときのトレースポイントと取得できるトレース情報についても説明します。

#### トレース取得ポイントと PRF トレース取得レベル

イベント ID、トレース取得ポイント、および PRF トレース取得レベルについて、次の表に示します。

表 A-7 Web コンテナでのトレース取得ポイントの詳細 (リクエスト処理のトレース)

イベント ID	図中の番号	トレース取得ポイント		レベル
0x8200	1	リクエスト取得・リクエストヘッダ解析完了	Web サーバ経由の場合	A
0x8201			簡易 Web サーバ経由の場合	A
0x8202	3	サーブレット / JSP の呼び出し		A



イベント ID	図中の番号	トレース取得ポイント	レベル
0x8203	2	フィルタ呼び出し	B
0x8206	4	RequestDispatcher 経由のサーブレット / JSP 呼び出し	B
0x8207	3	静的コンテンツの呼び出し	B
0x8300	8	リクエスト処理完了	Web サーバ経由の場合 A
0x8301		簡易 Web サーバ経由の場合 A	
0x8302	6	サーブレット / JSP の処理完了	A
0x8303	7	フィルタ処理完了	B
0x8306	5	RequestDispatcher 経由のサーブレット / JSP 処理完了	B
0x8307	6	静的コンテンツの処理完了	B

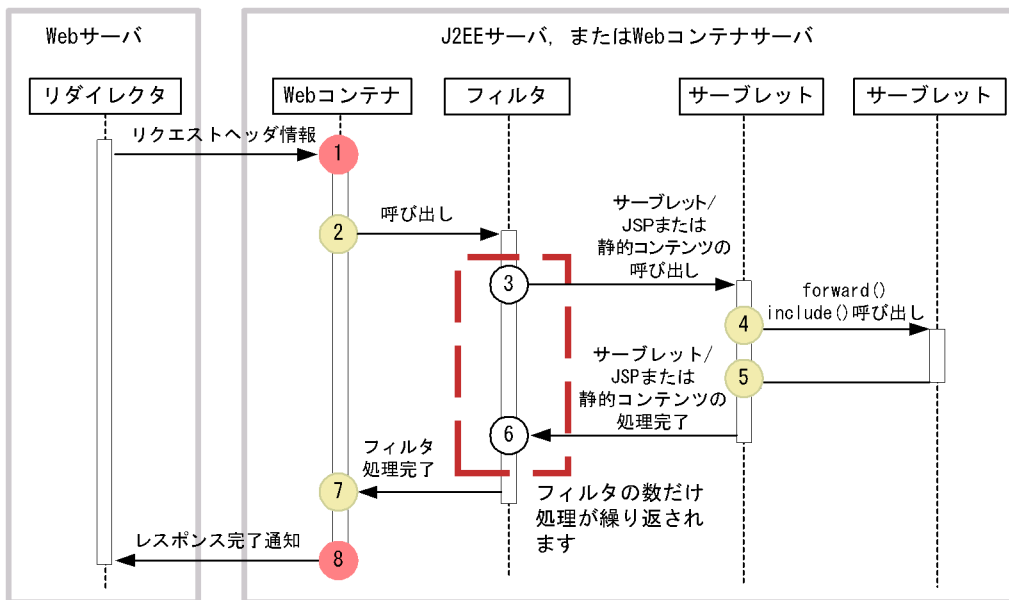
(凡例) A : 標準 B : 詳細

注

図 A-9 中の番号と対応しています。

Web コンテナでのトレース取得ポイントを次の図に示します。

図 A-9 Web コンテナのトレース取得ポイント (リクエスト処理のトレース)



- (凡例)
- : トレース取得ポイントを示します。PRF トレース取得レベルは「標準」です。
  - : トレース取得ポイントを示します。PRF トレース取得レベルは「詳細」です。
  - : トレース取得ポイントを示します。  
 サブレットの呼び出しの場合、PRF トレース取得レベルは「標準」です。  
 静的コンテンツの呼び出しの場合、PRF トレース取得レベルは「詳細」です。

取得できるトレース情報

Web コンテナで取得できるトレース情報を次の表に示します。

表 A-8 Web コンテナで取得できるトレース情報 (リクエスト処理のトレース)

図中の 番号	イベント ID	レベル	取得できる情報		
			インタフェース名	オペレーション名	オプション
1	0x8200	A	HTTP メソッド	URI	-
	0x8201	A	HTTP メソッド	URI	-
2	0x8203	B	クラス名	-	-
3	0x8202	A	クラス名または JSP 名	-	-
	0x8207	B	-	-	-
4	0x8206	B	クラス名	ディスパッチのタイプ	-
5	0x8306	B	クラス名	ディスパッチのタイプ	2

図中の 番号 1	イベント ID	レベル	取得できる情報		
			インタフェース名	オペレーション名	オプション
6	0x8302	A	クラス名または JSP 名	-	2
	0x8307	B	-	-	2
7	0x8303	B	クラス名	-	2
8	0x8300	A	HTTP メソッド	URI	3
	0x8301	A	HTTP メソッド	URI	3

( 凡例 ) A : 標準 B : 詳細 - : 該当なし

注 1

図 A-9 中の番号と対応しています。

注 2

正常に処理された場合、入り口時刻が表示されます。  
例外が発生した場合、入り口時刻と例外名が表示されます。

注 3

正常に処理された場合、入り口時刻およびステータスコードが表示されます。  
なお、Web サーバ経由の場合（イベント ID が 0x8300 の場合）、レスポンス送信時にリダイレクタとの接続が切れていた場合も、入り口時刻ステータスコードが表示されます。  
例外が発生した場合、入り口時刻、ステータスコード、および例外名が表示されます。

参考

SOAP クライアント以外のリクエストを受信した場合、トレース情報のキー情報である「クライアントアプリケーション情報」には、常に 0 が表示されます。SOAP クライアントのリクエストを受信した場合だけ、「クライアントアプリケーション情報」が出力されます。

トレース取得ポイントと PRF トレース取得レベル（インプロセス HTTP サーバを使用した場合）

インプロセス HTTP サーバを使用した場合のイベント ID、トレース取得ポイント、および PRF トレース取得レベルについて、次の表に示します。

表 A-9 インプロセス HTTP サーバでのトレース取得ポイントの詳細

イベント ID	図中の番 号	トレース取得ポイント	レベル
0x8202	5	サーブレット / JSP 呼び出し	A
0x8203	4	フィルタ呼び出し	B
0x8206	6	RequestDispatcher 経由のサーブレット / JSP 呼び出し	B
0x8207	5	静的コンテンツの呼び出し	B
0x8211	3	リクエスト取得時、リクエストヘッダ解析完了時	A

イベント ID	図中の番号	トレース取得ポイント	レベル
0x8212	1	Web クライアントからのデータ読み込み開始時	B
0x8213	8	Web クライアントへのデータ書き込み開始時	B
0x8302	10	サーブレット / JSP 処理完了	A
0x8303	11	フィルタ処理完了	B
0x8306	7	RequestDispatcher 経由のサーブレット / JSP 処理完了	B
0x8307	10	静的コンテンツの処理完了	B
0x8311	12	リクエスト処理完了時	A
0x8312	2	Web クライアントからのデータ読み込み完了時	B
0x8313	9	Web クライアントへのデータ書き込み完了時	B

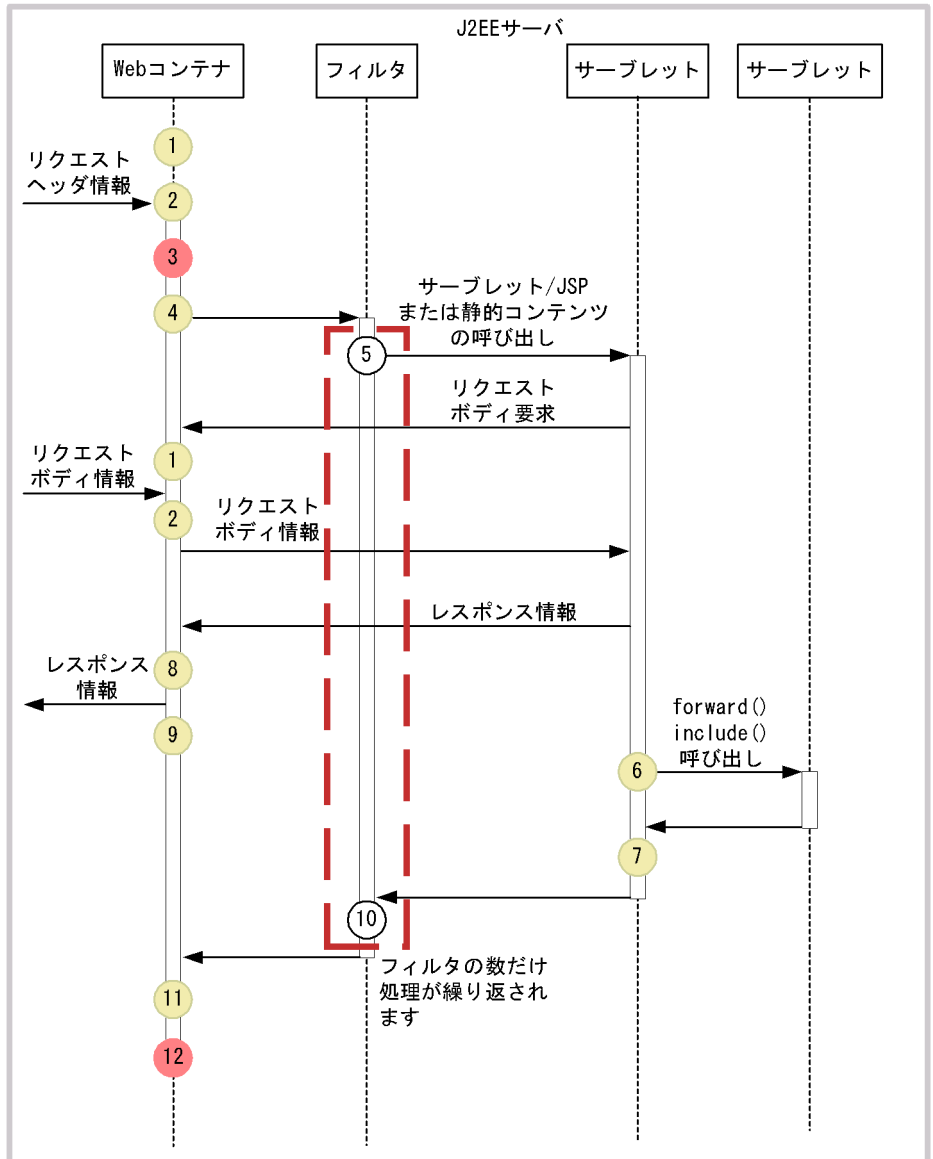
(凡例) A : 標準 B : 詳細

注

図 A-10 中の番号と対応しています。

インプロセス HTTP サーバを使用した場合のトレース取得ポイントを次の図に示します。

図 A-10 インプロセス HTTP サーバのトレース取得ポイント



- (凡例)
- (Red circle) : トレース取得ポイントを示します。PRF トレース取得レベルは「標準」です。
  - (Yellow circle) : トレース取得ポイントを示します。PRF トレース取得レベルは「詳細」です。
  - (White circle) : トレース取得ポイントを示します。  
 サブレットの呼び出しの場合、PRF トレース取得レベルは「標準」です。  
 静的コンテンツの呼び出しの場合、PRF トレース取得レベルは「詳細」です。

なお、1 および 2 のポイントはリクエスト情報をクライアントから複数回受信することがあるため、トレース情報も複数回出力されることがあります。

8 および 9 のポイントはレスポンス情報が複数回クライアントに送信されることがあるため、トレース情報も複数回出力されることがあります。

フィルタからリクエストボディ情報の要求や、レスポンスの送信を行なった場合も、1 および 2 または 8 および 9 のトレースは出力されず。

取得できるトレース情報

インプロセス HTTP サーバで取得できるトレース情報を次の表に示します。

表 A-10 インプロセス HTTP サーバで取得できるトレース情報（リクエスト処理のトレース）

図中の 番号 1	イベント ID	レベ ル	取得できる情報		
			インタフェース名	オペレーション名	オプション
1	0x8212	B	要求サイズ	-	-
2	0x8312	B	読み込んだサイズ	-	2
3	0x8211	A	HTTP メソッド	URI	-
4	0x8203	B	クラス名	コンテキストルート名	<セッション ID 文字数 :セッション ID: グロー バルセッション ID 文 字数: グローバルセッ ション ID>
5	0x8202	A	クラス名または JSP ファイル名	-	-
	0x8207	B	-	-	2
6	0x8206	B	クラス名	ディスパッチのタイプ コンテキストルート名	<セッション ID 文字数 :セッション ID: グロー バルセッション ID 文 字数: グローバルセッ ション ID>
7	0x8306	B	クラス名	ディスパッチのタイプ コンテキストルート名	3
8	0x8213	B	書き込みサイズ	-	-
9	0x8313	B	書き込んだサイズ	-	2
10	0x8302	A	クラス名または JSP ファイル名	-	3
	0x8307	B	-	-	2
11	0x8303	B	クラス名	コンテキストルート名	3
12	0x8311	A	HTTP メソッド	URI	入り口時刻ステータス コード

(凡例) A: 標準 B: 詳細 -: 該当なし

## 注 1

図 A-10 中の番号と対応しています。

## 注 2

正常に処理された場合、入り口時刻が表示されます。

例外が発生した場合、入り口時刻および例外が表示されます。

## 注 3

正常に処理された場合、入り口時刻とセッション ID 文字数: セッション ID が表示されます。

例外が発生した場合、入り口時刻と例外名に加えて、セッション ID 文字数: セッション ID が表示されます。

## 付録 A.4 Web コンテナのトレース取得ポイント (セッショントレース)

ここでは、Web コンテナのトレースのトレース取得ポイントと、取得できるトレース情報について説明します。なお、Web コンテナでは、リクエスト処理のトレースとセッショントレースが出力されます。ここでは、セッショントレース、およびグローバルセッションについてのトレース取得ポイントと取得できるトレース情報について説明します。

### トレース取得ポイントと PRF トレース取得レベル (セッショントレース)

セッショントレースに関連するトレースの、イベント ID、トレース取得ポイント、および PRF トレース取得レベルについて、次の表に示します。なお、「0x8203」、「0x8202」、「0x8207」、「0x8206」および「0x8300」では、グローバルセッションについての情報も出力されます。

表 A-11 Web コンテナでのトレース取得ポイントの詳細 (セッショントレース)

イベント ID	図中の番号 <sup>1</sup>	トレース取得ポイント	レベル <sup>2</sup>
0x8200	1	リクエスト取得・リクエストヘッダ解析完了時 (Web サーバ経由の場合)	A
0x8202	4, 9	サーブレット / JSP 呼び出し	A
0x8203	2, 3	リクエストを受信したサーブレット / JSP の実行前に実行されるフィルタの呼び出し (web.xml の <filter-mapping> タグの <dispatcher> タグを省略した場合、または <dispatcher> タグで "REQUEST" を指定したフィルタを呼び出した場合)	B
0x8206	7	RequestDispatcher 経由のサーブレット / JSP 呼び出し	B
0x8207	4, 9	静的コンテンツ呼び出し (DefaultServlet)	B
0x8208	5	セッション生成後	B
0x8209	6	セッション破棄後	B
0x8210	17	セッションタイムアウト後	B

イベント ID	図中の番号 <sup>1</sup>	トレース取得ポイント	レベル <sup>2</sup>
0x8211	1	リクエスト取得・リクエストヘッダ解析完了 (インプロセス HTTP サーバ経由の場合)	A
0x8214	8	フォワード時に実行されるフィルタの呼び出し ( web.xml の <filter-mapping> タグの <dispatcher> タグで "FORWARD" を指定したフィルタを呼び出した場合 )	B
0x8215	8	インクルード時に実行されるフィルタの呼び出し ( web.xml の <filter-mapping> タグの <dispatcher> タグで "INCLUDE" を指定したフィルタを呼び出した場合 )	B
0x8216	2	エラーページに転送される際に実行されるフィルタの呼び出し ( web.xml の <filter-mapping> タグの <dispatcher> タグで "ERROR" を指定したフィルタを呼び出した場合 )	B
0x8300	16	リクエスト処理完了 ( Web サーバ経由の場合 )	A
0x8302	10, 13	サーブレット / JSP 処理完了	A
0x8303	14, 15	リクエストを受信したサーブレット / JSP の実行前に実行されるフィルタの処理完了 ( web.xml の <filter-mapping> タグの <dispatcher> タグで ""REQUEST"" を指定したフィルタの処理が完了した場合 )	B
0x8306	12	RequestDispatcher 経由のサーブレット / JSP 処理完了	B
0x8307	10, 13	静的コンテンツ処理完了 ( DefaultServlet )	B
0x8311	16	リクエスト処理完了 ( インプロセス HTTP サーバ経由の場合 )	A
0x8314	11	フォワード時に実行されるフィルタの処理完了 ( web.xml の <filter-mapping> タグの <dispatcher> タグで "FORWARD" を指定したフィルタの処理が完了した場合 )	B
0x8315	11	インクルード時に実行されるフィルタの処理完了 ( web.xml の <filter-mapping> タグの <dispatcher> タグで "INCLUDE" を指定したフィルタの処理が完了した場合 )	B
0x8316	15	エラーページに転送される際に実行されるフィルタの処理完了 ( web.xml の <filter-mapping> タグの <dispatcher> タグで "ERROR" を指定したフィルタの処理が完了した場合 )	B

( 凡例 ) A : 標準 B : 詳細

注 1

図 A-11 中の番号と対応しています。

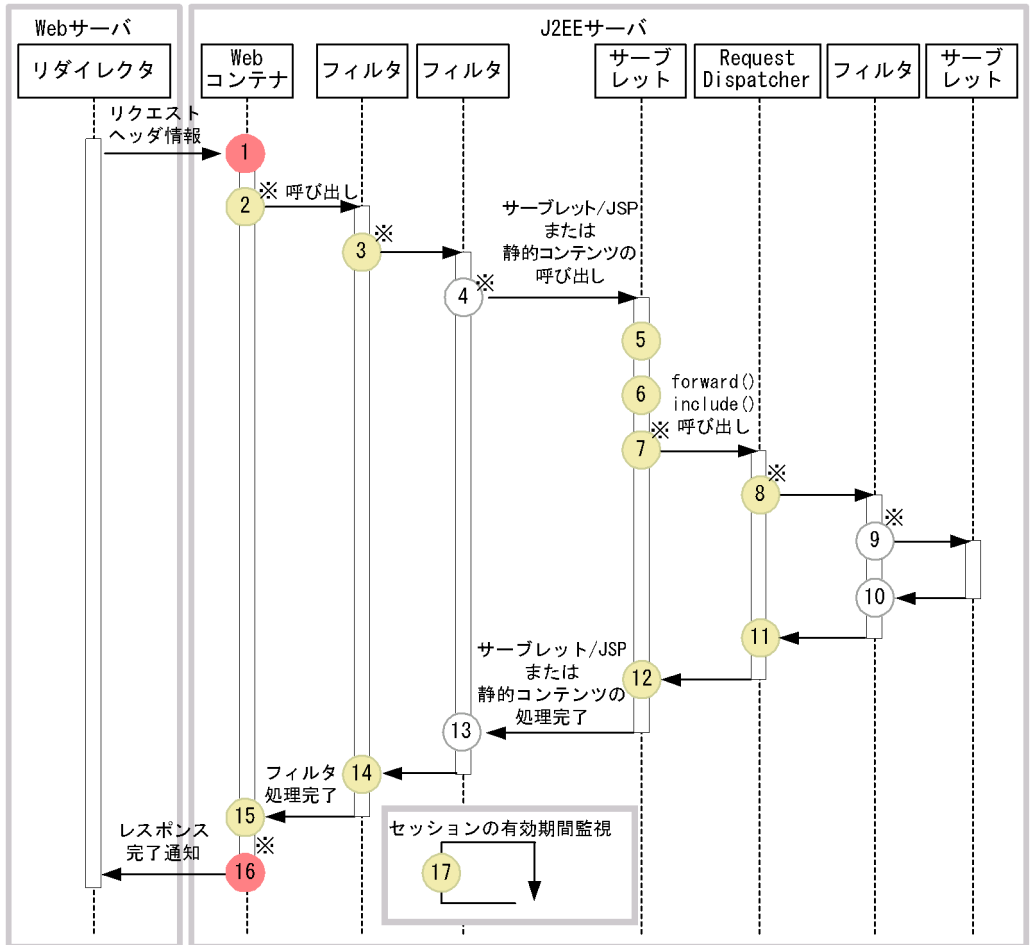
注 2

セッショントレースについての情報はレベルが「詳細」の場合だけ出力されます。

Web コンテナでのセッショントレースのトレース取得ポイントを次の図に示します。



図 A-11 Web コンテナのトレース取得ポイント (セッショントレース)



- (凡例)
- : トレース取得ポイントを示します。PRF トレース取得レベルは「標準」です。
  - : トレース取得ポイントを示します。PRF トレース取得レベルは「詳細」です。
  - : トレース取得ポイントを示します。PRF トレース取得レベルは、出力されるイベント ID によって異なります。

注※ グローバルセッション ID も取得できるトレース取得ポイントを示します。

それぞれのポイントで取得できるセッション ID について説明します。

### 1 のポイント

セッション ID を取得できます。ただし、リクエストヘッダの Cookie または URL からセッション ID を取得するため、無効なセッション ID (J2EE アプリケーションで破棄された HttpSession の ID または有効期限切れで破棄された HttpSession の ID) を取得する場合があります。

また、有効なセッション ID を取得した場合も、J2EE アプリケーションでセッションが破棄される場合があります。

#### 2, 3, 4, 7, 8, 9 のポイント

トレース取得ポイントで有効なセッション ID を取得できます。ただし、J2EE アプリケーションでセッションが破棄される場合があります。

また、これらのポイントでは、グローバルセッション ID も取得できます。取得できるグローバルセッション ID の内容は、トレース取得ポイントごとに異なります。

- 2 のポイントは、一つのリクエストで最初にイベント ID 「0x8203」が出力されるトレース取得ポイントです。このトレース取得ポイントでは、Web クライアントからリクエストとして送信されたグローバルセッション ID が取得できます。ただし、このポイントでは、すでに無効になっているグローバルセッション ID が出力される場合もあります。
- 3, 4, 7, 8, 9 のポイントで出力される、イベント ID が「0x8216」「0x8202」「0x8203」「0x8206」「0x8207」「0x8214」「0x8215」のトレースには、その時点で有効なグローバルセッション ID が取得できます。

#### 5 のポイント

J2EE アプリケーションでセッションが生成された場合だけ、トレース取得ポイントで有効なセッション ID を取得できます。ただし、J2EE アプリケーションでセッションが破棄される場合があります。

#### 6 のポイント

J2EE アプリケーションでセッションが破棄された場合だけ、トレース取得ポイントで無効になったセッション ID を取得できます。ただし、J2EE アプリケーションでセッションが破棄される場合があります。

#### 10, 11, 12, 13 のポイント

トレース取得ポイントで有効なセッション ID を取得できます。ただし、J2EE アプリケーションでセッションが破棄される場合があります。

#### 14, 15 のポイント

トレース取得ポイントで有効なセッション ID を取得できます。なお、このトレース取得ポイントでリクエスト処理が完了すると、以降の J2EE アプリケーションでセッションが破棄されることはありません。

#### 16 のポイント

トレース取得ポイントで有効なセッション ID を取得できます。なお、このトレース取得ポイントでリクエスト処理が完了すると、以降の J2EE アプリケーションでセッションが破棄されることはありません。

また、グローバルセッションを生成した場合、リクエスト処理が終了した時点で有効なグローバルセッション ID を取得できます。

#### 17 のポイント

有効期限を超えたセッションが破棄された場合だけ、無効になったセッション ID を取得できます。

#### 取得できるトレース情報

セッショントレースについて、Web コンテナで取得できるトレース情報を次の表に示します。なお、イベント ID が「0x8202」、「0x8203」、「0x8206」、「0x8207」、「0x8214」、「0x8215」、「0x8300」および「0x8311」のトレース取得ポイントでは、グローバルセッションについての情報も出力されます。

表 A-12 Web コンテナで取得できるトレース情報（セッショントレース）

図中の 番号 1	イベント ID	レベ ル	取得できる情報		
			インタフェース名	オペレーション名	オプション
1	0x8200	A	HTTP メソッド	URI	<セッション ID 文字数: セッション ID: 取得方法 > 2
4, 9	0x8202	A	クラス名 (JSP 呼び出し時は JSP ファイル名)	コンテキストルート名 <sup>2</sup>	<セッション ID 文字数: セッション ID: グローバル セッション ID 文字数: グ ローバルセッション ID> 2
2, 3	0x8203	B	クラス名	コンテキストルート名	<セッション ID 文字数: セッション ID: グローバル セッション ID 文字数: グ ローバルセッション ID>
7	0x8206	B	クラス名	ディスパッチのタイプ コンテキストルート名	<セッション ID 文字数: セッション ID: グローバル セッション ID 文字数: グ ローバルセッション ID>
4, 9	0x8207	B	-	コンテキストルート名	<セッション ID 文字数: セッション ID: グローバル セッション ID 文字数: グ ローバルセッション ID>
5	0x8208	B	コンテキストルート名	セッション有効期間	<セッション ID 文字数: セッション ID>
6	0x8209	B	コンテキストルート名	セッション生成時刻	<セッション ID 文字数: セッション ID>
17	0x8210	B	コンテキストルート名	セッション有効期間: セッション生成時刻	<セッション ID 文字数: セッション ID>
1	0x8211	A	HTTP メソッド	URI	<セッション ID 文字数: セッション ID: 取得方法 > 2

図中の 番号 1	イベント ID	レベ ル	取得できる情報		
			インタフェース名	オペレーション名	オプション
8	0x8214	B	クラス名	コンテキストルート 名	<セッション ID 文字数: セッション ID: グローバル セッション ID 文字数: グ ローバルセッション ID>
8	0x8215	B	クラス名	コンテキストルート 名	<セッション ID 文字数: セッション ID: グローバル セッション ID 文字数: グ ローバルセッション ID>
2	0x8216	B	クラス名	コンテキストルート 名	<セッション ID 文字数: セッション ID: グローバル セッション ID 文字数: グ ローバルセッション ID>
16	0x8300	A	HTTP メソッド	URI	3
10 , 13	0x8302	A	クラス名 (JSP 呼 び出し時は JSP ファイル名)	コンテキストルート 名 <sup>2</sup>	4
14 , 15	0x8303	B	クラス名	コンテキストルート 名	5
12	0x8306	B	クラス名	ディスパッチのタイ プ コンテキストルート 名	5
10 , 13	0x8307	B	-	コンテキストルート 名	5
16	0x8311	A	HTTP メソッド	URI	6
11	0x8314	B	クラス名	コンテキストルート 名	5
11	0x8315	B	クラス名	コンテキストルート 名	5
15	0x8316	B	クラス名	コンテキストルート 名	5

(凡例) A: 標準 B: 詳細 -: 該当なし

注 1

図 A-11 中の番号と対応しています。

注 2

レベルとして「詳細」を指定している場合だけ出力されます。

注 3

正常に処理された場合、またはレスポンス送信時にリダイレクタとの接続が切れていた場合は、指定しているレベルに応じて、次のように出力されます。

「標準」を指定しているときには、入り口時刻、ステータスコードが表示されます。

「詳細」を指定しているときには、入り口時刻、ステータスコード、セッション ID 文字数：  
 セッション ID: グローバルセッション ID 文字数: グローバルセッション ID が表示されます。  
 例外が発生した場合、指定しているレベルに応じて、次のよう出力されます。  
 「標準」を指定しているときには、入り口時刻、ステータスコード、例外名が表示されます。  
 「詳細」を指定しているときには、入り口時刻、ステータスコード、例外名: セッション文字数  
 : セッション ID: グローバルセッション ID 文字数: グローバルセッション ID が表示されます。

## 注 4

正常に処理された場合、指定しているレベルに応じて、次のよう出力されます。  
 「標準」を指定しているときには、入り口時刻が表示されます。  
 「詳細」を指定しているときには、入り口時刻とセッション ID 文字数: セッション ID が表示  
 されます。  
 例外が発生した場合、指定しているレベルに応じて、次のよう出力されます。  
 「標準」を指定しているときには、入り口時刻と例外名が表示されます。  
 「詳細」を指定しているときには、入り口時刻と例外名: セッション ID 文字数: セッション ID  
 が表示されます。

## 注 5

正常に処理された場合、入り口時刻とセッション ID 文字数: セッション ID が表示されます。  
 例外が発生した場合、入り口時刻と例外名: セッション ID 文字数: セッション ID が表示され  
 ます。

## 注 6

指定しているレベルに応じて、次のよう出力されます。  
 「標準」を指定している場合は、入り口時刻、ステータスコードが表示されます。  
 「詳細」を指定している場合は、入り口時刻、ステータスコード、セッション ID 文字数: セッ  
 ション ID: グローバルセッション ID 文字数: グローバルセッション ID が表示されます。

## 付録 A.5 Web コンテナのトレース取得ポイント (フィルタ のトレース)

ここでは、フォワード時、またはインクルード時に呼び出されるフィルタを設定した場合  
 の Web コンテナのトレースのトレース取得ポイントと、取得できるトレース情報につ  
 いて説明します。

なお、フォワード時、またはインクルード時に呼び出されるフィルタを設定した場合の  
 Web コンテナでは、正常に処理が終了した場合と障害が発生した場合で取得できるト  
 レース情報が異なります。ここでは、それぞれの場合について説明します。

なお、JSP の page ディレクティブで `errorPage` 属性を使用しエラーページを設定して、  
 JSP で例外が発生した場合、リクエストの `forward` でエラーページが表示されます。し  
 たがって、JSP でエラーページを表示する場合も、`forward` 時のトレースが出力されま  
 す。

## (1) 正常に処理が終了した場合の Web コンテナのトレース取得ポイント (フィルタのトレース)

## トレース取得ポイントと PRF トレース取得レベル

イベント ID, トレース取得ポイント, および PRF トレース取得レベルについて, 次の表に示します。

表 A-13 正常終了時の Web コンテナでのトレース取得ポイントの詳細 (フィルタのトレース)

イベント ID	図中の番号	トレース取得ポイント	レベル
0x8200	1	リクエスト取得・リクエストヘッダ解析完了時 (Web サーバ経由)	A
0x8201	1	リクエスト取得時・リクエストヘッダ解析完了時 (簡易 Web サーバ経由)	A
0x8211	1	リクエスト取得時・リクエストヘッダ解析完了時 (インプロセス HTTP サーバ経由)	A
0x8203	2	リクエストを受信したサーブレット / JSP の実行前に実行されるフィルタの呼出し (web.xml の <filter-mapping> タグの <dispatcher> タグを省略, または <dispatcher> タグで "REQUEST" を指定したフィルタ)	B
0x8202	3	サーブレット / JSP の呼び出し	A / B
0x8206	4	RequestDispatcher 経由のサーブレット / JSP 呼び出し	B
0x8214	5	フォワード時に実行されるフィルタの呼出し時 (web.xml の <filter-mapping> タグの <dispatcher> タグで "FORWARD" を指定したフィルタ)	B
0x9215	5	インクルード時に実行されるフィルタの呼出し (web.xml の <filter-mapping> タグの <dispatcher> タグで "INCLUDE" を指定したフィルタ)	B
0x8202	6	サーブレット / JSP の呼び出し	A / B
0x8207	6	静的コンテンツ呼び出し (DefaultServlet)	B
0x8302	7	サーブレット / JSP の処理完了	A / B
0x8307	7	静的コンテンツ処理完了 (DefaultServlet)	B
0x8314	8	フォワード時に実行されるフィルタの処理完了時 (web.xml の <filter-mapping> タグの <dispatcher> タグで "FORWARD" を指定したフィルタ)	B
0x8315	8	インクルード時に実行されるフィルタの処理完了時 (web.xml の <filter-mapping> タグの <dispatcher> タグで "INCLUDE" を指定したフィルタ)	B
0x8306	9	RequestDispatcher 経由のサーブレット / JSP 処理完了	B
0x8302	10	サーブレット / JSP の処理完了	A / B

イベント ID	図中の番号	トレース取得ポイント	レベル
0x8303	11	リクエストを受信したサーブレット / JSP の実行前に実行されるフィルタの処理完了時 ( web.xml の <filter-mapping> タグの <dispatcher> タグで "REQUEST" を指定したフィルタ )	B
0x8300	12	リクエスト処理完了 ( Web サーバ経由 )	A
0x8301	12	リクエスト処理完了 ( 簡易 Web サーバ経由 )	A
0x8311	12	リクエスト処理完了 ( インプロセス HTTP サーバ経由 )	A

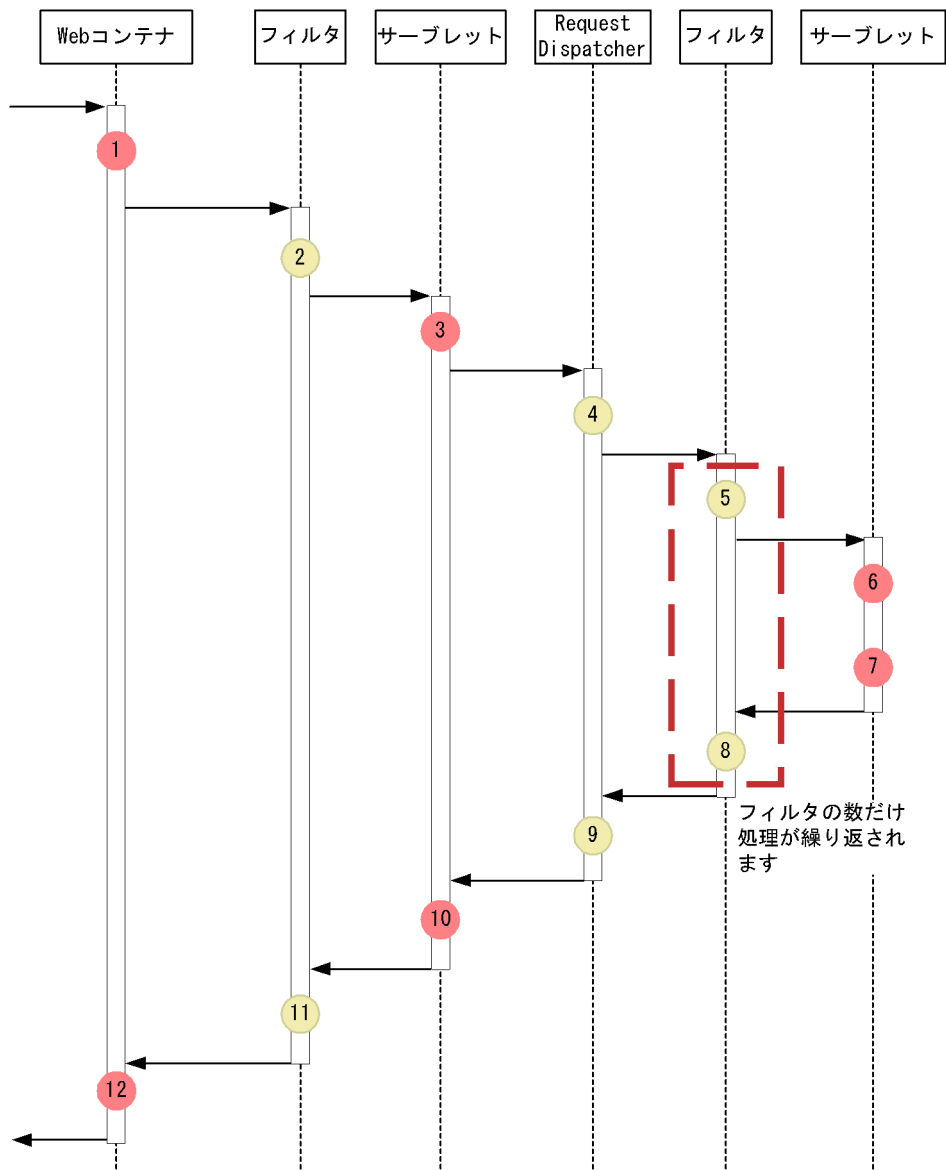
( 凡例 )    A : 標準    B : 詳細    A / B : 標準と詳細で異なる情報を取得

注

図 A-12 中の番号と対応しています。

フォワード時, またはインクルード時に呼び出されるフィルタを設定した場合の Web コンテナでのトレース取得ポイントを次の図に示します。

図 A-12 正常終了時の Web コンテナでのトレース取得ポイント (フィルタのトレース)



(凡例) ● : トレース取得ポイントを示します。PRF トレース取得レベルは「標準」です。

● : トレース取得ポイントを示します。PRF トレース取得レベルは「詳細」です。

#### 取得できるトレース情報

フォワード時, またはインクルード時に呼び出されるフィルタを設定した場合の Web コンテナで取得できるトレース情報を次の表に示します。



表 A-14 正常終了時の Web コンテナで取得できるトレース情報 (フィルタのトレース)

図中の 番号 1	イベント ID	レベ ル	取得できる情報		
			インタフェース名	オペレーション名	オプション
1	0x8200	A	HTTP メソッド	URI	-
	0x8201	A	HTTP メソッド	URI	-
	0x8211	A	HTTP メソッド	URI	-
2	0x8203	B	クラス名	コンテキストルート名	<セッション ID 文字数:セッション ID:グローバルセッション ID 文字数:グローバルセッション ID >
3	0x8202	A / B	クラス名 (JSP 呼び出しの場合は JSP 名)	レベルが A の場合 何も表示されない レベルが B の場合 コンテキストルート名	レベルが A の場合 何も表示されない レベルが B の場合 <セッション ID 文字数:セッション ID:グローバルセッション ID 文字数:グローバルセッション ID >
4	0x8206	B	クラス名	ディスパッチのタイプ コンテキストルート名	<セッション ID 文字数:セッション ID:グローバルセッション ID 文字数:グローバルセッション ID >
4	0x8214	B	クラス名	コンテキストルート名	<セッション ID 文字数:セッション ID:グローバルセッション ID 文字数:グローバルセッション ID >
5	0x9215	B	クラス名	コンテキストルート名	<セッション ID 文字数:セッション ID:グローバルセッション ID 文字数:グローバルセッション ID >
6	0x8202	A / B	クラス名 (JSP 呼び出しの場合は JSP 名)	レベルが A の場合 何も表示されない レベルが B の場合 コンテキストルート名	レベルが A の場合 何も表示されない レベルが B の場合 <セッション ID 文字数:セッション ID:グローバルセッション ID 文字数:グローバルセッション ID >

図中の 番号 1	イベント ID	レベ ル	取得できる情報		
			インタフェース名	オペレーション名	オプション
	0x8207	B	-	コンテキストルート名	<セッション ID 文字 数：セッション ID：グ ローバルセッション ID 文字数：グローバル セッション ID >
7	0x8302	A / B	クラス名（JSP 呼 び出しの場合は JSP 名）	レベルが A の場合 何も表示されない レベルが B の場合 コンテキストルー ト名	2
	0x8307	B	-	コンテキストルート名	2
8	0x8314	B	クラス名	コンテキストルート名	2
	0x8315	B	クラス名	コンテキストルート名	2
9	0x8306	B	クラス名	ディスパッチのタイプ コンテキストルート名	2
10	0x8302	A / B	クラス名（JSP 呼 び出しの場合は JSP 名）	レベルが A の場合 何も表示されない レベルが B の場合 コンテキストルー ト名	2
11	0x8303	B	クラス名	コンテキストルート名	2
12	0x8300	A	HTTP メソッド	URI	入り口時刻
	0x8301	A	HTTP メソッド	URI	入り口時刻
	0x8311	A	HTTP メソッド	URI	入り口時刻

（凡例） A：標準 B：詳細 A / B：標準と詳細で異なる情報を取得 -：該当なし

注 1

図 A-12 中の番号と対応しています。

注 2

正常に処理された場合、入り口時刻、セッション文字数：セッション ID が表示されます。  
例外が発生した場合、入り口時刻と例外名、セッション文字数：セッション ID が表示されま  
す。

## （2）例外が発生した場合の Web コンテナのトレース取得ポイント （フィルタのトレース）

### トレース取得ポイントと PRF トレース取得レベル

イベント ID、トレース取得ポイント、および PRF トレース取得レベルについて、次の  
表に示します。

表 A-15 例外が発生した場合の Web コンテナでのトレース取得ポイントの詳細（フィルタのトレース）

イベント ID	図中の番号	トレース取得ポイント	レベル
0x8200	1	リクエスト取得・リクエストヘッダ解析完了時（Web サーバ経由）	A
0x8201	1	リクエスト取得時・リクエストヘッダ解析完了時（簡易 Web サーバ経由）	A
0x8211	1	リクエスト取得時・リクエストヘッダ解析完了時（インプロセス HTTP サーバ経由）	A
0x8203	2	リクエストを受信したサーブレット / JSP の実行前に実行されるフィルタの呼出し（web.xml の <filter-mapping> タグの <dispatcher> タグを省略、または <dispatcher> タグで "REQUEST" を指定したフィルタ）	B
0x8202	3	サーブレット / JSP の呼び出し	A / B
0x8302	4	サーブレット / JSP の処理完了	A / B
0x8303	5	リクエストを受信したサーブレット / JSP の実行前に実行されるフィルタの処理完了時（web.xml の <filter-mapping> タグの <dispatcher> タグで "REQUEST" を指定したフィルタ）	B
0x8206	6	RequestDispatcher 経由のサーブレット / JSP 呼び出し	B
0x8216	7	エラーページに転送される際に実行されるフィルタの呼出し（web.xml の <filter-mapping> タグの <dispatcher> タグで "ERROR" を指定したフィルタ）	B
0x8202	8	サーブレット / JSP の呼び出し	A / B
0x8207	8	静的コンテンツ呼び出し（DefaultServlet）	B
0x8302	9	サーブレット / JSP の処理完了	A / B
0x8307	9	静的コンテンツ処理完了（DefaultServlet）	B
0x8316	10	エラーページに転送される際に実行されるフィルタの処理完了（web.xml の <filter-mapping> タグの <dispatcher> タグで "ERROR" を指定したフィルタ）	B
0x8306	11	RequestDispatcher 経由のサーブレット / JSP 処理完了	B
0x8300	12	リクエスト処理完了（Web サーバ経由）	A
0x8301	12	リクエスト処理完了（簡易 Web サーバ経由）	A
0x8311	12	リクエスト処理完了（インプロセス HTTP サーバ経由）	A

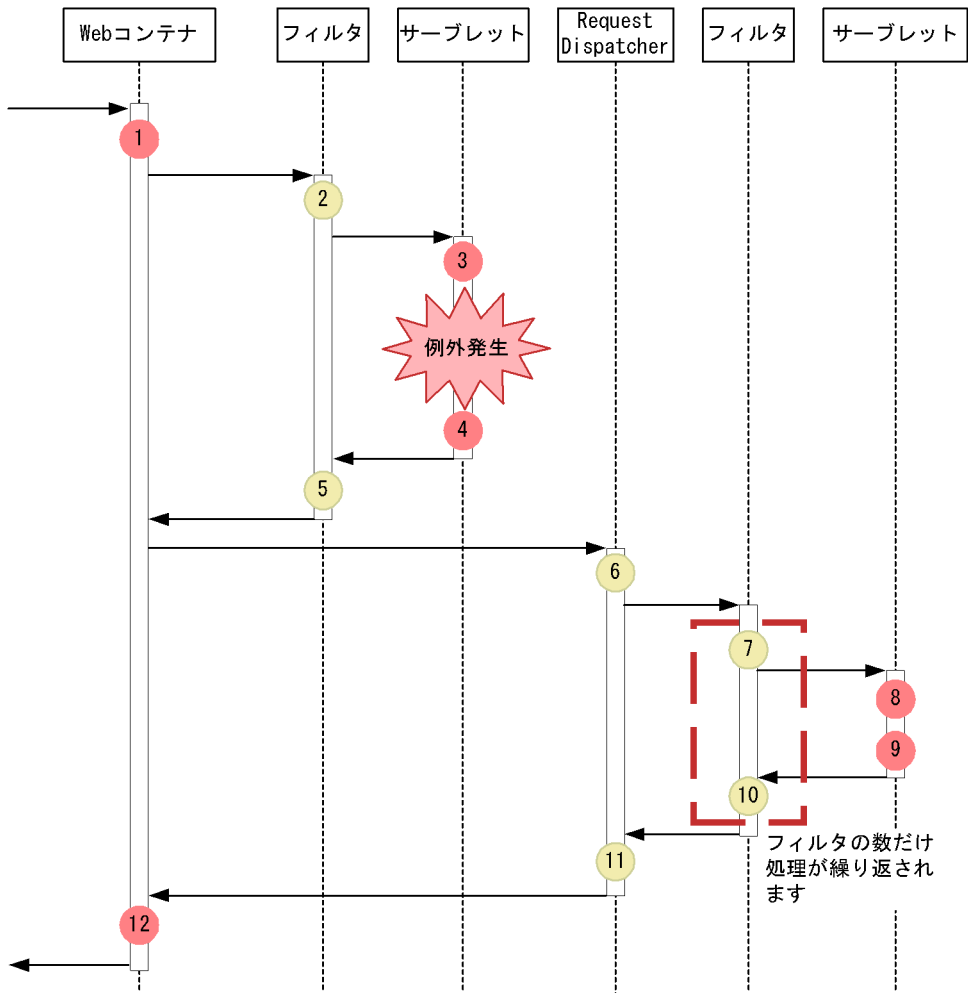
（凡例） A：標準 B：詳細 A / B：標準と詳細で異なる情報を取得

注

図 A-13 中の番号と対応しています。

例外が発生した場合の Web コンテナでのトレース取得ポイントを次の図に示します。

図 A-13 例外が発生した場合の Web コンテナでのトレース取得ポイント



(凡例) ● : トレース取得ポイントを示します。PRF トレース取得レベルは「標準」です。

● : トレース取得ポイントを示します。PRF トレース取得レベルは「詳細」です。

### 取得できるトレース情報

フォワード時，またはインクルード時に呼び出されるフィルタを設定した場合の Web コンテナで取得できるトレース情報を次の表に示します。

表 A-16 例外が発生した場合の Web コンテナで取得できるトレース情報 (フィルタのトレース)

図中の 番号 1	イベント ID	レベル	取得できる情報		
			インタフェース名	オペレーション名	オプション
1	0x8200	A	HTTP メソッド	URI	-
	0x8201	A	HTTP メソッド	URI	-
	0x8211	A	HTTP メソッド	URI	-
2	0x8203	B	クラス名	コンテキストルート名	<セッション ID 文字数:セッション ID:グローバルセッション ID 文字数:グローバルセッション ID >
3	0x8202	A / B	クラス名 (JSP 呼び出しの場合は JSP 名)	レベルが A の場合 何も表示されない レベルが B の場合 コンテキストルート名	レベルが A の場合 何も表示されない レベルが B の場合 <セッション ID 文字数:セッション ID:グローバルセッション ID 文字数:グローバルセッション ID >
4	0x8302	A / B	クラス名 (JSP 呼び出しの場合は JSP 名)	レベルが A の場合 何も表示されない レベルが B の場合 コンテキストルート名	2
5	0x8303	B	クラス名	コンテキストルート名	2
6	0x8206	B	クラス名	ディスパッチのタイプ コンテキストルート名	<セッション ID 文字数:セッション ID:グローバルセッション ID 文字数:グローバルセッション ID >
7	0x8216	B	クラス名	コンテキストルート名	<セッション ID 文字数:セッション ID:グローバルセッション ID 文字数:グローバルセッション ID >
8	0x8202	A / B	クラス名 (JSP 呼び出しの場合は JSP 名)	レベルが A の場合 何も表示されない レベルが B の場合 コンテキストルート名	レベルが A の場合 何も表示されない レベルが B の場合 <セッション ID 文字数:セッション ID:グローバルセッション ID 文字数:グローバルセッション ID >

図中の 番号 1	イベント ID	レベ ル	取得できる情報		
			インタフェース名	オペレーション名	オプション
	0x8207	B	-	コンテキストルート名	<セッション ID 文字数:セッション ID:グローバルセッション ID 文字数:グローバルセッション ID >
9	0x8302	A / B	クラス名 (JSP 呼び出しの場合は JSP 名)	レベルが A の場合 何も表示されない レベルが B の場合 コンテキストルート名	2
	0x8307	B	-	コンテキストルート名	2
10	0x8316	B	クラス名	コンテキストルート名	<セッション ID 文字数:セッション ID:グローバルセッション ID 文字数:グローバルセッション ID >
11	0x8306	B	クラス名	ディスパッチのタイプ コンテキストルート名	2
12	0x8300	A	HTTP メソッド	URI	入り口時刻
	0x8301	A	HTTP メソッド	URI	入り口時刻
	0x8311	A	HTTP メソッド	URI	入り口時刻

(凡例) A:標準 B:詳細 A/B:標準と詳細で異なる情報を取得 -:該当なし

注 1

図 A-13 中の番号と対応しています。

注 2

正常に処理された場合、入り口時刻、セッション文字数:セッション ID が表示されます。  
例外が発生した場合、入り口時刻と例外名、セッション文字数:セッション ID が表示されません。

## 付録 A.6 Web コンテナのトレース取得ポイント (セッションフェイルオーバー機能のトレース)

ここでは、セッションフェイルオーバー機能を使用した場合の Web コンテナのトレース取得ポイントと、取得できるトレース情報について説明します。

トレース取得ポイントと PRF トレース取得レベル

イベント ID、トレース取得ポイント、および PRF トレース取得レベルについて、次の表に示します。

表 A-17 Web コンテナでのトレース取得ポイントの詳細 (セッションフェイルオーバー機能)

イベント ID	図中の番号 <sup>1</sup>	トレース取得ポイント	レベル
0x8203	1	リクエストを受信したサーブレット / JSP の実行前に実行されるフィルタの呼び出し直前 ( web.xml の <filter-mapping> タグの <dispatcher> タグを省略した場合、または <dispatcher> タグで "REQUEST" を指定したフィルタを呼び出した場合 )	B
0x8E01	2	スタブでのリクエスト送信処理の始まり	A
0x8E05	3	サーバ側のリクエスト受信処理	A
0x8405	4	EJB コンテナがリクエストを受信した直後	A
0x8406	5	EJB コンテナがレスポンスを送信する直前	A
0x8E06	6	サーバ側のレスポンス送信処理	A
0x8E02	7	スタブでのレスポンス受信処理の終わり	A
0x8218 <sup>2</sup>	8	セッションフェイルオーバー機能のデシリアライズ開始直前	A
0x8318 <sup>2</sup>	9	セッションフェイルオーバー機能のデシリアライズ終了直後	A
0x8203	10	リクエストを受信したサーブレット / JSP の実行前に実行されるフィルタの呼び出し直前 ( web.xml の <filter-mapping> タグの <dispatcher> タグを省略した場合、または <dispatcher> タグで "REQUEST" を指定したフィルタを呼び出した場合 )	B
0x8202	11	サーブレット / JSP 呼び出し直前	A / B
0x8207	11	静的コンテンツ呼び出し直前 ( DefaultServlet )	B
0x8302	12	サーブレット / JSP 処理完了直後	A
0x8307	12	静的コンテンツ処理完了直後 ( DefaultServlet )	B
0x8303	13	リクエストを受信したサーブレット / JSP の実行前に実行されるフィルタの処理完了直前 ( web.xml の <filter-mapping> タグの <dispatcher> タグで "REQUEST" を指定したフィルタの処理が完了した場合 )	B
0x8217 <sup>2</sup>	14	セッションフェイルオーバー機能のシリアライズ開始直前	A
0x8317 <sup>2</sup>	15	セッションフェイルオーバー機能のシリアライズ終了直後	A
0x8E01	16	スタブでのリクエスト送信処理の始まり	A
0x8E05	17	サーバ側のリクエスト受信処理	A
0x8405	18	EJB コンテナがリクエストを受信した直後	A

イベント ID	図中の 番号 <sup>1</sup>	トレース取得ポイント	レベル
0x8406	19	EJB コンテナがレスポンスを送信する直前	A
0x8E06	20	サーバ側のレスポンス送信処理	A
0x8E02	21	スタブでのレスポンス受信処理の終わり	A
0x8203	22	リクエストを受信したサーブレット / JSP の実行前に実行されるフィルタの呼び出し直前 ( web.xml の <filter-mapping> タグの <dispatcher> タグを省略した場合、または <dispatcher> タグで "REQUEST" を指定したフィルタを呼び出した場合 )	B

( 凡例 )    A : 標準    B : 詳細    A / B : 標準と詳細で異なる情報を取得    - : 該当なし

注 1

図 A-14 中の番号と対応しています。

注 2

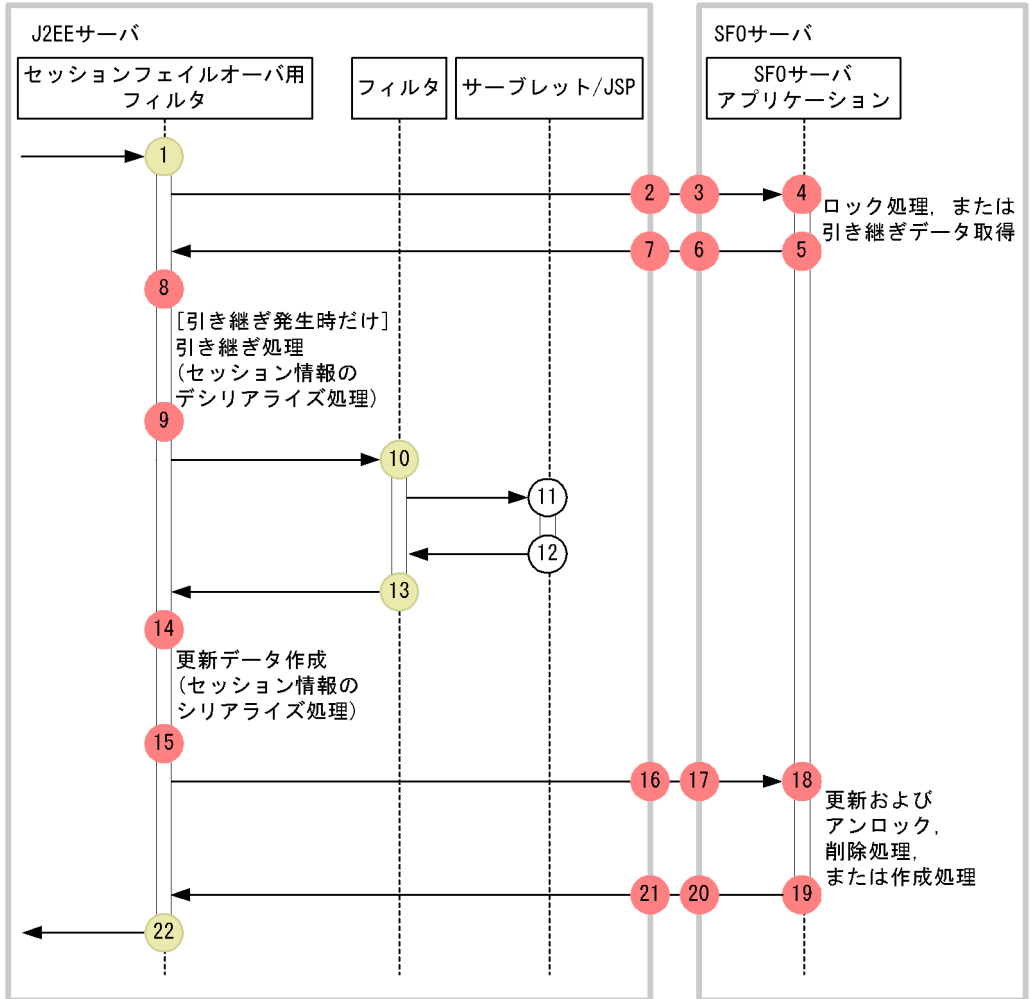
次の条件をすべて満たす場合に出力されます。

- ・ 直列化可能クラスのオブジェクトの引き継ぎが有効である。
- ・ 直列化対象のセッション情報が存在する。

セッションフェイルオーバー機能でのトレース取得ポイントを次の図に示します。



図 A-14 Web コンテナでのトレース取得ポイント（セッションフェイルオーバー機能）



- (凡例)
- (Red) : トレース取得ポイントを示します。PRFトレース取得レベルは「標準」です。
  - (Yellow) : トレース取得ポイントを示します。PRFトレース取得レベルは「詳細」です。
  - (White) : トレース取得ポイントを示します。  
 サブレットの呼び出しの場合、PRFトレース取得レベルは「標準」です。  
 静的コンテンツの呼び出しの場合、PRFトレース取得レベルは「詳細」です。

取得できるトレース情報

セッションフェイルオーバー機能で取得できるトレース情報を次の表に示します。

表 A-18 Web コンテナで取得できるトレース情報 (セッションフェイルオーバー機能)

図中の 番号 <sup>1</sup>	イベ ント ID	レベ ル	取得できる情報		
			インタフェース名	オペレーション名	オプション
1	0x8203	B	クラス名	コンテキストルート名	<セッション ID 文字数:セッション ID:グローバルセッション ID 文字数:グローバルセッション ID>
2	0x8E01	A	インタフェース名	オペレーション名	-
3	0x8E05	A	-	-	-
4	0x8405	A	Bean 名	メソッド名, 引数の数	-
5	0x8406	A	Bean 名	メソッド名, 引数の数	3
6	0x8E06	A	-	-	-
7	0x8E02	A	インタフェース名 <sup>2</sup>	オペレーション名 <sup>2</sup>	5
8	0x8218 <sup>2</sup>	A	リクエスト URL	デシリアライズ前のグローバルセッション情報のサイズ (バイト)	グローバルセッション ID 文字数:グローバルセッション ID
9	0x8318 <sup>2</sup>	A	リクエスト URL	デシリアライズ前のグローバルセッション情報のサイズ (バイト)	4
10	0x8203	B	クラス名	コンテキストルート名	<セッション ID 文字数:セッション ID:グローバルセッション ID 文字数:グローバルセッション ID>
11	0x8202	A / B	クラス名 (JSP 呼び出しの場合は JSP 名)	レベルが A の場合 何も表示されない レベルが B の場合 コンテキストルート名	レベルが A の場合 何も表示されない レベルが B の場合 <セッション ID 文字数:セッション ID:グローバルセッション ID 文字数:グローバルセッション ID >
11	0x8207	B	クラス名	コンテキストルート名	<セッション ID 文字数:セッション ID:グローバルセッション ID 文字数:グローバルセッション ID>
12	0x8302	A	クラス名または JSP 名	-	3
12	0x8307	B	-	コンテキストルート名	3

図中の 番号 <sup>1</sup>	イベント ID	レベル	取得できる情報		
			インタフェース名	オペレーション名	オプション
13	0x8303	B	クラス名	コンテキストルート名	3
14	0x8217 <sup>2</sup>	A	リクエスト URL	なし	<セッション ID 文字数:セッション ID:グローバルセッション ID 文字数:グローバルセッション ID>
15	0x8317 <sup>2</sup>	A	リクエスト URL	シリアライズ後のグローバルセッション情報のサイズ(バイト) <sup>6</sup>	4
16	0x8E01	A	インタフェース名	オペレーション名	-
17	0x8E05	A	-	-	-
18	0x8405	A	Bean 名	メソッド名, 引数の数	-
19	0x8406	A	Bean 名	メソッド名, 引数の数	4
20	0x8E06	A	-	-	-
21	0x8E02	A	インタフェース名 <sup>2</sup>	オペレーション名 <sup>2</sup>	5
22	0x8303	B	クラス名	コンテキストルート名	3

(凡例) A:標準 B:詳細 A/B:標準と詳細で異なる情報を取得 -:該当なし

注 1

図 A-14 中の番号と対応しています。

注 2

次の条件をすべて満たす場合に出力されます。

- ・直列化可能クラスのオブジェクトの引き継ぎが有効である。
- ・直列化対象のセッション情報が存在する。

注 3

正常に処理された場合, 入り口時刻, セッション文字数:セッション ID が表示されます。

例外が発生した場合, 入り口時刻と例外名, セッション文字数:セッション ID が表示されません。

注 4

正常に処理された場合, 入り口時刻が表示されます。

例外が発生した場合, 入り口時刻および例外が表示されます。

注 5

例外が発生した場合, インタフェース名およびオペレーション名は表示されません。また, オプションには発生した例外が表示されます。

注 6

例外が発生した場合, 0 が表示されます。

## 付録 A.7 EJB コンテナのトレース取得ポイント

ここでは、EJB コンテナのトレース取得ポイントと、取得できるトレース情報について、Session Bean、Entity Bean の場合、Message-driven Bean の場合、Timer Service を使用した場合、およびメソッドキャンセルが発生した場合に分けて説明します。

### (1) Session Bean、Entity Bean の場合

Session Bean、Entity Bean のトレース取得ポイントと、取得できるトレース情報について説明します。

#### トレース取得ポイントと PRF トレース取得レベル

イベント ID、トレース取得ポイント、および PRF トレース取得レベルについて、次の表に示します。

表 A-19 Session Bean、Entity Bean でのトレース取得ポイントの詳細

イベント ID	図中の番号	トレース取得ポイント		レベル
0x8401	1	リモートホームインタフェースの場合	EJB コンテナがリクエストを受信した直後	A
0x8402	4		EJB コンテナがレスポンスを送信する直前	A
0x8403	1	ローカルホームインタフェースの場合	EJB コンテナがリクエストを受信した直後	A
0x8404	4		EJB コンテナがレスポンスを送信する直前	A
0x8405	1	リモートコンポーネントインタフェースの場合	EJB コンテナがリクエストを受信した直後	A
0x8406	4		EJB コンテナがレスポンスを送信する直前	A
0x8407	1	ローカルコンポーネントインタフェースの場合	EJB コンテナがリクエストを受信した直後	A
0x8408	4		EJB コンテナがレスポンスを送信する直前	A
0x8409	2	EJB コンテナが EJB のビジネスメソッドをコールバックする直前		B
0x840A	3	EJB のビジネスメソッドのコールバックからリターンした直後		B
0x8453	2	EJB コンテナが EJB のホームメソッドをコールバックする直前		B
0x8454	3	EJB のホームメソッドのコールバックからリターンした直後		B

イベント ID	図中の番号	トレース取得ポイント		レベル
0x8470	1	リモートビジネス インタフェースの 場合	EJB コンテナがリクエストを受信した 直後	A
0x8471	4		EJB コンテナがレスポンスを送信する 直前	A
0x8472	1	ローカルビジネス インタフェースの 場合	EJB コンテナがリクエストを受信した 直後	A
0x8473	4		EJB コンテナがレスポンスを送信する 直前	A
0x8474	1	リモートビジネス インタフェースの 場合	EJB コンテナが Stateful Session Bean のインスタンス作成リクエストを受信 した直後	A
0x8475	1		EJB コンテナが Stateful Session Bean のインスタンス作成リクエストのレス ポンスを送信する直前	A
0x8476	1	ローカルビジネス インタフェースの 場合	EJB コンテナが Stateful Session Bean のインスタンス作成リクエストを受信 した直後	A
0x8477	1		EJB コンテナが Stateful Session Bean のインスタンス作成リクエストのレス ポンスを送信する直前	A

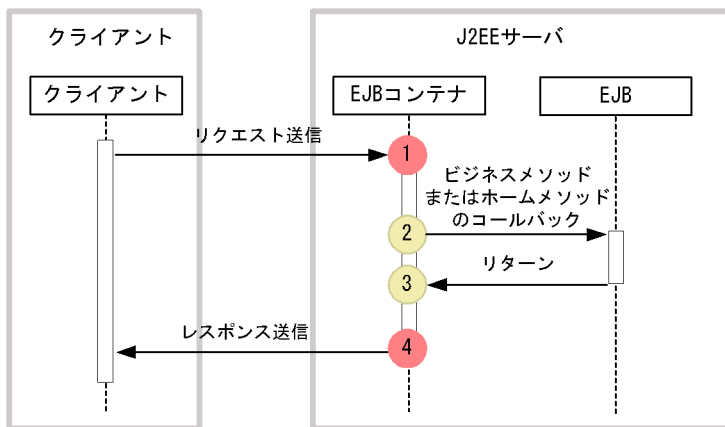
(凡例) A : 標準 B : 詳細

注

図 A-15 中の番号と対応しています。

Session Bean , Entity Bean でのトレース取得ポイントを次の図に示します。

図 A-15 Session Bean , Entity Bean のトレース取得ポイント



- (凡例)
- (Red) : トレース取得ポイントを示します。PRF トレース取得レベルは「標準」です。
  - (Yellow) : トレース取得ポイントを示します。PRF トレース取得レベルは「詳細」です。

取得できるトレース情報

Session Bean , Entity Bean で取得できるトレース情報を次の表に示します。

表 A-20 Session Bean , Entity Bean で取得できるトレース情報

図中の番号	イベント ID	レベル	取得できる情報		
			インタフェース名	オペレーション名	オプション
1	0x8401	A	Bean 名	メソッド名, 引数の数	-
	0x8403	A	Bean 名	メソッド名, 引数の数	-
	0x8405	A	Bean 名	メソッド名, 引数の数	-
	0x8407	A	Bean 名	メソッド名, 引数の数	-
	0x8470	A	Bean 名	メソッド名, 引数の数	-
	0x8472	A	Bean 名	メソッド名, 引数の数	-
	0x8474	A	Bean 名	-	-
	0x8475	A	Bean 名	-	2
	0x8476	A	Bean 名	-	-
2	0x8409	B	Bean 名	メソッド名, 引数の数	-
	0x8453	B	Bean 名	メソッド名, 引数の数	-
3	0x840A	B	Bean 名	メソッド名, 引数の数	2
	0x8454	B	Bean 名	メソッド名, 引数の数	2

図中の番号 1	イベント ID	レベル	取得できる情報		
			インタフェース名	オペレーション名	オプション
4	0x8402	A	Bean 名	メソッド名, 引数の数	2
	0x8404	A	Bean 名	メソッド名, 引数の数	2
	0x8406	A	Bean 名	メソッド名, 引数の数	2
	0x8408	A	Bean 名	メソッド名, 引数の数	2
	0x8471	A	Bean 名	メソッド名, 引数の数	2
	0x8473	A	Bean 名	メソッド名, 引数の数	2

(凡例) A: 標準 B: 詳細 - : 該当なし

注 1

図 A-15 中の番号と対応しています。

注 2

正常に処理された場合, 入り口時刻が表示されます。  
例外が発生した場合, 入り口時刻および例外が表示されます。

## (2) Message-driven Bean (EJB2.0) の場合

Message-driven Bean (EJB2.0) のトレース取得ポイントと, 取得できるトレース情報について説明します。

### トレース取得ポイントと PRF トレース取得レベル

イベント ID, トレース取得ポイント, および PRF トレース取得レベルについて, 次の表に示します。

表 A-21 Message-driven Bean (EJB2.0) でのトレース取得ポイントの詳細

イベント ID	図中の番号	トレース取得ポイント	レベル
0x8425	1	EJB コンテナでのメッセージ処理の始まり	A
0x8426	4	EJB コンテナでのメッセージ処理の終わり	A
0x8427	2	EJB コンテナが Message-driven Bean の onMessage メソッドをコールバックする直前	B
0x8428	3	Message-driven Bean の onMessage メソッドのコールバックからリターンした直後	B

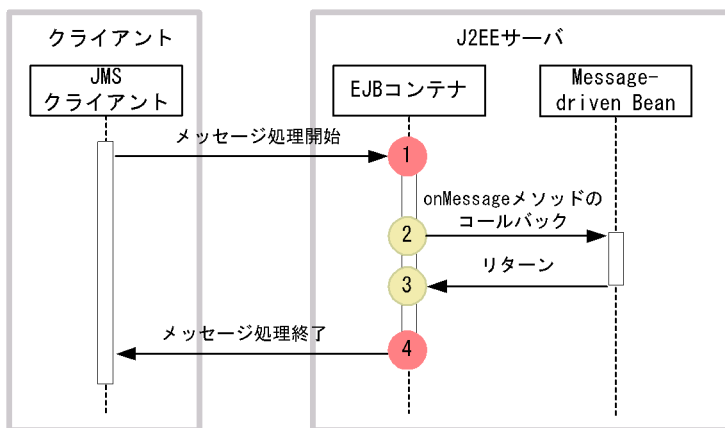
(凡例) A: 標準 B: 詳細

注

図 A-16 中の番号と対応しています。

Message-driven Bean (EJB2.0) でのトレース取得ポイントを次の図に示します。

図 A-16 Message-driven Bean (EJB2.0) のトレース取得ポイント



- (凡例)
- (赤) : トレース取得ポイントを示します。PRF トレース取得レベルは「標準」です。
  - (黄) : トレース取得ポイントを示します。PRF トレース取得レベルは「詳細」です。

### 取得できるトレース情報

Message-driven Bean (EJB2.0) で取得できるトレース情報を次の表に示します。

表 A-22 Message-driven Bean (EJB2.0) で取得できるトレース情報

図中の 番号 1	イベント ID	レベ ル	取得できる情報		
			インタフェース名	オペレーション名	オプション
1	0x8425	A	Bean 名	-	-
2	0x8427	B	-	-	-
3	0x8428	B	-	-	2
4	0x8426	A	Bean 名	-	2

- (凡例) A: 標準 B: 詳細 -: 該当なし

注 1

図 A-16 中の番号と対応しています。

注 2

正常に処理された場合、入り口時刻が表示されます。

例外が発生した場合、入り口時刻および例外が表示されず。

### (3) Message-driven Bean (EJB2.1) の場合

Message-driven Bean (EJB2.1) のトレース取得ポイントと、取得できるトレース情報について説明します。



## トレース取得ポイントと PRF トレース取得レベル

イベント ID, トレース取得ポイント, および PRF トレース取得レベルについて, 次の表に示します。

表 A-23 Message-driven Bean (EJB2.1) でのトレース取得ポイントの詳細

イベント ID	図中の番号	トレース取得ポイント	レベル
0x842D	1	Message-driven Bean の beforeDelivery() メソッドの呼び出し直後	A
0x842E	2	Message-driven Bean の beforeDelivery() メソッドのリターン直前	A
0x842F	3	リソースアダプタから Message-driven Bean のメッセージリスナのメソッドを呼び出した直後	A
0x8431	4	EJB コンテナが Message-driven Bean のメッセージリスナのメソッドをコールバックする直前	A
0x8432	5	Message-driven Bean のメッセージリスナのメソッドのコールバックからリターンした直後	A
0x8430	6	リソースアダプタから呼び出した Message-driven Bean のメッセージリスナのメソッドのリターン直前	A
0x8433	7	Message-driven Bean の afterDelivery() メソッドの呼び出し直後	A
0x8434	8	Message-driven Bean の afterDelivery() メソッドのリターン直前	A

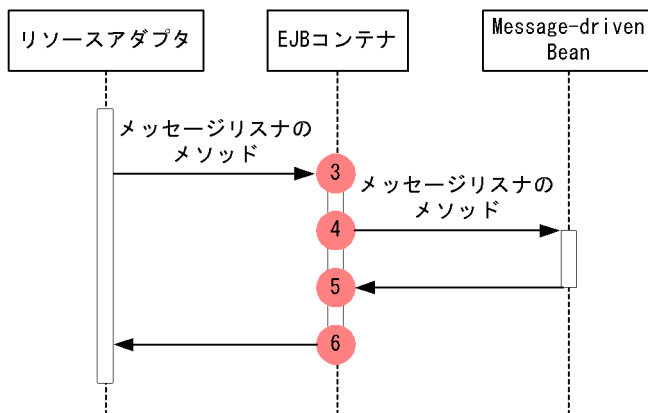
(凡例) A: 標準

注

図 A-17, および図 A-18 中の番号と対応しています。

Message-driven Bean (EJB2.1) でのトレース取得ポイントを次の図に示します。

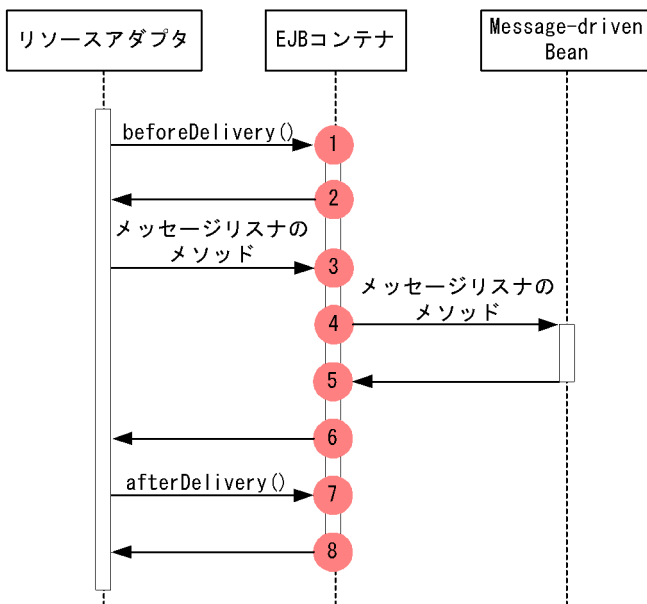
図 A-17 Message-driven Bean (EJB2.1) のトレース取得ポイント (optionA の場合)



(凡例) ● : トレース取得ポイントを示します。PRFトレース取得レベルは「標準」です。

注 Connector 1.5 仕様に記述されるメッセージ配送オプションを示します。

図 A-18 Message-driven Bean (EJB2.1) のトレース取得ポイント (optionB の場合)



(凡例) ● : トレース取得ポイントを示します。PRFトレース取得レベルは「標準」です。

注 Connector 1.5 仕様に記述されるメッセージ配送オプションを示します。

取得できるトレース情報

Message-driven Bean (EJB2.1) で取得できるトレース情報を次の表に示します。

表 A-24 Message-driven Bean (EJB2.1) で取得できるトレース情報

図中の 番号 1	イベント ID	レベ ル	取得できる情報		
			インタフェース名	オペレーション名	オプション
1	0x842D	A	Bean 名	-	-
2	0x842E	A	Bean 名	-	2
3	0x842F	A	Bean 名	メソッド名	-
4	0x8431	A	Bean 名	メソッド名	-
5	0x8432	A	Bean 名	メソッド名	2
6	0x8430	A	Bean 名	メソッド名	2
7	0x8433	A	Bean 名	-	-
8	0x8434	A	Bean 名	-	2

(凡例) A:標準 - :該当なし

注 1

図 A-17, および図 A-18 中の番号と対応しています。

注 2

正常に処理された場合, 入り口時刻が表示されます。

例外が発生した場合, 入り口時刻および例外が表示されます。

#### (4) Timer Service の場合

Timer Service のトレース取得ポイントと, 取得できるトレース情報について説明します。

##### (a) createTimer の場合

トレース取得ポイントと PRF トレース取得レベル

イベント ID, トレース取得ポイント, および PRF トレース取得レベルについて, 次の表に示します。

表 A-25 Timer Service でのトレース取得ポイントの詳細 (createTimer の場合)

イベント ID	図中の番 号	トレース取得ポイント	レベル
0x8460	1	TimerService.createTimer(Date initialExpiration, long intervalDuration, Serializable info) 処理開始	A
0x8462	1	TimerService.createTimer(Date expiration, Serializable info) 処理開始	A
0x8464	1	TimerService.createTimer(long initialDuration, long intervalDuration, Serializable info) 処理開始	A

イベント ID	図中の番号	トレース取得ポイント	レベル
0x8466	1	TimerService.createTimer(long duration, Serializable info) 処理開始	A
0x8461	2	TimerService.createTimer(Date initialExpiration, long intervalDuration, Serializable info) 処理終了	A
0x8463	2	TimerService.createTimer(Date expiration, Serializable info) 処理終了	A
0x8465	2	TimerService.createTimer(long initialDuration, long intervalDuration, Serializable info) 処理終了	A
0x8467	2	TimerService.createTimer(long duration, Serializable info) 処理終了	A

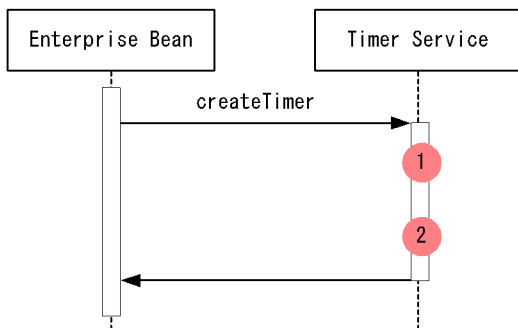
(凡例) A : 標準

注

図 A-19 中の番号と対応しています。

createTimer の場合の Timer Service でのトレース取得ポイントを次の図に示します。

図 A-19 Timer Service のトレース取得ポイント (createTimer の場合)



(凡例) ● : トレース取得ポイントを示します。PRF トレース取得レベルは「標準」です。

### 取得できるトレース情報

createTimer の場合の Timer Service で取得できるトレース情報を次の表に示します。

表 A-26 Timer Service で取得できるトレース情報 (createTimer の場合)

図中の番号	イベント ID	レベル	取得できる情報		
			インタフェース名	オペレーション名	オプション
1	0x8460	A	コールバック対象 Bean のクラス名	-	引数情報
1	0x8462	A	コールバック対象 Bean のクラス名	-	引数情報

図中の 番号 1	イベント ID	レベ ル	取得できる情報		
			インタフェース名	オペレーション名	オプション
1	0x8464	A	コールバック対象 Bean のク ラス名	-	引数情報
1	0x8466	A	コールバック対象 Bean のク ラス名	-	引数情報
2	0x8461	A	コールバック対象 Bean のク ラス名	-	2
2	0x8463	A	コールバック対象 Bean のク ラス名	-	2
2	0x8465	A	コールバック対象 Bean のク ラス名	-	2
2	0x8467	A	コールバック対象 Bean のク ラス名	-	2

(凡例) A: 標準 - : 該当なし

注 1

図 A-19 中の番号と対応しています。

注 2

正常に処理された場合、入り口時刻が表示されます。

例外が発生した場合、「入り口時刻：例外名：メッセージ」の形式で情報が表示されます。

(b) cancel の場合

トレース取得ポイントと PRF トレース取得レベル

イベント ID、トレース取得ポイント、および PRF トレース取得レベルについて、次の表に示します。

表 A-27 Timer Service でのトレース取得ポイントの詳細 (cancel の場合)

イベント ID	図中の番 号	トレース取得ポイント	レベル
0x8468	1	javax.ejb.Timer.cancel() 処理開始	A
0x8469	2	javax.ejb.Timer.cancel() 処理終了	A

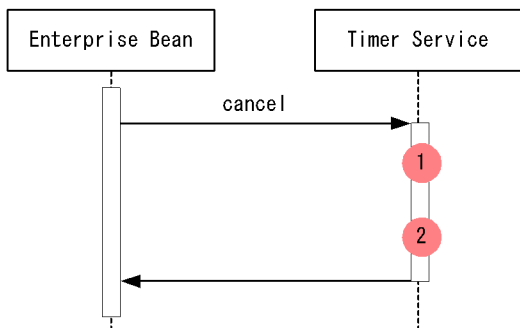
(凡例) A: 標準

注

図 A-20 中の番号と対応しています。

cancel の場合の Timer Service でのトレース取得ポイントを次の図に示します。

図 A-20 Timer Service のトレース取得ポイント (cancel の場合)



(凡例) ● : トレース取得ポイントを示します。PRF トレース取得レベルは「標準」です。

取得できるトレース情報

cancel の場合の Timer Service で取得できるトレース情報を次の表に示します。

表 A-28 Timer Service で取得できるトレース情報 (cancel の場合)

図中の番号	イベント ID	レベル	取得できる情報		
			インタフェース名	オペレーション名	オプション
1	0x8468	A	コールバック対象 Bean のクラス名	-	-
2	0x8469	A	コールバック対象 Bean のクラス名	-	2

(凡例) A : 標準 - : 該当なし

注 1  
図 A-20 中の番号と対応しています。

注 2  
正常に処理された場合、入り口時刻が表示されます。  
例外が発生した場合、「入り口時刻: 例外名: メッセージ」の形式で情報が表示されます。

(c) コールバックの場合

トレース取得ポイントと PRF トレース取得レベル

イベント ID, トレース取得ポイント, および PRF トレース取得レベルについて, 次の表に示します。

表 A-29 Timer Service でのトレース取得ポイントの詳細 (コールバックの場合)

イベント ID	図中の番号	トレース取得ポイント	レベル
0x846C	1	スレッドがコールバック処理開始	B

イベント ID	図中の番号	トレース取得ポイント	レベル
0x846A	2	Enterprise Bean のタイムアウトコールバックメソッドをコールバック開始	A
0x846B	3	Enterprise Bean のタイムアウトコールバックメソッドをコールバック終了	A
0x846D	4	スレッドがコールバック処理終了	B

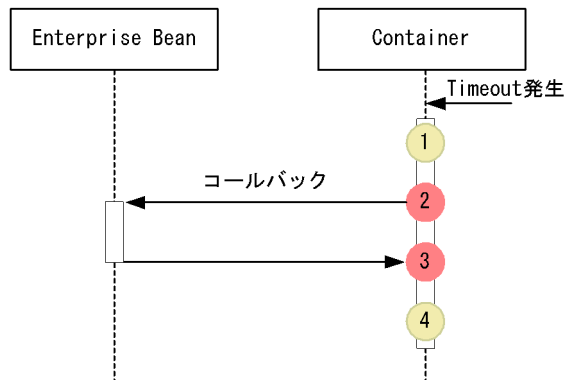
(凡例) A : 標準 B : 詳細

注

図 A-21 中の番号と対応しています。

コールバックの場合の Timer Service でのトレース取得ポイントを次の図に示します。

図 A-21 Timer Service のトレース取得ポイント (コールバックの場合)



(凡例) ● : トレース取得ポイントを示します。PRF トレース取得レベルは「標準」です。

● : トレース取得ポイントを示します。PRF トレース取得レベルは「詳細」です。

### 取得できるトレース情報

コールバックの場合の Timer Service で取得できるトレース情報を次の表に示します。

なお、ルートアプリケーション情報には、タイムアウトメソッドのコールバック時に取得する情報が出力されます。また、クライアントアプリケーション情報には「0」が出力されます。

表 A-30 Timer Service で取得できるトレース情報 (コールバックの場合)

図中の番号	イベント ID	レベル	取得できる情報		
			インタフェース名	オペレーション名	オプション
1	0x846C	B	コールバック対象 Bean のクラス名	-	-

図中の番号 1	イベント ID	レベル	取得できる情報		
			インタフェース名	オペレーション名	オプション
2	0x846A	A	コールバック対象 Bean のクラス名	-	-
3	0x846B	B	コールバック対象 Bean のクラス名	-	2
4	0x846D	A	コールバック対象 Bean のクラス名	-	2

(凡例) A:標準 B:詳細 -:該当なし

注 1

図 A-21 中の番号と対応しています。

注 2

正常に処理された場合、入り口時刻が表示されます。

例外が発生した場合、「入り口時刻:例外名:メッセージ」の形式で情報が表示されます。

### (5) メソッドキャンセルが発生した場合

メソッドキャンセルが発生した場合のトレース取得ポイントと、取得できるトレース情報について説明します。

トレース取得ポイントと PRF トレース取得レベル

イベント ID、トレース取得ポイント、および PRF トレース取得レベルについて、次の表に示します。

表 A-31 メソッドキャンセルが発生した場合のトレース取得ポイントの詳細

イベント ID	図中の番号	トレース取得ポイント	レベル
0x8490	1	メソッドキャンセル処理開始	A
0x8C41	2	障害調査用 SQL 出力	A
0x8491	3	メソッドキャンセル処理完了	A

(凡例) A:標準

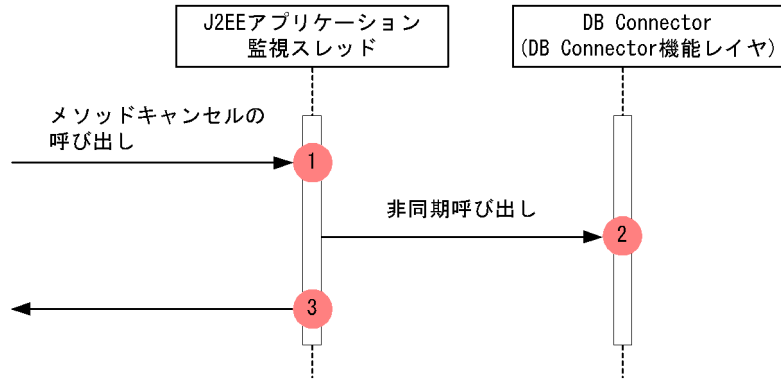
注

図 A-22 中の番号と対応しています。

メソッドキャンセルが発生した場合のトレース取得ポイントを次の図に示します。



図 A-22 メソッドキャンセルが発生した場合のトレース取得ポイント



(凡例) ● : トレース取得ポイントを示します。PRFトレース取得レベルは「標準」です。

取得できるトレース情報

メソッドキャンセルが発生した場合に取得できるトレース情報を次の表に示します。

表 A-32 メソッドキャンセルが発生した場合に取得できるトレース情報

図中の番号	イベント ID	レベル	取得できる情報		
			インタフェース名	オペレーション名	オプション
1	0x8490	A	メソッドキャンセル対象のアプリケーションのルートアプリケーション情報	-	-
2	0x8C41	A	トランザクションタイムアウト, J2EE アプリケーション強制停止, またはメソッドキャンセルされたコネクションのルートアプリケーション情報	-	SQL 文
3	0x8491	A	-	-	入り口時刻

(凡例) A: 標準 - : 該当なし

注

図 A-22 中の番号と対応しています。

## 付録 A.8 JNDI のトレース取得ポイント

ここでは, JNDI のトレース取得ポイントと, 取得できるトレース情報について説明します。

## トレース取得ポイントと PRF トレース取得レベル

イベント ID、トレース取得ポイント、および PRF トレース取得レベルについて、次の表に示します。

表 A-33 JNDI でのトレース取得ポイントの詳細

イベント ID	図中の番号	トレース取得ポイント		レベル		
0x8603	1	JNDI 内部の名前空間、または CORBA ネーミングサービスの検索時	javax.naming.Context.lookup	呼び出し直後	A	
0x8604	2			リターン直前	A	
0x8605	1		javax.naming.Context.list	呼び出し直後	B	
0x8606	2			リターン直前	B	
0x8607	1		javax.naming.Context.listBindings	呼び出し直後	B	
0x8608	2			リターン直前	B	
0x8609	1		javax.ejb.EJBLocalHome が保存されている名前空間の検索時	javax.naming.Context.lookup	呼び出し直後	A
0x860A	2				リターン直前	A
0x860B	1	javax.naming.Context.list		呼び出し直後	B	
0x860C	2			リターン直前	B	
0x860D	1	javax.naming.Context.listBindings		呼び出し直後	B	
0x860E	2			リターン直前	B	
0x860F	1	java: 名前空間検索時		javax.naming.Context.lookup	呼び出し直後	A
0x8610	2		リターン直前		A	
0x8611	1		javax.naming.Context.list	呼び出し直後	B	
0x8612	2			リターン直前	B	
0x8613	1		javax.naming.Context.listBindings	呼び出し直後	B	
0x8614	2			リターン直前	B	
0x8615	1		ラウンドロビン検索機能利用時	javax.naming.Context.lookup	呼び出し直後	A
0x8616	2	リターン直前			A	
0x8617	1	CORBA ネーミングサービス切り替え機能利用時	javax.naming.Context.lookup	呼び出し直後	A	
0x8618	2			リターン直前	A	
0x8619	1		javax.naming.Context.list	呼び出し直後	B	
0x861A	2			リターン直前	B	
0x861B	1		javax.naming.Context.listBindings	呼び出し直後	B	
0x861C	2			リターン直前	B	

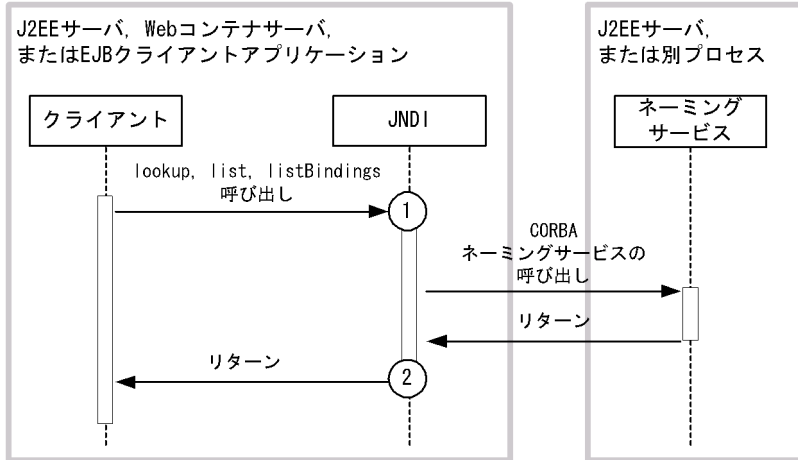
(凡例) A: 標準 B: 詳細

注

図 A-23 中の番号と対応しています。

JNDI でのトレース取得ポイントを次の図に示します。

図 A-23 JNDI のトレース取得ポイント



- (凡例) ○ : 性能解析トレース取得ポイントを示します。  
 javax.naming.Context.lookupメソッドの場合、取得レベルは「標準」です。  
 javax.naming.Context.listメソッドまたは  
 javax.naming.Context.list.Bindingsメソッドの場合、取得レベルは「詳細」です。

### 取得できるトレース情報

JNDI で取得できるトレース情報を次の表に示します。

表 A-34 JNDI で取得できるトレース情報

図中の 番号 1	イベント ID	レベ ル	取得できる情報		
			インタフェース名	オペレーション名	オプション
1	0x8603	A	-	指定された名前	-
	0x8609	A			
	0x860F	A			
	0x8615	A			
	0x8617	A			
	0x8605	B			
	0x860B	B			
	0x8611	B			
	0x8619	B			
	0x8607	B			
	0x860D	B			
	0x8613	B			
	0x861B	B			
2	0x8604	A	-		2
	0x860A	A			
	0x8610	A			
	0x8616	A			
	0x8618	A			
	0x8606	B			
	0x860C	B			
	0x8612	B			
	0x861A	B			
	0x8608	B			
	0x860E	B			
	0x8614	B			
	0x861C	B			

(凡例) A:標準 B:詳細 -:該当なし

注 1

図 A-23 中の番号と対応しています。

注 2

正常に処理された場合、入り口時刻が表示されます。

例外が発生した場合、「入り口時刻：例外名：メッセージ」の形式で情報が表示されます。

## 参考

JNDI でのトレース情報では、次に示す場合、キー情報である「ルートアプリケーション情報」および「クライアントアプリケーション情報」には、0 が表示されません。

- クライアントからの lookup メソッドを呼び出す場合
- サーバが開始処理中または停止処理中の場合

## 付録 A.9 JTA のトレース取得ポイント

ここでは、JTA のトレース取得ポイントと、取得できるトレース情報について、CMT の場合と、BMT の場合に分けて説明します。

### (1) CMT の場合

CMT の場合のトレース取得ポイントと、取得できるトレース情報について説明します。

#### トレース取得ポイントと PRF トレース取得レベル

イベント ID、トレース取得ポイント、および PRF トレース取得レベルについて、次の表に示します。

表 A-35 CMT でのトレース取得ポイントの詳細

イベント ID	図中の番号	トレース取得ポイント	レベル
0x8811	1	トランザクション開始処理直前	A
0x8812	2	トランザクション開始処理直後	A
0x8815	3	トランザクションコミット処理直前	A
0x8816	4	トランザクションコミット処理直後	A
0x8817	3	トランザクションロールバック処理直前	A
0x8818	4	トランザクションロールバック処理直後	A

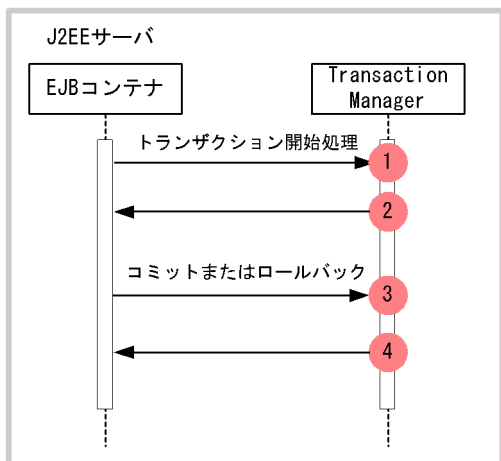
(凡例) A: 標準

注

図 A-24 中の番号と対応しています。

CMT のトレース取得ポイントを、次の図に示します。

図 A-24 CMT のトレース取得ポイント



(凡例) ● : トレース取得ポイントを示します。PRF トレース取得レベルは「標準」です。

取得できるトレース情報

CMT で取得できるトレース情報を次の表に示します。

表 A-36 CMT で取得できるトレース情報

図中の 番号 1	イベント ID	レベル	取得できる情報		
			インタフェース名	オペレーション名	オプション
1	0x8811	A	-	-	-
2	0x8812	A	-	-	2
3	0x8815	A	-	-	-
	0x8817	A	-	-	-
4	0x8816	A	-	-	2
	0x8818	A	-	-	2

(凡例) A : 標準 - : 該当なし

注 1  
図 A-24 中の番号と対応しています。

注 2  
正常に処理された場合、入り口時刻が表示されます。  
例外が発生した場合、入り口時刻および例外が表示されます。

(2) BMT の場合

BMT の場合のトレース取得ポイントと、取得できるトレース情報について説明します。

## トレース取得ポイントと PRF トレース取得レベル

イベント ID, トレース取得ポイント, および PRF トレース取得レベルについて, 次の表に示します。

表 A-37 BMT でのトレース取得ポイントの詳細

イベント ID	図中の番号	トレース取得ポイント	レベル
0x8813	1	トランザクション開始処理直前	A
0x8814	2	トランザクション開始処理直後	A
0x8815	3	トランザクションコミット処理直前	A
0x8816	4	トランザクションコミット処理直後	A
0x8817	3	トランザクションロールバック処理直前	A
0x8818	4	トランザクションロールバック処理直後	A

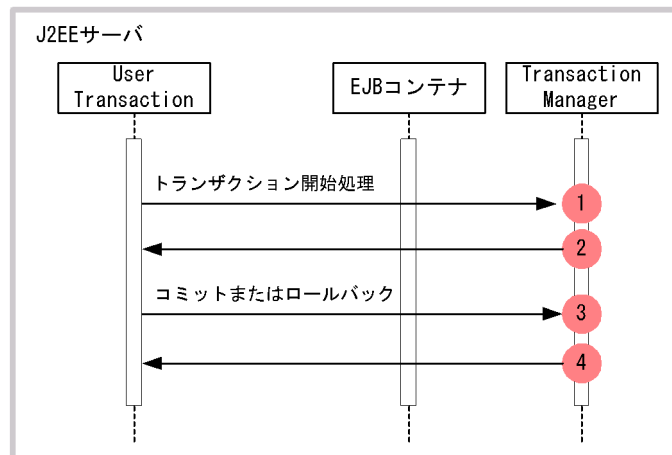
(凡例) A: 標準

注

図 A-25 中の番号と対応しています。

BMT のトレース取得ポイントを, 次の図に示します。

図 A-25 BMT のトレース取得ポイント



(凡例) ● : トレース取得ポイントを示します。PRF トレース取得レベルは「標準」です。

## 取得できるトレース情報

BMT で取得できるトレース情報を次の表に示します。

表 A-38 BMT で取得できるトレース情報

図中の 番号 1	イベント ID	レベ ル	取得できる情報		
			インタフェース名	オペレーション名	オプション
1	0x8813	A	-	-	-
2	0x8814	A	-	-	2
3	0x8815	A	-	-	-
	0x8817	A	-	-	-
4	0x8816	A	-	-	2
	0x8818	A	-	-	2

(凡例) A : 標準 - : 該当なし

注 1

図 A-25 中の番号と対応しています。

注 2

正常に処理された場合、入り口時刻が表示されます。

例外が発生した場合、入り口時刻および例外が表示されます。

### (3) トランザクションタイムアウトの場合

トランザクションタイムアウトの場合のトレース取得ポイントと、取得できるトレース情報について説明します。

トレース取得ポイントと PRF トレース取得レベル

イベント ID、トレース取得ポイント、および PRF トレース取得レベルについて、次の表に示します。

表 A-39 トランザクションタイムアウトでのトレース取得ポイントの詳細

イベント ID	図中の番 号	トレース取得ポイント	レベル
0x8819	1	トランザクションタイムアウト処理直前	A
0x8C41	2	障害調査用 SQL 出力	A
0x8820	3	トランザクションタイムアウト処理直後	A

(凡例) A : 標準

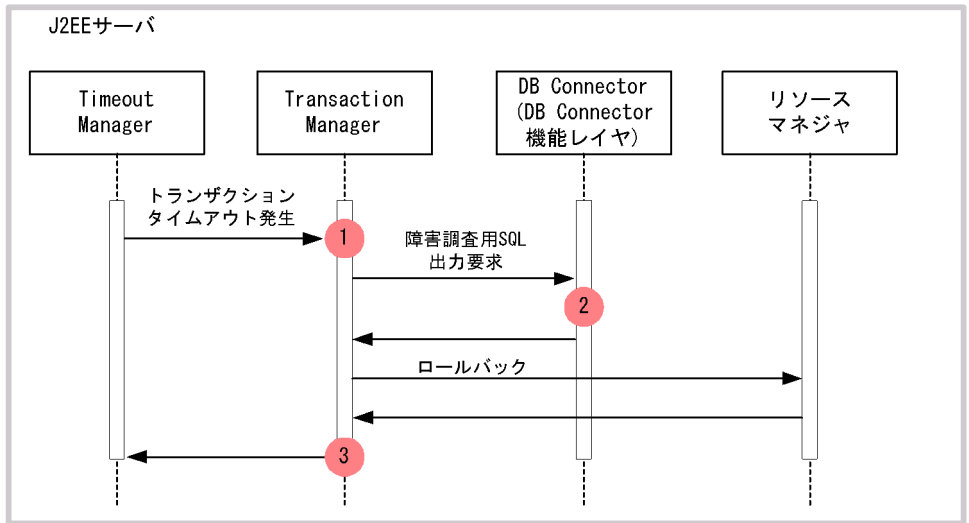
注

図 A-26 中の番号と対応しています。

トランザクションタイムアウトのトレース取得ポイントを、次の図に示します。



図 A-26 トランザクションタイムアウトのトレース取得ポイント



(凡例) ● : トレース取得ポイントを示します。PRF トレース取得レベルは「標準」です。

取得できるトレース情報

トランザクションタイムアウトで取得できるトレース情報を次の表に示します。

表 A-40 トランザクションタイムアウトで取得できるトレース情報

図中の 番号	イベント ID	レベ ル	取得できる情報		
			インタフェース名	オペレーション名	オプション
1	0x8819	A	タイムアウトしたトランザクションのルートアプリケーション情報	-	-
2	0x8C41	A	トランザクションタイムアウト, J2EE アプリケーション強制停止, またはメソッドキャンセルされたコネクションのルートアプリケーション情報	-	SQL 文
3	0x8820	A	-	-	入り口時刻

(凡例) A : 標準 - : 該当なし

注

図 A-26 中の番号と対応しています。

## 付録 A.10 DB Connector, JCA コンテナのトレース取得ポイント

ここでは、DB Connector および JCA コンテナのトレース取得ポイントと、取得できるトレース情報について説明します。

コネクション関連のトレース取得ポイントには、ローカルトランザクションを使用する場合だけ取得できるものがあります。ここでは、トランザクションサポートレベルに関係なく取得できるトレース取得ポイントと、ローカルトランザクションを使用する場合だけ取得できるトレース取得ポイントの二つに分けて説明します。なお、ここでは、トランザクションサポートレベルに関係なく取得できるトレース取得ポイントを、「コネクション関連のトレース取得ポイント」と呼びます。

### (1) コネクション関連のトレース取得ポイントと取得できるトレース情報

コネクション関連のトレース取得ポイントと取得できるトレース情報について説明します。

#### トレース取得ポイントと PRF トレース取得レベル

イベント ID, トレース取得ポイント, および PRF トレース取得レベルについて, イベント ID ごとに, 次の四つの表に示します。

- 0x8B00, 0x8B01, 0x8B80 ~ 0x8B83, 0x8C00 ~ 0x8C03, 0x8C10 ~ 0xC13, 0x8C20 ~ 0x8C29, 0x8C2A ~ 0x8C3F の場合 (コネクション関連の処理実行時)  
参照先: 表 A-41
- 0x8C80 ~ 0x8C93 の場合 (java.sql.Statement インタフェースのメソッド実行時)  
参照先: 表 A-42
- 0x8CC0 ~ 0x8CD9 の場合 (java.sql.PreparedStatement インタフェースのメソッド実行時)  
参照先: 表 A-43
- 0x8D00 ~ 0x8D19 の場合 (java.sql.CallableStatement インタフェースのメソッド実行時)  
参照先: 表 A-44

---

#### ポイント

イベント ID 「0x8C41」については、「付録 A.7 EJB コンテナのトレース取得ポイント」, および「付録 A.9 JTA のトレース取得ポイント」を参照してください。

---

表 A-41 DB Connector, JCA コンテナでのトレース取得ポイントの詳細 (コネクション関連の処理実行時) 1

イベント ID	図中の番号	トレース取得ポイント	レベル	
0x8B00	2	リソースアダプタからのコネクション取得要求呼び出し直後	B	
0x8B01	5	リソースアダプタからのコネクション取得要求リターン直前	B	
0x8B80	3	物理コネクション作成の呼び出し直前	B	
0x8B81	4	物理コネクション作成のリターン直後	B	
0x8B82	14	物理コネクション破棄の呼び出し直前	B	
0x8B83	15	物理コネクション破棄のリターン直後	B	
0x8C00	1	javax.sql.DataSource.getConnection() での, データベース接続の確立	処理開始	A
0x8C01	6		処理終了	A
0x8C02	1	javax.sql.DataSource.getConnection(String username, String password) での, データベース接続の確立	処理開始	A
0x8C03	6		処理終了	A
0x8C10	1	コネクションプールのクラスタ化機能使用時の javax.sql.DataSource.getConnection() での, データベース接続の確立	処理開始	A
0x8C11	6		処理終了	A
0x8C12	1	コネクションプールのクラスタ化機能使用時の javax.sql.DataSource.getConnection(String username, String password) での, データベース接続の確立	処理開始	A
0x8C13	6		処理終了	A
0x8C20	13	java.sql.Connection.close() での, Connection オブジェクトのデータベースと JDBC リソース の解除	処理開始	A
0x8C21	16		処理終了	A
0x8C22	11	java.sql.Connection.commit()	処理開始	B
0x8C23	12		処理終了	B
0x8C24	11	java.sql.Connection.rollback()	処理開始	B
0x8C25	12		処理終了	B
0x8C26	11	java.sql.Connection.rollback(Savepoint savepoint)	処理開始	B
0x8C27	12		処理終了	B
0x8C28	7	java.sql.Connection.createStatement()	処理開始	B
0x8C29	8		処理終了	B
0x8C2A	7	java.sql.Connection.createStatement(int resultSetType, int resultSetConcurrency)	処理開始	B
0x8C2B	8		処理終了	B
0x8C2C	7	java.sql.Connection.createStatement(int resultSetType, int resultSetConcurrency, int resultSetHoldability)	処理開始	B
0x8C2D	8		処理終了	B
0x8C2E	7	java.sql.Connection.prepareCall(String sql)	処理開始	B
0x8C2F	8		処理終了	B

イベント ID	図中の番号	トレース取得ポイント	レベル	
0x8C30	7	java.sql.Connection.prepareCall(String sql, int resultSetType, int resultSetConcurrency)	処理開始	B
0x8C31	8		処理終了	B
0x8C32	7	java.sql.Connection.prepareCall(String sql, int resultSetType, int resultSetConcurrency, int resultSetHoldability)	処理開始	B
0x8C33	8		処理終了	B
0x8C34	7	java.sql.Connection.prepareStatement(String sql)	処理開始	B
0x8C35	8		処理終了	B
0x8C36	7	java.sql.Connection.prepareStatement(String sql, int autoGeneratedKeys)	処理開始	B
0x8C37	8		処理終了	B
0x8C38	7	java.sql.Connection.prepareStatement(String sql, int[] columnIndexes)	処理開始	B
0x8C39	8		処理終了	B
0x8C3A	7	java.sql.Connection.prepareStatement(String sql, int resultSetType, int resultSetConcurrency)	処理開始	B
0x8C3B	8		処理終了	B
0x8C3C	7	java.sql.Connection.prepareStatement(String sql, int resultSetType, int resultSetConcurrency, int resultSetHoldability)	処理開始	B
0x8C3D	8		処理終了	B
0x8C3E	7	java.sql.Connection.prepareStatement(String sql, String[] columnNames)	処理開始	B
0x8C3F	8		処理終了	B

(凡例) A: 標準 B: 詳細

注

図 A-27 中の番号と対応しています。

表 A-42 DB Connector, JCA コンテナでのトレース取得ポイントの詳細  
(java.sql.Statement インタフェースのメソッド実行時) 2

イベント ID	図中の番号	トレース取得ポイント	レベル	
0x8C80	9	execute(String sql)	処理開始	B
0x8C81	10		処理終了	B
0x8C82	9	execute(String sql, int autoGeneratedKeys)	処理開始	B
0x8C83	10		処理終了	B
0x8C84	9	execute(String sql, int[] columnIndexes)	処理開始	B
0x8C85	10		処理終了	B
0x8C86	9	execute(String sql, String[] columnNames)	処理開始	B
0x8C87	10		処理終了	B
0x8C88	9	executeBatch()	処理開始	B
0x8C89	10		処理終了	B

イベント ID	図中の番号	トレース取得ポイント		レベル
0x8C8A	9	executeQuery(String sql)	処理開始	B
0x8C8B	10		処理終了	B
0x8C8C	9	executeUpdate(String sql)	処理開始	B
0x8C8D	10		処理終了	B
0x8C8E	9	executeUpdate(String sql, int autoGeneratedKeys)	処理開始	B
0x8C8F	10		処理終了	B
0x8C90	9	executeUpdate(String sql, int[] columnIndexes)	処理開始	B
0x8C91	10		処理終了	B
0x8C92	9	executeUpdate(String sql, String[] columnNames)	処理開始	B
0x8C93	10		処理終了	B

(凡例) B: 詳細

注

図 A-27 中の番号と対応しています。

表 A-43 DB Connector, JCA コンテナでのトレース取得ポイントの詳細  
(java.sql.PreparedStatement インタフェースのメソッド実行時) 3

イベント ID	図中の番号	トレース取得ポイント		レベル
0x8CC0	9	execute()	処理開始	B
0x8CC1	10		処理終了	B
0x8CC2	9	execute(String sql)	処理開始	B
0x8CC3	10		処理終了	B
0x8CC4	9	execute(String sql, int autoGeneratedKeys)	処理開始	B
0x8CC5	10		処理終了	B
0x8CC6	9	execute(String sql, int[] columnIndexes)	処理開始	B
0x8CC7	10		処理終了	B
0x8CC8	9	execute(String sql, String[] columnNames)	処理開始	B
0x8CC9	10		処理終了	B
0x8CCA	9	executeBatch()	処理開始	B
0x8CCB	10		処理終了	B
0x8CCC	9	executeQuery()	処理開始	B
0x8CCD	10		処理終了	B
0x8CCE	9	executeQuery(String sql)	処理開始	B
0x8CCF	10		処理終了	B

イベント ID	図中の番号	トレース取得ポイント	レベル
0x8CD0	9	executeUpdate()	処理開始
0x8CD1	10		処理終了
0x8CD2	9	executeUpdate(String sql)	処理開始
0x8CD3	10		処理終了
0x8CD4	9	executeUpdate(String sql, int autoGeneratedKeys)	処理開始
0x8CD5	10		処理終了
0x8CD6	9	executeUpdate(String sql, int[] columnIndexes)	処理開始
0x8CD7	10		処理終了
0x8CD8	9	executeUpdate(String sql, String[] columnNames)	処理開始
0x8CD9	10		処理終了

(凡例) B : 詳細

注

図 A-27 中の番号と対応しています。

表 A-44 DB Connector , JCA コンテナでのトレース取得ポイントの詳細  
(java.sql.CallableStatement インタフェースのメソッド実行時) 4

イベント ID	図中の番号	トレース取得ポイント	レベル
0x8D00	9	execute()	処理開始
0x8D01	10		処理終了
0x8D02	9	execute(String sql)	処理開始
0x8D03	10		処理終了
0x8D04	9	execute(String sql, int autoGeneratedKeys)	処理開始
0x8D05	10		処理終了
0x8D06	9	execute(String sql, int[] columnIndexes)	処理開始
0x8D07	10		処理終了
0x8D08	9	execute(String sql, String[] columnNames)	処理開始
0x8D09	10		処理終了
0x8D0A	9	executeBatch()	処理開始
0x8D0B	10		処理終了
0x8D0C	9	executeQuery()	処理開始
0x8D0D	10		処理終了
0x8D0E	9	executeQuery(String sql)	処理開始
0x8D0F	10		処理終了

イベント ID	図中の番号	トレース取得ポイント	レベル	
0x8D10	9	executeUpdate()	処理開始	B
0x8D11	10		処理終了	B
0x8D12	9	executeUpdate(String sql)	処理開始	B
0x8D13	10		処理終了	B
0x8D14	9	executeUpdate(String sql, int autoGeneratedKeys)	処理開始	B
0x8D15	10		処理終了	B
0x8D16	9	executeUpdate(String sql, int[] columnIndexes)	処理開始	B
0x8D17	10		処理終了	B
0x8D18	9	executeUpdate(String sql, String[] columnNames)	処理開始	B
0x8D19	10		処理終了	B

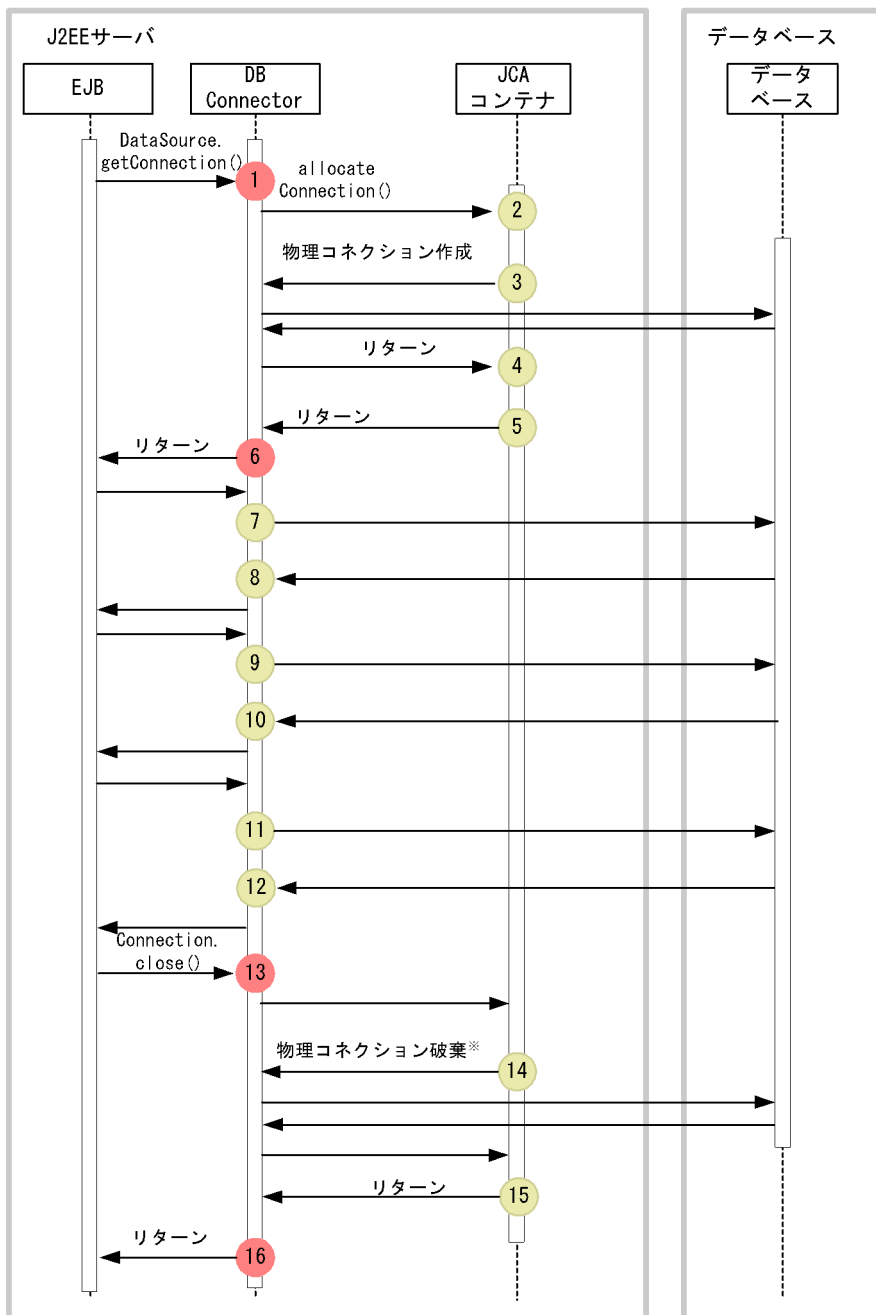
(凡例) B: 詳細

注

図 A-27 中の番号と対応しています。

DB Connector , JCA コンテナでのトレース取得ポイントを次の図に示します。

図 A-27 DB Connector, JCA コンテナのトレース取得ポイント (コネクション関連)



(凡例) ● : トレース取得ポイントを示します。PRFトレース取得レベルは「標準」です。

● : トレース取得ポイントを示します。PRFトレース取得レベルは「詳細」です。

注※ 物理コネクションの破棄は、発生する場合と発生しない場合があります。



## 取得できるトレース情報

DB Connector, JCA コンテナで取得できるトレース情報について, 次に示します。ここでは, 図 A-27 の番号と対応づけて説明しています。

番号 1 ~ 6・9 ~ 14 に対応するトレース情報

番号 1 ~ 6・9 ~ 14 に対応するイベント ID とトレース情報を, 次の表に示します。

表 A-45 DB Connector, JCA コンテナで取得できるトレース情報 (コネクション関連の処理実行時)

図中の 番号 1	イベント ID	レベ ル	取得できる情報		
			インタフェース名	オペレーション名	オプション
1	0x8C00	A	-	-	-
	0x8C02	A	-	-	-
	0x8C10	A	-	-	-
	0x8C12	A	-	-	-
2	0x8B00	B	-	-	-
3	0x8B80	B	-	-	-
4	0x8B81	B	-	-	2
5	0x8B01	B	-	-	2
6	0x8C01	A	コネクション ID	-	2
	0x8C03	A	コネクション ID	-	2
	0x8C11	A	コネクション ID	-	2
	0x8C13	A	コネクション ID	-	2
11	0x8C22	B	-	-	-
	0x8C24	B	-	-	-
	0x8C26	B	-	-	-
12	0x8C23	B	-	-	2
	0x8C25	B	-	-	2
	0x8C27	B	-	-	2
13	0x8C20	A	コネクション ID	-	-
14	0x8B82	B	-	-	-
15	0x8B83	B	-	-	2
16	0x8C21	A	-	-	2

(凡例) A: 標準 B: 詳細 -: 該当なし

注 1

図 A-27 中の番号と対応しています。

注 2

正常に処理された場合、入り口時刻が表示されます。  
例外が発生した場合、入り口時刻および例外が表示されます。

番号 7 に対応するトレース情報

番号 7 に対応するイベント ID は次のとおりです。

0x8C28, 0x8C2A, 0x8C2C, 0x8C2E, 0x8C30, 0x8C32, 0x8C34,  
0x8C36, 0x8C38, 0x8C3A, 0x8C3C, 0x8C3E

これらのイベント ID で取得できるトレース情報を示します。

- PRF トレース取得レベル  
すべて「詳細」です。
- インタフェース名, およびオペレーション名  
これらのイベント ID では出力されません。
- オプション  
メソッドの引数に sql を持つ場合, SQL 文が表示されます。

番号 8 に対応するトレース情報

番号 8 に対応するイベント ID は次のとおりです。

0x8C29, 0x8C2B, 0x8C2D, 0x8C2F, 0x8C31, 0x8C33, 0x8C35,  
0x8C37, 0x8C39, 0x8C3B, 0x8C3D, 0x8C3F

これらのイベント ID で取得できるトレース情報を示します。

- PRF トレース取得レベル  
すべて「詳細」です。
- インタフェース名, オペレーション名  
これらのイベント ID では出力されません。
- オプション  
これらのイベント ID では, 正常に処理された場合は入り口時刻が表示されます。  
例外が発生した場合は入り口時刻および例外が表示されます。

番号 9 に対応するトレース情報

番号 9 に対応するイベント ID は次のとおりです。

0x8C80, 0x8C82, 0x8C84, 0x8C86, 0x8C88, 0x8C8A, 0x8C8C, 0x8C8E,  
0x8C90, 0x8C92, 0x8CC0, 0x8CC2, 0x8CC4, 0x8CC6, 0x8CC8, 0x8CCA,  
0x8CCC, 0x8CCE, 0x8CD0, 0x8CD2, 0x8CD4, 0x8CD6, 0x8CD8, 0x8D00,  
0x8D02, 0x8D04, 0x8D06, 0x8D08, 0x8D0A, 0x8D0C, 0x8D0E, 0x8D10,  
0x8D12, 0x8D14, 0x8D16, 0x8D18

これらのイベント ID で取得できるトレース情報を示します。

- PRF トレース取得レベル  
すべて「詳細」です。
- インタフェース名, およびオペレーション名  
これらのイベント ID では出力されません。
- オプション  
メソッドの引数に sql を持つ場合, SQL 文が表示されます。

番号 10 に対応するトレース情報

番号 10 に対応するイベント ID は次のとおりです。

0x8C81, 0x8C83, 0x8C85, 0x8C87, 0x8C89, 0x8C8B, 0x8C8D, 0x8C8F,  
0x8C91, 0x8C93, 0x8CC1, 0x8CC3, 0x8CC5, 0x8CC7, 0x8CC9, 0x8CCB,  
0x8CCD, 0x8CCF, 0x8CD1, 0x8CD3, 0x8CD5, 0x8CD7, 0x8CD9, 0x8D01,  
0x8D03, 0x8D05, 0x8D07, 0x8D09, 0x8D0B, 0x8D0D, 0x8D0F, 0x8D11,  
0x8D13, 0x8D15, 0x8D17, 0x8D19

これらのイベント ID で取得できるトレース情報を示します。

- PRF トレース取得レベル  
すべて「詳細」です。
- インタフェース名, オペレーション名  
これらのイベント ID では出力されません。
- オプション  
これらのイベント ID では, 正常に処理された場合は入り口時刻が表示されます。  
例外が発生した場合は入り口時刻および例外が表示されます。

## (2) ローカルトランザクションを使用する場合のトレース取得ポイントと取得できるトレース情報

ローカルトランザクションを使用する場合のトレース取得ポイントと取得できるトレース情報トレース取得ポイントと取得できるトレース情報について説明します。

トレース取得ポイントと PRF トレース取得レベル

イベント ID, トレース取得ポイント, および PRF トレース取得レベルについて, 次の表に示します。

表 A-46 DB Connector, JCA コンテナでのトレース取得ポイントの詳細 (ローカルトランザクションの処理実行時)

イベント ID	図中の番号	トレース取得ポイント		レベル
0x8C60	1	ローカルトランザクションの開始	処理開始	B
0x8C61	2		処理終了	B
0x8C62	3	ローカルトランザクションのコミット	処理開始	B
0x8C63	4		処理終了	B
0x8C64	3	ローカルトランザクションのロールバック	処理開始	B
0x8C65	4		処理終了	B

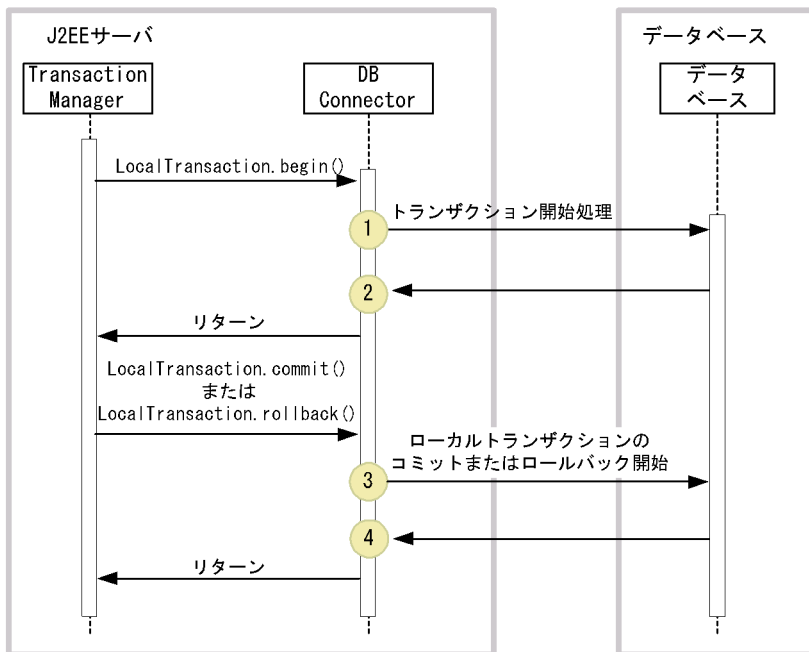
(凡例) B: 詳細

注

図 A-28 中の番号と対応しています。

トレース取得ポイントを次の図に示します。

図 A-28 DB Connector, JCA コンテナでのトレース取得ポイントの詳細 (ローカルトランザクション関連)



(凡例) ● : トレース取得ポイントを示します。PRF トレース取得レベルは「詳細」です。

取得できるトレース情報

取得できるトレース情報を次の表に示します。

表 A-47 DB Connector, JCA コンテナでのトレース取得ポイントの詳細 (ローカルトランザクションの処理実行時)

図中の番号	イベント ID	レベル	取得できる情報		
			インタフェース名	オペレーション名	オプション
1	0x8C60	B	-	-	-
2	0x8C61	B	-	-	2
3	0x8C62	B	-	-	-
	0x8C64	B	-	-	-
4	0x8C63	B	-	-	2
	0x8C65	B	-	-	2

(凡例) B: 詳細 - : 該当なし

注 1

図 A-28 中の番号と対応しています。

注 2

正常に処理された場合，入り口時刻が表示されます。

例外が発生した場合，入り口時刻および例外が表示されます。

### (3) コネクションアソシエーションを使用する場合のトレース取得ポイントと取得できるトレース情報

コネクションアソシエーションを使用する場合のトレース取得ポイントと取得できるトレース情報について説明します。

#### トレース取得ポイントと PRF トレース取得レベル

イベント ID，トレース取得ポイント，および PRF トレース取得レベルについて，次の表に示します。

表 A-48 DB Connector，JCA コンテナでのトレース取得ポイントの詳細（コネクションアソシエーションの処理実行時）

イベント ID	図中の番号	トレース取得ポイント	レベル
0x8C40	1	コネクションアソシエーション機能によって，論理コネクションに対する物理コネクションが差し替えられた時	A

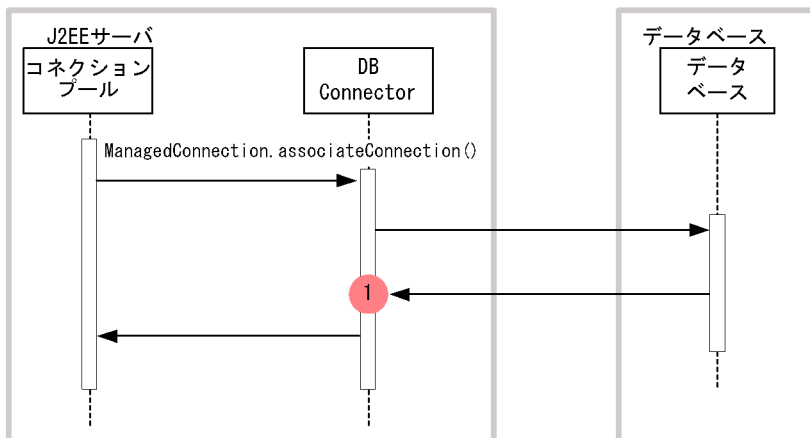
(凡例) A: 標準

注

図 A-29 中の番号と対応しています。

トレース取得ポイントを次の図に示します。

図 A-29 DB Connector, JCA コンテナでのトレース取得ポイントの詳細 (コネクションアソシエーション関連)



(凡例) ● : トレース取得ポイントを示します。PRF トレース取得レベルは「標準」です。

取得できるトレース情報

取得できるトレース情報を次の表に示します。

表 A-49 DB Connector, JCA コンテナでのトレース取得ポイントの詳細 (コネクションアソシエーション処理実行時)

図中の番号	イベント ID	レベル	取得できる情報		
			インタフェース名	オペレーション名	オプション
1	0x8C40	A	コネクション ID (コネクションアソシエーション機能によって差し替えられた先の物理コネクションのコネクション ID)	-	-

(凡例) A : 標準 - : 該当なし

注

図 A-29 中の番号と対応しています。

(4) コネクション自動クローズ機能を使用する場合のトレース取得ポイントと取得できるトレース情報

コネクション自動クローズ時のトレース取得ポイントと取得できるトレース情報について説明します。

トレース取得ポイントと PRF トレース取得レベル

イベント ID, トレース取得ポイント, および PRF トレース取得レベルについて, 次の

表に示します。

表 A-50 DB Connector, JCA コンテナでのトレース取得ポイントの詳細 (コネクション自動クローズ時)

イベント ID	図中の番号	トレース取得ポイント	レベル
0x8B84	1	コネクション自動クローズ機能によるクローズ処理呼び出し直後	B
0x8B82	2	物理コネクション破棄の呼び出し直前	B
0x8B83	3	物理コネクション破棄のリターン直後	B
0x8B85	4	コネクション自動クローズ機能によるクローズ処理リターン直前	B

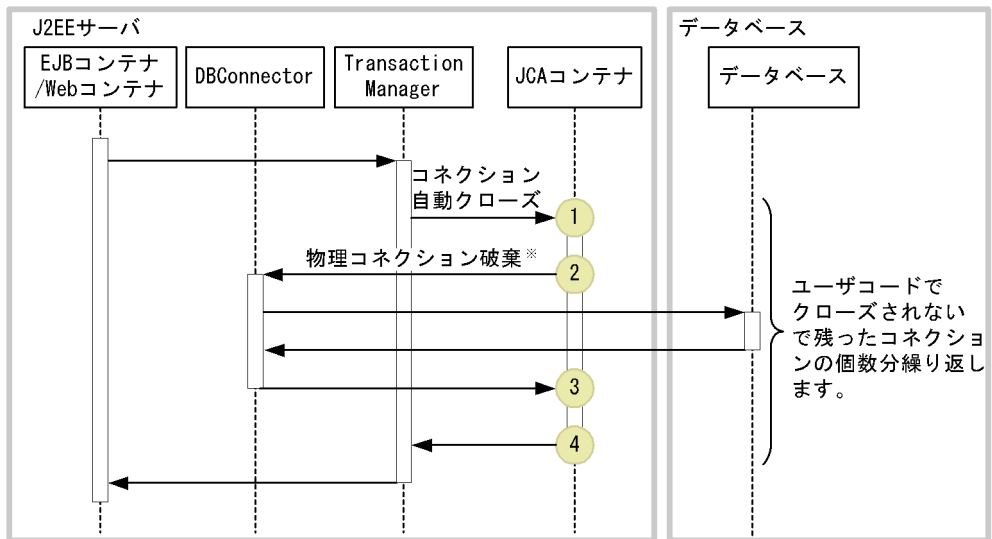
(凡例) B: 詳細

注

図 A-30 中の番号と対応しています。

トレース取得ポイントを次の図に示します。

図 A-30 DB Connector, JCA コンテナでのトレース取得ポイントの詳細 (コネクション自動クローズ時)



(凡例) ● : トレース取得ポイントを示します。PRFトレース取得レベルは「詳細」です。

注※ コネクションのクローズ時に、常に物理コネクション破棄が起こるわけではありません。

### 取得できるトレース情報

取得できるトレース情報を次の表に示します。

表 A-51 DB Connector , JCA コンテナでのトレース取得ポイントの詳細 (コネクション自動クローズ時)

図中の番号 1	イベント ID	レベル	取得できる情報		
			インタフェース名	オペレーション名	オプション
1	0x8B84	B	コネクション ID	-	-
2	0x8B82	B	-	-	-
3	0x8B83	B	-	-	2
4	0x8B85	B	-	-	入り口時刻

(凡例) B : 詳細 - : 該当なし

注 1

図 A-30 中の番号と対応しています。

注 2

正常に処理された場合、入り口時刻が表示されます。

例外が発生した場合、入り口時刻および例外が表示されます。

### (5) DB Connector for Cosminexus RM と連携する場合のトレース取得ポイントと取得できるトレース情報

DB Connector for Cosminexus RM と連携する場合のトレース取得ポイントと取得できるトレース情報について説明します。

なお、DB Connector for Cosminexus RM では、DB Connector を使用してデータベースに接続しています。そのため DB Connector for Cosminexus RM 利用時には DB Connector のトレースポイントも取得します。JDBC コネクションの生成物 (java.sql.Statement など) は、DB Connector と同じトレース取得ポイントになります。

#### トレース取得ポイントと PRF トレース取得レベル

イベント ID、トレース取得ポイント、および PRF トレース取得レベルについて、次の表に示します。

表 A-52 DB Connector , JCA コンテナでのトレース取得ポイントの詳細 (DB Connector for Cosminexus RM 連携時)

イベント ID	図中の番号	トレース取得ポイント		レベル
0x8D60	1	DB Connector for Cosminexus RM 利用時の javax.sql.DataSource.getConnection() でのデータベース接続の確立	処理開始	A
0x8D61	2		処理終了	A



イベント ID	図中の 番号	トレース取得ポイント		レベル
0x8D62	1	DB Connector for Cosminexus RM 利用時の javax.sql.DataSource.getConnection(String username, String password) での、データベース 接続の確立	処理開始	A
0x8D63	2		処理終了	A
0x8D80	5	DB Connector for Cosminexus RM 利用時の java.sql.Connection.close() での、Connection オ ブジェクトのデータベースと JDBC リソースの解 除	処理開始	A
0x8D81	6		処理終了	A
0x8D82	3	DB Connector for Cosminexus RM 利用時の Connection.createStatement()	処理開始	B
0x8D83	4		処理終了	B
0x8D84	3	DB Connector for Cosminexus RM 利用時の Connection.createStatement(int resultSetType, int resultSetConcurrency)	処理開始	B
0x8D85	4		処理終了	B
0x8D86	3	DB Connector for Cosminexus RM 利用時の Connection.createStatement(int resultSetType, int resultSetConcurrency, int resultSetHoldability)	処理開始	B
0x8D87	4		処理終了	B
0x8D88	3	DB Connector for Cosminexus RM 利用時の Connection.prepareCall(String sql)	処理開始	B
0x8D89	4		処理終了	B
0x8D8A	3	DB Connector for Cosminexus RM 利用時の (String sql, int resultSetType, int resultSetConcurrency)	処理開始	B
0x8D8B	4		処理終了	B
0x8D8C	3	DB Connector for Cosminexus RM 利用時の Connection.prepareCall(String sql, int resultSetType, int resultSetConcurrency, int resultSetHoldability)	処理開始	B
0x8D8D	4		処理終了	B
0x8D8E	3	DB Connector for Cosminexus RM 利用時の Connection.prepareStatement(String sql)	処理開始	B
0x8D8F	4		処理終了	B
0x8D90	3	DB Connector for Cosminexus RM 利用時の Connection.prepareStatement(String sql, int autoGeneratedKeys)	処理開始	B
0x8D91	4		処理終了	B
0x8D92	3	DB Connector for Cosminexus RM 利用時の Connection.prepareStatement(String sql, int[] columnIndexes)	処理開始	B
0x8D93	4		処理終了	B
0x8D94	3	DB Connector for Cosminexus RM 利用時の Connection.prepareStatement(String sql, int resultSetType, int resultSetConcurrency)	処理開始	B
0x8D95	4		処理終了	B
0x8D96	3	DB Connector for Cosminexus RM 利用時の Connection.prepareStatement(String sql, int resultSetType, int resultSetConcurrency, int resultSetHoldability)	処理開始	B
0x8D97	4		処理終了	B

イベント ID	図中の番号	トレース取得ポイント	レベル
0x8D98	3	DB Connector for Cosminexus RM 利用時の Connection.prepareStatement(String sql, String[] columnNames)	処理開始
0x8D99	4		処理終了

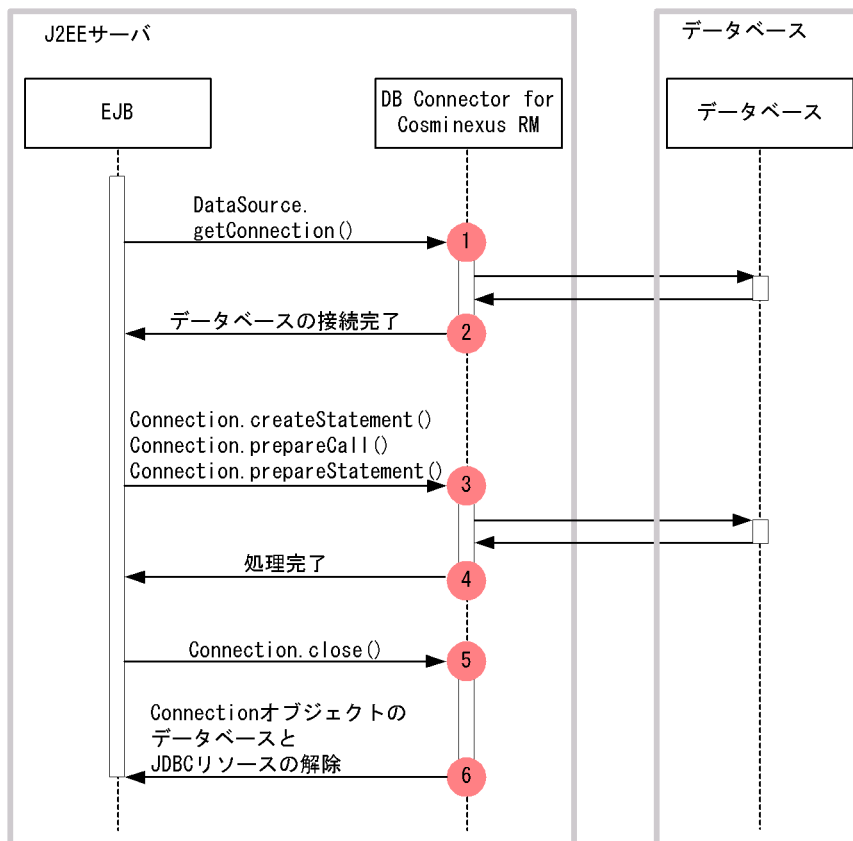
(凡例) A: 標準 B: 詳細

注

図 A-31 中の番号と対応しています。

トレース取得ポイントを次の図に示します。

図 A-31 DB Connector, JCA コンテナでのトレース取得ポイントの詳細 (DB Connector for Cosminexus RM 連携時)



(凡例) ● : トレース取得ポイントを示します。PRF トレース取得レベルは「標準」です。

取得できるトレース情報

取得できるトレース情報を次の表に示します。

表 A-53 DB Connector , JCA コンテナでのトレース取得ポイントの詳細 (DB Connector for Cosminexus RM 連携時)

図中の 番号 1	イベント ID	レベル	取得できる情報		
			インタフェース名	オペレーション名	オプション
1	0x8D60	A	-	-	-
2	0x8D61	A	コネクション ID	-	2
1	0x8D62	A	-	-	-
2	0x8D63	A	コネクション ID	-	2
5	0x8D80	A	コネクション ID	-	-
6	0x8D81	A	-	-	2
3	0x8D82	B	-	-	-
4	0x8D83	B	-	-	2
3	0x8D84	B	-	-	-
4	0x8D85	B	-	-	2
3	0x8D86	B	-	-	-
4	0x8D87	B	-	-	2
3	0x8D88	B	-	-	SQL 文
4	0x8D89	B	-	-	2
3	0x8D8A	B	-	-	SQL 文
4	0x8D8B	B	-	-	2
3	0x8D8C	B	-	-	SQL 文
4	0x8D8D	B	-	-	2
3	0x8D8E	B	-	-	SQL 文
4	0x8D8F	B	-	-	2
3	0x8D90	B	-	-	SQL 文
4	0x8D91	B	-	-	2
3	0x8D92	B	-	-	SQL 文
4	0x8D93	B	-	-	2
3	0x8D94	B	-	-	SQL 文
4	0x8D95	B	-	-	2
3	0x8D96	B	-	-	SQL 文
4	0x8D97	B	-	-	2
3	0x8D98	B	-	-	SQL 文
4	0x8D99	B	-	-	2

(凡例) A: 標準 B: 詳細 - : 該当なし

注 1

図 A-31 中の番号と対応しています。

注 2

正常に処理された場合、入り口時刻が表示されます。

例外が発生した場合、入り口時刻および例外が表示されます。

## (6) ワーク管理を使用する場合のトレース取得ポイントと取得できるトレース情報

Connector1.5 に準拠したリソースアダプタでワーク管理を使用する場合のトレース取得ポイントと取得できるトレース情報について説明します。

トレース取得ポイントと PRF トレース取得レベル

イベント ID, トレース取得ポイント, および PRF トレース取得レベルについて, 次の表に示します。

表 A-54 DB Connector, JCA コンテナでのトレース取得ポイントの詳細 (ワーク管理を使用する場合)

イベント ID	図中の番号	トレース取得ポイント	レベル
0x8B86	1	javax.resource.spi.work.WorkManager のメソッドの呼び出し直後	A
0x8B88	2	javax.resource.spi.work.WorkListener のメソッドの呼び出し直前	B
0x8B89	3	javax.resource.spi.work.WorkListener のメソッドのリターン直後	B
0x8B88	4	javax.resource.spi.work.WorkListener のメソッドの呼び出し直前	B
0x8B89	5	javax.resource.spi.work.WorkListener のメソッドのリターン直後	B
0x8B87	6	javax.resource.spi.work.WorkManager のメソッドのリターン直前	A
0x8B88	7	javax.resource.spi.work.WorkListener のメソッドの呼び出し直前	B
0x8B89	8	javax.resource.spi.work.WorkListener のメソッドのリターン直後	B
0x8B87	9	javax.resource.spi.work.WorkManager のメソッドのリターン直前	A
0x8B8A	10	javax.resource.spi.work.Work のメソッド呼び出し直前	A
0x8B8B	11	javax.resource.spi.work.Work のメソッドのリターン直後	A

イベント ID	図中の番号	トレース取得ポイント	レベル
0x8B88	12	javax.resource.spi.work.WorkListener のメソッドの呼び出し直前	B
0x8B89	13	javax.resource.spi.work.WorkListener のメソッドのリターン直後	B
0x8B87	14	javax.resource.spi.work.WorkManager のメソッドのリターン直前	A

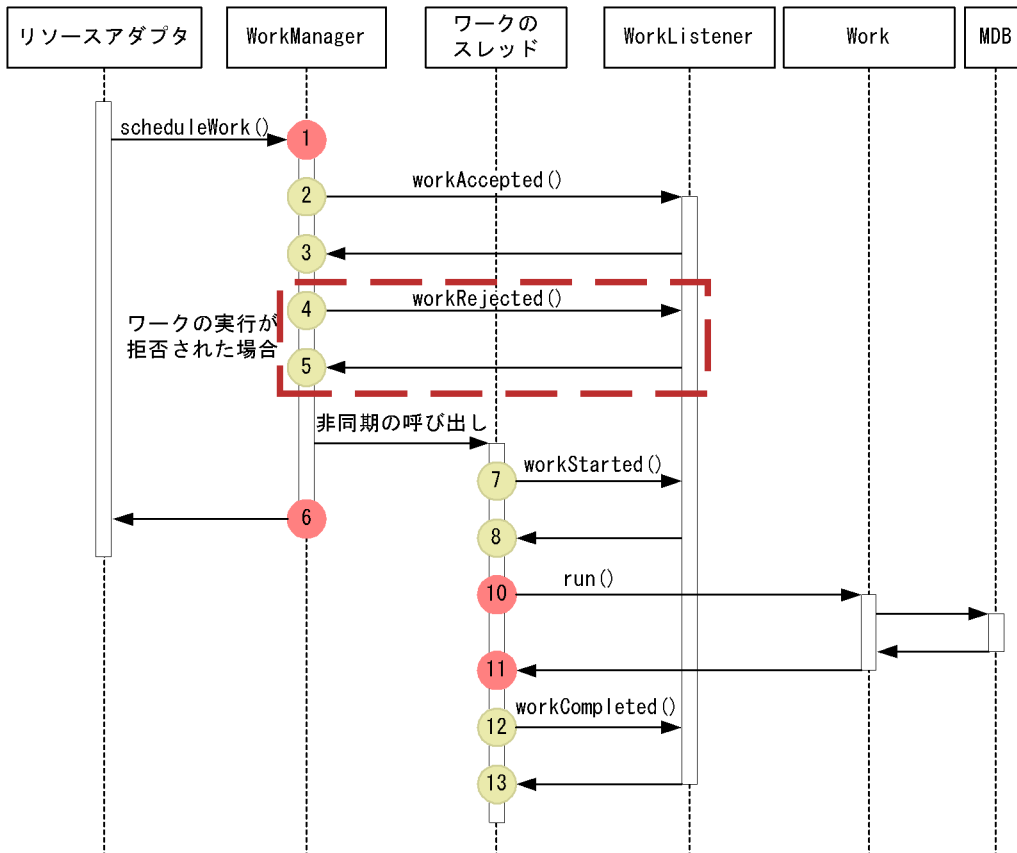
( 凡例 ) A : 標準 B : 詳細

注

図 A-32 ~ 図 A-34 中の番号と対応しています。

トレース取得ポイントを次の図に示します。

図 A-32 DB Connector, JCA コンテナでのトレース取得ポイントの詳細  
(scheduleWork() 呼び出し時)



- (凡例)
- : トレース取得ポイントを示します。PRF トレース取得レベルは「標準」です。
  - : トレース取得ポイントを示します。PRF トレース取得レベルは「詳細」です。

図 A-33 DB Connector , JCA コンテナでのトレース取得ポイントの詳細 ( startWork() 呼び出し時 )

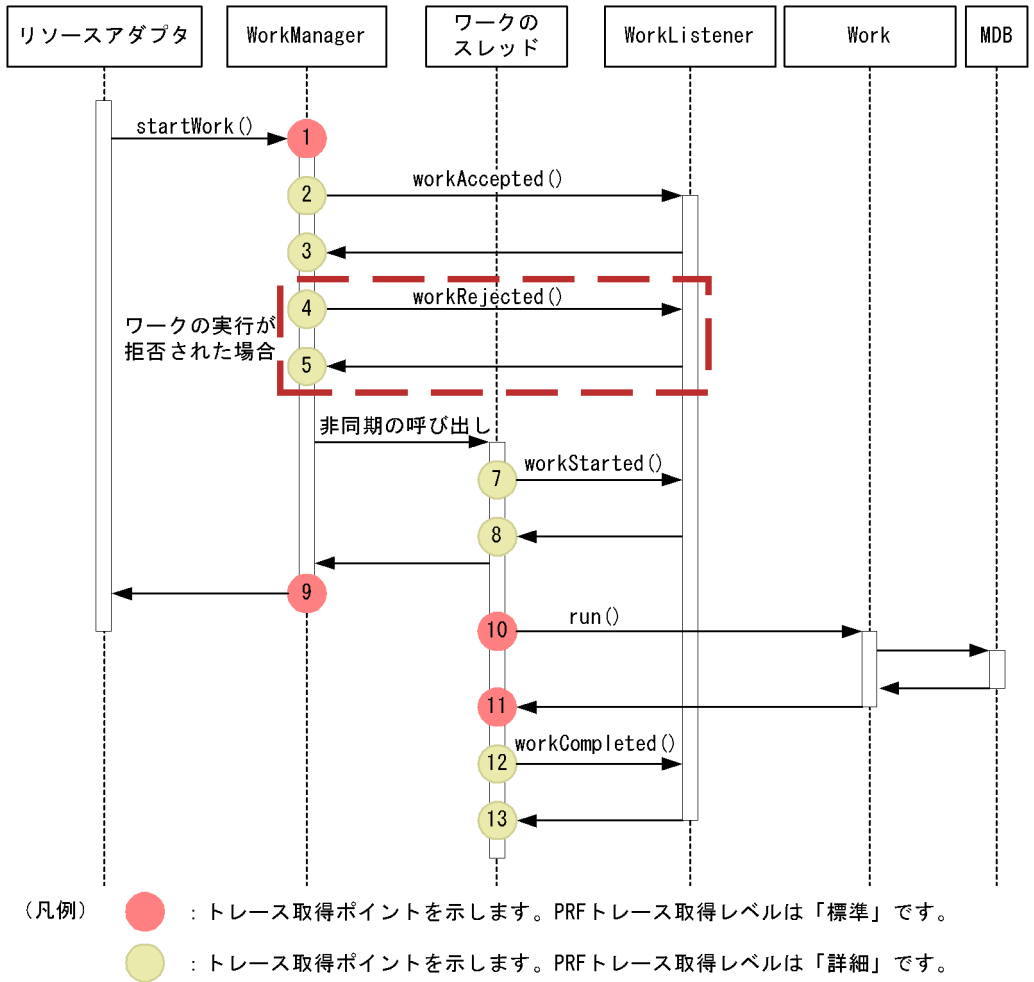
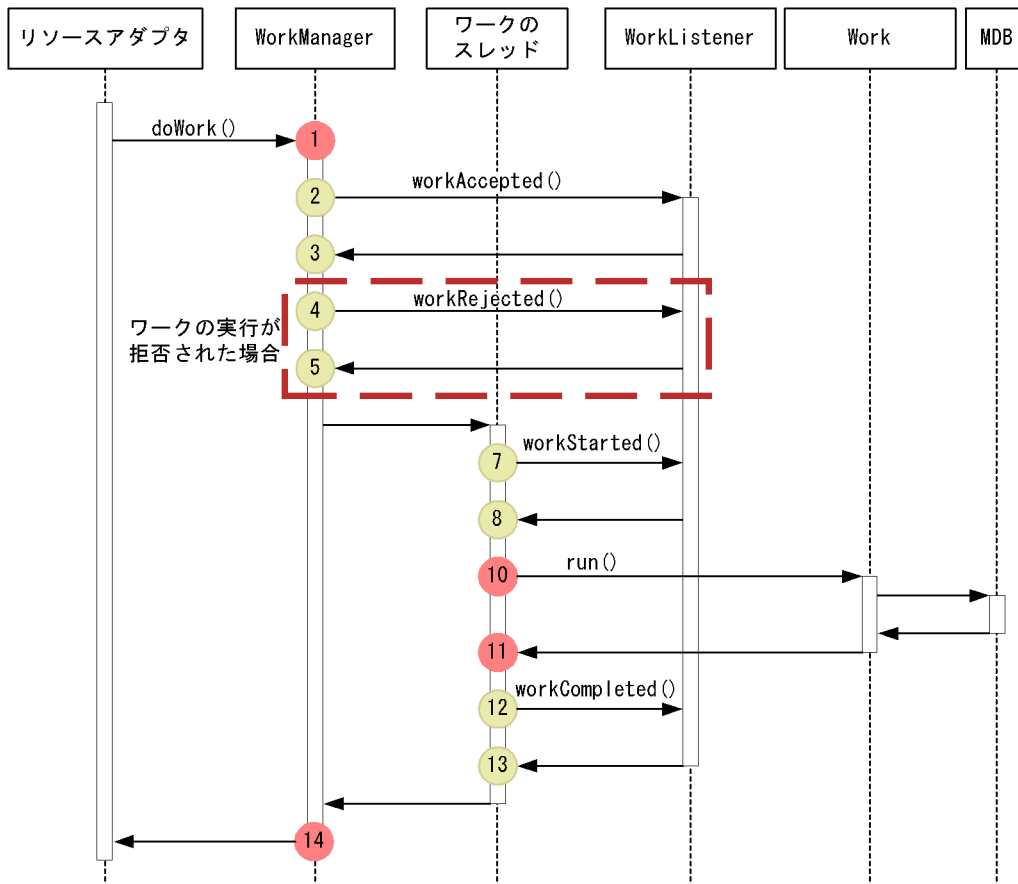


図 A-34 DB Connector, JCA コンテナでのトレース取得ポイントの詳細 (doWork() 呼び出し時)



- (凡例)
- (Red): トレース取得ポイントを示します。PRFトレース取得レベルは「標準」です。
  - (Green): トレース取得ポイントを示します。PRFトレース取得レベルは「詳細」です。

取得できるトレース情報

取得できるトレース情報を次の表に示します。

表 A-55 DB Connector, JCA コンテナでのトレース取得ポイントの詳細 (ワーク管理を使用する場合)

図中の番号	イベント ID	レベル	取得できる情報		
			インタフェース名	オペレーション名	オプション
1	0x8B86	A	Work のクラス名	メソッド名	リソースアダプタの表示名



図中の 番号 1	イベント ID	レベル	取得できる情報		
			インタフェース名	オペレーション名	オプション
2	0x8B88	B	Work のクラス名	メソッド名	リソースアダプタの表示名 <sup>2</sup>
3	0x8B89	B	Work のクラス名	メソッド名	3
4	0x8B88	B	Work のクラス名	メソッド名	リソースアダプタの表示名 <sup>2</sup>
5	0x8B89	B	Work のクラス名	メソッド名	3
6	0x8B87	A	Work のクラス名	メソッド名	3
7	0x8B88	B	Work のクラス名	メソッド名	リソースアダプタの表示名 <sup>2</sup>
8	0x8B89	B	Work のクラス名	メソッド名	3
9	0x8B87	A	Work のクラス名	メソッド名	3
10	0x8B8A	A	Work のクラス名	メソッド名	リソースアダプタの表示名
11	0x8B8B	A	Work のクラス名	メソッド名	3
12	0x8B88	B	Work のクラス名	メソッド名	リソースアダプタの表示名 <sup>2</sup>
13	0x8B89	B	Work のクラス名	メソッド名	3
14	0x8B87	A	Work のクラス名	メソッド名	3

(凡例) A: 標準 B: 詳細

注 1

図 A-32 ~ 図 A-34 中の番号と対応しています。

注 2

アプリケーションに含まれているリソースアダプタの場合、「アプリケーション名: リソースアダプタの表示名」と表示されます。

注 3

正常に処理された場合、入り口時刻が表示されます。  
例外が発生した場合、入り口時刻および例外が表示されます。

## 付録 A.11 RMI のトレース取得ポイント

ここでは、RMI のトレース取得ポイントと、取得できるトレース情報について説明します。

トレース取得ポイントとPRF トレース取得レベル

イベント ID, トレース取得ポイント, および PRF トレース取得レベルについて, 次の表に示します。

表 A-56 RMI でのトレース取得ポイントの詳細

イベント ID	図中の番号	トレース取得ポイント	レベル
0x8E01	1	スタブでのリクエスト送信処理の始まり	A
0x8E02	6	スタブでのレスポンス受信処理の終わり	A
0x8E03	2	クライアント側のリクエスト送信処理	A
0x8E04	5	クライアント側のレスポンス受信処理	A
0x8E05	3	サーバ側のリクエスト受信処理	A
0x8E06	4	サーバ側のレスポンス送信処理	A

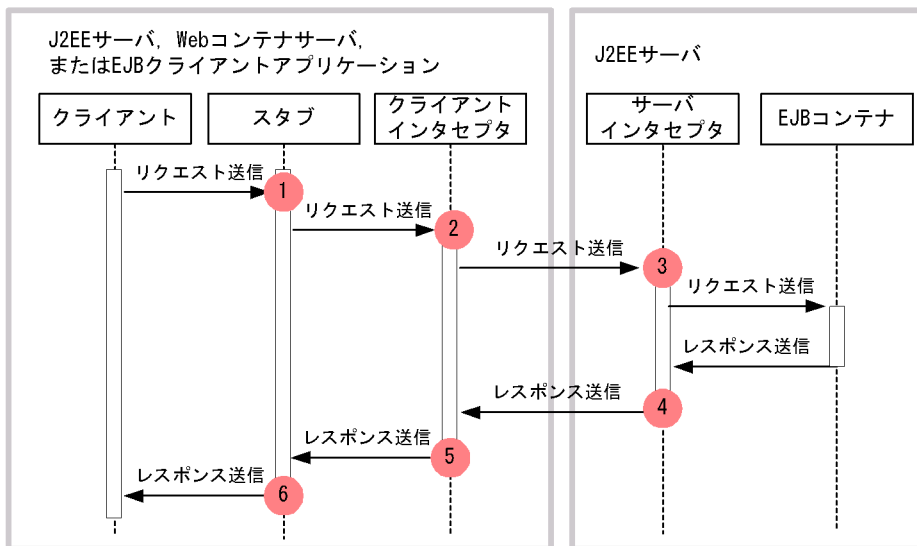
(凡例) A: 標準

注

図 A-35 中の番号と対応しています。

RMI でのトレース取得ポイントを次の図に示します。

図 A-35 RMI のトレース取得ポイント



(凡例) ● : トレース取得ポイントを示します。PRF トレース取得レベルは「標準」です。

取得できるトレース情報

RMI で取得できるトレース情報を次の表に示します。

表 A-57 RMI で取得できるトレース情報

図中の 番号 1	イベント ID	レベ ル	取得できる情報		
			インタフェース名	オペレーション名	オプション
1	0x8E01	A	インタフェース名	オペレーション名	-
2	0x8E03	A	-	-	-
3	0x8E05	A	-	-	-
4	0x8E06	A	-	-	-
5	0x8E04	A	-	-	-
6	0x8E02	A	インタフェース名 <sup>2</sup>	オペレーション名 <sup>2</sup>	- <sup>2</sup>

(凡例) A:標準 - :該当なし

注 1

図 A-35 中の番号と対応しています。

注 2

例外が発生した場合、インタフェース名およびオペレーション名は表示されません。また、オプションには発生した例外が表示されます。

## 付録 A.12 OTS のトレース取得ポイント

ここでは、OTS のトレース取得ポイントと、取得できるトレース情報について説明します。

### トレース取得ポイントと PRF トレース取得レベル

イベント ID、トレース取得ポイント、および PRF トレース取得レベルについて、次の表に示します。

表 A-58 OTS でのトレース取得ポイントの詳細

イベント ID	図中の番 号	トレース取得ポイント	レベル
0x9400	1	トランザクション生成直後	A
0x9401	2	トランザクションの状態が MarkedRollback へ遷移した直後	A
0x9402	3	トランザクションの状態が RollingBack へ遷移した直後	A
0x9403	12	トランザクション決着直後	A
0x9404	4	javax.transaction.xa.XAResource への決着処理直前	B
0x9405	5	javax.transaction.xa.XAResource への決着処理直後	B
0x9406	6	サブオーディネートトランザクションへの決着処理直前	B
0x9407	9	サブオーディネートトランザクションへの決着処理直後	B
0x9408	7	スベリアトランザクションからの決着処理要求受信直後	B

イベント ID	図中の番号	トレース取得ポイント	レベル
0x9409	8	スベリアトランザクションからの決着処理要求返信直前	B
0x9410	13	javax.transaction.xa.XAResource オブジェクト取得処理直前	B
0x9411	14	javax.transaction.xa.XAResource オブジェクト取得処理直後	B
0x9412	10	ステータスファイルへの書き込み・読み込み直前	B
0x9413	11	ステータスファイルへの書き込み・読み込み直後	B

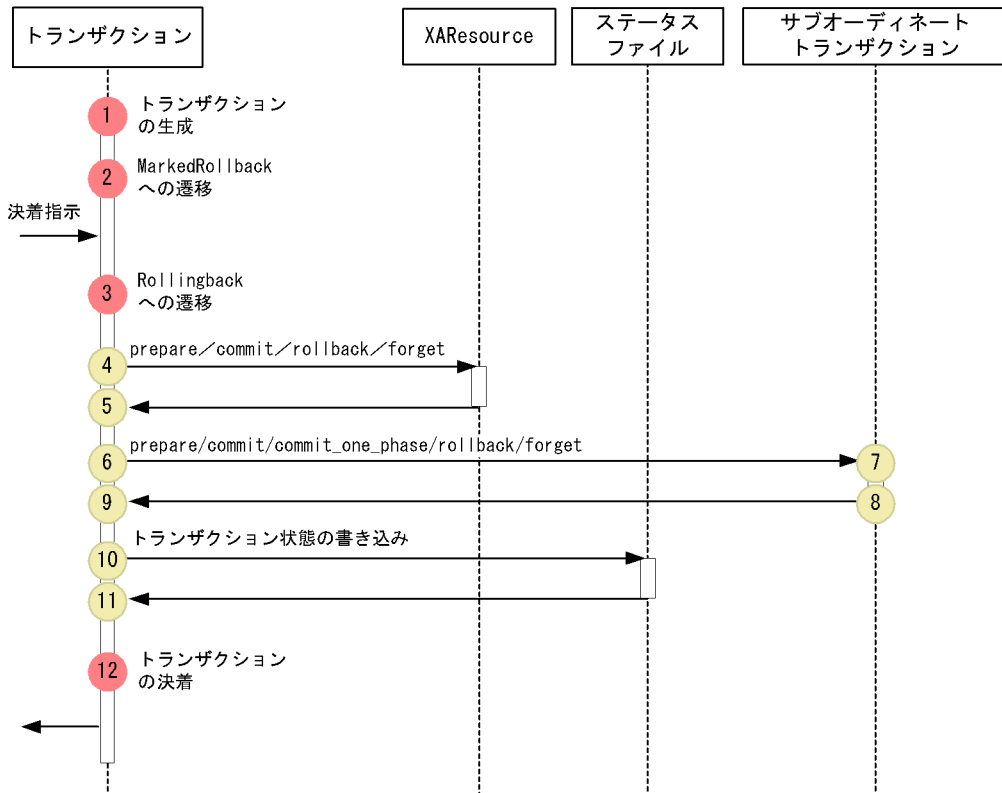
(凡例) A: 標準 B: 詳細

注

図 A-36 ~ A-40 中の番号と対応しています。

OTS でのトレース取得ポイントを次の図に示します。

図 A-36 トランザクション生成から決着までのトレース取得ポイント



(凡例) ● : トレース取得ポイントを示します。PRF トレース取得レベルは「標準」です。

● : トレース取得ポイントを示します。PRF トレース取得レベルは「詳細」です。

図 A-37 インプロセス OTS 初期化時のステータスファイルへの読み込み・書き込みに関連するトレース取得ポイント

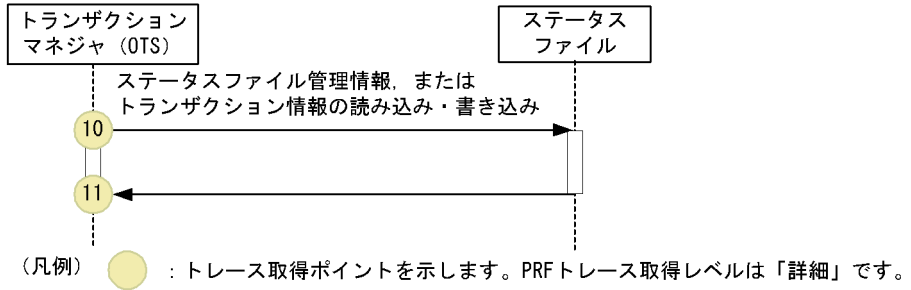


図 A-38 javax.transaction.xa.Xid の回復処理でのトレース取得ポイント

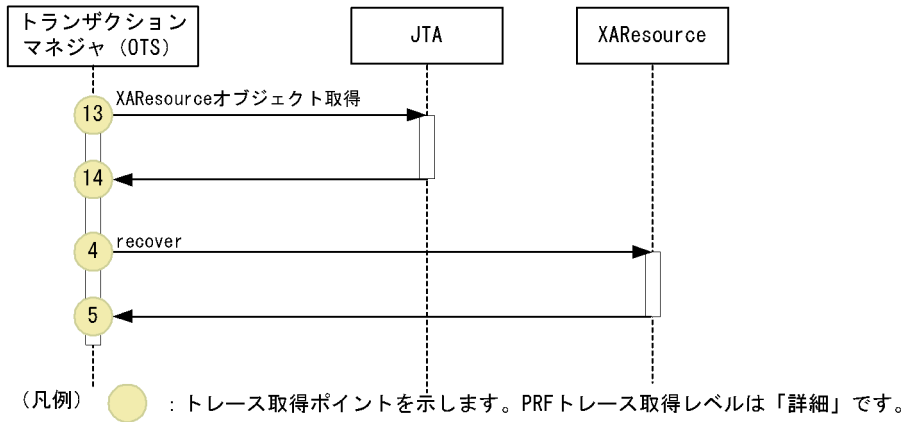


図 A-39 典型的な 1 フェーズコミットモデルでのトレース取得ポイント

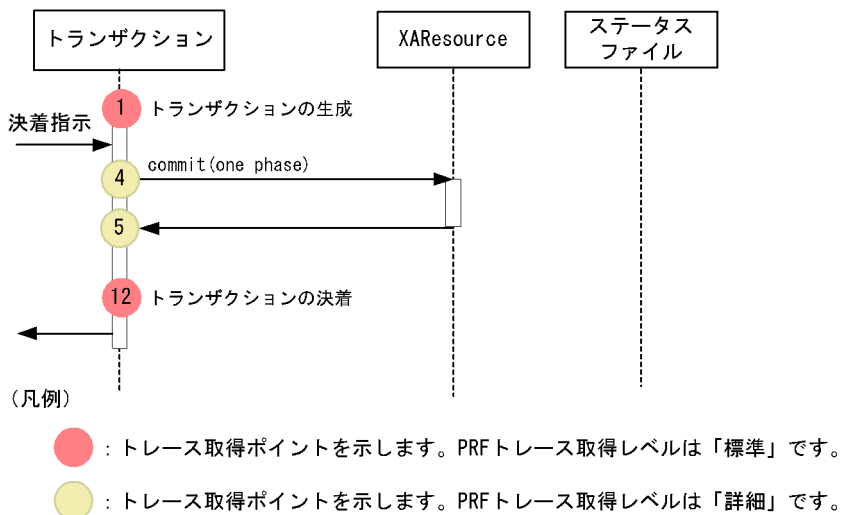
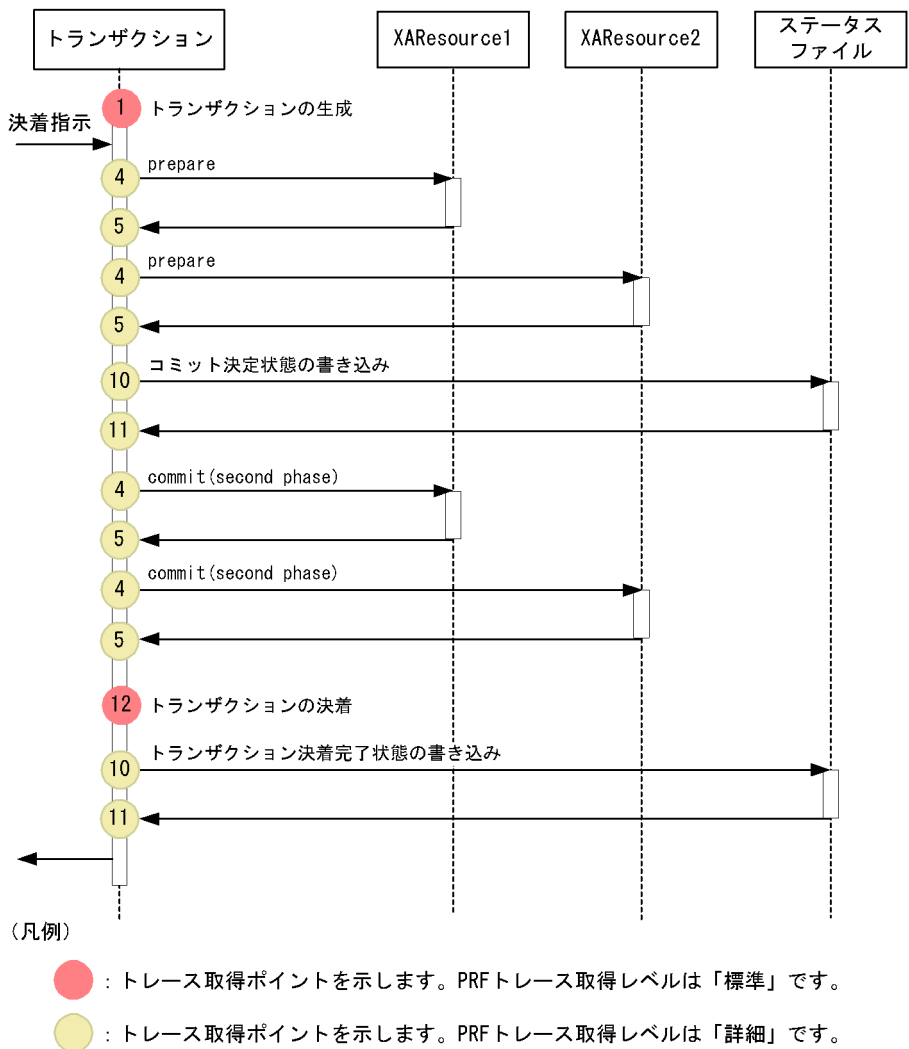


図 A-40 典型的な 2 フェーズコミットモデルでのトレース取得ポイント



取得できるトレース情報

OTS で取得できるトレース情報を次の表に示します。なお、次の表の中で、一つの項目に複数の情報が個条書きで記載してある場合は、そのどれかが出力されることを意味します。

表 A-59 OTS で取得できるトレース情報

図中の 番号 1	イベント ID	レベ ル	取得できる情報		
			インタフェース名	オペレーション名	オプション
1	0x9400	A	global transaction	<ul style="list-style-type: none"> <li>•   created トランザクション開始指示を受け、生成されました。</li> <li>•   recreated 別ノードからトランザクション処理への参加指示を受け、生成されました。</li> <li>•   recovered ステータスファイルから回復されました。</li> <li>•   recovered(orphan) javax.transaction.xa.XA Resource から Xid が回復されましたが、これに対応するトランザクションが存在しないため、トランザクションが新たに生成されました。</li> </ul>	<グローバルトランザクション ID>
2	0x9401	A	global transaction	marked rollback(< 遷移した理由 > <sup>2)</sup> )	<グローバルトランザクション ID>
3	0x9402	A	global transaction	rolling back(< 遷移した理由 > <sup>3)</sup> )	<グローバルトランザクション ID>
12	0x9403	A	global transaction	<ul style="list-style-type: none"> <li>•   committed コミットしました。</li> <li>•   rolled back ロールバックしました。</li> <li>•   heuristic commit 強制コミットしました。</li> <li>•   heuristic rollback 強制ロールバックしました。</li> <li>•   heuristic mixed 部分的にコミットおよびロールバックされました。</li> <li>•   heuristic hazard コミットしたかロールバックしたか不明です。</li> <li>•   unknown コミットしたかロールバックしたか不明です。</li> <li>•   invalid status 上記以外の状態で決着しました。</li> </ul>	<グローバルトランザクション ID>

図中の 番号 1	イベント ID	レベ ル	取得できる情報		
			インタフェース名	オペレーション名	オプション
4	0x9404	B	<ul style="list-style-type: none"> <li>• &lt;XAResource 識別子 &gt;</li> <li>• なし</li> </ul>	<ul style="list-style-type: none"> <li>• &gt; prepare prepare を発行します。</li> <li>• &gt; commit(second phase) 2 相目の commit を発行し ます。</li> <li>• &gt; commit(one phase) 1 相目の commit を発行し ます。</li> <li>• &gt; rollback rollback を発行します。</li> <li>• &gt; forget forget を発行します。</li> <li>• &gt; recover recover を発行します。</li> </ul>	<ul style="list-style-type: none"> <li>• &lt; グローバ ルトラン ザクシ ョン ID &gt;</li> <li>• なし</li> </ul>
5	0x9405	B	<ul style="list-style-type: none"> <li>• &lt;XAResource 識別子 &gt;</li> <li>• なし</li> </ul>	<p>正常リターンの場合</p> <ul style="list-style-type: none"> <li>• &lt; prepare</li> <li>• &lt; commit(second phase)</li> <li>• &lt; commit(one phase)</li> <li>• &lt; rollback</li> <li>• &lt; forget</li> <li>• &lt; recover</li> </ul> <p>異常リターンの場合（想定 外の値がリターンされた場 合，または例外が発生した 場合）</p> <ul style="list-style-type: none"> <li>• &lt;!prepare</li> <li>• &lt;!commit(second phase)</li> <li>• &lt;!commit(one phase)</li> <li>• &lt;!rollback</li> <li>• &lt;!forget</li> <li>• &lt;!recover</li> </ul>	<ul style="list-style-type: none"> <li>• &lt; グローバ ルトラン ザクシ ョン ID &gt; : &lt; 結果 &gt; <sup>4</sup></li> <li>• なし</li> </ul>
6	0x9406	B	resource	<ul style="list-style-type: none"> <li>• &gt; prepare prepare を発行します。</li> <li>• &gt; commit(second phase) 2 相目の commit を発行し ます。</li> <li>• &gt; commit(one phase) 1 相目の commit を発行し ます。</li> <li>• &gt; rollback rollback を発行します。</li> <li>• &gt; forget forget を発行します。</li> </ul>	< グローバル トランザク ション ID >



図中の 番号 1	イベント ID	レベル	取得できる情報		
			インタフェース名	オペレーション名	オプション
9	0x9407	B	resource	<p>正常リターンの場合</p> <ul style="list-style-type: none"> <li>• &lt; prepare</li> <li>• &lt; commit(second phase)</li> <li>• &lt; commit(one phase)</li> <li>• &lt; rollback</li> <li>• &lt; forget</li> </ul> <p>異常リターンの場合（想定外の値がリターンされた場合，または例外が発生した場合）</p> <ul style="list-style-type: none"> <li>• &lt;!prepare</li> <li>• &lt;!commit(second phase)</li> <li>• &lt;!commit(one phase)</li> <li>• &lt;!rollback</li> <li>• &lt;!forget</li> </ul>	<ul style="list-style-type: none"> <li>• &lt;グローバルトランザクション ID&gt;</li> <li>• &lt;グローバルトランザクション ID&gt;:&lt;結果&gt; <sup>5</sup></li> </ul>
7	0x9408	B	subordinate transaction	<ul style="list-style-type: none"> <li>• &gt; prepare prepare 指示が入ってきました。</li> <li>• &gt; commit(second phase) 2 相目の commit 指示が入ってきました。</li> <li>• &gt; commit(one phase) 1 相目の commit 指示が入ってきました。</li> <li>• &gt; rollback rollback 指示が入ってきました。</li> <li>• &gt; forget forget 指示が入ってきました。</li> </ul>	<グローバルトランザクション ID>
8	0x9409	B	subordinate transaction	<p>正常リターンの場合</p> <ul style="list-style-type: none"> <li>• &lt; prepare</li> <li>• &lt; commit(second phase)</li> <li>• &lt; commit(one phase)</li> <li>• &lt; rollback</li> <li>• &lt; forget</li> </ul> <p>異常リターンの場合（想定外の値がリターンされた場合，または例外が発生した場合）</p> <ul style="list-style-type: none"> <li>• &lt;!prepare</li> <li>• &lt;!commit(second phase)</li> <li>• &lt;!commit(one phase)</li> <li>• &lt;!rollback</li> <li>• &lt;!forget</li> </ul>	<ul style="list-style-type: none"> <li>• &lt;グローバルトランザクション ID&gt;</li> <li>• &lt;グローバルトランザクション ID&gt;:&lt;結果&gt; <sup>6</sup></li> </ul>

図中の 番号 1	イベント ID	レベ ル	取得できる情報		
			インタフェース名	オペレーション名	オプション
13	0x9410	B	<ul style="list-style-type: none"> <li>なし</li> <li>&lt;XAResource 識別子 &gt;</li> </ul>	> get xaresource	<ul style="list-style-type: none"> <li>なし</li> <li>&lt;グローバル ラン ザクシ ョン ID&gt;</li> </ul>
14	0x9411	B	<ul style="list-style-type: none"> <li>なし</li> <li>&lt;XAResource 識別子 &gt;</li> </ul>	正常リターンの場合 <ul style="list-style-type: none"> <li>&lt; get xaresource</li> </ul> 異常リターン場合（想定外 の値がリターンされた場合、 または例外が発生した場合） <ul style="list-style-type: none"> <li>&lt;!get xaresource</li> </ul>	<ul style="list-style-type: none"> <li>なし</li> <li>&lt;結果 &gt; 7</li> <li>&lt;グローバル ラン ザクシ ョン ID&gt;</li> <li>&lt;グローバル ラン ザクシ ョン ID&gt;:&lt; 結果 &gt; 7</li> </ul>
10	0x9412	B	<ul style="list-style-type: none"> <li>&lt;ステータスファイ ル名 &gt;</li> <li>&lt;ステータスファイ ル名 &gt;:&lt;エントリ番 号 &gt; 8</li> </ul>	<ul style="list-style-type: none"> <li>&gt; write(&lt;書き込む内容 &gt; 9)</li> <li>&gt; read(&lt;読み込む内容 &gt; 10)</li> </ul>	<ul style="list-style-type: none"> <li>なし</li> <li>&lt;グローバル ラン ザクシ ョン ID&gt;</li> </ul>
11	0x9413	B	<ul style="list-style-type: none"> <li>&lt;ステータスファイ ル名 &gt;</li> <li>&lt;ステータスファイ ル名 &gt;:&lt;エントリ番 号 &gt; 8</li> </ul>	正常リターンの場合 <ul style="list-style-type: none"> <li>&lt; write</li> <li>&lt; read</li> </ul> 異常リターン（想定外の値 がリターンされた場合、ま たは例外が発生した場合） <ul style="list-style-type: none"> <li>&lt;!write</li> <li>&lt;!read</li> </ul>	<ul style="list-style-type: none"> <li>なし</li> <li>&lt;結果 &gt; 11</li> <li>&lt;グローバル ラン ザクシ ョン ID&gt;</li> <li>&lt;グローバル ラン ザクシ ョン ID&gt;:&lt; 結果 &gt; 11</li> </ul>

(凡例) A:標準 B:詳細

注 1

図 A-36 ~ A-40 中の番号と対応しています。

注 2

遷移した理由として、次のどれかが出力されます。

- operation  
インプロセス OTS 外部から指示されました。
- server call

別ノードにあるサーバへの呼び出しに失敗しました。

- superior  
別ノードからトランザクション処理への参加指示を受けましたが、すでにそのトランザクションが MarkedRollback 状態でした。
- sync before  
決着処理中の JTA へのコールバック処理に失敗しました。

注 3

遷移した理由として、次のどれかが出力されます。

- operation  
インプロセス OTS 外部から指示されました。
- timeout  
グローバルトランザクションタイムアウトが発生しました。
- superior  
スペリアトランザクションから指示されました。
- forgotten  
サブオーディネートトランザクション、または `javax.transaction.xa.XAResource` に対して決着指示を出すトランザクションが存在しないと判断しました。
- end  
`javax.transaction.xa.XAResource` に対する end に失敗しました。
- prepare  
`javax.transaction.xa.XAResource` に対する prepare に失敗しました。
- write prepared  
ステータスファイルへの prepared 書き込みに失敗しました。
- write committing  
ステータスファイルへの committing 書き込みに失敗しました。

注 4

prepare, commit, rollback または forget の場合、結果として次のどれかが出力されます。

- リターン値
- XAException のエラーコード
- 例外 (=XAException 以外の例外) の toString()

recover の場合、結果として次のどれかが出力されます。

- 回復された Xid の個数
- null ( Xid 配列自体が null の場合 )
- XAException のエラーコード
- 例外 (=XAException 以外の例外) の toString()

注 5

prepare の場合、結果として次のどちらかが出力されます。

- リターンされた値
- 例外の toString()

prepare 以外の場合、結果として例外の toString() が出力されます。

注 6

prepare の場合、結果として次のどちらかが出力されます。

- リターンする値
- 例外の toString()

prepare 以外の場合、結果として例外の toString() が出力されます。

注 7

結果として次のどちらかが出力されます。

- 例外の toString()
- null (リターン値が null の場合)

注 8

内部情報です。

注 9

書き込み内容として、次のどれかが出力されます。

- management info  
ステータスファイル管理情報
- status file body  
ステータスファイル本体
- prepared  
ブリペア完了状態
- committing  
コミット決定状態
- heuristic commit  
強制コミットされた状態
- heuristic rollback  
強制ロールバックされた状態
- heuristic mixed  
部分的にコミットおよびロールバックされた状態
- heuristic hazard  
コミットしたかロールバックしたか不明な状態
- forgotten  
トランザクション決着完了状態

注 10

読み込む内容として、次のどちらかが出力されます。

- management info  
ステータスファイル管理情報

- status file body  
ステータスファイル本体

注 11

結果として、次のどれかが出力されます。

- 書き込みサイズ (単位: バイト)
- 読み込みサイズ (単位: バイト)
- 例外の toString()

## 付録 A.13 標準出力 / 標準エラー出力 / ユーザログのトレース取得ポイント

ここでは、標準出力 / 標準エラー出力 / ユーザログのトレース取得ポイントと、取得できるトレース情報について説明します。

### (1) 標準出力 / 標準エラー出力のトレース取得ポイント

トレース取得ポイントと PRF トレース取得レベル

イベント ID、トレース取得ポイント、および PRF トレース取得レベルについて、次の表に示します。

表 A-60 標準出力 / 標準エラー出力でのトレース取得ポイントの詳細

イベント ID	図中の番号	トレース取得ポイント	レベル
0x9C00	1	標準出力 / 標準エラー出力への出力開始時	B
0x9C01	2	標準出力 / 標準エラー出力への出力完了時	B

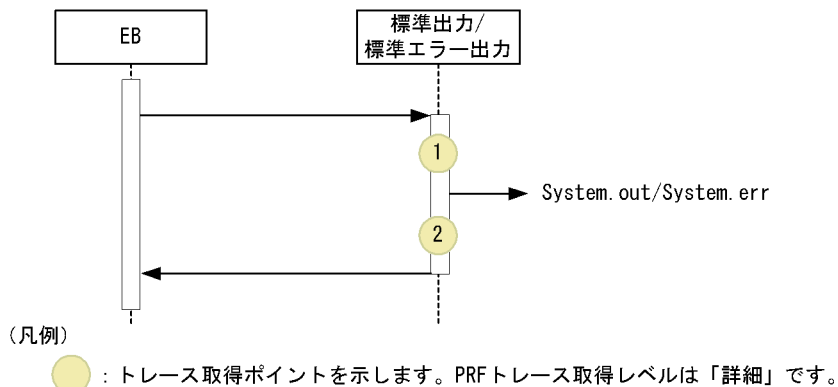
(凡例) B: 詳細

注

図 A-41 中の番号と対応しています。

標準出力 / 標準エラー出力のトレース取得ポイントを次の図に示します。

図 A-41 標準出力 / 標準エラー出力のトレース取得ポイント



取得できるトレース情報

標準出力 / 標準エラー出力で取得できるトレース情報を次の表に示します。

表 A-61 標準出力 / 標準エラー出力で取得できるトレース情報

図中の番号	イベント ID	レベル	取得できる情報		
			インタフェース名	オペレーション名	オプション
1	0x9C00	B	ストリーム名 (「out」 / 「err」)	-	-
2	0x9C01	B	ストリーム名 (「out」 / 「err」)	-	-

(凡例) B: 詳細 - : 該当なし

注

図 A-41 中の番号と対応しています。

(2) ユーザログのトレース取得ポイント

トレース取得ポイントと PRF トレース取得レベル

イベント ID, トレース取得ポイント, および PRF トレース取得レベルについて, 次の表に示します。

表 A-62 ユーザログのトレース取得ポイントの詳細

イベント ID	図中の番号	トレース取得ポイント	レベル
0x9C02	1	ユーザログを使用したメッセージ出力開始時	B
0x9C03	2	ユーザログを使用したメッセージ出力完了時	B

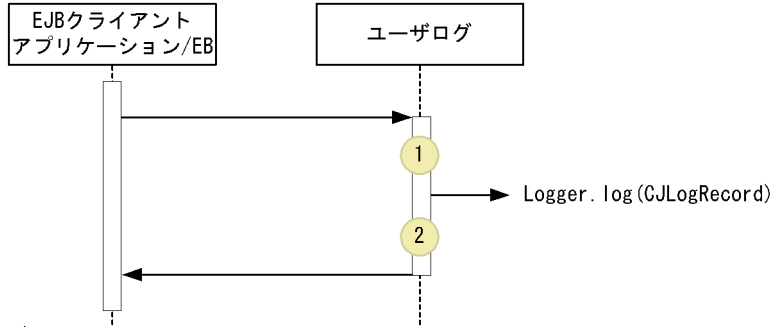
(凡例) B: 詳細

注

図 A-42 中の番号と対応しています。

ユーザログのトレース取得ポイントを次の図に示します。

図 A-42 ユーザログのトレース取得ポイント



(凡例)

● : トレース取得ポイントを示します。PRF トレース取得レベルは「詳細」です。

### 取得できるトレース情報

ユーザログで取得できるトレース情報を次の表に示します。

表 A-63 ユーザログで取得できるトレース情報

図中の 番号 1	イベント ID	レベル	取得できる情報		
			インタフェース名	オペレーション名	オプション
1	0x9C02	B	アプリケーション出力 名	メッセージ ID	-
2	0x9C03	B	アプリケーション出力 名	メッセージ ID	-

(凡例) B: 詳細 - : 該当なし

注

図 A-42 中の番号と対応しています。

## 付録 A.14 DI のトレース取得ポイント

ここでは、DI のトレース取得ポイントと、取得できるトレース情報について説明します。

### PRF トレース取得レベル

イベント ID、トレース取得ポイント、および PRF トレース取得レベルについて、次の

表に示します。

表 A-64 DI でのトレース取得ポイントの詳細

イベント ID	図中の番号	トレース取得ポイント	レベル
0x9D00	1	依存性を注入する直前	A
0x9D01	2	依存性を注入した直後	A

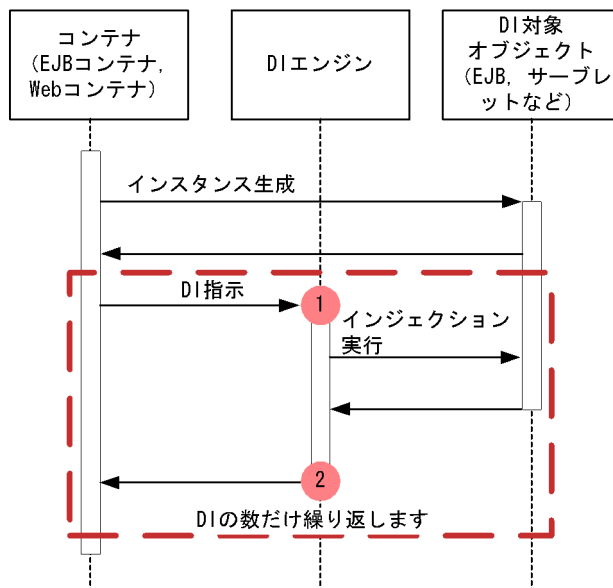
(凡例) A: 標準

注

図 A-43 中の番号と対応しています。

DI のトレース取得ポイントを次の図に示します。

図 A-43 DI のトレース取得ポイント



(凡例)

● : トレース取得ポイントを示します。PRFトレース取得レベルは「標準」です。

取得できるトレース情報

DI で取得できるトレース情報を次の表に示します。



表 A-65 DI で取得できるトレース情報

図中の 番号 1	イベント ID	レベ ル	取得できる情報		
			インタフェース名	オペレーション名	オプション
1	0x9D00	A	依存性を注入する先の ターゲット名	注入するリファレンス名	-
2	0x9D01	A	依存性を注入した先の ターゲット名	注入したリファレンス名	2

(凡例) A:標準 - :該当なし

注 1

図 A-43 中の番号と対応しています。

注 2

正常に処理された場合、入り口時刻が表示されます。

例外が発生した場合、入り口時刻および例外が表示されます。

## 付録 A.15 バッチアプリケーション実行機能のトレース取得 ポイント

ここでは、バッチアプリケーション実行機能のトレース取得ポイントと、取得できるトレース情報について説明します。

### PRF トレース取得レベル

イベント ID、トレース取得ポイント、および PRF トレース取得レベルについて、次の表に示します。

表 A-66 バッチアプリケーション実行機能でのトレース取得ポイントの詳細

イベント ID	図中の番 号	トレース取得ポイント	レベル
0xA100	1	バッチアプリケーションの実行直前	A
0xA101	2	バッチアプリケーションの終了直後	A

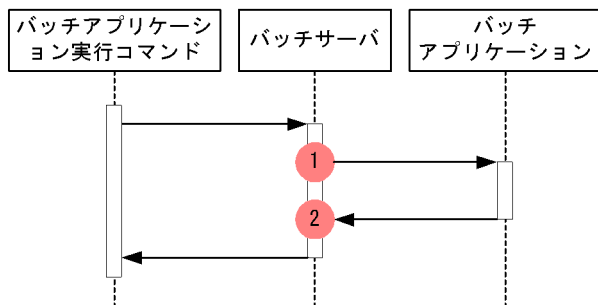
(凡例) A:標準

注

図 A-44 中の番号と対応しています。

バッチアプリケーション実行機能のトレース取得ポイントを次の図に示します。

図 A-44 バッチアプリケーション実行機能のトレース取得ポイント



(凡例) ● : トレース取得ポイントを示します。PRFトレース取得レベルは「標準」です。

取得できるトレース情報

バッチアプリケーション実行機能で取得できるトレース情報を次の表に示します。

表 A-67 バッチアプリケーション実行機能で取得できるトレース情報

図中の 番号 1	イベント ID	レベ ル	取得できる情報		
			インタフェース名	オペレーション名	オプション
1	0xA100	A	バッチアプリケーションのクラス名 (パッケージ名を含む)	-	-
2	0xA101	A	バッチアプリケーションのクラス名 (パッケージ名を含む)	-	2

(凡例) A: 標準 - : 該当なし

注 1

図 A-44 中の番号と対応しています。

注 2

正常に処理された場合、入り口時刻が表示されます。  
例外が発生した場合、入り口時刻および例外が表示されます。

## 付録 A.16 J2EE サーバの開始・終了時のトレース取得ポイント

J2EE サーバの開始処理の完了時、および J2EE サーバの終了処理の開始時に、トレース情報を取得できます。

トレース取得ポイントと PRF トレース取得レベル

イベント ID、トレース取得ポイント、および PRF トレース取得レベルについて、次の

表に示します。

表 A-68 J2EE サーバでのトレース取得ポイントの詳細

イベント ID	トレース取得ポイント	レベル
0x8FFE	J2EE サーバの開始処理完了 (KDJE90028-I メッセージ出力) 時	A
0x8FFF	J2EE サーバのシャットダウン開始 (KDJE90031-I メッセージ出力) 時	A

(凡例) A: 標準

取得できるトレース情報

J2EE サーバの開始・終了時に取得できるトレース情報を次に示します。

- イベント ID  
0x8FFE, 0x8FFF
- PRF トレース取得レベル  
すべて「標準」になります。
- インタフェース名, オペレーション名, およびオプション  
情報は出力されません。

---

## 付録 B snapshot ログの収集対象一覧

ここでは、snapshot ログの一括収集を実行する場合に収集対象となる構成ソフトウェアの実行環境ディレクトリについて説明します。なお、収集した ZIP ファイルを展開すると、収集元と同じディレクトリ構成で展開されます。

snapshot ログの一括収集時、Cosminexus で構築したシステムの構成ソフトウェアおよびライブラリのログファイルおよび定義ファイルが収集されます。収集対象になるディレクトリパスは、次のファイルを編集すると変更できます。snapshot ログ取得の設定については、マニュアル「Cosminexus システム構築ガイド」を参照してください。

snapshot ログは、snapshotlog.conf と snapshotlog.2.conf という二つのファイルに分けて収集されます。

### snapshotlog.conf

一次送付資料の収集対象ディレクトリが記載されているファイルです。ファイルの格納場所を次に示します。

- Windows の場合  
    <Cosminexus のインストールディレクトリ >%manager%\config\snapshotlog.conf
- UNIX の場合  
    /opt/Cosminexus/manager/config/snapshotlog.conf

### snapshotlog.2.conf

二次送付資料の収集対象ディレクトリが記載されているファイルです。ファイルの格納場所を次に示します。

- Windows の場合  
    <Cosminexus のインストールディレクトリ  
    >%manager%\config\snapshotlog.2.conf
- UNIX の場合  
    /opt/Cosminexus/manager/config/snapshotlog.2.conf

ここでは、デフォルトの状態でのそれぞれの構成ソフトウェアおよびライブラリの収集対象ディレクトリを示します。

収集対象ディレクトリパスの共通の記述方法を次に示します。

- ディレクトリパス中の「\*」は 0 文字以上の任意の文字列を示します。
- Windows での「%\*%\*%\*」や UNIX での「/\*/\*/\*」のような記述は、何階層下のファイルを収集するかを示します。例えば、Windows での「log%\*%\*」や UNIX での「log/\*/\*」の場合は、log ディレクトリの 2 階層下のファイルが収集対象になります。

**!** 注意事項

snapshot ログの一括収集は、デフォルトの状態では構成ソフトウェアのインストール時にデフォルトで作成されたディレクトリを対象に実行します。ログの出力先および作業ディレクトリを変更している場合は、収集先をカスタマイズしてください。

## 付録 B.1 Cosminexus Component Container

Cosminexus Component Container に関連する収集対象ディレクトリパスを次の表に示します。

表 B-1 Cosminexus Component Container に関連する収集対象ディレクトリパス  
(Windows の場合)

分類	収集対象ディレクトリパス	一次 / 二次	収集方法
J2EE サーバ (J2EE サーバモードで実行している場合)	<Cosminexus のインストールディレクトリ >%CC%\server\usrconf\%ejb%\<サーバ名称>*.cfg	一次	
	<Cosminexus のインストールディレクトリ >%CC%\server\usrconf\%ejb%\<サーバ名称>*.properties	一次	
	<Cosminexus のインストールディレクトリ >%CC%\server\usrconf\%ejb%\<サーバ名称>*.policy	一次	
	<J2EE サーバ作業ディレクトリ (ejb.public.directory)>%ejb%\<サーバ名称>%javacore*	一次	
	<J2EE サーバ作業ディレクトリ (ejb.public.directory)>%ejb%\<サーバ名称>%hs_err_*	一次	
	<J2EE サーバ作業ディレクトリ (ejb.public.directory)>%ejb%\<サーバ名称>%** <sup>1</sup>	二次	×
	<J2EE サーバ作業ディレクトリ (ejb.public.directory)>%web%\<サーバ名称>%** <sup>1</sup>	二次	×
	<JSP 用一時ディレクトリ (webserver.work.directory)>%** <sup>1</sup>	二次	×
	<J2EE サーバログ出力ディレクトリ (ejb.server.log.directory)>%*	一次	
	<J2EE サーバログ出力ディレクトリ (ejb.server.log.directory)>%*%*	一次	
<J2EE サーバログ出力ディレクトリ (ejb.server.log.directory)>%*%*%*	一次		

分類	収集対象ディレクトリパス	一次 / 二次	収集方法
	<J2EE サーバログ出力ディレクトリ (ejb.server.log.directory)>¥*¥*¥*¥*	二次	
	<インプロセス HTTP サーバのアクセスログファイル名 (webserver.logger.access_log.inprocess_http.filename)>¥*.log	一次	
	<J2EE サーバログ出力ディレクトリ (ejb.server.log.directory)>¥http¥maintenance¥thr¥*	一次	
	<J2EE サーバログ出力ディレクトリ (ejb.server.log.directory)>¥http¥maintenance¥comm¥*	一次	
	<稼働情報ファイル出力先ディレクトリ (ejbserver.management.stats_file.dir)>¥*	二次	
Web コンテナサーバ (サブプレットエンジンモードで実行している場合)	<Web コンテナサーバログ出力ディレクトリ (web.server.log.directory)>¥*	一次	
	<Web コンテナサーバログ出力ディレクトリ (web.server.log.directory)>¥*¥*	一次	
	<Web コンテナサーバログ出力ディレクトリ (web.server.log.directory)>¥*¥*¥*	一次	
	<Web コンテナサーバログ出力ディレクトリ (web.server.log.directory)>¥*¥*¥*¥*	一次	
	<Cosminexus のインストールディレクトリ >¥CC¥web¥containers¥<サーバ名称 >¥usrconf¥*	一次	
	<Cosminexus のインストールディレクトリ >¥CC¥web¥containers¥<サーバ名称 >¥javacore*	一次	
	<Cosminexus のインストールディレクトリ >¥CC¥web¥containers¥<サーバ名称 >¥hs_err*	一次	
	<Cosminexus のインストールディレクトリ >¥CC¥web¥containers¥<サーバ名称 >¥core*	二次	
サーバ管理コマンド <sup>2</sup>	<Cosminexus のインストールディレクトリ >¥CC¥admin¥usrconf¥*	一次	
	<Cosminexus のインストールディレクトリ >¥CC¥admin¥logs¥*	一次	
	<Cosminexus のインストールディレクトリ >¥CC¥admin¥logs¥*¥*	一次	
	<Cosminexus のインストールディレクトリ >¥CC¥admin¥logs¥*¥*¥*	一次	

分類	収集対象ディレクトリパス	一次 / 二次	収集方法
	<Cosminexus のインストールディレクトリ >%CC¥admin¥logs¥*¥*¥*¥*	二次	
リダイレクタ (Web サーバ)	<Cosminexus のインストールディレクトリ >%CC¥web¥redirector¥logs¥*	一次	
	<Cosminexus のインストールディレクトリ >%CC¥web¥redirector¥*.conf	一次	
	<Cosminexus のインストールディレクトリ >%CC¥web¥redirector¥*.properties	一次	
	<リダイレクタログ出力ディレクトリ (JkLogFileDir)>¥*	一次	
	<リダイレクタログ出力ディレクトリ (JkLogFileDir)>¥*¥*	一次	
	<リダイレクタログ出力ディレクトリ (JkTraceLogFileDir)>¥*	一次	
	<リダイレクタログ出力ディレクトリ (JkTraceLogFileDir)>¥*¥*	一次	
	<Cosminexus のインストールディレクトリ >%CC¥web¥redirector¥servers¥<Web サーバ名 >¥*.conf	一次	
	<Cosminexus のインストールディレクトリ >%CC¥web¥redirector¥servers¥<Web サーバ名 >¥*.properties	一次	
JCA/JDBC コネクション管理	<J2EE サーバログ出力ディレクトリ (ejb.server.log.directory)>¥*	一次	
	<J2EE サーバログ出力ディレクトリ (ejb.server.log.directory)>¥*¥*	一次	
TP1/ Message Queue - Access	<J2EE サーバログ出力ディレクトリ (ejb.server.log.directory)>¥*	一次	
	<J2EE サーバログ出力ディレクトリ (ejb.server.log.directory)>¥*¥*	一次	
	<J2EE サーバ作業ディレクトリ (ejb.public.directory)>¥ejb¥<サーバ名称 >mqc.api.*.*	一次	
uCosminexus TP1 Connector (TP1/Client/ J)	<J2EE サーバログ出力ディレクトリ (ejb.server.log.directory)>¥*¥*	一次	
	<J2EE サーバログ出力ディレクトリ (ejb.server.log.directory)>¥*¥*¥*	一次	
	¥{jp.co.hitachi_system.tp1connector.logdestination}¥tp1connector*.log	一次	

分類	収集対象ディレクトリパス	一次 / 二次	収集方法
	<code>\${user.home}\¥tp1connector*.log</code>	一次	
	<code>\${user.home}\¥TP1clientJ¥dcClit*.dmp</code>	二次	
Cosminexus RM	<code>&lt;J2EE サーバログ出力ディレクトリ (ejb.server.log.directory)&gt;¥*</code>	一次	
	<code>&lt;J2EE サーバログ出力ディレクトリ (ejb.server.log.directory)&gt;¥*¥*</code>	一次	
	<code>&lt;J2EE サーバログ出力ディレクトリ (ejb.server.log.directory)&gt;¥*¥*¥*</code>	一次	
	<code>&lt;Cosminexus RM インストールディレクトリ &gt;¥logs¥cmd¥*</code>	一次	
	<code>&lt;Cosminexus RM インストールディレクトリ &gt;¥logs¥cmd¥*¥*¥*</code>	一次	
	<code>&lt;Cosminexus RM インストールディレクトリ &gt;¥logs¥cmd¥*¥*¥*¥*</code>	一次	
	<code>&lt;Cosminexus RM インストールディレクトリ &gt;¥logs¥cmd¥*¥*¥*¥*¥*</code>	一次	
	<code>&lt;Cosminexus インストールディレクトリ (cosminexus.home)&gt;¥RM¥logs¥cmd¥*</code>	一次	
	<code>&lt;Cosminexus インストールディレクトリ (cosminexus.home)&gt;¥RM¥ogs¥cmd¥*¥*¥*</code>	一次	
	<code>&lt;Cosminexus インストールディレクトリ (cosminexus.home)&gt;¥RM/ logs¥cmd¥*¥*¥*¥*</code>	一次	
	<code>&lt;Cosminexus インストールディレクトリ (cosminexus.home)&gt;¥RM¥ogs¥cmd¥*¥*¥*¥*¥*</code>	一次	
<code>&lt;ユーザ指定ディレクトリ (RMLogTraceFileDir)&gt;¥*</code>	一次		
EJB クライアント	<code>&lt;EJB0 クライアントログ出力ディレクトリ (ejbserver.client.log.directory)&gt;¥*</code>	一次	
	<code>&lt;EJB クライアントログ出力ディレクトリ (ejbserver.client.log.directory)&gt;¥*¥*¥*</code>	一次	
	<code>&lt;EJB クライアントログ出力ディレクトリ (ejbserver.client.log.directory)&gt;¥*¥*¥*¥*</code> <sup>3</sup>	一次	
	<code>&lt;EJB クライアントログ出力ディレクトリ (ejbserver.client.log.directory)&gt;¥*¥*¥*¥*¥*</code> <sup>3</sup>	二次	
Management Server	<code>&lt;Cosminexus のインストールディレクトリ &gt;¥manager¥config¥*.bak</code>	一次	



分類	収集対象ディレクトリパス	一 次 / 二 次	収 集 方 法
	<Cosminexus のインストールディレクトリ >¥manager¥config¥*.cfg	一 次	
	<Cosminexus のインストールディレクトリ >¥manager¥config¥*.conf	一 次	
	<Cosminexus のインストールディレクトリ >¥manager¥config¥*.xml	一 次	
	<Cosminexus のインストールディレクトリ >¥manager¥config¥*.properties	一 次	
	<Cosminexus のインストールディレクトリ >¥manager¥config¥*.server_name	一 次	
	<Cosminexus のインストールディレクトリ >¥manager¥config¥filterlist	一 次	
	<Cosminexus のインストールディレクトリ >¥manager¥config¥.+	二 次	
	<Cosminexus のインストールディレクトリ >¥manager¥tmp¥* <sup>4</sup>	一 次	
	<Cosminexus のインストールディレクトリ >¥manager¥tmp¥*.zip	二 次	
	<Manager ログ出力ディレクトリ (com.cosminexus.manager.log.dir)>¥*	一 次	
	<Manager ログ出力ディレクトリ (com.cosminexus.manager.log.dir)>¥message¥*	一 次	
	<Manager ログ出力ディレクトリ (com.cosminexus.manager.log.dir)>¥message¥mmap¥*	一 次	
	<Manager ログ出力ディレクトリ (com.cosminexus.manager.log.dir)>¥trace¥*	一 次	
	<Manager ログ出力ディレクトリ (com.cosminexus.manager.log.dir)>¥trace¥mmap¥*	一 次	
	<Manager ログ出力ディレクトリ (com.cosminexus.manager.log.dir)>¥maintenance¥*	二 次	
	<Manager ログ出力ディレクトリ (com.cosminexus.manager.log.dir)>¥maintenance¥mmap¥*	二 次	
	<HNTRLib2 のインストールディレクトリ >¥spool¥hnr2*.log	一 次	
	< システム環境変数 %SystemRoot%>¥system32¥drivers¥etc¥hosts	一 次	
	<Cosminexus のインストールディレクトリ >¥manager¥setup¥config¥*	一 次	

分類	収集対象ディレクトリパス	一次 / 二次	収集方法
	<Cosminexus のインストールディレクトリ >¥manager¥setup¥log¥*.log	一次	
	<Cosminexus のインストールディレクトリ >¥manager¥setup¥log¥maintenance¥*.log	一次	
	<Cosminexus のインストールディレクトリ >¥manager¥setup¥tmp¥*	一次	
Cosminexus Service Coordinator 5	<J2EE サーバログ出力ディレクトリ (ejb.server.log.directory)>¥csc¥*	一次	
	<J2EE サーバログ出力ディレクトリ (ejb.server.log.directory)>¥csc¥maintenance¥*	一次	
	<HCSC サーバのプロパティ (requesttrace-filepath)>¥*	一次	
	<HCSC サーバのプロパティ (methodtrace-filepath)>¥*	一次	
	<Cosminexus のインストールディレクトリ >¥CSC¥system¥msg¥*	一次	
	<J2EE サーバ作業ディレクトリ (ejb.public.directory)>¥ejb¥< サーバ名称 >¥apps¥*	二次	
	<J2EE サーバ作業ディレクトリ (ejb.public.directory)>¥ejb¥< サーバ名称 >¥apps¥*¥*	二次	
	<J2EE サーバ作業ディレクトリ (ejb.public.directory)>¥ejb¥< サーバ名称 >¥apps¥*¥*¥*	二次	
	<J2EE サーバ作業ディレクトリ (ejb.public.directory)>¥ejb¥< サーバ名称 >¥apps¥*¥*¥*¥*	二次	
	<Cosminexus のインストールディレクトリ >¥CSC¥config¥manager¥*	一次	
	<リポジトリルート (cscmng.repository.root)>¥**	二次	×
<ログディレクトリ (cscmng.log.dir)>¥message*¥*	一次	×	
Server Plug-in	<Eclipse ワークスペースディレクトリ >¥.metadata¥.plugins¥com.cosminexus.plugin.server.explorer¥.logs¥*.log	一次	×
	<Eclipse ワークスペースディレクトリ >¥.metadata¥.plugins¥com.cosminexus.plugin.server.core¥.logs¥*.log	一次	×
	<Eclipse ワークスペースディレクトリ >¥.metadata¥.log	二次	×

分類	収集対象ディレクトリパス	一次 / 二次	収集方法
バッチアプリケーション実行コマンド	<Cosminexus のインストールディレクトリ >¥CC¥batch¥logs¥*	一次	
	<Cosminexus のインストールディレクトリ >¥CC¥batch¥logs¥*¥*	一次	
	<Cosminexus のインストールディレクトリ >¥CC¥batch¥logs¥*¥*¥*	一次	
	<Cosminexus のインストールディレクトリ >¥CC¥batch¥logs¥*¥*¥*¥*	一次	
SOAP	<Cosminexus のインストールディレクトリ >¥c4web¥logs¥*	一次	
	<Cosminexus のインストールディレクトリ >¥c4web¥conf¥*	一次	
	<Cosminexus のインストールディレクトリ >¥CC¥server¥public¥web¥< サーバ名称 >¥< コンテキストルート >¥WEB-INF¥server-config.xml	一次	
	<Cosminexus のインストールディレクトリ >¥CC¥server¥public¥web¥< サーバ名称 >¥< コンテキストルート >¥WEB-INF¥classes¥c4webcl.properties	一次	
	<Cosminexus のインストールディレクトリ >¥CC¥web¥containers¥< サーバ名称 >¥webapps¥< コンテキストルート >¥WEB-INF¥server-config.xml	一次	
	<Cosminexus のインストールディレクトリ >¥CC¥web¥containers¥< サーバ名称 >¥webapps¥< コンテキストルート >¥WEB-INF¥classes¥c4webcl.properties	一次	
	<J2EE サーバ作業ディレクトリ (ejb.public.directory)>¥ejb¥< サーバ名称 >¥cjmessage*	一次	
	<J2EE サーバ作業ディレクトリ (ejb.public.directory)>¥ejb¥< サーバ名称 >¥cjexception*	一次	
	<J2EE サーバ作業ディレクトリ (ejb.public.directory)>¥ejb¥< サーバ名称 >¥WS¥*	一次	
	<J2EE サーバ作業ディレクトリ (ejb.public.directory)>¥ejb¥< サーバ名称 >¥WS¥maintenance¥*	二次	
	<EJB クライアントログ出力ディレクトリ (ejbserver.client.log.directory)>¥*¥*¥*	一次	
	<Cosminexus のインストールディレクトリ >¥CC¥web¥containers¥< サーバ名称 >¥logs¥cjmessage*	一次	
	<Cosminexus のインストールディレクトリ >¥CC¥web¥containers¥< サーバ名称 >¥logs¥cjexception*	一次	

分類	収集対象ディレクトリパス	一 次 / 二 次	収 集 方 法
	<Cosminexus のインストールディレクトリ>¥CC¥web¥containers¥<サーバ名称>¥logs¥WS¥*	一 次	
	<Cosminexus のインストールディレクトリ>¥CC¥web¥containers¥<サーバ名称>¥logs¥WS¥maintenance¥*	二 次	

(凡例)

：デフォルトで収集されます。出力ディレクトリを変更した場合も、収集対象になります。

：デフォルトで収集されます。ただし、出力ディレクトリを設定を変更した場合、snapshot ログとして収集するには、snapshot ログ収集対象定義ファイル (snapshotlog.conf, または snapshotlog.2.conf) を編集して、変更後の出力ディレクトリを収集対象ディレクトリとして設定し直す必要があります。

×：デフォルトでは収集されません。snapshot ログとして収集するには、snapshot ログ収集対象定義ファイル (snapshotlog.conf, または snapshotlog.2.conf) を編集して、収集対象ディレクトリに設定する必要があります。

注 表内のそれぞれのディレクトリの意味は次のとおりです。

- <Cosminexus のインストールディレクトリ>  
Cosminexus のインストールディレクトリです。デフォルトは、「C:¥Program Files¥HITACHI¥Cosminexus」です。
- <J2EE サーバ作業ディレクトリ (ejb.public.directory)>  
簡易構築定義ファイルの論理 J2EE サーバ (j2ee-server) の <configuration> タグ内に、ejb.public.directory パラメタで指定したディレクトリです。
- <JSP 用一時ディレクトリ (webserver.work.directory)>  
簡易構築定義ファイルの論理 J2EE サーバ (j2ee-server) の <configuration> タグ内に、webserver.work.directory パラメタで指定したディレクトリです。
- <J2EE サーバログ出力ディレクトリ (ejb.server.log.directory)>  
簡易構築定義ファイルの論理 J2EE サーバ (j2ee-server) の <configuration> タグ内に、ejb.server.log.directory パラメタで指定したディレクトリです。
- <インプロセス HTTP サーバのアクセスログファイル名 (webserver.logger.access\_log.inprocess\_http.filename)>  
簡易構築定義ファイルの論理 J2EE サーバ (j2ee-server) の <configuration> タグ内に、webserver.logger.access\_log.inprocess\_http.filename パラメタで指定したディレクトリです。
- <稼働情報ファイル出力先ディレクトリ (ejbserver.management.stats\_file.dir)>  
簡易構築定義ファイルの論理 J2EE サーバ (j2ee-server) または論理 SFO サーバ (sfo-server) の <configuration> タグ内に、ejbserver.management.stats\_file.dir パラメタで指定したディレクトリです。
- <Web コンテナサーバログ出力ディレクトリ (web.server.log.directory)>  
J2EE サーバ用の usrconf.cfg の web.server.log.directory キーに指定したディレクトリです。
- <リダイレクタログ出力ディレクトリ (JkLogFileDir)>  
簡易構築定義ファイルの論理 Web サーバ (web-server) の <configuration> タグ内に、JkLogFileDir パラメタで指定したディレクトリです。

- <リダイレクタログ出力ディレクトリ (JkTraceLogFileDir)>  
簡易構築定義ファイルの論理 Web サーバ (web-server) の <configuration> タグ内に、  
JkTraceLogFileDir パラメタで指定したディレクトリです。
- <Cosminexus RM インストールディレクトリ>  
環境変数 HRMDIR に指定したディレクトリです。
- <EJB クライアントログ出力ディレクトリ (ejbserver.client.log.directory)>  
EJB クライアントアプリケーションのシステムプロパティ `ejbserver.client.log.directory` キーに  
指定したディレクトリ、または `ejclstartap` コマンドのカレントディレクトリです。
- <Manager ログ出力ディレクトリ (com.cosminexus.manager.log.dir)>  
`manager.cfg` の `com.cosminexus.manager.log.dir` キーに指定したディレクトリです。
- <HNTRLib2 のインストールディレクトリ>  
日立トレース共通ライブラリのインストールディレクトリです。デフォルトは、「`C:\Program  
Files\Hitachi\HNTRLib2`」です。
- <HCSC サーバのプロパティ (requesttrace-filepath)>  
HCSC サーバ定義コマンド (`cscsvconfig`) で `requesttrace-filepath` プロパティに指定したディ  
レクトリです。
- <HCSC サーバのプロパティ (methodtrace-filepath)>  
HCSC サーバ定義コマンド `cscsvconfig` で `methodtrace-filepath` プロパティに指定したディレク  
トリーです。
- <リポジトリルート (cscmng.repository.root)>  
HCSC-Manager 定義 (`cscmng.properties`) の `cscmng.repository.root` キーに指定したリポジト  
リルートです。
- <ログディレクトリ (cscmng.log.dir)>  
HCSC-Manager 定義 (`cscmng.properties`) の `cscmng.log.dir` キーに指定したログディレクトリ  
です。
- <Eclipse ワークスペースディレクトリ>  
Eclipse を利用するユーザが任意で作成するディレクトリです。

注 1

21 階層分までのディレクトリを収集対象にします。

注 2

環境に合わせて収集対象になるディレクトリパスを変更してから収集してください。

注 3

Cosminexus TPBroker で環境変数が設定されている場合は、環境変数の設定が優先されます。

注 4

\*.zip 以外のファイルが収集されます。

注 5

uCosminexus Service Platform の場合だけ出力されます。

表 B-2 Cosminexus Component Container に関連する収集対象ディレクトリパス (UNIX の場合)

分類	収集対象ディレクトリパス	一次 / 二次	収集方法
J2EE サーバ (J2EE サーバ モードで 実行して いる場合)	/opt/Cosminexus/CC/server/usrconf/ejb/<サーバ名称>/*.cfg	一次	
	/opt/Cosminexus/CC/server/usrconf/ejb/<サーバ名称>/*.properties	一次	
	/opt/Cosminexus/CC/server/usrconf/ejb/<サーバ名称>/*.policy	一次	
	<J2EE サーバ作業ディレクトリ (ejb.public.directory)>/ejb/<サーバ名称>/javacore*	一次	
	<J2EE サーバ作業ディレクトリ (ejb.public.directory)>/ejb/<サーバ名称>/hs_err_*	一次	
	<J2EE サーバ作業ディレクトリ (ejb.public.directory)>/ejb/<サーバ名称>/core*	二次	
	<J2EE サーバ作業ディレクトリ (ejb.public.directory)>/ejb/<サーバ名称>/** 1	二次	×
	<J2EE サーバ作業ディレクトリ (ejb.public.directory)>/web/<サーバ名称>/** 1	二次	×
	<JSP 用一時ディレクトリ (webserver.work.directory)>/** 1	二次	×
	<J2EE サーバログ出力ディレクトリ (ejb.server.log.directory)>/*	一次	
	<J2EE サーバログ出力ディレクトリ (ejb.server.log.directory)>/*/*	一次	
	<J2EE サーバログ出力ディレクトリ (ejb.server.log.directory)>/*/*/*	一次	
	<J2EE サーバログ出力ディレクトリ (ejb.server.log.directory)>/*/*/*/*	二次	
	<インプロセス HTTP サーバのアクセスログファイル名 (webserver.logger.access_log.inprocess_http.filename)>/*.log	一次	
	<J2EE サーバログ出力ディレクトリ (ejb.server.log.directory)>/http/maintenance/thr/*	一次	
<J2EE サーバログ出力ディレクトリ (ejb.server.log.directory)>/http/maintenance/comm/*	一次		

分類	収集対象ディレクトリパス	一次 / 二次	収集方法
	<稼働情報ファイル出力先ディレクトリ (ejbserver.management.stats_file.dir)>/*	二次	
Web コンテナサーバ(サブプレットエンジンモードで実行している場合)	<Web コンテナサーバログ出力ディレクトリ (web.server.log.directory)>/*	一次	
	<Web コンテナサーバログ出力ディレクトリ (web.server.log.directory)>/*/*	一次	
	<Web コンテナサーバログ出力ディレクトリ (web.server.log.directory)>/*/*/*	一次	
	/opt/Cosminexus/CC/web/containers/<サーバ名称>/usrconf/*	一次	
	/opt/Cosminexus/CC/web/containers/<サーバ名称>/javacore*	一次	
	/opt/Cosminexus/CC/web/containers/<サーバ名称>/hs_err*	一次	
	/opt/Cosminexus/CC/web/containers/<サーバ名称>/core*	二次	
サーバ管理コマンド <sup>2</sup>	/opt/Cosminexus/CC/admin/usrconf/*	一次	
	/opt/Cosminexus/CC/admin/logs/*	一次	
	/opt/Cosminexus/CC/admin/logs/*/*	一次	
	/opt/Cosminexus/CC/admin/logs/*/*/*	一次	
	/opt/Cosminexus/CC/admin/logs/*/*/*/*	二次	
リダイレクタ (Webサーバ)	/opt/Cosminexus/CC/web/redirector/logs/*	一次	
	/opt/Cosminexus/CC/web/redirector/*.conf	一次	

分類	収集対象ディレクトリパス	一次 / 二次	収集方法
	/opt/Cosminexus/CC/web/redirector/*.properties	一次	
	<リダイレクタログ出力ディレクトリ (JkLogFileDir)>/*	一次	
	<リダイレクタログ出力ディレクトリ (JkLogFileDir)>/*/*	一次	
	<リダイレクタログ出力ディレクトリ (JkTraceLogFileDir)>/*	一次	
	<リダイレクタログ出力ディレクトリ (JkTraceLogFileDir)>/*/*	一次	
	/opt/Cosminexus/CC/web/redirector/servers/<Web サーバ名 >/*.conf	一次	
	/opt/Cosminexus/CC/web/redirector/servers/<Web サーバ名 >/*.properties	一次	
JCA/ JDBC コ ネクション 管理	<J2EE サーバログ出力ディレクトリ (ejb.server.log.directory)>/*	一次	
	<J2EE サーバログ出力ディレクトリ (ejb.server.log.directory)>/*/*	一次	
TP1/ Message Queue Access	<J2EE サーバログ出力ディレクトリ (ejb.server.log.directory)>/*	一次	
	<J2EE サーバログ出力ディレクトリ (ejb.server.log.directory)>/*/*	一次	
	<J2EE サーバ作業ディレクトリ (ejb.public.directory)>/ejb/<サーバ名称 >/mqc.api.*.*	一次	
uCosmine xus TP1 Connector (TP1/ Client/J)	<J2EE サーバログ出力ディレクトリ (ejb.server.log.directory)>/*/*	一次	
	<J2EE サーバログ出力ディレクトリ (ejb.server.log.directory)>/*/*/*	一次	
	\${jp.co.hitachi_system.tp1connector.logdestination}/tp1connector*.log	一次	
	\${user.home}/tp1connector*.log	一次	
	\${jp.co.hitachi_system.tp1connector.logdestination}/TP1clientJdcClnt*.dmp	二次	



分類	収集対象ディレクトリパス	一次 / 二次	収集方法
Cosminexus RM	<J2EE サーバログ出力ディレクトリ (ejb.server.log.directory)>/*	一次	
	<J2EE サーバログ出力ディレクトリ (ejb.server.log.directory)>/*/*	一次	
	<J2EE サーバログ出力ディレクトリ (ejb.server.log.directory)>/*/*/*	一次	
	<Cosminexus RM インストールディレクトリ >/logs/cmd/*	一次	
	<Cosminexus RM インストールディレクトリ >/logs/cmd/*/*/*	一次	
	<Cosminexus RM インストールディレクトリ >/logs/cmd/*/*/*/*	一次	
	<Cosminexus RM インストールディレクトリ >/logs/cmd/*/*/*/*/*	一次	
	<Cosminexus インストールディレクトリ (cosminexus.home)>/RM/logs/cmd/*	一次	
	<Cosminexus インストールディレクトリ (cosminexus.home)>/RM/logs/cmd/*/*/*	一次	
	<Cosminexus インストールディレクトリ (cosminexus.home)>/RM/logs/cmd/*/*/*/*	一次	
	<Cosminexus インストールディレクトリ (cosminexus.home)>/RM/logs/cmd/*/*/*/*/*	一次	
	<ユーザ指定ディレクトリ (RMLogTraceFileDir)>/*	一次	
EJB クライアント	<EJB クライアントログ出力ディレクトリ (ejbserver.client.log.directory)>/*	一次	
	<EJB クライアントログ出力ディレクトリ (ejbserver.client.log.directory)>/*/*/*	一次	
	<EJB クライアントログ出力ディレクトリ (ejbserver.client.log.directory)>/*/*/*/* 3	一次	
	<EJB クライアントログ出力ディレクトリ (ejbserver.client.log.directory)>/*/*/*/*/* 3	二次	
Management Server	/opt/Cosminexus/manager/config/* .bak	一次	
	/opt/Cosminexus/manager/config/* .cfg	一次	
	/opt/Cosminexus/manager/config/* .conf	一次	

分類	収集対象ディレクトリパス	一 次 / 二 次	収 集 方 法
	/opt/Cosminexus/manager/config/*.xml	一 次	
	/opt/Cosminexus/manager/config/*.properties	一 次	
	/opt/Cosminexus/manager/config/*.server_name	一 次	
	/opt/Cosminexus/manager/config/filterlist	一 次	
	/opt/Cosminexus/manager/config/.+	二 次	
	/opt/Cosminexus/manager/tmp/* <sup>4</sup>	一 次	
	/opt/Cosminexus/manager/tmp/*.zip	二 次	
	<Manager ログ出力ディレクトリ (com.cosminexus.manager.log.dir)>/*	一 次	
	<Manager ログ出力ディレクトリ (com.cosminexus.manager.log.dir)>/message/*	一 次	
	<Manager ログ出力ディレクトリ (com.cosminexus.manager.log.dir)>/message/mmap/*	一 次	
	<Manager ログ出力ディレクトリ (com.cosminexus.manager.log.dir)>/trace/*	一 次	
	<Manager ログ出力ディレクトリ (com.cosminexus.manager.log.dir)>/trace/mmap/*	一 次	
	<Manager ログ出力ディレクトリ (com.cosminexus.manager.log.dir)>/maintenance/*	二 次	
	<Manager ログ出力ディレクトリ (com.cosminexus.manager.log.dir)>/maintenance/mmap/*	二 次	
	<コアダンプ出力ディレクトリ (CoreDumpDirectory)>/*	二 次	
	<HNTRLib2 のインストールディレクトリ >/spool/hntr2*.log	一 次	
	/var/opt/hitachi/HNTRLib2/spool/*	一 次	
	/etc/hosts	一 次	
	/opt/Cosminexus/manager/setup/config/*	一 次	
	/opt/Cosminexus/manager/setup/log/*.log	一 次	

分類	収集対象ディレクトリパス	一次 / 二次	収集方法
	/opt/Cosminexus/manager/setup/log/maintenance/*.log	一次	
	/opt/Cosminexus/manager/setup/tmp/*	一次	
Cosminexus Service Coordinator <sup>5</sup>	<J2EE サーバログ出力ディレクトリ (ejb.server.log.directory)>/csc/*	一次	
	<J2EE サーバログ出力ディレクトリ (ejb.server.log.directory)>/csc/maintenance/*	一次	
	<HCSC サーバのプロパティ (requesttrace-filepath)>/*	一次	
	<HCSC サーバのプロパティ (methodtrace-filepath)>/*	一次	
	/opt/Cosminexus/CSC/system/msg/*	一次	
	<J2EE サーバ作業ディレクトリ (ejb.public.directory)>/ejb/<サーバ名称>/apps/*	二次	
	<J2EE サーバ作業ディレクトリ (ejb.public.directory)>/ejb/<サーバ名称>/apps/*/*	二次	
	<J2EE サーバ作業ディレクトリ (ejb.public.directory)>/ejb/<サーバ名称>/apps/*/*/*	二次	
	<J2EE サーバ作業ディレクトリ (ejb.public.directory)>/ejb/<サーバ名称>/apps/*/*/*/*	二次	
	/opt/Cosminexus/CSC/config/manager/*	一次	
	<リポジトリルート (cscmng.repository.root)>/**	二次	×
<ログディレクトリ (cscmng.log.dir)>/*	一次		
Server Plug-in	<Eclipse ワークスペースディレクトリ >/.metadata/.plugins/com.cosminexus.plugin.server.explorer/.logs/*.log	一次	×
	<Eclipse ワークスペースディレクトリ >/.metadata/.plugins/com.cosminexus.plugin.server.core/.logs/*.log	一次	×
	<Eclipse ワークスペースディレクトリ >/.metadata/.log	二次	×

分類	収集対象ディレクトリパス	一次 / 二次	収集方法
バッチアプリケーション実行コマンド	/opt/Cosminexus/CC/batch/logs/*	一次	
	/opt/Cosminexus/CC/batch/logs/*/*	一次	
	/opt/Cosminexus/CC/batch/logs/*/*/*	一次	
	/opt/Cosminexus/CC/batch/logs/*/*/*/*	一次	
SOAP	/opt/Cosminexus/c4web/logs/*	一次	
	/opt/Cosminexus/c4web/conf/*	一次	
	/opt/Cosminexus/CC/server/public/web/<サーバ名称>/<コンテキストルート>/WEB-INF/server-config.xml	一次	
	/opt/Cosminexus/CC/server/public/web/<サーバ名称>/<コンテキストルート>/WEB-INF/classes/c4webcl.properties	一次	
	/opt/Cosminexus/CC/web/containers/<サーバ名称>/webapps/<コンテキストルート>/WEB-INF/server-config.xml	一次	
	/opt/Cosminexus/CC/web/containers/<サーバ名称>/webapps/<コンテキストルート>/WEB-INF/classes/c4webcl.properties	一次	
	<J2EE サーバ作業ディレクトリ (ejb.public.directory)>/ejb/<サーバ名称>/cjmessage*	一次	
	<J2EE サーバ作業ディレクトリ (ejb.public.directory)>/ejb/<サーバ名称>/cjexception*	一次	
	<J2EE サーバ作業ディレクトリ (ejb.public.directory)>/ejb/<サーバ名称>/WS/*	一次	
	<J2EE サーバ作業ディレクトリ (ejb.public.directory)>/ejb/<サーバ名称>/WS/maintenance/*	二次	
	<EJB クライアントログ出力ディレクトリ (ejbserver.client.log.directory)>/**/*	一次	
	/opt/Cosminexus/CC/web/containers/<サーバ名称>/logs/cjmessage*	一次	
	/opt/Cosminexus/CC/web/containers/<サーバ名称>/logs/cjexception*	一次	
	/opt/Cosminexus/CC/web/containers/<サーバ名称>/logs/WS/*	一次	

分類	収集対象ディレクトリパス	一 次 / 二 次	収 集 方 法
	/opt/Cosminexus/CC/web/containers/<サーバ名称>/logs/WS/maintenance/*	二 次	

## ( 凡例 )

：デフォルトで収集されます。出力ディレクトリを変更した場合も、収集対象になります。

：デフォルトで収集されます。ただし、出力ディレクトリを設定を変更した場合、snapshot ログとして収集するには、snapshot ログ収集対象定義ファイル ( snapshotlog.conf, または snapshotlog.2.conf ) を編集して、変更後の出力ディレクトリを収集対象ディレクトリとして設定し直す必要があります。

×：デフォルトでは収集されません。snapshot ログとして収集するには、snapshot ログ収集対象定義ファイル ( snapshotlog.conf, または snapshotlog.2.conf ) を編集して、収集対象ディレクトリに設定する必要があります。

注 表内のそれぞれのディレクトリの意味は次のとおりです。

- <J2EE サーバ作業ディレクトリ (ejb.public.directory)>  
簡易構築定義ファイルの論理 J2EE サーバ ( j2ee-server ) の <configuration> タグ内に、  
ejb.public.directory パラメタで指定したディレクトリです。
- <JSP 用一時ディレクトリ (webserver.work.directory)>  
簡易構築定義ファイルの論理 J2EE サーバ ( j2ee-server ) の <configuration> タグ内に、  
webserver.work.directory パラメタで指定したディレクトリです。
- <J2EE サーバログ出力ディレクトリ (ejb.server.log.directory)>  
簡易構築定義ファイルの論理 J2EE サーバ ( j2ee-server ) の <configuration> タグ内に、  
ejb.server.log.directory パラメタで指定したディレクトリです。
- <インプロセス HTTP サーバのアクセスログファイル名  
(webserver.logger.access\_log.inprocess\_http.filename)>  
簡易構築定義ファイルの論理 J2EE サーバ ( j2ee-server ) の <configuration> タグ内に、  
webserver.logger.access\_log.inprocess\_http.filename パラメタで指定したディレクトリです。
- <稼働情報ファイル出力先ディレクトリ (ejbserver.management.stats\_file.dir)>  
簡易構築定義ファイルの論理 J2EE サーバ ( j2ee-server ) または論理 SFO サーバ ( sfo-server )  
の <configuration> タグ内に、ejbserver.management.stats\_file.dir パラメタで指定したディレ  
クトリです。
- <Web コンテナサーバログ出力ディレクトリ (web.server.log.directory)>  
J2EE サーバ用の usrconf.cfg の web.server.log.directory キーに指定したディレクトリです。
- <リダイレクタログ出力ディレクトリ (JkLogFileDir)>  
簡易構築定義ファイルの論理 Web サーバ ( web-server ) の <configuration> タグ内に、  
JkLogFileDir パラメタで指定したディレクトリです。
- <リダイレクタログ出力ディレクトリ (JkTraceLogFileDir)>  
簡易構築定義ファイルの論理 Web サーバ ( web-server ) の <configuration> タグ内に、  
JkTraceLogFileDir パラメタで指定したディレクトリです。
- <Cosminexus RM インストールディレクトリ >  
環境変数 HRMDIR に指定したディレクトリです。
- <EJB クライアントログ出力ディレクトリ (ejbserver.client.log.directory)>

EJB クライアントアプリケーションのシステムプロパティ `ejbserver.client.log.directory` キーに指定したディレクトリ, または `ejelstartap` コマンドのカレントディレクトリです。

- <Manager ログ出力ディレクトリ (`com.cosminexus.manager.log.dir`)>  
`manager.cfg` の `com.cosminexus.manager.log.dir` キーに指定したディレクトリです。
- <コアダンプ出力ディレクトリ (`CoreDumpDirectory`)>  
簡易構築定義ファイルの論理 Web サーバ ( `web-server` ) の <configuration> タグ内に, `CoreDumpDirectory` パラメタで指定したディレクトリです。
- <HNTRLib2 のインストールディレクトリ >  
日立トレース共通ライブラリのインストールディレクトリです。デフォルトは, 「`/opt/Hitachi/HNTRLib2`」です。
- <HCSC サーバのプロパティ (`requesttrace-filepath`)>  
HCSC サーバ定義コマンド (`cscsvconfig`) で `requesttrace-filepath` プロパティに指定したディレクトリです。
- <HCSC サーバのプロパティ (`methodtrace-filepath`)>  
HCSC サーバ定義コマンド `cscsvconfig` で `methodtrace-filepath` プロパティに指定したディレクトリです。
- <リポジトリルート (`cscmng.repository.root`)>  
HCSC-Manager 定義 ( `cscmng.properties` ) の `cscmng.repository.root` キーに指定したリポジトリルートです。
- <ログディレクトリ (`cscmng.log.dir`)>  
HCSC-Manager 定義 ( `cscmng.properties` ) の `cscmng.log.dir` キーに指定したログディレクトリです。
- <Eclipse ワークスペースディレクトリ >  
Eclipse を利用するユーザが任意で作成するディレクトリです。

注 1

21 階層分までのディレクトリを収集対象にします。

注 2

環境に合わせて収集対象になるディレクトリパスを変更してから収集してください。

注 3

Cosminexus TPBroker で環境変数が設定されている場合は, 環境変数の設定が優先されます。

注 4

\*.zip 以外のファイルが収集されます。

注 5

uCosminexus Service Platform の場合だけ出力されます。

## 付録 B.2 Cosminexus Component Transaction Monitor

Cosminexus Component Transaction Monitor に関連する収集対象ディレクトリパスを次の表に示します。

表 B-3 Cosminexus Component Transaction Monitor に関連する収集対象ディレクトリパス (Windows の場合)

収集対象ディレクトリパス	一次 / 二次	収集方法
<スプールディレクトリ (ctmspool)>¥log¥ctmdmlog*	一次	
<スプールディレクトリ (ctmspool)>¥log¥<CTM 識別子 >¥ctmlog*	一次	

(凡例)

: デフォルトで収集されます。出力ディレクトリを変更した場合も、収集対象になります。

注 &lt;スプールディレクトリ (ctmspool)&gt; は、環境変数 CTMSPOOL に指定したディレクトリです。

表 B-4 Cosminexus Component Transaction Monitor に関連する収集対象ディレクトリパス (UNIX の場合)

収集対象ディレクトリパス	一次 / 二次	収集方法
<スプールディレクトリ (ctmspool)>/log/ctmdmlog*	一次	
<スプールディレクトリ (ctmspool)>/log/<CTM 識別子 >/ctmlog*	一次	

(凡例)

: デフォルトで収集されます。出力ディレクトリを変更した場合も、収集対象になります。

注 &lt;スプールディレクトリ (ctmspool)&gt; は、環境変数 CTMSPOOL に指定したディレクトリです。

## 付録 B.3 Cosminexus DABroker Library

Cosminexus DABroker Library に関連する収集対象ディレクトリパスを次の表に示します。

表 B-5 Cosminexus DABroker Library に関連する収集対象ディレクトリパス (Windows の場合)

収集対象ディレクトリパス	一次 / 二次	収集方法
<Cosminexus のインストールディレクトリ >¥DAB¥conf¥*	一次	
<Cosminexus のインストールディレクトリ >¥DAB¥spool¥*	一次	
<Cosminexus のインストールディレクトリ >¥DAB¥spool¥**¥*	二次	

(凡例)

: デフォルトで収集されます。出力ディレクトリを変更した場合も、収集対象になります。

注 &lt;Cosminexus のインストールディレクトリ &gt; は、Cosminexus のインストールディレクトリです。デフォルトは、「C:¥Program Files¥HITACHI¥Cosminexus」です。

表 B-6 Cosminexus DABroker Library に関連する収集対象ディレクトリパス (UNIX の場合)

収集対象ディレクトリパス	一次 / 二次	収集方法
/opt/DABroker/conf/*	一次	
/opt/DABroker/spool/*	一次	
/opt/DABroker/spool/*/*	二次	
<DABroker 運用ディレクトリ >/DAB/conf/*	一次	
<DABroker 運用ディレクトリ >/DAB/spool/*	一次	
<DABroker 運用ディレクトリ >/DAB/spool/*/*	二次	

(凡例)

: デフォルトで収集されます。出力ディレクトリを変更した場合も、収集対象になります。

: デフォルトで収集されます。ただし、出力ディレクトリを設定を変更した場合、snapshot ログとして収集するには、snapshot ログ収集対象定義ファイル (snapshotlog.conf, または snapshotlog.2.conf) を編集して、変更後の出力ディレクトリを収集対象ディレクトリとして設定し直す必要があります。

注 <DABroker 運用ディレクトリ > は、デフォルトでは <Cosminexus DABroker Library のインストールディレクトリ > です。dabsetup コマンドを使用して変更することもできます。

## 付録 B.4 Cosminexus Developer's Kit for Java

Windows の場合の Cosminexus Developer's Kit for Java に関連する収集対象ディレクトリパスを次の表に示します。

表 B-7 Cosminexus Developer's Kit for Java に関連する収集対象ディレクトリパス (Windows の場合)

収集対象ディレクトリパス	一次 / 二次	収集方法
C:\¥Documents and Settings¥All Users¥Application Data¥Microsoft¥Dr Watson¥drwtsn32.log	一次	
C:\¥Documents and Settings¥All Users¥Application Data¥Microsoft¥Dr Watson¥user.dmp	二次	

(凡例)

: デフォルトで収集されます。ただし、出力ディレクトリを設定を変更した場合、snapshot ログとして収集するには、snapshot ログ収集対象定義ファイル (snapshotlog.conf, または snapshotlog.2.conf) を編集して、変更後の出力ディレクトリを収集対象ディレクトリとして設定し直す必要があります。

注 収集対象ディレクトリパスは、drwtsn32 のデフォルトです。変更されている可能性があります。



## 付録 B.5 Cosminexus Performance Tracer

Cosminexus Performance Tracer に関連する収集対象ディレクトリパスを次の表に示します。

表 B-8 Cosminexus Performance Tracer に関連する収集対象ディレクトリパス  
(Windows の場合)

収集対象ディレクトリパス	一次 / 二次	収集 方法
<スプールディレクトリ (prfspool)>%log%<PRF 識別子>%*	一次	
<スプールディレクトリ (prfspool)>%shm%*	二次	
<スプールディレクトリ (prfspool)>%utt%prf%<PRF 識別子>%dcopltrc%*	二次	
<スプールディレクトリ (prfspool)>%utt%prf%<PRF 識別子>%spool%dcopltrcinf%*	二次	
<スプールディレクトリ (prfspool)>%utt%prf%<PRF 識別子>%spool%save%*	二次	
<スプールディレクトリ (prfspool)>%utt%prf%<PRF 識別子>%*	二次	

(凡例)

: デフォルトで収集されます。出力ディレクトリを変更した場合も、収集対象になります。

注 <スプールディレクトリ (prfspool)> は、環境変数 PRFSPOOL に指定したディレクトリです。

表 B-9 Cosminexus Performance Tracer に関連する収集対象ディレクトリパス (UNIX  
の場合)

収集対象ディレクトリパス	一次 / 二次	収集 方法
<スプールディレクトリ (prfspool)>/log/<PRF 識別子>/*	一次	
<スプールディレクトリ (prfspool)>/utt/prf/<PRF 識別子>/dcopltrc/*	二次	
<スプールディレクトリ (prfspool)>/utt/prf/<PRF 識別子>/spool/dcopltrcinf/*	二次	
<スプールディレクトリ (prfspool)>/utt/prf/<PRF 識別子>/spool/save/*	二次	
<スプールディレクトリ (prfspool)>/utt/prf/<PRF 識別子>/*	二次	

(凡例)

: デフォルトで収集されます。出力ディレクトリを変更した場合も、収集対象になります。

注 <スプールディレクトリ (prfspool)> は、環境変数 PRFSPOOL に指定したディレクトリです。

## 付録 B.6 Cosminexus TPBroker

Cosminexus TPBroker に関連する収集対象ディレクトリパスを次の表に示します。

表 B-10 Cosminexus TPBroker に関連する収集対象ディレクトリパス ( Windows の場合 )

収集対象ディレクトリパス	一次 / 二次	収集方法
<Cosminexus のインストールディレクトリ >¥TPB¥log¥*	一次	
<Cosminexus のインストールディレクトリ >¥TPB¥log¥*¥*	二次	
<Cosminexus のインストールディレクトリ >¥TPB¥log¥*	一次	
<Cosminexus のインストールディレクトリ >¥TPB¥log¥*¥*	二次	
<Cosminexus のインストールディレクトリ >¥TPB¥adm¥*	一次	
<Cosminexus のインストールディレクトリ >¥TPB¥spool¥log¥admlog	一次	
<Cosminexus のインストールディレクトリ >¥TPB¥spool¥admsts	二次	
<Cosminexus のインストールディレクトリ >¥TPB¥spool¥admproc	二次	
<Cosminexus のインストールディレクトリ >¥TPB¥spool¥*	二次	
<Cosminexus のインストールディレクトリ >¥TPB¥spool¥*¥*	二次	
<Cosminexus のインストールディレクトリ >¥TPB¥spool¥*¥*¥*	二次	
<Cosminexus のインストールディレクトリ >¥TPB¥spool¥*¥*¥*¥*	二次	
<ADM スプールディレクトリ >¥log¥admlog	一次	
<ADM スプールディレクトリ >¥admsts	二次	
<ADM スプールディレクトリ >¥admproc	二次	
<ADM スプールディレクトリ >¥*	二次	
<ADM スプールディレクトリ >¥*¥*	二次	
<ADM スプールディレクトリ >¥*¥*¥*	二次	
<ADM スプールディレクトリ >¥*¥*¥*¥*	二次	
<ADMFS ディレクトリ >¥admsts	二次	
<ADMFS ディレクトリ >¥admproc	二次	
<ADMFS ディレクトリ >¥*	二次	
<Cosminexus のインストールディレクトリ >¥TPB¥otsspool¥log¥*	一次	
<Cosminexus のインストールディレクトリ >¥TPB¥otsspool¥tmsh	二次	
<Cosminexus のインストールディレクトリ >¥TPB¥otsspool¥*	二次	
<Cosminexus のインストールディレクトリ >¥TPB¥otsspool¥*¥*	二次	
<Cosminexus のインストールディレクトリ >¥TPB¥otsspool¥*¥*¥*	二次	
<Cosminexus のインストールディレクトリ >¥TPB¥otsspool¥*¥*¥*¥*	二次	
<スプールディレクトリ (tpspool)>¥log¥*	一次	
<スプールディレクトリ (tpspool)>¥tmsh	二次	

収集対象ディレクトリパス	一次 / 二次	収集方法
<スプールディレクトリ (tpspool)>¥*	二次	
<スプールディレクトリ (tpspool)>¥*¥*	二次	
<スプールディレクトリ (tpspool)>¥*¥*¥*	二次	
<スプールディレクトリ (tpspool)>¥*¥*¥*¥*	二次	
<TPFS ディレクトリ (tpfs) >¥*	二次	
<VBROKER_ADM ディレクトリ >¥..¥log¥*	一次	
<VBROKER_ADM ディレクトリ >¥..¥log¥*¥*	二次	
<VBROKER_ADM ディレクトリ >¥..¥logj¥*	一次	
<VBROKER_ADM ディレクトリ >¥..¥logj¥*¥*	二次	
<Cosminexus のインストールディレクトリ >¥CC¥server¥public¥ejb¥< サーバ名称 >¥otsstatus¥*	二次	
<Cosminexus のインストールディレクトリ >¥CC¥server¥public¥ejb¥< サーバ名称 >¥otsstatus¥*¥*	二次	
<ステータスファイルのディレクトリ (ejbserver.distributedtx.ots.status.directory1) >¥*	二次	
<ステータスファイルのディレクトリ (ejbserver.distributedtx.ots.statusfile.directory1) >¥*¥*	二次	
<予備ステータスファイルのディレクトリ (ejbserver.distributedtx.ots.statusfile.directory2) >¥*	二次	
<予備ステータスファイルのディレクトリ (ejbserver.distributedtx.ots.statusfile.directory2) >¥*¥*	二次	
イベントログ	一次	×

## ( 凡例 )

：デフォルトで収集されます。出力ディレクトリを変更した場合も、収集対象になります。

：デフォルトで収集されます。ただし、出力ディレクトリを設定を変更した場合、snapshot ログとして収集するには、snapshot ログ収集対象定義ファイル ( snapshotlog.conf, または snapshotlog.2.conf ) を編集して、変更後の出力ディレクトリを収集対象ディレクトリとして設定し直す必要があります。

×：デフォルトでは収集されません。snapshot ログとして収集するには、snapshot ログ収集対象定義ファイル ( snapshotlog.conf, または snapshotlog.2.conf ) を編集して、収集対象ディレクトリに設定する必要があります。

注 表内のそれぞれのディレクトリの意味は次のとおりです。

- <Cosminexus のインストールディレクトリ >  
Cosminexus のインストールディレクトリです。デフォルトは、「 C:¥Program Files¥HITACHI¥Cosminexus 」です。
- <ADM スプールディレクトリ >  
環境変数 ADMSPOOL に指定したディレクトリです。

- <ADMFS ディレクトリ>  
環境変数 ADMFS に指定したディレクトリです。
- <スプールディレクトリ (tpspool)>  
環境変数 TPSPOOL に指定したディレクトリです。
- <TPFS ディレクトリ (tpfs)>  
環境変数 TPFS に指定したディレクトリです。
- <VBROKER\_ADM ディレクトリ>  
環境変数 VBROKER\_ADM に指定したディレクトリです。
- <ステータスファイルのディレクトリ (ejbserver.distributedtx.ots.status.directory1)>  
簡易構築定義ファイルの論理 J2EE サーバ (j2ee-server) の <configuration> タグ内に ,  
ejbserver.distributedtx.ots.status.directory1 パラメタで指定したディレクトリです。
- <予備ステータスファイルのディレクトリ (ejbserver.distributedtx.ots.status.directory2)>  
簡易構築定義ファイルの論理 J2EE サーバ (j2ee-server) の <configuration> タグ内に ,  
ejbserver.distributedtx.ots.status.directory2 パラメタで指定したディレクトリです。

注

環境に合わせて収集対象になるディレクトリパスを変更してから収集してください。

表 B-11 Cosminexus TPBroker に関連する収集対象ディレクトリパス (UNIX の場合)

収集対象ディレクトリパス	一次 / 二次	収集方法
/opt/Cosminexus/TPB/log/*	一次	
/opt/Cosminexus/TPB/log/*/*	二次	
/opt/Cosminexus/TPB/logj*	一次	
/opt/Cosminexus/TPB/logj/*/*	二次	
/opt/Cosminexus/TPB/adm/*	一次	
/opt/Cosminexus/TPB/conf/*	一次	
/opt/Cosminexus/TPB/conf/*/*	一次	
/opt/Cosminexus/TPB/conf/*/*/*	一次	
/opt/Cosminexus/TPB/spool/log/admlog	一次	
/opt/Cosminexus/TPB/spool/admsts	二次	
/opt/Cosminexus/TPB/spool/admproc	二次	
/opt/Cosminexus/TPB/spool/*	二次	
/opt/Cosminexus/TPB/spool/*/*	二次	
/opt/Cosminexus/TPB/spool/*/*/*	二次	
<ADM スプールディレクトリ >/log/admlog	一次	
<ADM スプールディレクトリ >/admsts	二次	
<ADM スプールディレクトリ >/admproc	二次	

収集対象ディレクトリパス	一次 / 二次	収集方法
<ADM スプールディレクトリ >/*	二次	
<ADM スプールディレクトリ >/*/*	二次	
<ADM スプールディレクトリ >/*/*/*	二次	
<ADM スプールディレクトリ >/*/*/*/*	二次	
<ADMFS ディレクトリ >/admsts	二次	
<ADMFS ディレクトリ >/admproc	二次	
<ADMFS ディレクトリ >/*	二次	
/opt/Cosminexus/TPB/otsspool/log/*	一次	
/opt/Cosminexus/TPB/otsspool/tmshm	二次	
/opt/Cosminexus/TPB/otsspool/*	二次	
/opt/Cosminexus/TPB/otsspool/*/*	二次	
/opt/Cosminexus/TPB/otsspool/*/*/*	二次	
/opt/Cosminexus/TPB/otsspool/*/*/*/*	二次	
<スプールディレクトリ (tpspool)>/log/*	一次	
<スプールディレクトリ (tpspool)>/tmshm	一次	
<スプールディレクトリ (tpspool)>/*	二次	
<スプールディレクトリ (tpspool)>/*/*	二次	
<スプールディレクトリ (tpspool)>/*/*/*	二次	
<スプールディレクトリ (tpspool)>/*/*/*/*	二次	
<スプールディレクトリ (tpspool)>/*/*/*/*/*	二次	
<TPFS ディレクトリ ( tpfs ) >/*	二次	
<VBROKER_ADM ディレクトリ >./log/*	一次	
<VBROKER_ADM ディレクトリ >./log/*/*	二次	
<VBROKER_ADM ディレクトリ >./logj/*	一次	
<VBROKER_ADM ディレクトリ >./logj/*/*	二次	
/opt/Cosminexus/CC/server/public/ejb/<サーバ名称 >/otsstatus/*	二次	
/opt/Cosminexus/CC/server/public/ejb/<サーバ名称 >/otsstatus/*/*	二次	
<ステータスファイルのディレクトリ (ejbserver.distributedtx.ots.status.directory1) >/*	二次	
<ステータスファイルのディレクトリ (ejbserver.distributedtx.ots.status.directory1) >/*/*	二次	

収集対象ディレクトリパス	一次 / 二次	収集方法
< 予備ステータスファイルのディレクトリ (ejbserver.distributedtx.ots.status.directory2) >/*	二次	
< 予備ステータスファイルのディレクトリ (ejbserver.distributedtx.ots.status.directory2) >/*/*	二次	
syslog	一次	×

(凡例)

：デフォルトで収集されます。出力ディレクトリを変更した場合も、収集対象になります。

：デフォルトで収集されます。ただし、出力ディレクトリの設定を変更した場合、snapshot ログとして収集するには、snapshot ログ収集対象定義ファイル (snapshotlog.conf, または snapshotlog.2.conf) を編集して、変更後の出力ディレクトリを収集対象ディレクトリとして設定し直す必要があります。

×：デフォルトでは収集されません。snapshot ログとして収集するには、snapshot ログ収集対象定義ファイル (snapshotlog.conf, または snapshotlog.2.conf) を編集して、収集対象ディレクトリに設定する必要があります。

注 表内のそれぞれのディレクトリの意味は次のとおりです。

- <ADM スプールディレクトリ >  
環境変数 ADMSPPOOL に指定したディレクトリです。
- <ADMFS ディレクトリ >  
環境変数 ADMFS に指定したディレクトリです。
- <TPFS ディレクトリ (tpfs) >  
環境変数 TPFS に指定したディレクトリです。
- <VBROKER\_ADM ディレクトリ >  
環境変数 VBROKER\_ADM に指定したディレクトリです。
- <スプールディレクトリ (tpspool)>  
環境変数 TPSPOOL に指定したディレクトリです。
- <ステータスファイルのディレクトリ (ejbserver.distributedtx.ots.status.directory1)>  
簡易構築定義ファイルの論理 J2EE サーバ (j2ee-server) の <configuration> タグ内に、  
ejbserver.distributedtx.ots.status.directory1 パラメタで指定したディレクトリです。
- <予備ステータスファイルのディレクトリ (ejbserver.distributedtx.ots.status.directory2)>  
簡易構築定義ファイルの論理 J2EE サーバ (j2ee-server) の <configuration> タグ内に、  
ejbserver.distributedtx.ots.status.directory2 パラメタで指定したディレクトリです。

注

環境に合わせて収集対象になるディレクトリパスを変更してから収集してください。

## 付録 B.7 Cosminexus Web Services - Security

Cosminexus Web Services - Security に関連する収集対象ディレクトリパスを次の表に示します。

表 B-12 Cosminexus Web Services - Security に関連する収集対象ディレクトリパス  
(Windows の場合)

収集対象ディレクトリパス	一次 / 二 次	収集 方法
<Cosminexus のインストールディレクトリ >¥wss¥logs¥*	一次	
<Cosminexus のインストールディレクトリ >¥wss¥conf¥*	一次	
<Cosminexus のインストールディレクトリ >¥CC¥server¥public¥web¥< サーバ名称 >¥< コンテキストルート >¥WEB-INF¥classes¥client-config.xml	一次	
<Cosminexus のインストールディレクトリ >¥CC¥server¥public¥web¥< サーバ名称 >¥< コンテキストルート >¥WEB-INF¥classes¥policy-config.xml	一次	
<Cosminexus のインストールディレクトリ >¥CC¥server¥public¥web¥< サーバ名称 >¥< コンテキストルート >¥WEB-INF¥classes¥security-config.xml	一次	
<Cosminexus のインストールディレクトリ >¥CC¥web¥containers¥< サーバ名称 >¥webapps¥< コンテキストルート >¥WEB-INF¥classes¥client-config.xml	一次	
<Cosminexus のインストールディレクトリ >¥CC¥web¥containers¥< サーバ名称 >¥webapps¥< コンテキストルート >¥WEB-INF¥classes¥policy-config.xml	一次	
<Cosminexus のインストールディレクトリ >¥CC¥web¥containers¥< サーバ名称 >¥webapps¥< コンテキストルート >¥WEB-INF¥classes¥security-config.xml	一次	
<J2EE サーバ作業ディレクトリ (ejb.public.directory)>¥ejb¥< サーバ名称 >¥cjmessage*	一次	
<J2EE サーバ作業ディレクトリ (ejb.public.directory)>¥ejb¥< サーバ名称 >¥cjexception*	一次	
<J2EE サーバ作業ディレクトリ (ejb.public.directory)>¥ejb¥< サーバ名称 >¥WS¥*	一次	
<J2EE サーバ作業ディレクトリ (ejb.public.directory)>¥ejb¥< サーバ名称 >¥WS¥maintenance¥*	二次	
<EJB クライアントログ出力ディレクトリ (ejbserver.client.log.directory)>¥*¥*¥*	一次	
<Cosminexus のインストールディレクトリ >¥CC¥web¥containers¥< サーバ名称 >¥logs¥cjmessage*	一次	
<Cosminexus のインストールディレクトリ >¥CC¥web¥containers¥< サーバ名称 >¥logs¥cjexception*	一次	
<Cosminexus のインストールディレクトリ >¥CC¥web¥containers¥< サーバ名称 >¥logs¥WS¥*	一次	
<Cosminexus のインストールディレクトリ >¥CC¥web¥containers¥< サーバ名称 >¥logs¥WS¥maintenance¥*	二次	
<Cosminexus のインストールディレクトリ >¥XMLSEC¥logs¥*	一次	

## ( 凡例 )

：デフォルトで収集されます。出力ディレクトリを変更した場合も、収集対象になります。

：デフォルトで収集されます。ただし、出力ディレクトリを設定を変更した場合、snapshot ログとして収集するには、snapshot ログ収集対象定義ファイル ( snapshotlog.conf, または

snapshotlog.2.conf) を編集して、変更後の出力ディレクトリを収集対象ディレクトリとして設定し直す必要があります。

注 表内のそれぞれのディレクトリの意味は次のとおりです。

- <Cosminexus のインストールディレクトリ>  
Cosminexus のインストールディレクトリです。デフォルトは、「C:\Program Files\HITACHI\Cosminexus」です。
- <J2EE サーバ作業ディレクトリ (ejb.public.directory)>  
簡易構築定義ファイルの論理 J2EE サーバ (jee-server) の <configuration> タグ内に、  
ejb.public.directory パラメタで指定されたディレクトリです。
- <EJB クライアントログ出力ディレクトリ (ejbserver.client.log.directory)>  
EJB クライアントアプリケーションのシステムプロパティ ejbserver.client.log.directory キーに  
指定したディレクトリです。

表 B-13 Cosminexus Web Services - Security に関連する収集対象ディレクトリパス  
(UNIX の場合)

収集対象ディレクトリパス	一次 /二 次	収集 方法
/opt/Cosminexus/wss/logs/*	一次	
/opt/Cosminexus/wss/conf/*	一次	
/opt/Cosminexus/CC/server/public/web/<サーバ名称>/<コンテキストルート>/WEB-INF/classes/client-config.xml	一次	
/opt/Cosminexus/CC/server/public/web/<サーバ名称>/<コンテキストルート>/WEB-INF/classes/policy-config.xml	一次	
/opt/Cosminexus/CC/server/public/web/<サーバ名称>/<コンテキストルート>/WEB-INF/classes/security-config.xml	一次	
/opt/Cosminexus/CC/web/containers/<サーバ名称>/webapps/<コンテキストルート>/WEB-INF/classes/client-config.xml	一次	
/opt/Cosminexus/CC/web/containers/<サーバ名称>/webapps/<コンテキストルート>/WEB-INF/classes/policy-config.xml	一次	
/opt/Cosminexus/CC/web/containers/<サーバ名称>/webapps/<コンテキストルート>/WEB-INF/classes/security-config.xml	一次	
<J2EE サーバ作業ディレクトリ (ejb.public.directory)>/ejb/<サーバ名称>/cjmessage*	一次	
<J2EE サーバ作業ディレクトリ (ejb.public.directory)>/ejb/<サーバ名称>/cjexception*	一次	
<J2EE サーバ作業ディレクトリ (ejb.public.directory)>/ejb/<サーバ名称>/WS/*	一次	
<J2EE サーバ作業ディレクトリ (ejb.public.directory)>/ejb/<サーバ名称>/WS/maintenance/*	二次	
<EJB クライアントログ出力ディレクトリ (ejbserver.client.log.directory)>/*/*/*	一次	
/opt/Cosminexus/CC/web/containers/<サーバ名称>/logs/cjmessage*	一次	
/opt/Cosminexus/CC/web/containers/<サーバ名称>/logs/cjexception*	一次	



収集対象ディレクトリパス	一次 / 二次	収集方法
/opt/Cosminexus/CC/web/containers/<サーバ名称>/logs/WS/*	一次	
/opt/Cosminexus/CC/web/containers/<サーバ名称>/logs/WS/maintenance/*	二次	
/opt/Cosminexus/XMLSEC/logs/*	一次	

(凡例)

：デフォルトで収集されます。出力ディレクトリを変更した場合も、収集対象になります。

：デフォルトで収集されます。ただし、出力ディレクトリを設定を変更した場合、snapshot ログとして収集するには、snapshot ログ収集対象定義ファイル (snapshotlog.conf, または snapshotlog.2.conf) を編集して、変更後の出力ディレクトリを収集対象ディレクトリとして設定し直す必要があります。

注 表内のそれぞれのディレクトリの意味は次のとおりです。

- <J2EE サーバ作業ディレクトリ (ejb.public.directory)>  
簡易構築定義ファイルの論理 J2EE サーバ (j2ee-server) の <configuration> タグ内に、  
ejb.public.directory パラメータで指定されたディレクトリです。
- <EJB クライアントログ出力ディレクトリ (ejbserver.client.log.directory)>  
EJB クライアントアプリケーションのシステムプロパティ ejbserver.client.log.directory キーに  
指定したディレクトリです。

## 付録 B.8 Hitachi Web Server

Hitachi Web Server に関連する収集対象ディレクトリパスを次の表に示します。

表 B-14 Hitachi Web Server に関連する収集対象ディレクトリパス (Windows の場合)

収集対象ディレクトリパス	一次 / 二次	収集方法
<HWS のインストールディレクトリ>%conf%\*.conf	一次	
<HWS のインストールディレクトリ>%logs\*	二次	
<HWS のインストールディレクトリ>%servers%HWS_<サーバ名称>%logs\*	二次	
<HWSTraceLogFile ディレクティブ>%*	二次	
<HWS ログ出力ディレクトリ (hws.logfile.dir)>%*	二次	
<HWS のインストールディレクトリ>%servers%HWS_<サーバ名称>%conf%\*.conf	一次	

(凡例)

：デフォルトで収集されます。出力ディレクトリを変更した場合も、収集対象になります。

：デフォルトで収集されます。ただし、出力ディレクトリを設定を変更した場合、snapshot ログとして収集するには、snapshot ログ収集対象定義ファイル (snapshotlog.conf, または snapshotlog.2.conf) を編集して、変更後の出力ディレクトリを収集対象ディレクトリとして設定し直す必要があります。

注 表内のそれぞれのディレクトリの意味は次のとおりです。

- <HWS のインストールディレクトリ>  
Hitachi Web Server のインストールディレクトリです。デフォルトは、「C:\Program Files\HITACHI\Cosminexus\httpsd」です。
- <HWSTraceLogFile ディレクティブ>  
httpsd.conf の HWSTraceLogFile ディレクティブに指定したディレクトリです。
- <HWS ログ出力ディレクトリ (hws.logfile.dir)>  
httpsd.conf の HWSTraceLogFile, ErrorLog, CustomLog, および HWSRequestLog ディレクティブに指定したファイル名のディレクトリ部分です。

表 B-15 Hitachi Web Server に関連する収集対象ディレクトリパス (UNIX の場合)

収集対象ディレクトリパス	一次 / 二次	収集方法
opt/hitachi/httpsd/conf/* .conf	一次	
opt/hitachi/httpsd/logs/*	二次	
opt/hitachi/httpsd/servers/HWS_<サーバ名称>/logs/*	二次	
<HWSTraceLogFile ディレクティブ>/*	二次	
<コアダンプ出力ディレクトリ (CoreDumpDirectory)>/*	二次	
<HWS ログ出力ディレクトリ (hws.logfile.dir)>/*	二次	
opt/hitachi/httpsd/servers/HWS_<サーバ名称>/conf/* .conf	一次	

(凡例)

：デフォルトで収集されます。出力ディレクトリを変更した場合も、収集対象になります。

：デフォルトで収集されます。ただし、出力ディレクトリを設定を変更した場合、snapshot ログとして収集するには、snapshot ログ収集対象定義ファイル (snapshotlog.conf, または snapshotlog.2.conf) を編集して、変更後の出力ディレクトリを収集対象ディレクトリとして設定し直す必要があります。

注 表内のそれぞれのディレクトリの意味は次のとおりです。

- <サーバ名称>  
Management Server で設定した値です。
- <HWSTraceLogFile ディレクティブ>  
httpsd.conf の HWSTraceLogFile ディレクティブに指定したディレクトリです。
- <HWS ログ出力ディレクトリ (hws.logfile.dir)>  
httpsd.conf の HWSTraceLogFile, ErrorLog, CustomLog, および HWSRequestLog ディレクティブに指定したファイル名のディレクトリ部分です。

## 付録 B.9 Cosminexus cFramework

Cosminexus cFramework に関連する収集対象ディレクトリパスを次の表に示します。

表 B-16 Cosminexus cFramework に関連する収集対象ディレクトリパス (Windows の場合)

分類	収集対象ディレクトリパス	一次 / 二次	収集方法
J2EE サーバ (J2EE サーバモードで実行している場合)	<J2EE サーバ作業ディレクトリ (ejb.public.directory)>¥web¥< サーバ名称 >¥< コンテキストルート >¥WEB-INF¥props¥*	一次	
	<J2EE サーバ作業ディレクトリ (ejb.public.directory)>¥ejb¥< サーバ名称 >¥cframework*	一次	
Web コンテナサーバ (サブレットエンジンモードで実行している場合)	<Cosminexus のインストールディレクトリ >CC¥web¥containers¥< サーバ名称 >¥webapps¥ < コンテキストルート > ¥WEB-INF¥props¥*	一次	
	<Cosminexus のインストールディレクトリ >¥CC¥web¥containers¥< サーバ名称 >¥cframework*	一次	

## (凡例)

: デフォルトで収集されます。出力ディレクトリを変更した場合も、収集対象になります。

注 表内のそれぞれのディレクトリの意味は次のとおりです。

- <Cosminexus のインストールディレクトリ >  
Cosminexus のインストールディレクトリです。デフォルトは、「C:¥Program Files¥HITACHI¥Cosminexus」です。
- <J2EE サーバ作業ディレクトリ (ejb.public.directory)>  
簡易構築定義ファイルの論理 J2EE サーバ (j2ee-server) の <configuration> タグ内に、ejb.public.directory パラメタで指定されたディレクトリです。

表 B-17 Cosminexus cFramework に関連する収集対象ディレクトリパス (UNIX の場合)

分類	収集対象ディレクトリパス	一次 / 二次	収集方法
J2EE サーバ (J2EE サーバモードで実行している場合)	<J2EE サーバ作業ディレクトリ (ejb.public.directory)>/web/< サーバ名称 >/< コンテキストルート >/WEB-INF/props/*	一次	
	<J2EE サーバ作業ディレクトリ (ejb.public.directory)>/ejb/< サーバ名称 >/cframework*	一次	

分類	収集対象ディレクトリパス	一次 / 二次	収集方法
Web コンテナサーバ (サブレットエンジンモードで実行している場合)	/opt/Cosminexus/CC/web/containers/<サーバ名称>/webapps/ <コンテキストルート> /WEB-INF/props/*	一次	
	/opt/Cosminexus/CC/web/containers/<サーバ名称>/cframework*	一次	

(凡例)

: デフォルトで収集されます。出力ディレクトリを変更した場合も、収集対象になります。

注 表内の <J2EE サーバ作業ディレクトリ (ejb.public.directory)> は簡易構築定義ファイルの論理 J2EE サーバ (j2ee-server) の <configuration> タグ内に、ejb.public.directory パラメータで指定されたディレクトリです。

## 付録 B.10 uCosminexus Service Adapter for TP1

uCosminexus Service Adapter for TP1 の情報に関連する収集対象ディレクトリパスを次の表に示します。

表 B-18 uCosminexus Service Adapter for TP1 の情報に関連する収集対象ディレクトリパス (Windows の場合)

収集対象ディレクトリパス	一次 / 二次	収集方法
<J2EE サーバ作業ディレクトリ (ejb.public.directory)>%ejb%<サーバ名称>%logs%¥CSCADP¥TP1ADP¥maintenance¥*	二次	

(凡例)

: デフォルトで収集されます。ただし、出力ディレクトリを設定を変更した場合、snapshot ログとして収集するには、snapshot ログ収集対象定義ファイル (snapshotlog.conf、または snapshotlog.2.conf) を編集して、変更後の出力ディレクトリを収集対象ディレクトリとして設定し直す必要があります。

注 表内の <J2EE サーバ作業ディレクトリ (ejb.public.directory)> は簡易構築定義ファイルの論理 J2EE サーバ (j2ee-server) の <configuration> タグ内に、ejb.public.directory パラメータで指定されたディレクトリです。

表 B-19 uCosminexus Service Adapter for TP1 の情報に関連する収集対象ディレクトリパス (UNIX の場合)

収集対象ディレクトリパス	一次 / 二次	収集方法
<J2EE サーバ作業ディレクトリ (ejb.public.directory)>/ejb/<サーバ名称>/logs/CSCADP/TP1ADP/maintenance/*	二次	

( 凡例 )

: デフォルトで収集されます。ただし、出力ディレクトリを設定を変更した場合、snapshot ログとして収集するには、snapshot ログ収集対象定義ファイル ( snapshotlog.conf, または snapshotlog.2.conf ) を編集して、変更後の出力ディレクトリを収集対象ディレクトリとして設定し直す必要があります。

注 表内の <J2EE サーバ作業ディレクトリ (ejb.public.directory)> は簡易構築定義ファイルの論理 J2EE サーバ ( j2ee-server ) の <configuration> タグ内に、ejb.public.directory パラメタで指定されたディレクトリです。

## 付録 B.11 監査ログ

監査ログの情報に関連する収集対象ディレクトリパスを次の表に示します。

表 B-20 監査ログ ( Windows の場合 )

分類	収集対象ディレクトリパス	一次 / 二次	収集方法
監査ログの例外情報	<Cosminexus のインストールディレクトリ>%auditlog%rasexception*.log	一次	
監査ログのメッセージログ	<Cosminexus のインストールディレクトリ>%auditlog%rasmessage*.log	一次	
監査ログ	<Cosminexus のインストールディレクトリ>%auditlog%*	二次	
	<Cosminexus のインストールディレクトリ>%auditlog%*%*	二次	

( 凡例 )

: デフォルトで収集されます。ただし、出力ディレクトリを設定を変更した場合、snapshot ログとして収集するには、snapshot ログ収集対象定義ファイル ( snapshotlog.conf, または snapshotlog.2.conf ) を編集して、変更後の出力ディレクトリを収集対象ディレクトリとして設定し直す必要があります。

表 B-21 監査ログ (UNIX の場合)

分類	収集対象ディレクトリパス	一次 / 二次	収集方法
監査ログの例外情報	/opt/Cosminexus/auditlog/rasexception*.log	一次	
監査ログのメッセージログ	/opt/Cosminexus/auditlog/rasmesssage*.log	一次	
監査ログ	/opt/Cosminexus/auditlog/*	二次	
	/opt/Cosminexus/auditlog/*/*	二次	

(凡例)

：デフォルトで収集されます。ただし、出力ディレクトリを設定を変更した場合、snapshot ログとして収集するには、snapshot ログ収集対象定義ファイル (snapshotlog.conf、または snapshotlog.2.conf) を編集して、変更後の出力ディレクトリを収集対象ディレクトリとして設定し直す必要があります。

## 付録 B.12 そのほかの情報

そのほかの情報に関連する収集対象ディレクトリパスを次の表に示します。

表 B-22 そのほかの情報に関連する収集対象ディレクトリパス (UNIX の場合)

内容	収集対象のディレクトリパス	一次 / 二次	収集方法
プログラムプロダクト情報	/etc/.hitachi/pplistd	一次	

(凡例)

：デフォルトで収集されます。出力ディレクトリを変更した場合も、収集対象になります。

## 付録 C データベースと接続中にトラブルが発生したコネクションの特定

データベース関連のトラブルに対処するためには、どのコネクションを使用してデータベースと接続しているかを特定することが重要です。

ここでは、データベース（HiRDB または Oracle）と接続中にトラブルが発生した場合に、Cosminexus から出力される情報（性能解析トレースやログなど）とデータベースから出力される情報（トレース情報や pdls コマンドの実行結果、動的パフォーマンスビューの内容など）を使用して、トラブルが発生したコネクションを特定する方法について説明します。

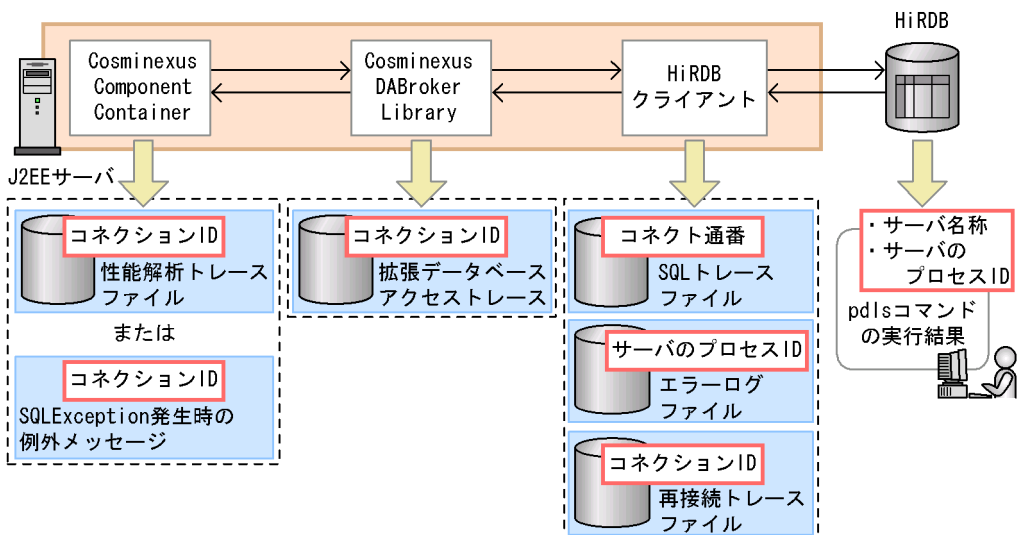
Cosminexus では、HiRDB および Oracle との接続に使用しているコネクションを一意に識別するための接続情報（コネクション ID）を性能解析トレースに出力しています。J2EE サーバからデータベースに至るまで、およびデータベースから J2EE サーバに処理結果が返却されるまでの一連の処理では、関連する構成ソフトウェアのログやトレース情報などにコネクション ID やデータベースサーバで割り振られるコネクト通番などが出力されます。これらの情報を突き合わせて確認することで、トラブルが発生したコネクションを特定できます。

HiRDB を使用する場合、および Oracle を使用する場合のコネクション ID の出力の概要、およびトラブルが発生したコネクションの特定に使用する情報を次に示します。

HiRDB を使用する場合

コネクション ID の出力の概要を次に示します。

図 C-1 コネクション ID の出力の概要（HiRDB の場合）



トラブルが発生したコネクションの特定に使用する情報を次の表に示します。

表 C-1 トラブルが発生したコネクションの特定に使用する情報 (HiRDB の場合)

項番	出力元	情報の種類	参照先
1	Cosminexus Component Container	性能解析トレースファイル	付録 C.1
2		SQLException 発生時の例外メッセージ	
3	Cosminexus DABroker Library	拡張データベースアクセストレース	付録 C.2
4	HiRDB クライアント	SQL トレースファイル	付録 C.3
5		エラーログ情報	
6		再接続トレース	
7	HiRDB サーバ	pdls コマンドの実行結果	付録 C.4

コネクション ID には次に示す情報が含まれています。

- サーバ名称  
フロントエンドサーバ名 (HiRDB/Parallel Server 使用時) またはシングルサーバ名 (HiRDB/Single Server 使用時) が表示されます。
- コネクト通番  
サーバ名称に表示された HiRDB サーバで割り振られるコネクト通番が表示されません。
- サーバのプロセス ID  
サーバ名称に表示された HiRDB サーバのプロセス ID が表示されます。

コネクション ID の出力形式と出力例を次に示します。

コネクション ID の出力形式 (HiRDB の場合)

サーバ名称 : コネクト通番 : サーバのプロセス ID

コネクション ID の出力例 (HiRDB の場合)

fes01:15:2351

### ! 注意事項

07-01 以前のバージョンの HiRDB に接続していて、グローバルトランザクションとローカルトランザクションが混在している場合、性能解析トレースに出力されるコネクション ID と実際の接続状態に不整合が生じます。

例えば、コネクションの障害検知機能を無効にして、次のようなグローバルトランザクションとローカルトランザクションが混在している処理を実行する場合、6. ~ 8. および 9. ~ 13. の処理で性能解析トレースに出力される情報と実際の接続状態に不整合が生じます。



```

1. DataSource.getConnection() : xa_open
2. UserTransaction.begin()   : xa_start
3. SQL実行                   --- グローバルトランザクション
4. UserTransaction.commit()  : xa_end
5. Connection.close()

6. DataSource.getConnection()
7. SQL実行                   --- ローカルトランザクション
8. Connection.close()

9. DataSource.getConnection() : xa_open
10. UserTransaction.begin()   : xa_start
11. SQL実行                   --- グローバルトランザクション
12. UserTransaction.commit()  : xa_end
13. Connection.close()

```

- 6. ~ 8. の処理では、ローカルトランザクションのコネクション ID は出力されずに、グローバルトランザクションのコネクション ID が出力されます。
- 9. ~ 13. の処理では、グローバルトランザクションのコネクション ID は出力されずに、ローカルトランザクションのコネクション ID が出力されます。

このように、ローカルトランザクションのコネクション ID や、ローカルトランザクション実行後のグローバルトランザクションのコネクション ID を追跡することはできません。なお、7. のローカルトランザクションの処理で、コネクションが切り替わりますが Cosminexus には通知されません。

グローバルトランザクションとローカルトランザクションの混在は、使用する DBConnector のトランザクションサポートレベルが XATransaction の場合に、トランザクションを開始する前にコネクションを取得して SQL を実行したときにも発生します。

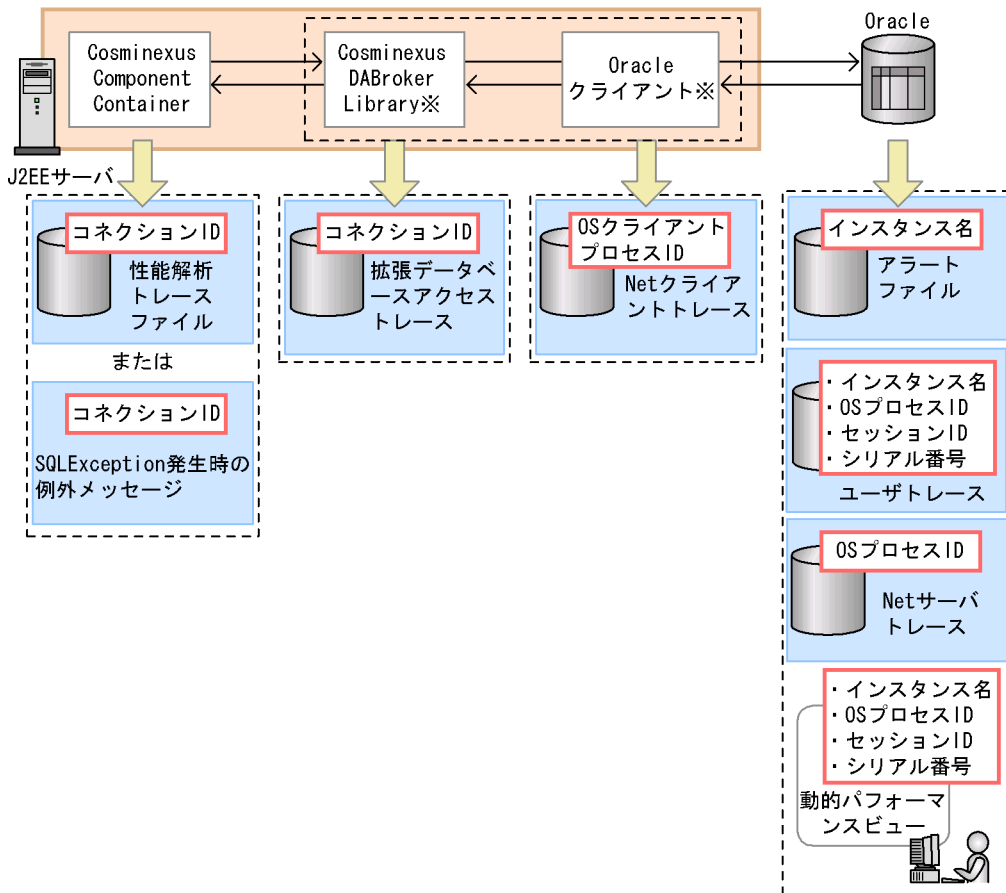
## 参考

グローバルトランザクションの処理中に、エラーが発生してコネクションが切断されると自動的に再接続されます。しかし、この再接続処理は再接続トレースには出力されません。この場合、再接続トレースに出力されるコネクション ID と実際の接続情報に不整合が生じます。ただし、基本的にグローバルトランザクションの処理中にコネクションが切断されることはありません。

## Oracle を使用する場合

コネクション ID の出力の概要を次に示します。

図 C-2 コネクション ID の出力の概要 (Oracle の場合)



注※ Oracle JDBC Thin Driverを使用する場合、Cosminexus DABroker LibraryとOracleクライアントは使用しません。

トラブルが発生したコネクションの特定に使用する情報を次の表に示します。

表 C-2 トラブルが発生したコネクションの特定に使用する情報 (Oracle の場合)

項番	出力元	情報の種類	参照先
1	Cosminexus Component Container	性能解析トレースファイル	付録 C.1
2		SQLException 発生時の例外メッセージ	
3	Cosminexus DABroker Library	拡張データベースアクセストレース	付録 C.2
4	Oracle クライアント	Net クライアントトレース	付録 C.5

項番	出力元	情報の種類	参照先
5	Oracle サーバ	アラートファイル	付録 C.6
6		ユーザトレース	
7		Net サーバトレース	
8		動的パフォーマンスビュー	

コネクション ID には次に示す情報が含まれています。

- インスタンス名  
Oracle サーバのインスタンス名が表示されます。
- セッション ID  
Oracle サーバで割り振られるセッション ID が表示されます。
- セッションシリアル番号  
Oracle サーバで割り振られるセッションのシリアル番号が表示されます。
- OS プロセス ID  
OS のプロセス ID が表示されます。

コネクション ID の出力形式 (Oracle の場合)

インスタンス名 : セッション ID : セッションシリアル番号 : OS プロセス名

コネクション ID の出力例 (Oracle の場合)

ORCL:17:5:920

コネクション ID の生成には Oracle の動的パフォーマンスビューを使用するため、Oracle に接続するユーザが動的パフォーマンスビューの参照権を持っている必要があります。次のどちらかの方法で Oracle に接続するユーザに参照権を設定してください。

- 「GRANT SELECT\_CATALOG\_ROLE TO <ユーザ名>;」を実行する。
- 「GRANT SELECT ON V\_\$INSTANCE TO <ユーザ名>;」、「GRANT SELECT ON V\_\$PROCESS TO <ユーザ名>;」、および「GRANT SELECT ON V\_\$SESSION TO <ユーザ名>;」を実行する。

また、Oracle を使用する場合、DB Connector のプロパティ定義で、プロパティ項目「ConnectionIDUpdate」の値に「true」を設定すると、コネクションを取得するたびにコネクション ID を生成することができます。ただし、コネクションを取得するたびにコネクション ID 生成用の SQL が発行されるため、性能に影響を与える可能性があります。再接続が行われない環境では「false」を設定してください。DB Connector のプロパティ定義の設定方法については、マニュアル「Cosminexus アプリケーション設定操作ガイド」の J2EE リソースの設定に関する説明を参照してくだ

さい。

**!** 注意事項

Oracle の透過的アプリケーションフェイルオーバーを使用している場合、PRF トレースに出力されるコネクション ID と実際に接続しているコネクション ID が異なることがあります。これは Oracle 内でコネクションが再接続されているためです。PRF トレースに出力されるコネクション ID と実際に接続しているコネクション ID が異なる場合、コネクション ID で Oracle のトレース情報を追跡することはできません。

## 付録 C.1 Cosminexus Component Container

トランザクション処理の状態（正常または異常）によって、Cosminexus Component Container から出力されるコネクション ID の出力先が異なります。

### 正常の場合

性能解析トレースファイルに出力されます。

### 異常の場合

SQLException 発生時の例外メッセージに出力されます。

性能解析トレースファイルには、次の三つのタイミングでコネクション ID が出力されません。

- データベースとのコネクション取得時  
HiRDB を使用している場合、DataSource.getConnection() メソッドまたは DataSource.getConnection(String username, String password) メソッドの終了直前に、取得したコネクションのコネクション ID を出力します。  
Oracle を使用している場合、DataSource.getConnection() メソッドの終了直前に、取得したコネクションのコネクション ID を出力します。
- データベースのコネクション解放時  
Connection.close() メソッドの開始直後に、このメソッドに対応する getConnection メソッドで取得したコネクションのコネクション ID を出力します。
- アソシエーション機能によってデータベースとのコネクションが差し替えられた時  
ManagedConnection.associateConnection() メソッドの呼び出し時に、アソシエーション機能によって差し替えられた先のコネクションのコネクション ID を出力します。

上記のタイミングで取得したコネクション ID は、次の表に示すイベント ID のトレース情報のインタフェース名に出力されています。

表 C-3 コネクション ID が出力される性能解析トレース

イベント ID	説明
0x8C01	DataSource.getConnection() メソッドの終了直前の処理で出力された情報です。
0x8C03	DataSource.getConnection(String username, String password) メソッドの終了直前の処理で出力された情報です。
0x8C20	Connection.close() メソッドの開始直後の処理で出力された情報です。
0x8C40	ManagedConnection.associateConnection() メソッドの呼び出し時の処理で出力された情報です。

なお、トレース取得ポイントと取得できるトレース情報の詳細については、「付録 A.10 DB Connector, JCA コンテナのトレース取得ポイント」を参照してください。

性能解析トレース、および SQLException 発生時の例外メッセージの取得方法と出力形式について説明します。

### (1) 性能解析トレースファイル

PRF トレースファイルに出力された、クライアントからデータベースなどの EIS に至るまで、およびその処理結果がクライアントに返却されるまでのリクエストの一連の処理で出力されるトレース情報を CSV 形式で編集出力したファイルです。

#### コネクション ID が出力される条件

次の条件をすべて満たしている場合、性能解析トレースファイルにコネクション ID が出力されます。

- 使用しているデータベースが、次のどれかである。  
バージョンが 07-01 以降の HiRDB  
Oracle9i  
Oracle 10g
- 論理パフォーマンストレーサが起動中である。
- リソースアダプタとして DB Connector を使用している。

#### (a) 留意事項

性能解析トレースを参照するときに留意することを次に示します。

##### コネクションの解放に関する留意事項

同一コネクションに対して Connection.close() メソッドを複数回呼び出した場合、その数だけ性能解析トレースが出力されます。

##### アソシエーション機能を使用する場合の留意事項

アソシエーション機能によってデータベースとのコネクションが差し替えられた場合、getConnection() メソッドで出力されたコネクション ID とは異なる物理コネクションでデータベースに接続されます。この場合、性能解析トレースには、ManagedConnection.associateConnection() メソッドの呼び出し時に、アソシエー

ション機能によって差し替えられた先の物理コネクションに対応するコネクション ID が出力されます。したがって、実際にデータベースに接続しているコネクション ID を特定するためには、イベント ID が「0x8C40」の性能解析トレースに出力されたコネクション ID も追跡する必要があります。

なお、ManagedConnection.associateConnection() メソッドを呼び出しても、コネクションが差し替えられない場合があります。この場合、コネクション ID は性能解析トレースに出力されません。

---

参考

アソシエーション機能は、通常、1:1 に対応する論理コネクションと物理コネクションの対応を差し替えて、複数の論理コネクションで一つの物理コネクションを共有する機能です。

---

自動再接続機能を使用する場合の留意事項

HiRDB の自動再接続機能を使用する場合、性能解析トレースファイルに出力されるコネクション ID と、実際の接続に使用しているコネクションのコネクション ID が異なることがあります。この場合、HiRDB クライアントの再接続トレースも参照する必要があります。

(b) 取得方法

運用管理コマンド (mngsvrutil) を実行して性能解析トレースファイルを取得します。性能解析トレースファイルは、「<Manager のログ出力ディレクトリ >%prf」(Windows の場合)、または「<Manager のログ出力ディレクトリ >/prf」(UNIX の場合) に出力されます。取得方法については、「3.2.1 性能解析トレースファイルの収集方法」を参照してください。

(c) 出力形式

性能解析トレースの出力例を次に示します。コネクション ID はインタフェース名(「INT」欄)に出力されます。

図 C-3 性能解析トレースの出力例

	A	B	C	D	E	F	G	T	U	V	
1	PRF	Process	Thread(hashcode)	Trace	ProcessName	Event	Date	ReceiveSCD	INT	OPR	L
2	Rec	1048	2580(5990238)	54	MyServer	0x8c00	2005/	****			*
3	Rec	1048	2580(5990238)	55	MyServer	0x8b00	2005/	****			*
4	Rec	1048	2580(5990238)	56	MyServer	0x8b80	2005/	****			*
5	Rec	1048	2580(5990238)	57	MyServer	0x8b81	2005/	****			*
6	Rec	1048	2580(5990238)	58	MyServer	0x8b01	2005/	****			*
7	Rec	1048	2580(5990238)	59	MyServer	0x8c01	2005/	****	sds01-2-3380		*
8	Rec	1048	2580(5990238)	60	MyServer	0x8c34	2005/	****			*
9	Rec	1048	2580(5990238)	61	MyServer	0x8c35	2005/	****			*
10	Rec	1048	2580(5990238)	62	MyServer	0x8ccc	2005/	****			*
11	Rec	1048	2580(5990238)	63	MyServer	0x8ccd	2005/	****			*
12	Rec	1048	2580(5990238)	64	MyServer	0x8d40	2005/	****			*
13	Rec	1048	2580(5990238)	65	MyServer	0x8d41	2005/	****			*
14	Rec	1048	2580(5990238)	66	MyServer	0x8c20	2005/	****	sds01-2-3380		*
15	Rec	1048	2580(5990238)	67	MyServer	0x8c21	2005/	****			*

## (2) SQLException 発生時の例外メッセージ

データベースアクセス，Cosminexus DABroker Library または JDBC ドライバでエラーが発生した場合に，例外として SQLException がスローされたことを示すメッセージです。

コネクション ID が出力される条件

使用しているデータベースが次のどれかである場合，SQLException 発生時の例外メッセージにコネクション ID が出力されます。

- バージョンが 07-01 以降の HiRDB
- Oracle9i
- Oracle 10g

### (a) 留意事項

SQLException 発生時の例外メッセージを参照するときに留意することを次に示します。

- 出力されるコネクション ID に関する留意事項

SQLException 発生時の例外メッセージに出力されるコネクション ID は常に最新です。HiRDB の自動再接続機能によって再接続が実行された場合，再接続後のコネクション ID が出力されます。

### (b) 取得方法

SQLException 発生時に，次のログファイルに例外メッセージが出力されます。

- Windows の場合
  - <作業ディレクトリ>%ejb%\<サーバ名称>\logs\%cjexception[n].log
  - <作業ディレクトリ>%ejb%\<サーバ名称>\logs\%connectors%\<リソースアダプタ

の表示名 >[n].log

- UNIX の場合
  - <作業ディレクトリ>/ejb/<サーバ名称>/logs/cjexception[n].log
  - <作業ディレクトリ>/ejb/<サーバ名称>/logs/connectors/<リソースアダプタの表示名>[n].log

ログファイル名の [n] の部分には、面の番号（1 から面数（最大 16）まで）が付きます。

### (c) 出力形式

#### HiRDB の場合

メッセージ KFDJ00001-E の ErrMsg の最後にコネクション ID が出力されます。出力例を次に示します。太字はコネクション ID です。

```
JP.co.Hitachi.soft.DBPSV_Driver.SQLException: KFDJ00001-E Error occurred at
server.
[JdbcDbpsvResultSQLExecute.SQLExecute]
OperationType : 2002
ReturnCode    : -100
ErrorCode     : -404
WarningInfo   : 0
ErrorMsg      : KFPAl1404-E Input data too long for column or assignment target
in variable 1 [HiRDB_CONNECTION_ID(sds01:7:2988)]
```

#### Oracle の場合

メッセージ ORA-00942 の ErrMsg の最後にコネクション ID が出力されます。出力例を次に示します。太字はコネクション ID です。

```
JP.co.Hitachi.soft.DBPSV_Driver.SQLException: KFDJ00001-E Error occurred at
server.
[JdbcDbpsvResultSQLExecute.SQLPrepare]
OperationType : 2002
ReturnCode    : -200
ErrorCode     : 942
WarningInfo   : 0
ErrorMsg      : ORA-00942: 表またはビューが存在しません。
[ORACLE_CONNECTION_ID(ORCL:17:5:920)]
PreparedSQL   : selectSectionID      : 2
```

## 付録 C.2 Cosminexus DABroker Library

Cosminexus DABroker Library では、拡張データベースアクセストレースにコネクション ID が出力されます。

拡張データベースアクセストレースの取得方法と出力形式について説明します。

### (1) 拡張データベースアクセストレース

データベースへの接続から切断までのアクセス情報を出力したトレースです。データベースへの接続ごとに出力されます。



拡張データベースアクセストレースにコネクション ID が出力されるためには、次の条件を満たしている必要があります。

コネクション ID が出力される条件 (HiRDB の場合)

- HiRDB クライアントのバージョンが 07-01 以降である。
- Cosminexus DABroker Library 動作環境定義ファイルの DABEXSQL\_TRC\_LINE に 0 以上の値を指定している。

なお、Cosminexus DABroker Library の動作環境の設定については、マニュアル「Cosminexus 簡易構築・運用ガイド」を参照してください。Cosminexus DABroker Library 動作環境定義ファイルの詳細については、マニュアル「Cosminexus リファレンス 定義編」を参照してください。

コネクション ID が出力される条件 (Oracle の場合)

- Oracle9i, または Oracle10g を使用している。
- Cosminexus DABroker Library 動作環境定義ファイルの DABEXSQL\_TRC\_LINE E に -1, 0, または 1024 から 32767 の値を指定している。

なお、Windows の場合、Cosminexus DABroker Library 動作環境定義ファイルの設定は、環境設定ユティリティを使用します。環境設定ユティリティを使用した Cosminexus DABroker Library の動作環境の設定については、マニュアル「Cosminexus 簡易構築・運用ガイド」を参照してください。Cosminexus DABroker Library 動作環境定義ファイルの詳細については、マニュアル「Cosminexus リファレンス 定義編」を参照してください。

(a) 取得方法

拡張データベースアクセストレースの格納場所は次のとおりです。

- Windows の場合  
 <Cosminexus DABroker Library の運用ディレクトリ>¥spool¥db\_access ディレクトリ下のファイル
- UNIX の場合  
 /opt/DABroker/spool/db\_access ディレクトリ下のファイル

(b) 出力形式

拡張データベースアクセストレースの出力形式を次に示します。

なお、拡張データベースアクセストレースの出力形式の詳細については、「5.5.3(2) 拡張データベースアクセストレース」を参照してください。

HiRDB の場合

「HiRDB\_CONNECTION\_ID」欄にコネクション ID が出力されます。

- Windows の場合

```

*-----*
*- DataBase Access Information (DRV)                -*
*- Date  YYYY/MM/DD hh:mm:ss.nnnnnn                -*
*-----*
DABroker Connect ID   : AAAAA(BBBBB)
Process ID            : CCCCC
UserID               : DDDDD
Client Name          : EEEEE
(DB別出力情報)
Lang Mode            : GGGGG

THREAD-ID  CID  EVT  START-TIME  END-TIME  RETCODE  BLOCKCNT  (Windows
QueryPerformance Counter) HiRDB_CONNECTION_ID
(トレース情報)
(SQL)SS...SS
(トレース情報)

```

• UNIX の場合

```

*-----*
*- DataBase Access Information (DRV)                -*
*- Date  YYYY/MM/DD hh:mm:ss.nnnnnn                -*
*-----*
DABroker Connect ID   : AAAAA(BBBBB)
Process ID            : CCCCC
UserID               : DDDDD
Client Name          : EEEEE
(DB別出力情報)
Lang Mode            : GGGGG

THREAD-ID  CID  EVT  START-TIME  END-TIME  RETCODE
BLOCKCNT  HiRDB_CONNECTION_ID
(トレース情報)
(SQL)SS・S
(トレース情報)

```

Oracle の場合

「ORACLE\_CONNECTION\_ID」欄にコネクション ID が出力されます。

```

*-----*
*- DataBase Access Infomation (ORACLE8i Driver)    -*
*- Date  YYYY/MM/DD hh:mm:ss.nnnnnn                -*
*-----*
DABroker Connect ID   : AAAAAAAA(BBBBB)
Process ID            : CCCC
UserID               : DDDDDD
Client Name          : EEEEE
SQLNET              : FFFFFF
Lang Mode            : GGGG
*-----*
THREAD-ID  CID  EVT  START-TIME  END-TIME
RETCODE  BLOCKCNT  ORACLE_CONNECTION_ID
(トレース情報)
(SQL) SELECT H FROM I
(トレース情報)

```

## 付録 C.3 HiRDB クライアント

HiRDB クライアントでは、SQL トレースファイル、エラーログファイル、および再接続トレースファイルに、コネクション ID やコネクト通番などの情報が出力されます。

SQL トレースファイル、エラーログファイル、および再接続トレースファイルの取得方法と出力形式について説明します。

### (1) SQL トレースファイル

実行した UAP の SQL トレース情報を出力したトレースファイルです。SQL の実行終了時に出力されます。

UAP 実行時に SQL エラーが発生した場合、SQL トレースファイルを参照すると、エラーの原因となる SQL を特定できます。

コネクション ID が出力される条件

次の条件を満たしている場合、SQL トレースファイルにコネクション ID が出力されます。

- ・ クライアント環境定義の PDCLTPATH および PDSQLTRACE に値を指定している。

なお、クライアント環境定義の設定方法については、マニュアル「HiRDB UAP 開発ガイド」を参照してください。

#### (a) 取得方法

クライアント環境定義の PDCLTPATH に指定したディレクトリに格納されます。

#### (b) 出力形式

次に示す形式で、サーバ名称、コネクト通番、およびサーバのプロセス ID が出力されます。

```
(省略)
:
CONNECTION STATUS :
  CURHOST ( 接続先ホスト名 )  CURPORT ( 接続ポート番号 )  SRVNAME ( サーバ名称 )
  CNCTNO ( コネクト通番 )  SVRPID ( サーバのプロセス ID )  CLTPID ( UAP のプロセス ID )
  CLTTID ( UAP のスレッド番号 )
:
(省略)
```

SQL トレースファイルには、SQL 文、SQL 実行時刻や SQL 文中の変数に設定した値なども出力されます。出力形式の詳細については、マニュアル「HiRDB UAP 開発ガイド」を参照してください。

## (2) エラーログファイル

クライアントと HiRDB サーバ間の通信処理中、または X/Open で規定した XA インタフェースでエラーが発生した場合のエラー情報を出力したログファイルです。SQL 実行時、通信処理時、または X/Open で規定した XA インタフェース関数実行時でエラーが発生したときに出力されます。

### コネクション ID が出力される条件

次の条件を満たしている場合、エラーログファイルにコネクション ID が出力されません。

- クライアント環境定義の PDCLTPATH および PDUAPERLOG に値を指定している。

なお、設定方法については、マニュアル「HiRDB UAP 開発ガイド」を参照してください。

### (a) 取得方法

クライアント環境定義の PDCLTPATH に指定したディレクトリに格納されます。

### (b) 出力形式

次に示す形式で、SQL トレースにサーバのプロセス ID が出力されます。

エラーログ先頭識別子 ('>>'または'>') UAPのプロセスID UAPのスレッド番号 サーバのプロセスID エラーログカウンタ (以降省略)
---

出力形式の詳細については、マニュアル「HiRDB UAP 開発ガイド」を参照してください。

エラーログ情報には、接続情報のほか、エラー取得時刻、SQLCODE やエラーが発生した SQL のオペレーションコードなどが出力されます。

## (3) 再接続トレースファイル

HiRDB クライアントの自動再接続機能で再接続が実行された場合、HiRDB が内部的に管理している接続ハンドルの値、再接続前の接続情報、再接続後の接続情報、および再接続時刻を出力したトレースファイルです。自動再接続機能で自動的に接続が実行された場合にも出力されます。

### コネクション ID が出力される条件

次の条件を満たしている場合、再接続トレースファイルにコネクション ID が出力されます。

- クライアント環境定義の PDCLTPATH および PDSQLTRACE に値を指定している。
- HiRDB の自動再接続機能を使用している。

なお、設定方法については、マニュアル「HiRDB UAP 開発ガイド」を参照してください。

(a) 取得方法

クライアント環境定義 PDCLTPATH に指定したディレクトリに再接続トレースファイルが格納されます。格納されるファイル名称は、pdrenct1.trc および pdrenct2.trc です。

(b) 出力形式

再接続トレースは次の形式で出力されます。

```
接続ハンドルの値 再接続結果 (SまたはF) 再接続開始日時 - 再接続完了日時 再接続前の接続情報
=> 再接続後の接続情報
```

コネクション ID は再接続後の接続情報として出力されます。

出力形式の詳細については、マニュアル「HiRDB UAP 開発ガイド」を参照してください。

再接続トレースファイルの出力例を次に示します。**太字**はコネクション ID です。

```
40004250 S 2004/04/12 11:10:36.766 - 2004/04/12 11:10:41.846 sds:9:23763 =>
sds:10:23750
40004250 S 2004/04/12 11:11:07.491 - 2004/04/12 11:11:12.547 sds:10:23750 =>
sds:11:23765
40004850 F 2004/04/12 11:17:58.285 - 2004/04/12 11:18:23.395 sds:14:23751 =>
40005050 S 2004/04/12 11:27:35.098 - 2004/04/12 11:27:40.152 sds:1:24414 =>
sds:2:24418
```

## 付録 C.4 HiRDB サーバ

HiRDB サーバでは、pdls コマンドの実行結果を参照してサーバの状態を確認します。

pdls コマンドの実行結果に表示されるサーバ名称やサーバのプロセス ID と、コネクション ID に含まれる情報を突き合わせて、接続した HiRDB サーバの状態を確認します。

pdls コマンドの実行形式と、実行結果の出力形式について説明します。

### (1) pdls コマンドの実行結果

HiRDB サーバの排他制御の状態、プロセスの状態、通信制御情報などを表示できるコマンドです。

(a) 実行形式

次に示す pdls コマンドを実行して、サーバの状態を確認します。

サーバの排他制御の状態表示する場合

```
pdls -d lck
```

サーバのプロセスの状態表示する場合

```
pdls -d prc
```

サーバの通信制御情報の表示する場合

```
pdls -d rpc
```

pdls コマンドの詳細については、マニュアル「HiRDB コマンドリファレンス」を参照してください。

(b) 出力形式

実行結果の「SVID」欄にサーバ名称が出力されて、「PID」欄にサーバのプロセス ID が出力されます。

出力形式の詳細については、マニュアル「HiRDB コマンドリファレンス」を参照してください。

## 付録 C.5 Oracle クライアント

Oracle クライアントでは、Net クライアントトレースファイルに、実行されたネットワークイベントの詳細が出力されます。Net クライアントトレースファイルのファイル名にはクライアントの OS クライアントプロセス ID が含まれています。

Net クライアントトレースファイルは次の場所に出力されます。

- Windows の場合  
<sqlnet.ora ファイルで指定した場所 >¥CLI\_<OS クライアントプロセス ID>.trc
- UNIX の場合  
<sqlnet.ora ファイルで指定した場所 >/CLI\_<OS クライアントプロセス ID>.trc

注 Net クライアントトレースファイルの出力先は、Oracle の sqlnet.ora ファイルで設定します。sqlnet.ora ファイルの記述形式を次に示します。なお、sqlnet.ora ファイルの詳細については、Oracle のドキュメントを参照してください。

```
TRACE_LEVEL_CLIENT=16
TRACE_DIRECTORY_CLIENT= <ディレクトリ名>
TRACE_UNIQUE_CLIENT=ON
TRACE_TIMESTAMP_CLIENT=ON
```

Net クライアントトレースファイルを参照するには、出力された Net クライアントトレースファイルの中から、目的の Net クライアントトレースファイルを特定する必要があります。目的の Net クライアントトレースファイルを特定する手順を次に示します。

1. 次の SQL を実行して、OS クライアントプロセス ID を取得します。

```
SELECT PROCESS FROM V$SESSION WHERE (SID = <セッションID>) AND
(SERIAL# = <セッションシリアル番号>);
```

2. 取得した OS クライアントプロセス ID と Net クライアントトレースファイルのファイル名に含まれる OS クライアントプロセス ID を突き合わせて、目的の Net クライアントトレースファイルを特定します。

なお、Net クライアントトレースファイルの出力内容の詳細については、Oracle のドキュメントを参照してください。

### ! 注意事項

Net クライアントトレースは、大量のディスク領域を消費するため、システムの性能が低下する場合があります。必要なときにだけ Net クライアントトレースを参照してください。

## 付録 C.6 Oracle サーバ

Oracle サーバでは、次の方法で Oracle サーバのエラー情報、サーバの状態などの情報が出力されます。

- アラートファイルの取得
- ユーザトレースの取得
- Net サーバトレースファイルの取得
- 動的パフォーマンスビューの使用

### (1) アラートファイル

アラートファイルでは、性能解析トレースのインスタンス名をキーにして、次の情報を確認できます。

- 発生したすべての内部エラー、ブロック破損エラー、およびデッドロックエラー
- CREATE/ALTER/DROP 文、STARTUP/SHUTDOWN 文、ARCHIVELOG 文などの管理、操作
- 共有サーバとディスパッチャプロセスの機能に関するメッセージ、およびエラー
- マテリアライズドビューの自動リフレッシュ中に発生したエラー
- データベースおよびインスタンス起動時の、すべての初期化パラメタの値

アラートファイルは、次の場所に出力されます。

- Windows の場合

<BACKGROUND\_DUMP\_DEST で指定した場所 >¥ALERT\_< インスタンス名

>.LOG

- UNIX の場合

<BACKGROUND\_DUMP\_DEST で指定した場所 >/ALERT\_< インスタンス名

>.LOG

注 Oracle の初期化パラメタ USER\_DUMP\_DEST で設定します。

USER\_DUMP\_DEST の詳細については、Oracle のドキュメントを参照してください。

出力されたアラートファイルの中から、性能解析トレースに出力されたインスタンス名とアラートファイル名に含まれるインスタンス名を突き合わせて、目的のアラートファイルを特定してください。

## (2) ユーザトレース

ユーザトレースでは、性能解析トレースのインスタンス名、OS プロセス ID、セッション ID、セッションシリアル番号をキーにして、次の情報を確認できます。

- サーバプロセスで発生したエラー情報
- SQL 分の実行計画と統計情報

## (3) Net サーバトレースファイル

Oracle サーバでは、Net サーバトレースファイルに、実行されたネットワークイベントの詳細が出力されます。Net サーバトレースファイルのファイル名にはサーバの OS プロセス ID が含まれています。

Net サーバトレースファイルは次の場所に出力されます。

- Windows の場合

<sqlnet.ora ファイルで指定した場所 >¥CLI\_<OS プロセス ID>.TRC

- UNIX の場合

<sqlnet.ora ファイルで指定した場所 >/CLI\_<OS プロセス ID>.TRC

注 Net サーバトレースファイルの出力先は、Oracle の sqlnet.ora ファイルで設定します。sqlnet.ora ファイルの記述形式を次に示します。なお、sqlnet.ora ファイルの詳細については、Oracle のドキュメントを参照してください。

```
TRACE_LEVEL_CLIENT=16
TRACE_DIRECTORY_CLIENT=<ディレクトリ名>
TRACE_UNIQUE_CLIENT=ON
TRACE_TIMESTAMP_CLIENT=ON
```

出力された Net サーバトレースファイルの中から、性能解析トレースの OS プロセス ID と Net サーバトレースファイル名に含まれる OS プロセス ID 名を突き合わせて、目的の Net サーバトレースファイルを特定してください。



なお、Net サーバトレースファイルの出力内容の詳細については、Oracle のドキュメントを参照してください。

### ! 注意事項

Net サーバトレースファイルは、大量のディスク領域を消費するため、システムの性能が低下する場合があります。必要なときにだけ Net クライアントトレースを参照してください。

## (4) 動的パフォーマンスビュー

動的パフォーマンスビューを調査して、セッションを特定したり、プロセスの詳細についての情報を参照したりできます。動的パフォーマンスビューは、性能解析トレースファイルに出力されるコネクション ID と突き合わせて調査します。

動的パフォーマンスビューと性能解析トレースのコネクション ID の関係を次の表に示します。

表 C-4 動的パフォーマンスビューと性能解析トレースのコネクション ID の関係

項番	動的パフォーマンスビューに表示される項目	性能解析トレースのコネクション ID の項目
1	V\$INSTANCE INSTANCE_NAME	インスタンス名
2	V\$SESSION SID	セッション ID
3	V\$SESSION SERIAL#	セッションシリアル番号
4	V\$PROCESS SPID	OS プロセス ID

動的パフォーマンスビューの詳細については、Oracle のドキュメントを参照してください。

---

## 付録 D 機能ごとに出力されるログ情報

特定の機能を利用したときに出力されるログ情報について説明します。

ここでは、次の機能を利用したときに出力されるログ情報について説明します。

J2EE アプリケーション実行時間監視

### 付録 D.1 J2EE アプリケーション実行時間監視で出力されるログ情報

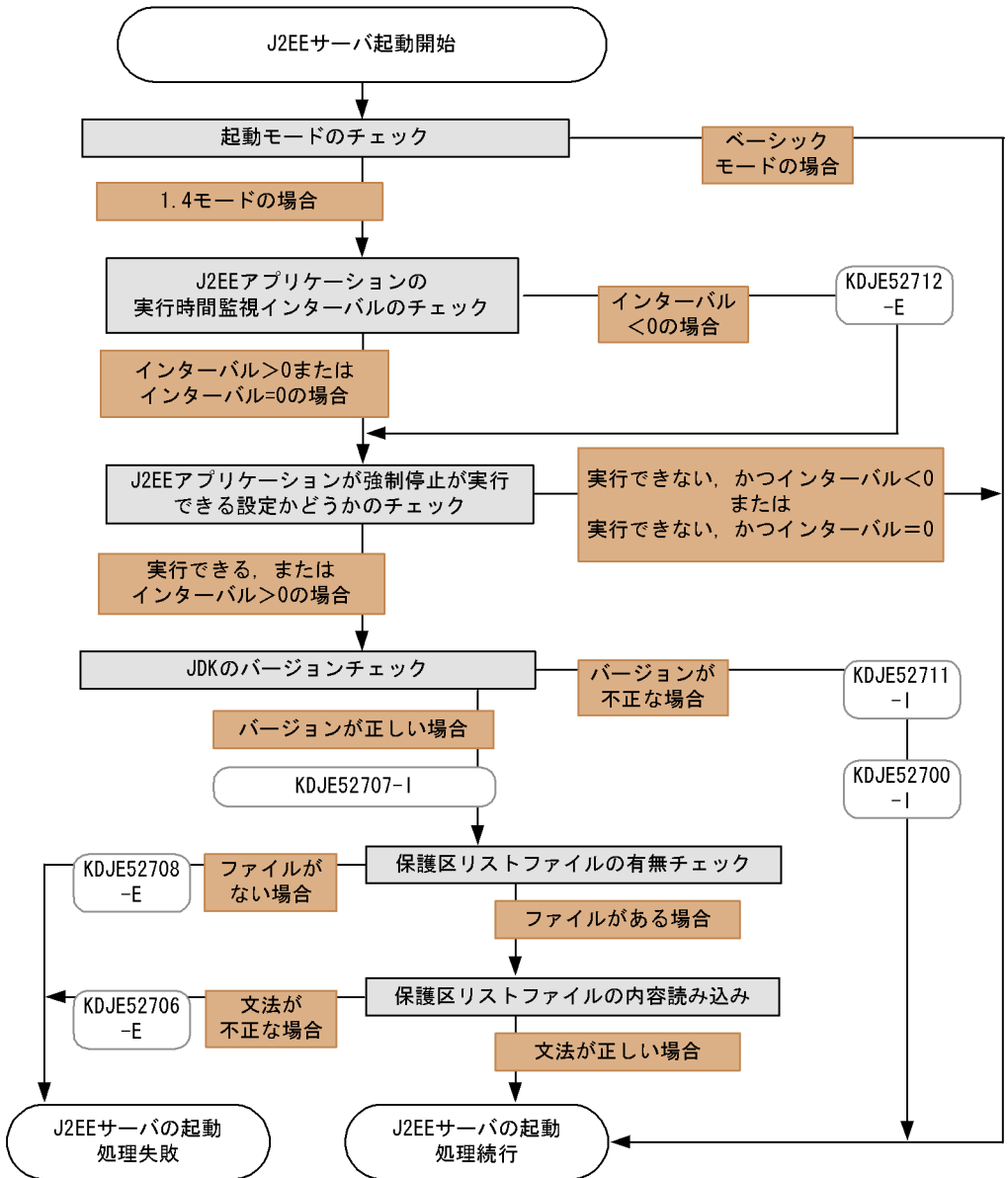
ここでは、J2EE アプリケーションの実行時間監視機能で出力されるログ情報について説明します。

#### (1) J2EE サーバ起動時に出力されるログ情報

J2EE サーバ起動時には、プロパティの読み込み、JavaVM のバージョンチェック、保護区リストファイルの読み込みが実行されます。

このときに出力されるログ情報を次の図に示します。

図 D-1 J2EE サーバ起動時に出力されるログ情報



(凡例)

: ログ出力のトリガ

## (2) タイムアウトの延長でメソッドキャンセルを実行した場合に出力されるログ情報

タイムアウトが発生して、その処理の延長でメソッドキャンセルが実行された場合に出力されるログ情報を次の図に示します。どのメッセージが出力された場合も、処理は続

行されます。なお、メソッドキャンセルモードが有効か無効かによって、スレッドダンプの出力回数が異なります。

メソッドキャンセルモードが有効の場合

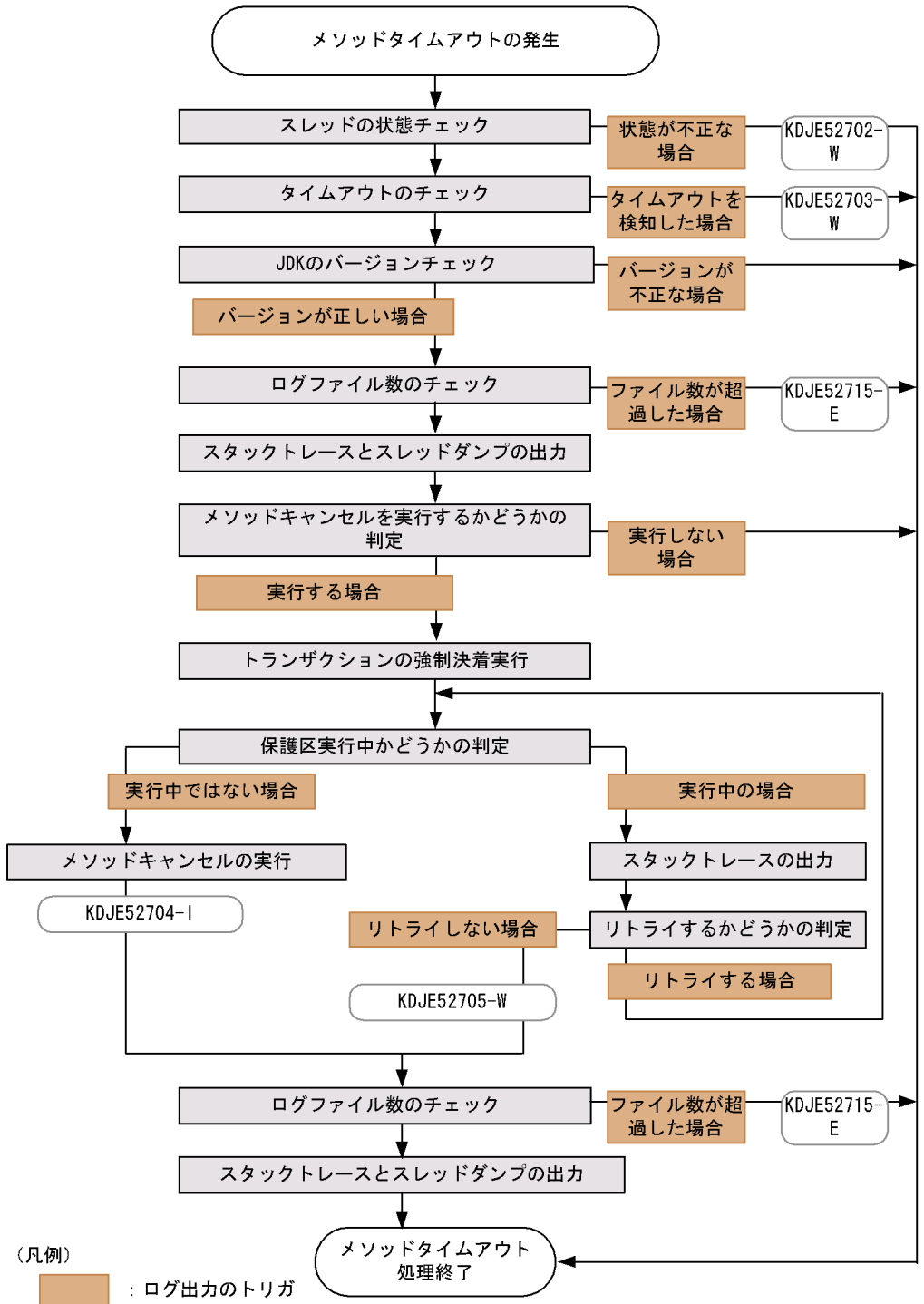
タイムアウト発生時と、メソッドキャンセルの成功時または失敗時に、2 回出力されます。

メソッドキャンセルモードが無効の場合

タイムアウト発生時に 1 回だけ出力されます。

また、タイムアウトを検知したメソッドが終了した場合には、KDJE52716-I が出力されます。これは、メソッドキャンセルの実行が成功したかどうかとは関係ありません。また、実行しないで終了した場合にも出力されます。つまり、一つのメソッドの実行について KDJE52703-W と KDJE52716-I は、1 対 1 で出力されます。

図 D-2 タイムアウトの延長でメソッドキャンセルを実行した場合に出力されるログ情報

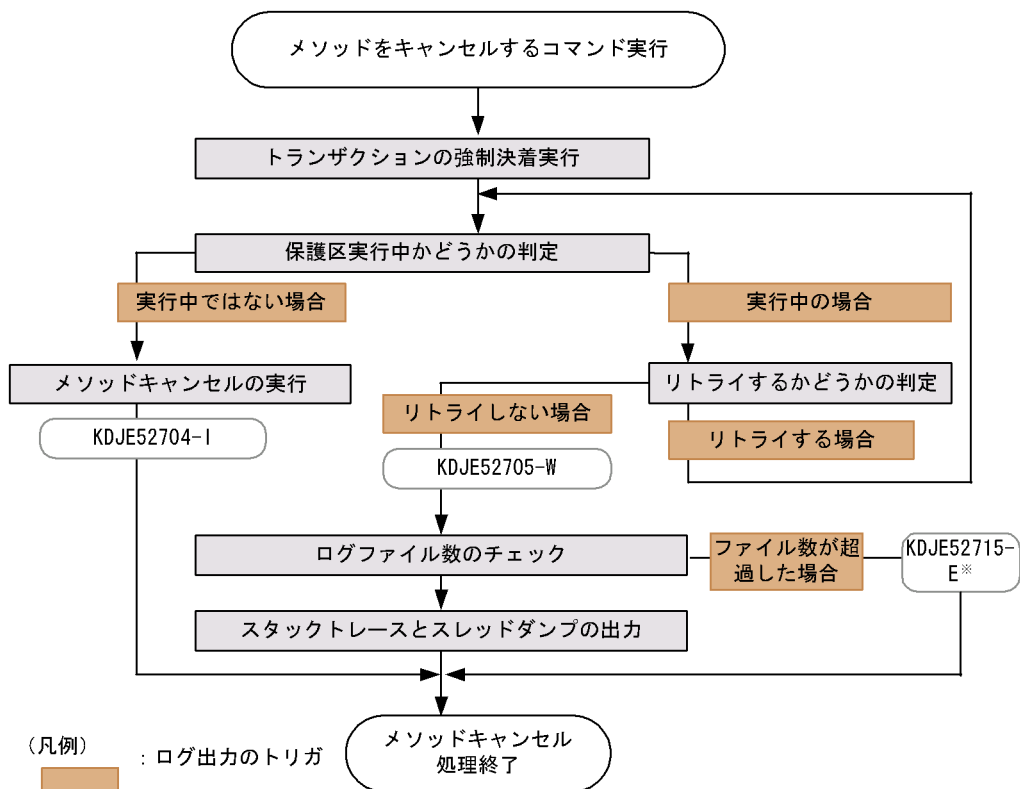


### (3) コマンドによってメソッドキャンセルを実行した場合に出力されるログ情報

タイムアウトが発生したメソッドを、J2EE アプリケーションの強制停止またはメソッドキャンセルコマンド (cjstophthread) を実行してキャンセルする場合に出力されるログ情報を次の図に示します。どのメッセージが出力された場合も、処理は続行されます。

なお、スレッドダンプはメソッドキャンセルが失敗した場合に 1 回だけ出力されます。また、メソッドキャンセルコマンドの実行に失敗した場合は、メッセージ KDJE52713-E が出力されます。

図 D-3 コマンドによってメソッドキャンセルを実行した場合に出力されるログ情報



注※

usrconf.properties で ejbserver.server.threaddump.filenum に指定した値よりも多くのログファイルを出力しようとした場合、メッセージ「KDJE52715-E」が出力されて、スタックトレースとスレッドダンプは出力されません。

### (4) 注意事項

J2EE サーバでのスレッドダンプファイル数の上限チェック後、スレッドダンプ出力要求と JavaVM でのスレッドダンプ出力処理は非同期に実行されるため、タイムラグが発生することがあります。このため、スレッドダンプファイルが上限値を超えて出力される

ことがあります。

スレッドダンプの出力数の上限値は、「設定したスレッドダンプの上限値 +9」になります。

また、JavaVM のスレッドダンプ出力処理では、同時に複数の出力要求があった場合、スレッドダンプを出力できないことがあります。この場合には、スレッドダンプの数が、実際に出力しようとした数よりも少なくなります。

## 付録 E 障害発生時の CMR 用の表の回復

ここでは、CMR を含むアプリケーションをデプロイしたあと J2EE サーバで障害が発生した場合に、障害発生前に使用していた CMR 用の表を回復させる手順について説明します。なお、CMR 用の表の詳細については、マニュアル「Cosminexus アプリケーション設定操作ガイド」を参照してください。

CMR を含むアプリケーションがデプロイされ、運用していた状態で保守などのために J2EE サーバを停止、起動したときに障害が発生すると、CMR を含むアプリケーションがデプロイされた状態で立ち上がらないことが考えられます。障害を解決し、再度 CMR を含むアプリケーションをデプロイしようとしても、作成する表と同名の表がデータベース上にある場合、デプロイができません（アプリケーション間で表の共有を避けるため）。

ここで、SQL 生成を再度実行すると、新しい CMR 用の表名を使用した SQL が生成され、新しい CMR 用の表を使用するようにデプロイができます。しかし、J2EE サーバを停止する以前に使用していた関係を使用したい場合、データベースに残っている表を使うようにデプロイする必要があります。

簡易構築定義ファイルの論理 J2EE サーバ (j2ee-server) の <configuration> タグ内に ejbserver.ejb.cmp20.cmr.use.existing\_table パラメタで指定します。

ejbserver.ejb.cmp20.cmr.use.existing\_table は、アプリケーションの起動障害発生時、それまで使用していた関係の情報を回復させるためのオプションです。true を指定すると、データベース既存の表を使用するようにデプロイができます。このオプションを使用して、既存の表を使用する場合の手順を次に示します。

1. 障害発生前に使用していた CMR 用の表がデータベース上に残っていることを、SQL を使用して確認してください。
2. J2EE サーバを停止します（アプリケーションがデプロイ状態で立ち上がることに失敗した原因の対処をしてください）。
3. 簡易構築定義ファイルの論理 J2EE サーバ (j2ee-server) の <configuration> タグ内に ejbserver.ejb.cmp20.cmr.use.existing\_table パラメタで次のように指定します。

```
<param-name> タグ
    ejbserver.ejb.cmp20.cmr.use.existing_table

<param-value> タグ
    true
```

4. J2EE サーバを起動します。
5. デプロイ状態での開始に失敗した CMR を含むアプリケーションを再度デプロイします。



**!** 注意事項

ここで SQL 生成を再実行しないでください。再実行すると新しい表名で SQL が生成され、以前の表が使用できなくなります。

---

6. J2EE サーバを停止します。
7. 簡易構築定義ファイルの論理 J2EE サーバ (j2ee-server) の <configuration> タグ内に ejbserver.ejb.cmp20.cmr.use.existing\_table パラメタで false を指定するか、またはこのオプションを設定しない状態に戻します。
8. 再度 J2EE サーバを起動します。

## 付録 F ベーシックモードの利用（互換機能）

ベーシックモードの場合、プロセスの依存関係とプロセスの再起動方法が J2EE サーバモードの場合と異なります。プロセスの依存関係とプロセスの再起動方法を次に示します。

### 付録 F.1 ベーシックモードの場合のプロセスの依存関係

ベーシックモードの場合のプロセスの依存関係を次の表に示します。

表 F-1 ベーシックモードの場合のプロセスの依存関係

プロセスの種類	前提プロセス
パフォーマンスストレサ	-
スマートエージェント <sup>1</sup>	-
CTM ドメインマネージャ <sup>1</sup>	スマートエージェント
CORBA ネーミングサービス	-
CTM デーモン <sup>1</sup>	<ul style="list-style-type: none"> <li>• パフォーマンスストレサ</li> <li>• スマートエージェント</li> <li>• CTM ドメインマネージャ</li> <li>• ネーミングサービス</li> </ul>
SFO サーバ	<ul style="list-style-type: none"> <li>• パフォーマンスストレサ</li> </ul>
J2EE サーバ	<ul style="list-style-type: none"> <li>• パフォーマンスストレサ</li> <li>• スマートエージェント<sup>2</sup></li> <li>• CTM ドメインマネージャ<sup>2</sup></li> <li>• CORBA ネーミングサービス</li> <li>• CTM デーモン<sup>2</sup></li> </ul>
Web サーバ	-

（凡例）

- : 前提プロセスがない。

注 1

CTM を使用する場合に起動するプロセスです。

注 2

CTM を使用する場合に前提となるプロセスです。

### 付録 F.2 ベーシックモードの場合のプロセスの再起動方法

ベーシックモードの場合のシステムでプロセスが異常終了時のプロセスの再起動方法について説明します。

なお、CTM に関連するプロセスが異常終了して再起動できない場合は、次の手順で対処

してください。

ベーシックモードの場合のプロセスの再起動 (回復) の手順を次に示します。

表 F-2 ベーシックモードの場合の再起動の手順

異常プロセス	開始コマンド	再起動 (回復) 手順
データベースサーバ	-	DB サーバを再起動してください。
パフォーマンストレーサ	cprfstart	パフォーマンストレーサを再起動してください。
スマートエージェント <sup>1</sup>	osagent	スマートエージェントを再起動してください。
CTM ドメインマネージャ 1	ctmdmstart	CTM ドメインマネージャを再起動してください。
CORBA ネーミングサービス	nameserv	次に示す手順で対処してください。なお、CTM を使用しない場合、手順 1 および手順 4 は不要です。 1. CTM デーモンの強制停止 2. J2EE サーバの強制停止 3. CORBA ネーミングサービスの再起動 4. CTM デーモンの再起動 5. J2EE サーバ再起動
CTM デモン <sup>1</sup>	ctmstart	次に示す手順で対処してください。 1. CORBA ネーミングサービスの強制停止 2. J2EE サーバの強制停止 3. CORBA ネーミングサービスの再起動 4. CTM デーモンの再起動 5. J2EE サーバの再起動
CTM レギュレータ	-	CTM デモンによって自動再起動されるため、再起動は不要です。
SFO サーバ <sup>2</sup>	cjstartsv	SFO サーバを再起動してください。
J2EE サーバ	cjstartsv	J2EE サーバを再起動してください。
Web サーバ	-	Web サーバを再起動してください。

(凡例)

- : 使用する製品によって開始コマンドが異なる, または該当する開始コマンドがない。

注 1

CTM を使用する場合に起動するプロセスです。

注 2

セッションフェイルオーバー機能を使用する場合に起動するプロセスです。

## 付録 G サブレットエンジンモードでのシステム運用

ここでは、サブレットエンジンモードを使用してシステムを運用する場合の方法について説明します。

### 付録 G.1 Component Container 管理者による運用に関する留意事項

UNIX で Component Container 管理者を設定している場合の留意事項について説明します。Component Container 管理者とは、UNIX を使用している場合に、Web コンテナサーバの構築・運用に関する操作を実行できる権限を持つユーザです。Component Container 管理者を設定することで、root 権限を持たないユーザでも、Web コンテナサーバの起動・停止やサーバ管理コマンドの実行など、システム運用時の操作を実行できるようになります。

#### ！ 注意事項

Component Container 管理者を設定している場合、Component Container 管理者が実行できる操作は、一部の操作を除き、スーパーユーザでは操作できなくなるので注意してください。詳細については、マニュアル「Cosminexus 機能解説」の、J2EE サーバの運用とシステムの運用管理者に関する説明を参照してください。

なお、Component Container 管理者の設定については、マニュアル「Cosminexus システム構築ガイド」の Component Container 管理者の設定に関する説明を参照してください。

また、Component Container 管理者を設定している場合にパーミッションが変更されるディレクトリについては、「1.8.1 (2) Component Container 管理者を設定している場合にパーミッションが変更されるディレクトリ」を参照してください。

Component Container 管理者が実施できるシステム運用時の操作について説明します。

#### (1) Component Container 管理者が実施できるシステム運用時の操作

Component Container 管理者を設定した場合に、スーパーユーザおよび Component Container 管理者が実施できるシステム運用時の操作を次の表に示します。

表 G-1 Component Container 管理者が実施できるシステム運用時の操作

操作	スーパーユーザ	Component Container 管理者
Web コンテナサーバの起動 / 停止 (cjstartweb コマンド, cjstopweb コマンド)	×	

操作	スーパーユーザ	Component Container 管理者
サーバ管理コマンドの実行		
スレッドダンプの取得 (cjdumpweb コマンド)		
OS 状態情報の取得 (cjgetsysinfo コマンド)		

(凡例) : できる    × : できない

なお, Component Container 管理者を設定しない場合は, スーパーユーザが運用管理者となるため, Component Container 管理者の操作もスーパーユーザが実施します。

## 付録 G.2 システムの起動

ここでは, サブレットエンジンモードでのシステムの起動の手順について説明します。

なお, データベース, LDAP ディレクトリサーバなどの関連製品は起動済みであることを前提に説明します。データベース, LDAP ディレクトリサーバなどを使用している場合は, 必要に応じてあらかじめ起動しておいてください。システムの起動手順を次に示します。

1. PRF デーモンを起動します。  
cprfstart コマンドで PRF デーモンを起動します。起動方法については, 「(1) PRF デーモンの起動」を参照してください。
2. Web コンテナサーバを起動します。  
cjstartweb コマンドで Web コンテナサーバを起動します。起動方法については, 「(2) Web コンテナサーバの起動」を参照してください。
3. Web サーバを起動します。  
起動方法については, ご使用の Web サーバのマニュアルを参照してください。
4. Web アプリケーションの動作を確認します。  
システムが起動できたら, Web アプリケーションが正常に動作することを確認してください。

### (1) PRF デーモンの起動

PRF デーモンの起動手順を次に示します。

1. cd コマンドでカレントディレクトリを移動します。
  - Windows の場合

```
cd "<Cosminexusのインストールディレクトリ>%PRF%bin"
```

- UNIX の場合

```
# cd /opt/Cosminexus/PRF/bin
```

2. cprfstart コマンドで PRF デーモンを起動します。

- Windows の場合

```
cprfstart
```

- UNIX の場合

```
# cprfstart
```

なお、cprfstart コマンドの詳細については、マニュアル「Cosminexus リファレンス コマンド編」を参照してください。

## (2) Web コンテナサーバの起動

Web コンテナサーバの起動方法について説明します。

Web コンテナサーバを起動するには、次に示すコマンドを実行します。

- Windows の場合

```
"<Cosminexusのインストールディレクトリ>%CC%web%bin%cjstartweb サーバ名称"
```

なお、Windows の場合、コマンドの実行には、Administrator 権限が必要です。

- UNIX の場合

```
"/opt/Cosminexus/CC/web/bin/cjstartweb <サーバ名称>"
```

なお、UNIX の場合、コマンドの実行には、root 権限 (Component Container 管理者を設定していない場合)、または Component Container 管理者の権限 (Component Container 管理者を設定している場合) が必要です。

同一ホストで複数の Web コンテナを同時に起動する場合のポート番号については、Web コンテナサーバのセットアップで設定します。Web コンテナサーバのセットアップについては、マニュアル「Cosminexus システム構築ガイド」を参照してください。

なお、SecurityManager による Web コンテナの保護をするには、-security オプションを追加してください。

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

## 付録 G.3 システムの停止

ここでは、サブレットエンジンモードのシステムの停止手順について説明します。

なお、データベース、LDAP ディレクトリサーバなどを使用している場合は、必要に応じて停止してください。システムの停止手順を次に示します。

1. Web サーバを停止します。  
停止方法については、ご使用の Web サーバのマニュアルを参照してください。
2. Web コンテナサーバを停止します。  
cjstopweb コマンドで Web コンテナサーバを停止します。停止方法については、「(1) Web コンテナサーバの停止」を参照してください。
3. PRF デーモンを停止します。  
cprfstop コマンドで PRF デーモンを停止します。停止方法については、「(2) PRF デーモンの停止」を参照してください。

### (1) Web コンテナサーバの停止

Web コンテナサーバの停止方法について説明します。

Web コンテナサーバを停止するには、次に示すコマンドを実行します。

- Windows の場合

```
"<Cosminexusのインストールディレクトリ>%CC%\web\bin\cjstopweb サーバ名称"
```

なお、Windows の場合、コマンドの実行には、Administrator 権限が必要です。

- UNIX の場合

```
"/opt/Cosminexus/CC/web/bin/cjstopweb <サーバ名称>"
```

なお、UNIX の場合、コマンドの実行には、root 権限 (Component Container 管理者を設定していない場合)、または Component Container 管理者の権限 (Component Container 管理者を設定している場合) が必要です。

Web コンテナサーバを停止しても、Web コンテナサーバの停止処理が完了しない場合があります。この場合、Web コンテナサーバを強制的に停止する必要があります。Web コンテナサーバを強制停止する場合、cjstopweb コマンドに -f オプションまたは -fd オプションを指定して実行します。-fd オプションを指定すると、Web コンテナサーバの障害情報を取得したあと、Web コンテナサーバを強制停止します。

なお、cjstopweb コマンドの詳細については、マニュアル「Cosminexus リファレンス コマンド編」を参照してください。

## (2) PRF デーモンの停止

PRF デーモンの停止手順を次に示します。

1. cd コマンドでカレントディレクトリを移動します。

- Windows の場合

```
cd "<Cosminexusのインストールディレクトリ>%PRF%bin"
```

- UNIX の場合

```
# cd /opt/Cosminexus/PRF/bin
```

2. cprfstop コマンドで PRF デーモンを停止します。

cprfstart コマンドを実行したユーザが実行してください

- Windows の場合

```
cprfstop
```

- UNIX の場合

```
# cprfstop
```

PRF デーモンを停止しても、PRF デーモンの停止処理が完了しない場合があります。この場合、PRF デーモンを強制的に停止する必要があります。PRF デーモンを強制停止する場合、cprfstop コマンドに -Force オプションを指定して実行します。

なお、cprfstop コマンドの詳細については、マニュアル「Cosminexus リファレンス コマンド編」を参照してください。

## 付録 G.4 システムの処理性能の解析

ここでは、サブレットエンジンモードで動作するシステムでの処理性能の解析方法について説明します。

Cosminexus で構築したシステムの処理性能は、クライアントからデータベースなどの EIS まで、およびその処理結果がクライアントに返却されるまでのリクエストの一連の処理で、リダイレクタや Web コンテナなどから出力されるトレース情報を基に解析できます。トレース情報を出力するリダイレクタや Web コンテナなどを、機能レイヤといいます。サブレットエンジンモードで動作するシステムでのリクエストの処理過程では、次に示す機能レイヤからトレース情報が出力されます。

- リダイレクタ



- Web コンテナ

各機能レイヤでは決まった処理のポイント（トレース取得ポイント）でトレース情報を出力します。各機能レイヤが出力するトレース情報は、パフォーマンストレーサ（Cosminexus Performance Tracer）の PRF デーモンによって、トレースファイル（PRF トレースファイル）に出力されます。PRF デーモンにはトレース取得レベル（標準または詳細）を設定できます。PRF デーモンに設定するトレース取得レベルを PRF トレース取得レベルといいます。各機能レイヤのトレース取得ポイントおよび PRF トレース取得レベルの詳細については、「付録 A トレース取得ポイントと PRF トレース取得レベル」を参照してください。

なお、各機能レイヤのトレース情報はバイナリ形式で PRF トレースファイルに出力されます。システムの処理性能を解析する際は、PRF トレースファイルをテキスト形式（CSV 形式）に変換したファイル（性能解析トレースファイル）を使用します。

サブレットエンジンモードの場合、システムの処理性能を解析する作業は、パフォーマンストレーサのコマンドで実行します。

システムの処理性能を解析する場合に実施する作業を次に示します。詳細については、参照先の説明を参照してください。

表 G-2 システムの処理性能を解析する場合に実施する作業（サブレットエンジンモードの場合）

作業の目的	作業の概要	パフォーマンストレーサのコマンド	参照先
性能解析トレースファイルを収集する	PRF トレースファイルを出力して、CSV 形式に変換します。	cprfflush cprfed	-
収集した性能解析トレースファイルを利用してシステムの処理性能を解析する	Web サーバのレスポンスタイムを解析します。	-	3.3.1
	アプリケーションサーバ内でのリクエストの処理状況を調査します。	-	3.3.2
	セッションのライフサイクルを調査します。	-	3.3.3
	タイムアウトしたトランザクションを特定します。	-	3.3.4
	タイムアウトしたリクエストを特定します。	-	3.3.5
	ルートアプリケーション情報を利用してログを調査します。	-	3.3.6
	データベースとの接続中にトラブルが発生したコネクションを特定します。	-	3.3.7

（凡例） - : 該当しません。

注 cprfed コマンドを使用して出力された性能解析トレースファイルを使用します。

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

## 付録 G.5 トラブルシューティング

ここでは、サブレットエンジンモードでシステムを運用している場合に発生したトラブルに対処する方法について説明します。

### (1) 対処の手順 (サブレットエンジンモードの場合)

ここでは、Cosminexus のシステムの運用中にトラブルが発生した場合の、対処の手順について説明します。

なお、トラブル発生時に取得する情報のうち、次に示す情報を取得するためには、運用開始前の準備が必要です。

日立 JavaVM 固有ログ

このファイルには、JavaVM のガーベージコレクションのログも出力されます。

ユーザダンプ (Windows の場合)

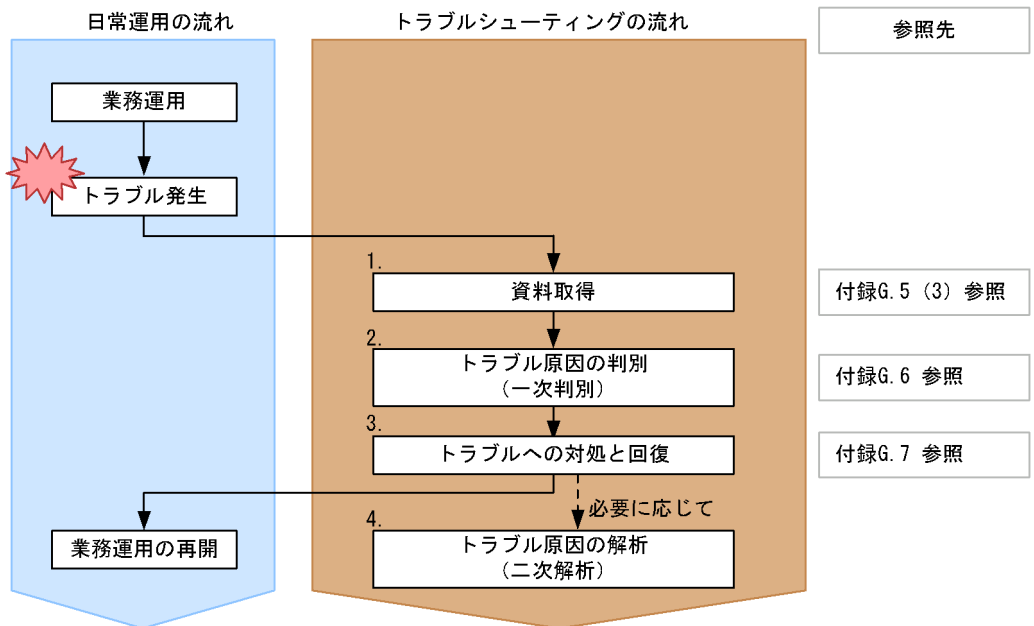
core ダンプ (UNIX の場合)

OS の統計情報 (Windows の場合)

これらの情報については、必要に応じてシステム構築時に取得準備をしてください。トラブルシューティングの資料取得の設定については、マニュアル「Cosminexus システム構築ガイド」を参照してください。

システムの運用中にトラブルが発生した場合、次の図に示す手順で対処します。

図 G-1 トラブル発生時の対処の流れ



#### 1. 資料取得

トラブルシューティングに必要な資料をすべて取得します。

取得が必要な資料については、「(2) 取得が必要な資料の種類」を参照してください。なお、メモリダンプは、Web コンテナサーバを再起動する場合にだけ必要な資料です。

また、資料の取得方法については、「(3) 資料の取得」を参照してください。

#### 2. トラブル原因の判別 (一次判別)

トラブルの原因を判別します。判別には、取得した資料を使用します。資料の調査方法については、「付録 G.6 資料の調査」を参照してください。

#### 3. トラブルへの対処と回復

トラブルに対処して、回復します。

なお、トラブルの種類ごとの対処方法については、「付録 G.7 トラブルへの対処と回復」を参照してください。

#### 4. トラブル原因の解析 (二次解析)

必要に応じてトラブル原因をさらに分析します。なお、この解析作業は保守員が実施します。

### (2) 取得が必要な資料の種類

ここでは、トラブル発生時に取得が必要な資料の種類を示します。

トラブル発生時に取得が必要な資料の種類については、「5.2 取得が必要な資料の種類」を参照してください。

ただし、サブレットエンジンモードの場合、snapshot ログの取得方法と、デフォルトの設定で snapshot ログとして取得できる情報の種類が異なります。

また、資料を個別で取得するとき、次の資料の取得内容または取得方法が、運用管理ポータルを使用して構築したシステムの場合と異なります。

Cosminexus のログ

性能解析トレース

### (3) 資料の取得

ここでは、トラブル発生時の資料の取得方法について説明します。

トラブルシュートに必要な資料は、次のどちらかの方法で取得できます。

snapshotlog コマンドを利用して一括取得する

それぞれの情報を個別に収集する

snapshot ログは snapshotlog コマンドを実行して収集します。snapshotlog コマンドの詳細については、マニュアル「Cosminexus リファレンス コマンド編」を参照してください。

それぞれの情報を個別に収集する場合、次の資料を取得します。

Cosminexus のログ

- Cosminexus Component Container のログ
- Cosminexus Performance Tracer のログ
- Cosminexus DABroker Library のログ
- Web アプリケーションのログ

性能解析トレース

JavaVM のスレッドダンプ

JavaVM のガーベージコレクションログ

メモリダンプ

Cosminexus の定義情報

Cosminexus の作業ディレクトリの内容

Web サーバログ

そのほかの情報

それぞれの情報の取得方法については「付録 G.9 資料の取得方法」を参照してください。

## 付録 G.6 資料の調査

ここでは、トラブルシューティングで、取得した資料を調査して、トラブルの原因を判別する方法について説明します。

トラブルシューティングに必要な資料の調査方法は、J2EE サーバモードの場合と同様です。

資料の調査方法については、「5.5 資料の調査」を参照してください。

## 付録 G.7 トラブルへの対処と回復

ここでは、発生したトラブルの内容に応じた対処方法および回復方法について説明します。

ここでは、次のようなトラブルが発生した場合の対処について説明します。

構成ソフトウェアのプロセスが異常終了した場合

データベースと接続中にトラブルが発生した場合

JavaVM が異常終了した場合

### (1) 構成ソフトウェアのプロセスが異常終了した場合（サブレットエンジンモードの場合）

システムの運用中に構成ソフトウェアが異常終了して再起動を実行する場合、異常終了した構成ソフトウェアの起動順序に依存関係があるときは、前提となる構成ソフトウェアが起動していることを確認してから再起動する必要があります。前提となる構成ソフトウェアは、システムの運用形態によって異なります。以降で、構成ソフトウェアの起動順序の依存関係、および構成ソフトウェアの再起動方法について説明します。

#### (a) 構成ソフトウェアの起動順序の依存関係

Cosminexus で構築したシステムを構成する構成ソフトウェアの起動順序の依存関係について説明します。

サブレットエンジンモードの場合のプロセスの依存関係を次に示します。

表 G-3 サブレットエンジンモードの場合のプロセスの依存関係

プロセスの種類	前提プロセス
PRF デーモン	-
Web コンテナサーバ	PRF デーモン
Web サーバ	-

#### (凡例)

- : 前提プロセスがない。

(b) プロセスの再起動方法

システムでプロセスが異常終了した場合のプロセスの再起動方法について説明します。

サブレットエンジンモードの場合のプロセスの再起動（回復）の手順は次のとおりです。

表 G-4 サブレットエンジンモードの場合の再起動の手順

異常プロセス	開始コマンド	再起動（回復）手順
データベースサーバ	-	DBサーバを再起動してください。
PRF デモン	cprfstart	パフォーマンストレーサを再起動してください。
Web コンテナサーバ	cjstartweb	Web コンテナサーバを再起動してください。
Web サーバ	-	Web サーバを再起動してください。

(凡例)

- :使用する製品によって開始コマンドが異なる,または該当する開始コマンドがない。

(2) データベースと接続中にトラブルが発生した場合（サブレットエンジンモードの場合）

データベースとの接続中に, Cosminexus の構成ソフトウェアである Cosminexus DABroker Library でトラブルが発生した場合, Cosminexus DABroker Library によってデータベースとの接続は切り離されます。また, Cosminexus DABroker Library を強制終了した場合も同様です。

これらのトラブルへの対処方法は, J2EE サーバモードの場合と同様です。対処方法については, 「5.6.4 データベースと接続中にトラブルが発生した場合」を参照してください。

(3) JavaVM が異常終了した場合（サブレットエンジンモードの場合）

JavaVM では, 次のようなトラブルが発生する場合があります。

C ヒープ不足

内部論理エラー

OutOfMemoryError

これらのトラブルへの対処方法は, J2EE サーバモードの場合と同様です。対処方法については, 「5.6.5 JavaVM が異常終了した場合」を参照してください。

## 付録 G.8 トラブルシューティングに関連する留意事項

トラブルシューティングに関連する留意事項は、J2EE サーバモードの場合と同様です。留意事項については、「5.7 トラブルシューティングに関連する留意事項」を参照してください。

## 付録 G.9 資料の取得方法

### (1) Cosminexus Component Container のログ

Cosminexus Component Container のログとして取得する資料の出力先を次に示します。それぞれの資料の詳細については、「5.4.1 Cosminexus のログの取得 (J2EE アプリケーションを実行するシステム)」または「5.4.2 Cosminexus のログの取得 (バッチアプリケーションを実行するシステム)」を参照してください。

表 G-5 Web コンテナサーバおよびリダイレクタのログの出力先 (デフォルト)

分類	内容	ログ出力先およびログファイル名 <sup>1</sup>	デフォルトのサイズ×面数	チャンネル名
メッセージログ	稼働ログ	<ul style="list-style-type: none"> <li>Windows の場合 &lt;Cosminexus のインストールディレクトリ&gt;¥CC¥web¥containers¥&lt;サーバ名称&gt;¥logs¥cjweb_message[n].log</li> <li>UNIX の場合 /opt/Cosminexus/CC/web/containers/&lt;サーバ名称&gt;/logs/cjweb_message[n].log</li> </ul>	1MB × 2	WebMessageLogFile
	リダイレクタのメッセージログ	ディレクトリ <sup>2</sup> <ul style="list-style-type: none"> <li>Windows の場合 &lt;Cosminexus のインストールディレクトリ&gt;¥CC¥web¥redirector¥logs</li> <li>UNIX の場合 /opt/Cosminexus/CC/web/redirector/logs</li> </ul> ファイル名 <ul style="list-style-type: none"> <li>Windows の場合 isapi_redirect[n].log (Microsoft IIS) hws_redirect[n].log (Hitachi Web Server)</li> <li>UNIX の場合 hws_redirect[n].log (Hitachi Web Server)</li> </ul>	4MB × 5	-
ユーザログ	Web サブレットログ <sup>3</sup>	<ul style="list-style-type: none"> <li>Windows の場合 &lt;Cosminexus のインストールディレクトリ&gt;¥CC¥web¥containers¥&lt;サーバ名称&gt;¥logs¥web_servlet[n].log</li> <li>UNIX の場合 /opt/Cosminexus/CC/web/containers/&lt;サーバ名称&gt;/logs/web_servlet[n].log</li> </ul>	4MB × 4	WebServletLogFile

分類	内容	ログ出力先およびログファイル名 <sup>1</sup>	デフォルトのサイズ×面数	チャンネル名
	ユーザ出力ログ <sup>4</sup>	<ul style="list-style-type: none"> <li>Windows の場合 &lt;Cosminexus のインストールディレクトリ&gt;%CC%web%containers%\&lt;サーバ名称&gt;%logs%\user_out[n].log</li> <li>UNIX の場合 /opt/Cosminexus/CC/web/containers/&lt;サーバ名称&gt;/logs/user_out[n].log</li> </ul>	1MB × 2	UserOutLogFile
	ユーザエラーログ <sup>4</sup>	<ul style="list-style-type: none"> <li>Windows の場合 &lt;Cosminexus のインストールディレクトリ&gt;%CC%web%containers%\&lt;サーバ名称&gt;%logs%\user_err[n].log</li> <li>UNIX の場合 /opt/Cosminexus/CC/web/containers/&lt;サーバ名称&gt;/logs/user_err[n].log</li> </ul>	1MB × 2	UserErrLogFile
	JavaVMの保守情報およびガーベージコレクションのログ	<ul style="list-style-type: none"> <li>Windows の場合 &lt;Cosminexus のインストールディレクトリ&gt;%CC%web%containers%\&lt;サーバ名称&gt;%logs%\javalog[nn].log</li> <li>UNIX の場合 /opt/Cosminexus/CC/web/containers/&lt;サーバ名称&gt;/logs/javalog[nn].log</li> </ul>	256KB × 4	-
例外ログ	障害発生時の例外情報	<ul style="list-style-type: none"> <li>Windows の場合 &lt;Cosminexus のインストールディレクトリ&gt;%CC%web%containers%\&lt;サーバ名称&gt;%logs%\cjweb_exception[n].log</li> <li>UNIX の場合 /opt/Cosminexus/CC/web/containers/&lt;サーバ名称&gt;/logs/cjweb_exception[n].log</li> </ul>	12MB × 4	WebExceptionLogFile
保守用ログ	保守情報	<ul style="list-style-type: none"> <li>Windows の場合 &lt;Cosminexus のインストールディレクトリ&gt;%CC%web%containers%\&lt;サーバ名称&gt;%logs%\cjweb_maintenance[n].log</li> <li>UNIX の場合 /opt/Cosminexus/CC/web/containers/&lt;サーバ名称&gt;/logs/cjweb_maintenance[n].log</li> </ul>	1MB × 2	WebMaintenanceLogFile
	Webコンテナの保守情報	<ul style="list-style-type: none"> <li>Windows の場合 &lt;Cosminexus のインストールディレクトリ&gt;%CC%web%containers%\&lt;サーバ名称&gt;%logs%\cjwebcontainer[n].log</li> <li>UNIX の場合 /opt/Cosminexus/CC/web/containers/&lt;サーバ名称&gt;/logs/cjwebcontainer[n].log</li> </ul>	1MB × 2	WebContainerLogFile



分類	内容	ログ出力先およびログファイル名 <sup>1</sup>	デフォルトのサイズ×面数	チャンネル名
起動プロセス標準出力情報 <sup>5</sup>	<ul style="list-style-type: none"> <li>Windows の場合 &lt;Cosminexus のインストールディレクトリ&gt;¥CC¥web¥containers¥&lt; サーバ名称&gt;¥logs¥cjstdout.log</li> <li>UNIX の場合 /opt/Cosminexus/CC/web/containers/&lt; サーバ名称 &gt;/logs/cjstdout.log</li> </ul>	-	-	
起動プロセス標準エラー情報 <sup>5</sup>	<ul style="list-style-type: none"> <li>Windows の場合 &lt;Cosminexus のインストールディレクトリ&gt;¥CC¥web¥containers¥&lt; サーバ名称&gt;¥logs¥cjstderr.log</li> <li>UNIX の場合 /opt/Cosminexus/CC/web/containers/&lt; サーバ名称 &gt;/logs/cjstderr.log</li> </ul>	-	-	
終了プロセス情報	<ul style="list-style-type: none"> <li>Windows の場合 &lt;Cosminexus のインストールディレクトリ&gt;¥CC¥web¥containers¥&lt; サーバ名称&gt;¥logs¥cjweb_shutdown[n].log</li> <li>UNIX の場合 /opt/Cosminexus/CC/web/containers/&lt; サーバ名称 &gt;/logs/cjweb_shutdown[n].log</li> </ul>	1MB × 2	WebShutdownLogFile	
Web コンテナ保守用トレースログ	<ul style="list-style-type: none"> <li>Windows の場合 &lt;Cosminexus のインストールディレクトリ&gt;¥CC¥web¥containers¥&lt; サーバ名称&gt;¥logs¥cjweb_access[n].log</li> <li>UNIX の場合 /opt/Cosminexus/CC/web/containers/&lt; サーバ名称 &gt;/logs/cjweb_access[n].log</li> </ul>	2MB × 16	WebAccessLogFile	
リダイレクタの保守用トレースログ	<p>ディレクトリ<sup>2</sup></p> <ul style="list-style-type: none"> <li>Windows の場合 &lt;Cosminexus のインストールディレクトリ&gt;¥CC¥web¥redirector¥logs</li> <li>UNIX の場合 /opt/Cosminexus/CC/web/redirector/logs</li> </ul> <p>ファイル名<sup>2</sup></p> <ul style="list-style-type: none"> <li>Windows の場合 hws_rd_trace[n].log ( Hitachi Web Server ) isapi_rd_trace[n].log ( Microsoft IIS )</li> <li>UNIX の場合 hws_rd_trace[n].log</li> </ul>	16MB × 2	-	

注

チャンネル名は、ログの出力先を識別する名称です。ログの属性（サイズ、面数）を変更する場合のキー値として使用します。

注 1

[ n ]: ログファイル名の [ n ] 部分には、面の番号（1 から面数（最大 16）まで）が付きます。また、[ nn ] の部分には 01 ~ 99 の通し番号が付きます。

注 2

ディレクトリ、ファイル名は、isapi\_redirect.conf（Microsoft IIS 用リダイレクタ動作定義ファイル）、または mod\_jk.conf（Hitachi Web Server 用リダイレクタ動作定義ファイル）で設定できます。リダイレクタのログ取得の設定については、マニュアル「Cosminexus システム構築ガイド」を参照してください。各ファイルについては、マニュアル「Cosminexus リファレンス 定義編」を参照してください。

注 3

サブレット、JSP で発生した例外のスタックトレースについても出力されます。

注 4

標準出力にも出力されます。また、サブレット、JSP での System.out、および System.err への出力はユーザ出力ログ、およびユーザエラーログに出力されます。

注 5

起動プロセス情報だけを取得するログです。主に Web コンテナサーバの起動または終了時に出力されるため、オンライン中に出力されることはほとんどありません。ファイルのサイズが上限に達したときは、<Cosminexus のインストールディレクトリ>¥CC¥web¥containers¥<サーバ名称>¥logs 下（Windows の場合）、または /opt/Cosminexus/CC/web/containers/<サーバ名称>/logs 下（UNIX の場合）の cjstdout\_save.log または cjstderr\_save.log に退避されます。すでに cjstdout\_save.log または cjstderr\_save.log があつたときは、上書きされます。

**!** 注意事項

Web サーバとして Microsoft IIS 6.0 を使用している場合、リダイレクタのログファイルは、Web サーバへの初回アクセス時に、isapi\_redirect.conf で指定したファイルサイズで、指定した面数分、すべて作成されます。このとき、トレースファイルは空白（0x20）で初期化されます。デフォルトのファイルサイズと面数で運用している場合、ログの出力先に指定したディレクトリに、52 メガバイトのディスク容量が必要になります。

なお、Web コンテナサーバおよびリダイレクタのログの出力先を変更している場合は、次の表に示す、ログの出力先を設定するユーザ定義ファイルを参照して出力先を確認してください。

表 G-6 ログの出力先を設定するユーザ定義ファイル（サブレットエンジンモードの場合）

分類	ユーザ定義ファイルの格納場所
Web コンテナサーバ	格納場所は、usrconf.cfg ファイルの web.server.log.directory キーで指定します。キーのデフォルトは、「<Cosminexus のインストールディレクトリ>¥CC¥web¥containers¥cosmi_m¥logs」（Windows の場合）、または「/opt/Cosminexus/CC/web/containers/cosmi_m/logs」（UNIX の場合）です。

分類	ユーザ定義ファイルの格納場所
リダイレクタ (Windows の場合)	<p>Hitachi Web Server の場合</p> <p>[ メッセージログ ]  &lt;Cosminexus のインストールディレクトリ  &gt;%CC%web¥redirector¥mod_jk.conf の JkLogFileDir キー  キーのデフォルトは、「&lt;Cosminexus のインストールディレクトリ  &gt;%CC%web¥redirector¥logs」です。</p> <p>[ 保守用トレースログ ]  &lt;Cosminexus のインストールディレクトリ  &gt;%CC%web¥redirector¥mod_jk.conf の JkTraceLogFileDir キー  キーのデフォルトは、「&lt;Cosminexus のインストールディレクトリ  &gt;%CC%web¥redirector¥logs」です。</p> <p>Microsoft IIS の場合</p> <p>[ メッセージログ ]  &lt;Cosminexus のインストールディレクトリ  &gt;%CC%web¥redirector¥isapi_redirect.conf の log_file_dir キー  キーのデフォルトは、「&lt;Cosminexus のインストールディレクトリ  &gt;%CC%web¥redirector¥logs」です。</p> <p>[ 保守用トレースログ ]  &lt;Cosminexus のインストールディレクトリ  &gt;%CC%web¥redirector¥isapi_redirect.conf の trace_log_file_dir  キー  キーのデフォルトは、「&lt;Cosminexus のインストールディレクトリ  &gt;%CC%web¥redirector¥logs」です。</p>
リダイレクタ (UNIX の場合)	<p>メッセージログ  /opt/Cosminexus/CC/web/redirector/mod_jk.conf の JkLogFileDir  キー  キーのデフォルトは、「/opt/Cosminexus/CC/web/redirector/logs」で  す。</p> <p>保守用トレースログ  /opt/Cosminexus/CC/web/redirector/mod_jk.conf の  JkTraceLogFileDir キー  キーのデフォルトは、「/opt/Cosminexus/CC/web/redirector/logs」で  す。</p>

ログの出力先の変更方法など、トラブルシューティングの資料取得の設定については、マニュアル「Cosminexus システム構築ガイド」を参照してください。

#### (a) Cosminexus Performance Tracer のログ

Cosminexus Performance Tracer のログの出力先を次に示します。

それぞれの資料の詳細については、「5.4.1 Cosminexus のログの取得 (J2EE アプリケーションを実行するシステム)」または「5.4.2 Cosminexus のログの取得 (バッチアプリケーションを実行するシステム)」を参照してください。

表 G-7 Cosminexus Performance Tracer のログの出力先

内容	出力先ディレクトリ
PRF デーモンおよび PRF コマンドのログ	<ul style="list-style-type: none"> <li>Windows の場合 &lt;環境変数 PRFSPOOL の設定ディレクトリ &gt;%log%\PRF 識別子 &gt;%ctmlog[n]</li> <li>UNIX の場合 \$PRFSPOOL/log/&lt;PRF 識別子 &gt;/ctmlog[n]</li> </ul>
モジュールトレース	<ul style="list-style-type: none"> <li>Windows の場合 &lt;環境変数 PRFSPOOL の設定ディレクトリ &gt;%utt%\umt</li> <li>UNIX の場合 \$PRFSPOOL/utt/umt</li> </ul>
構造化例外発生ログ	<ul style="list-style-type: none"> <li>Windows の場合 &lt;環境変数 PRFSPOOL の設定ディレクトリ &gt;%osltrc</li> <li>UNIX の場合 \$PRFSPOOL/osltrc</li> </ul>

注

[n] には「01」または「02」が表示されます。

#### (b) Cosminexus DABroker Library のログ

それぞれの資料の詳細については、「5.4.1 Cosminexus のログの取得 (J2EE アプリケーションを実行するシステム)」または「5.4.2 Cosminexus のログの取得 (バッチアプリケーションを実行するシステム)」を参照してください。

Cosminexus DABroker Library では、トラブルシューティングに使用するログとして、スプール情報を取得します。

次のディレクトリ下のすべてのファイルをコピーして保存してください。

- Windows の場合  
<Cosminexus DABroker Library の運用ディレクトリ >%spool
- UNIX の場合  
<Cosminexus DABroker Library の運用ディレクトリ >/spool

## 参考

Cosminexus DABroker Library では、スプール情報のほかに次の JDBC ドライバのトレースも取得できます。

- JDBC インタフェースメソッドトレース
- Cosminexus DABroker Library とのイベントトレース

ただし、JDBC ドライバで次のトレースを取得するためには、アプリケーション開発時にあらかじめ指定されている必要があります。トレースを取得するための情報の設定については、マニュアル「Cosminexus 機能解説」の Cosminexus DABroker Library を使用したデータベース接続の実装に関する説明を参照してください。

## (c) Web アプリケーションのログ

Web アプリケーションのログは、日立トレース共通ライブラリ形式で出力するように設定している場合に取得できます。

Web アプリケーションのログの出力先については、usrconf.properties (Web コンテナサーバ用ユーザプロパティファイル) の ejbserver.application.userlog.CJLogHandler.<ハンドラ名>.path キーの値を参照してください。なお、<ハンドラ名>には、キーの値を区別するためのハンドラ名が指定されています。

usrconf.properties ファイルの格納場所を次に示します。

- Windows の場合  
 <Cosminexus のインストールディレクトリ>%CC%\web\containers\<サーバ名称>\usrconf\usrconf.properties
- UNIX の場合  
 /opt/Cosminexus/CC/web/containers/<サーバ名称>/usrconf/usrconf.properties

なお、Web アプリケーションのユーザログ出力の設定については、マニュアル「Cosminexus システム構築ガイド」のサブレットエンジンモードの利用に関する説明を参照してください。

## (2) 性能解析トレースの取得 (サブレットエンジンモードの場合)

性能解析トレースの収集方法については、「3. システムの処理性能の解析」を参照してください。セッショントレースも同様に取得できます。

性能解析トレースの格納場所は次のとおりです。

- Windows の場合  
 <環境変数 PRFSPOOL の設定ディレクトリ>%utt%\prf\- UNIX の場合  
 \$PRFSPOOL/utt/prf/<PRF 識別子>

### (3) JavaVM のスレッドダンプの取得 (サブレットエンジンモードの場合)

ここでは、JavaVM のスレッドダンプを取得する方法について説明します。

なお、Web コンテナサーバの起動オプションとして `-XX:+HitachiThreadDump` が設定されている場合、拡張スレッドダンプを取得できます。

`-XX:+HitachiThreadDumpToStdout` が設定されている場合、スレッドダンプは標準出力にも出力されます。

また、`-XX:+HitachiOutOfMemoryAbort` が設定されている場合に、`-XX:+HitachiOutOfMemoryAbortThreadDump` が設定されていると、`OutOfMemoryError` によって強制終了した時に、スレッドダンプが出力されます。ただし、次の場合は除きます。

- J2SE クラスライブラリ内で C ヒープ不足が発生した場合
- JavaVM の処理中に C ヒープ不足が発生した場合

`cjstartweb` プロセスが存在する場合のスレッドダンプの取得方法を次に示します。

#### Windows の場合

`cjstartweb` プロセスが存在する場合、次を実行します (3 秒おきに 10 回程度)。

```
<Cosminexusのインストールディレクトリ>%CC%web%bin%cjdumpweb <Webコンテナサーバ名>
```

#### UNIX の場合

`cjstartweb` プロセスが存在する場合、`cjdumpweb` コマンドまたは `kill` コマンドを使用して Web コンテナサーバのスレッドダンプを取得します (3 秒おきに 10 回程度)。

- `cjdumpweb` コマンドを使用する場合

```
/opt/Cosminexus/CC/web/bin/cjdumpweb <Webコンテナサーバ名>
```

- `kill` コマンドを使用する場合

```
ps -ef | grep cjstartweb  
kill -3 <WebコンテナサーバのプロセスID>
```

スレッドダンプは、「`/opt/Cosminexus/CC/web/containers/<サーバ名称>/javacore*.txt`」、およびサーバ標準出力ログ (`cjstdout.log`) に出力されます。

また、javacore<プロセス番号>.<コマンド実行日時>.txt ファイルの出力先は、環境変数 JAVACOREDİR で変更できます。ただし、指定したディレクトリへの書き込みに失敗した場合は、カレントディレクトリに出力されます。このディレクトリにも出力できなかった場合は、標準エラー出力にだけ出力されます。

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

#### 参考

スレッドダンプが出力されると、標準出力に、次のメッセージが出力されます。そのあと、java プログラムの実行が継続されます。このメッセージは、-XX:+HitachiThreadDumpToStdout の設定とは関係なく出力されます。

```
Writing Java core to <ファイル名(フルパス)>...OK
```

#### (4) JavaVM のガーベージコレクションログの取得 (サブレットエンジンモードの場合)

JavaVM のガーベージコレクションログは、JavaVM または Web コンテナサーバ起動前にログ出力先が設定されている場合にだけ取得できます。

ユーザ定義ファイルを使用して設定している場合は、usrconf.cfg ファイルの web.server.log.directory のキーを確認してください。

また、実行中の Java プロセスでガーベージコレクションを発生させたい場合は、javagc コマンドを実行してください。javagc コマンドは、トラブル発生時の要因調査のほか、1 トランザクション当たりのメモリ使用量の測定、メモリリークの調査、アプリケーションのデバッグなどの目的でも使用できます。

実行中の Java プロセスを指定してガーベージコレクションを発生させる場合、javagc コマンドは、次の形式で実行します。なお、これ以外に指定できるオプションについては、マニュアル「Cosminexus リファレンス コマンド編」を参照してください。

```
javagc -p プロセスID
```

実行結果として、次のログが出力されます。なお、これは、オプションとして、-XX:+HitachiVerboseGCPrintCause が指定されている場合の例です。

- Windows の場合

```
[VGC]<Wed Mar 17 00:42:30 2004>(Skip Full:0,Copy:0)[ Full GC
149K->149K(1984K),0.0786038
secs] [ DefNew::Eden:264K->0K(512K)] [ DefNew::Survivor:0K->63K(64K)] [
Tenured:
85K->149K(1408K)] [ Perm:1300K->1300K(1536K)] [ cause:System.gc]
```

- UNIX の場合

```
[VGC]<Wed Mar 17 00:42:30 2004>(Skip Full:0,Copy:0)[ Full GC
149K->149K(1984K),0.0786038
secs] [ DefNew::Eden:264K->0K(512K)] [ DefNew::Survivor:0K->63K(64K)] [
Tenured:
85K->149K(1408K)] [ Perm:1300K->1300K(1536K)] [ cause:JavaGC Command]
```

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

なお、-XX:+HitachiVerboseGCPrintCause オプションは、usrconf.cfg に指定します。オプションの設定方法については、マニュアル「Cosminexus リファレンス 定義編」を参照してください。

### (5) メモリダンプの取得 (Web コンテナサーバを再起動する場合)

メモリダンプの取得の取得方法について OS ごとに説明します。

#### (a) Windows の場合

Web コンテナサーバを再起動するときに、メモリダンプを取得します。メモリダンプは、システムでトラブルが発生したときに、保守員が障害解析するために使用します。

Web コンテナサーバが稼働している場合 (cjstartweb プロセスが存在する場合)、タスクマネージャから cjstartweb のプロセス ID を取得します。userdump.exe <プロセス ID> を実行し、メモリダンプ (user.dump) を採取します。

#### (b) UNIX の場合

Web コンテナサーバを再起動するときに、メモリダンプとして、Web コンテナサーバの core ダンプを取得します。メモリダンプは、システムでトラブルが発生したときに、保守員が障害解析するために使用します。

cjstartweb プロセスがダウンした場合、「/opt/Cosminexus/CC/web/containers/cosmi\_m/core\*」に出力された core ダンプを取得します。

core ダンプのファイル名は、cjstartweb プロセスの再起動時に「core.<出力日時 >」(AIX, HP-UX, Solaris の場合)または「core.<プロセス ID>.<出力日時 >」(Linux の場合)にリネームされます。cjstartweb プロセスの再起動時に core ダンプが上書き保存されないため、障害発生時の core ダンプを保存できます。

注



出力日時は「YYMMDDhhmmss」の形式で出力されます。

YY：西暦（下2けた） MM：月（2けた） DD：日（2けた）

hh：時（2けた [24時間表記]） mm：分（2けた） ss：秒（2けた）

なお、保存する core ダンプは上限値を設定できます。cjstartweb プロセスの再起動時に「/opt/Cosminexus/CC/web/containers/<サーバ名称>/」に出力される core ダンプファイルの合計が上限値を超えた場合、出力日時が古い順に削除されます。上限値は、usrconf.cfg の web.server.corefilenum キーに指定します。なお、ファイルの削除は cjstartweb プロセスの再起動時に実行されます。usrconf.cfg の詳細については、マニュアル「Cosminexus リファレンス 定義編」を参照してください。

core ダンプを取得したあとに、core ダンプからスタックトレース情報だけを取得する場合は、javatrace コマンドを実行してください。スタックトレース情報は、JavaVM の異常終了の原因を究明するために必要な情報です。スタックトレース情報の取得方法は、J2EE サーバモードの場合と同様です。「5.4.16 JavaVM のスタックトレース情報の取得」を参照してください。

core ダンプは次のような場合にも取得できます。それぞれの場合の取得方法について説明します。

cjstartweb プロセスが稼働している状態で core ダンプを取得する場合

実行中の Java プロセスで core ダンプとスレッドダンプを同時に取得する場合

cjstartweb プロセスが稼働している状態で core ダンプを取得する場合

cjstartweb プロセス（Web コンテナサーバ）が稼働している状態で core ダンプを取得する場合、cjstartweb プロセスのプロセス ID を確認して、kill コマンドを実行します。kill コマンドは次の形式で実行してください。なお、kill コマンドを実行するとプロセスが終了するため、再起動する前に kill コマンドを実行することを推奨します。

```
kill -6 <cjstartwebプロセスID>
```

実行中の Java プロセスで core ダンプとスレッドダンプを同時に取得する場合

実行中の Java プロセスで core ダンプとスレッドダンプを同時に取得する方法は、J2EE サーバモードの場合と同様です。「5.4.7(2) J2EE サーバのメモリダンプの取得」を参照してください。

## (6) Cosminexus の定義情報の取得（サブレットエンジンモードの場合）

Cosminexus の定義を取得します。この情報を使用して、発生した障害によって定義の設定を確認します。

Web コンテナサーバに関する定義情報

次のディレクトリ下に格納されているファイル一式を取得します。

Windows の場合

- <Cosminexus のインストールディレクトリ >%CC%\web\containers\<サーバ名称 >%usrconf 下すべて

UNIX の場合

- /opt/Cosminexus/CC/web/containers/<サーバ名称 >/usrconf 下すべて

Cosminexus DABroker Library に関する定義情報

次のファイルを取得します。

Windows の場合

- <Cosminexus DABroker Library 運用ディレクトリ >%conf%\dasys.conf  
(Cosminexus DABroker Library 動作環境定義ファイル)
- <Cosminexus DABroker Library 運用ディレクトリ >%conf%\dabenv (接続先データベース定義ファイル)

UNIX の場合

- <Cosminexus DABroker Library 運用ディレクトリ >/conf/dasys.conf  
(Cosminexus DABroker Library 動作環境定義ファイル)
- <Cosminexus DABroker Library 運用ディレクトリ >/conf/dabenv (接続先データベース定義ファイル)

なお、トラブルシューティング用に保存するファイルからは、パスワードなどの公開できない情報を削除することをお勧めします。

## (7) Cosminexus の作業ディレクトリの内容の取得 (サブレットエンジンモードの場合)

システムでトラブルが発生した場合に、保守員による原因究明のため作業ディレクトリを調査に使用する場合があります。

環境変数「PRFSPOOL」の設定ディレクトリ下すべて

<Cosminexus のインストールディレクトリ >%CC%\web\containers\<サーバ名称 >%webapps 下すべて (Windows の場合)

/opt/Cosminexus/CC/web/containers/<サーバ名称 >/webapps 下すべて (UNIX の場合)

注

環境変数「PRFSPOOL」未設定の場合は、「<Cosminexus のインストールディレクトリ >%PRF%\spool 下すべて」(Windows の場合)、または「/opt/Cosminexus/PRF/spool 下すべて」(UNIX の場合)になります。

設定を確認の上、トラブルが発生した場合に保存しておいてください。

## (8) Web サーバログの取得

システムで使用している Web サーバ (Hitachi Web Server または Microsoft IIS) のログを取得します。

Hitachi Web Server を使用している場合、ログの格納場所は「<Cosminexus のインストールディレクトリ>¥httpsd¥logs」(Windows の場合)、「/opt/hitachi/httpsd/logs」(UNIX の場合)です。

Microsoft IIS を使用している場合、ログの格納場所は「C:¥WINNT¥system32¥LogFiles¥W3SVC1」です (C: の部分にはシステムドライブを指定します)。

## (9) そのほかの情報の取得 (サブレットエンジンモードの場合)

そのほかの情報を取得するための方法は、すべて J2EE サーバモードの場合と同様です。参照先を次に示します。

表 G-8 取得する情報の種類と参照先

取得する情報	参照先
日立固有の JavaVM ログ	5.4.8
JavaVM 出力メッセージログ (標準出力またはエラーレポートファイル)	5.4.9
OS の状態情報と OS のログ	5.4.10
OS の統計情報	5.4.11

## 付録 H 代替手段で実行できる運用作業

ここでは、Cosminexus のシステムで実行できる運用作業を一覧で示します。実行できる運用作業には、システムで推奨する手段での作業、推奨外の作業および代替手段での作業を含みます。各運用作業の詳細やコマンドの使用方法については、参照先に示す箇所、マニュアルを確認してください。

### 付録 H.1 システムの起動と停止

システムの起動と停止で実行できる運用作業とその手段について、次の表に示します。

表 H-1 システムの起動と停止で実行できる運用作業とその手段

運用作業	手段	推奨	参照先
サービスユニットを一括起動する	cmx_start_target コマンド		簡易構築
サービスユニットを一括停止する	cmx_stop_target コマンド		簡易構築
論理サーバを個別に、または一括で起動する	運用管理ポータル、運用管理コマンド (mngsvrutil)		運用管理操作
論理サーバを個別に、または一括で停止する	運用管理ポータル、運用管理コマンド (mngsvrutil)		運用管理操作
コマンドでサーバを起動する	サーバ管理コマンド (ejstartsv)		リファレンス
コマンドでサーバを停止する	サーバ管理コマンド (ejstopsv)		リファレンス
クラスタソフトウェアを使用したシステムを起動する	Cosminexus のコマンド、クラスタソフトウェアの機能		8 章, 9 章
クラスタソフトウェアを使用したシステムを停止する	Cosminexus のコマンド、クラスタソフトウェアの機能		8 章, 9 章

(凡例)

: 推奨する作業

: 推奨外の作業、または代替手段での作業

運用管理操作: マニュアル「Cosminexus 運用管理操作ガイド」

簡易構築: マニュアル「Cosminexus 簡易構築・運用ガイド」

リファレンス: マニュアル「Cosminexus リファレンス コマンド編」

### 付録 H.2 システムの運用監視

システムの運用監視で実行できる運用作業とその手段について、次の表に示します。

表 H-2 システムの運用監視で実行できる運用作業とその手段

運用作業	手段	推奨	参照先
サービスユニットのステータスを確認する	cmx_list_status コマンド		簡易構築
論理サーバのステータスを確認する	運用管理コマンド (mngsvrutil)		2.2
	運用管理ポータル		運用管理操作
Web システム内のサービスユニットの構成を確認する	cmx_list_model コマンド		簡易構築
稼働情報ファイルを参照して、J2EE サーバおよび SFO サーバの稼働情報を監視する	-		2.3
J2EE サーバおよび SFO サーバの稼働情報を監視する <sup>1</sup>	運用管理コマンド (mngsvrutil)		2.4
	運用管理ポータル		運用管理操作
リソースの使用率や使用数を監視して、しきい値を超えた場合に原因を調査する	-		2.5
稼働情報ファイルに出力される稼働情報、またはリソースの枯渇状況に応じてメッセージが出力された場合の処理を自動化する (Management アクション)	-		-
運用管理ドメイン内の CTM の稼働統計情報を確認する <sup>2</sup>	運用管理コマンド (mngsvrutil)		2.6
CTM デーモン単位の稼働統計情報を確認する <sup>2</sup>	CTM のコマンド (ctmstsed)		リファレンス
CTM のスケジュールキューの滞留状況を監視する <sup>2 3</sup>	CTM のコマンド (ctmstart)		リファレンス
トランザクションの状態を確認する	サーバ管理コマンド (cjlisttrn)		2.7

## (凡例)

- : 推奨する作業
- : 推奨外の作業、または代替手段での作業
- : 該当しない

運用管理操作: マニュアル「Cosminexus 運用管理操作ガイド」

簡易構築: マニュアル「Cosminexus 簡易構築・運用ガイド」

リファレンス: マニュアル「Cosminexus リファレンス コマンド編」

注 1 稼働情報ファイルを参照した稼働情報の監視で取得できる情報よりも詳細な情報を参照したい場合に、運用管理コマンドを使用した稼働情報の監視を行ってください。

注 2 CTM は、構成ソフトウェアに Cosminexus Component Transaction Monitor を含む製品だけで利用できます。利用できる製品については、マニュアル「Cosminexus 概説」を参照してください。

注 3 Smart Composer 機能を使用して CTM のスケジュールキューの滞留監視をする場合は、簡易構築定義ファイルで定義します。そのため、運用時の作業は必要ありません。

### 付録 H.3 システムの処理性能解析

システムの処理性能解析で実行できる運用作業とその手段について、次の表に示します。

表 H-3 システムの処理性能解析で実行できる運用作業とその手段

運用作業	手段	推奨	参照先
性能解析トレースファイルを収集する	運用管理コマンド (mngsvrutil)		3.2
	PRF のコマンド (cprfed)		リファレンス
収集した性能解析トレースファイルを利用してシステムの処理性能を解析する	-		3.3
PRF トレース取得レベルを確認する	運用管理ポータル		運用管理操作
	PRF のコマンド (cprflevel)		リファレンス
PRF トレース取得レベルを変更する	運用管理コマンド (mngsvrutil)		リファレンス
	運用管理ポータル		運用管理操作
	PRF のコマンド (cprflevel)		リファレンス

(凡例)

- : 推奨する作業
- : 推奨外の作業, または代替手段での作業
- : 該当しない
- 運用管理操作: マニュアル「Cosminexus 運用管理操作ガイド」
- リファレンス: マニュアル「Cosminexus リファレンス コマンド編」

### 付録 H.4 統合ユーザ管理のリソースの監視

統合ユーザ管理のリソースの監視で実行できる運用作業とその手段について、次の表に示します。

表 H-4 統合ユーザ管理のリソースの監視で実行できる運用作業とその手段

運用作業	手段	推奨	参照先
統合ユーザ管理のリソースの使用状況を監視する	運用管理ポータル		運用管理操作

(凡例)

- : 推奨外の作業, または代替手段での作業

運用管理操作：マニュアル「Cosminexus 運用管理操作ガイド」

## 付録 H.5 J2EE アプリケーションとリソースの管理

J2EE アプリケーションとリソースの管理で実行できる運用作業とその手段について、次の表に示します。

表 H-5 J2EE アプリケーションとリソースの管理で実行できる運用作業とその手段

運用作業	手段	推奨	参照先
Web アプリケーションの稼働状況を確認する	運用管理コマンド (mngsvrutil)		4.2.1
	運用管理ポータル		運用管理操作
アクセス状況に応じて、Web アプリケーションの同時実行スレッド数を動的に変更する	運用管理コマンド (mngsvrutil)		4.2.2
CTM のスケジュールキューの稼働状況を確認する <sup>1</sup>	運用管理コマンド (mngsvrutil)		4.3.1
	CTM のコマンド (ctmlsque)		4.3.2
CTM のスケジュールキューの同時実行数を動的に変更する <sup>1</sup>	CTM のコマンド (ctmchpara)		4.3.2
J2EE アプリケーションの実行状態を確認する	サーバ管理コマンド (cjisthread)		4.4.1
J2EE アプリケーションでタイムアウトが発生したリクエストをキャンセルする	サーバ管理コマンド (cjstopthread)		4.4.2
J2EE アプリケーションを通常開始する	サーバ管理コマンド (cjstartapp)		リファレンス
	運用管理コマンド (mngsvrutil)		リファレンス
	Smart Composer 機能のコマンド (cmx_stop_application コマンド)		簡易構築
	運用管理ポータル		運用管理操作

運用作業	手段	推奨	参照先
J2EE アプリケーションを通常停止する	サーバ管理コマンド (cjstopapp)		4.5
	運用管理コマンド (mngsvrutil)		リファレンス
	Smart Composer 機能のコマンド (cmx_stop_application コマンド)		簡易構築
	運用管理ポータル		運用管理操作
J2EE アプリケーションを強制停止する	サーバ管理コマンド (cjstopapp)		4.5.5
	運用管理コマンド (mngsvrutil)		リファレンス
負荷分散機を使用してサービスを閉塞する <sup>2</sup>	負荷分散機の機能		4.5.3
Web サーバのサービスを閉塞する	運用管理コマンド (mngsvrutil)		リファレンス
	運用管理ポータル		運用管理操作
	Hitachi Web Server の機能		Hitachi Web Server
CTM を利用してサービスを閉塞する <sup>1</sup>	運用管理コマンド (mngsvrutil)		4.5.4
	CTM のコマンド (ctmholdque)		リファレンス
J2EE サーバを停止してサービスを閉塞する	運用管理コマンド (mngsvrutil)		リファレンス
	運用管理ポータル		運用管理操作
	cjstopsv コマンド		リファレンス
J2EE アプリケーションの閉塞を解除する	サーバ管理コマンド (cjstartapp)		リファレンス
	運用管理コマンド (mngsvrutil)		リファレンス
	運用管理ポータル		運用管理操作



運用作業		手段	推奨	参照先
J2EE アプリケーションを入れ替える	通常の入替	サーバ管理コマンド (cjstopapp, cjimportapp など)		4.6.1
		運用管理ポータル		運用管理操作
	リデプロイによる入替	サーバ管理コマンド (cjreplaceapp)		4.6.2
	リロードによる入替	サーバ管理コマンド (cjreloadapp)		4.6.3
JSP を事前にコンパイルしてから、J2EE アプリケーションを入れ替える		cjjspc コマンド		4.6.4
J2EE アプリケーションの名称を変更する		サーバ管理コマンド (cjrenameapp)		4.6.5
		Server Plug-in		運用管理操作
Web アプリケーションのサービス部分閉塞を利用して J2EE アプリケーションを入れ替える		運用管理ポータル		運用管理操作
CTM を利用して、オンライン状態で J2EE アプリケーションを入れ替える <sup>1</sup>		運用管理コマンド (mngsvrutil)		4.6.6
J2EE アプリケーションを本番環境でテストしてから本番運用を開始する		サーバ管理コマンド (cjimportapp, cjstartapp)		4.6.7
コネクションプールの情報を表示する		サーバ管理コマンド (cjlistpool)		解説, リファレンス
コネクションプールをクリアする		サーバ管理コマンド (cjclearpool)		解説, リファレンス
JavaBeans リソースを入れ替える		サーバ管理コマンド (cjdeletejb, cjimportjb)		4.7
クラスタ構成のデータベースの場合に、コネクションプールの状態を表示する		サーバ管理コマンド (cjlistrar)		4.8.1
クラスタ構成のデータベースの場合に、コネクションプールを一時停止する		サーバ管理コマンド (cjsuspendpool)		4.8.2
クラスタ構成のデータベースの場合に、コネクションプールを再開する		サーバ管理コマンド (cjresumepool)		4.8.3
リソースの接続テストをする		サーバ管理コマンド (cjtestres)		解説, リファレンス

( 凡例 )

- : 推奨する作業
- : 推奨外の作業，または代替手段での作業

Hitachi Web Server：マニュアル「Hitachi Web Server」

運用管理操作：マニュアル「Cosminexus 運用管理操作ガイド」

解説：マニュアル「Cosminexus 機能解説」

簡易構築：マニュアル「Cosminexus 簡易構築・運用ガイド」

リファレンス：マニュアル「Cosminexus リファレンス コマンド編」

注 1 CTM は，構成ソフトウェアに Cosminexus Component Transaction Monitor を含む製品だけで利用できます。利用できる製品については，マニュアル「Cosminexus 概説」を参照してください。

注 2 負荷分散機を Smart Composer 機能で管理していない場合に実行します。負荷分散機を Smart Composer 機能で管理している場合は，cmx\_stop\_target コマンドを使用してサービスを閉塞してください。cmx\_stop\_target コマンドの詳細については，マニュアル「Cosminexus 簡易構築・運用ガイド」を参照してください。

## 付録 H.6 システムの監査

システムの監査で実行できる運用作業とその手段について，次の表に示します。

表 H-6 システムの監査で実行できる運用作業とその手段

運用作業	手段	推奨	参照先
監査ログを出力して，管理する	-		6.2
監査ログを収集して調査する	-		6.2
データベース監査証跡に出力された情報を基に，データベースにアクセスしたリクエストの情報を特定する	-		7.2

(凡例)

- : 推奨する作業
- : 該当しない

## 付録 H.7 トラブルシューティング

トラブルシューティングで実行できる運用作業とその手段について，次の表に示します。

表 H-7 トラブルシューティングで実行できる運用作業とその手段

運用作業	手段	推奨	参照先
トラブルシューティングに必要な資料を収集する	運用管理コマンド (mngsvrutil)		5.3.1
	運用管理ポータル		運用管理操作
	snapshotlog コマンド		リファレンス
論理サーバに異常が発生した場合の処理を自動化する	-		5.3.2

運用作業	手段	推奨	参照先
トラブルシューティングに必要な資料を調査する	-		5.5

(凡例)

- : 推奨する作業
  - : 推奨外の作業, または代替手段での作業
  - : 該当しない
- 運用管理操作: マニュアル「Cosminexus 運用管理操作ガイド」  
リファレンス: マニュアル「Cosminexus リファレンス コマンド編」

---

## 付録 I 用語解説

Application Server , uCosminexus Client , Web Redirector , および uCosminexus Operator で使用する用語について説明します。

### ( 数字 )

---

#### 1.4 モード

サーバの動作モードです。J2EE 1.4 の機能を使用できます。データベースを含む複数のリソースのトランザクション管理ができます。

#### 1:1 系切り替えシステム

実行系のアプリケーションサーバと待機系のアプリケーションサーバが 1:1 で対応しているシステムです。実行系のアプリケーションサーバに障害が発生すると、待機系のアプリケーションサーバで処理を引き継ぎます。

### ( 英字 )

---

#### Application Server

Cosminexus の実行環境を構築する基盤製品です。Application Server Standard と、Application Server Enterprise の総称です。

#### BJEX

uCosminexus Batch Job Execution Server のことです。

#### BMP ( Bean-Managed Persistence )

Enterprise Bean のビジネスメソッド内でデータの永続化を管理するモデルです。

#### BMT ( Bean-Managed Transaction )

Enterprise Bean でトランザクションを管理するモデルです。Session Bean および Message-driven Bean が対象になります。Entity Bean は対象外です。

#### CMP ( Container-Managed Persistence )

EJB コンテナがデータの永続化を管理するモデルです。

#### CMT ( Container-Managed Transaction )

EJB コンテナがトランザクションを管理するモデルです。Session Bean , Entity Bean および Message-driven Bean が対象になります。

#### Component Container 管理者

スーパーユーザ以外の、J2EE サーバやバッチサーバのセットアップ、J2EE サーバやバッチサーバの起動・停止などを実施できるユーザを指します。Component Container 管理者は UNIX の場合に設定できます。

## Cookie スイッチング機能

負荷分散機で提供しているサーバスイッチング機能の一つです。Cookie スイッチング機能を使用すると、クライアントからのリクエストバケット内の Cookie 情報に基づいて、同じサーバにセッションをリダイレクトします。

## CORBA ネーミングサービス

CORBA の仕様に準拠した、リモートオブジェクトの格納場所を管理するためのネーミングサービスです。Application Server の構成ソフトウェアである Cosminexus TPBroker によって提供される機能です。

## Cosminexus

アプリケーションサーバを中核とした、性能および信頼性の高いアプリケーションを実行および開発するためのシステム構築基盤製品です。

## Cosminexus Component Container

サーバ・サイドの業務処理プログラム（ビジネスロジック）をコンポーネントとして実行するための構成ソフトウェアです。また、Cosminexus システムの運用管理をするための Management Server、統合ユーザ管理、snapshot ログ収集などの機能も提供しています。

## Cosminexus Component Transaction Monitor

クライアントからのリクエストのスケジューリングを実現するための構成ソフトウェアです。

## Cosminexus Operator Plug-in

実行環境を運用管理操作するための構成ソフトウェアです。Server Plug-in と HCSC-Manager を提供しています。HCSC-Manager とは、Cosminexus サービスプラットフォームの運用環境で利用する運用管理機能です。なお、HCSC-Manager については、マニュアル「Cosminexus サービスプラットフォーム システム構築・運用ガイド」を参照してください。

## Cosminexus Performance Tracer

リクエストが処理されるときに、決められたポイントごとに各機能が出力する性能解析情報をファイルに出力するための構成ソフトウェアです。

## Cosminexus Reliable Messaging

Cosminexus システム上の J2EE アプリケーションがメッセージを使用して非同期に通信するためのミドルウェアです。JMS インタフェースでのメッセージ通信機能を J2EE アプリケーションに提供します。

## Cosminexus システム

Application Server を実行基盤として構築されているシステムのことで。

## Cosminexus 標準ログインモジュール

統合ユーザ管理のユーザ認証処理で使用するログインモジュールです。javax.security.auth.spi に含まれる LoginModule インタフェースの実装クラスです。

次のログインモジュールがあります。

- WebPasswordLoginModule  
パスワードを使用してユーザを認証します。
- WebCertificateLoginModule

クライアント証明書を使用してユーザを認証します。

- WebPasswordLDAPLoginModule  
LDAP ディレクトリサーバの認証機能を使用してユーザを認証します。
- WebPasswordJDBCLoginModule  
RDB に格納された情報を使用してユーザを認証します。
- DelegationLoginModule  
カスタムログインモジュールを呼び出します。
- WebSSOLoginModule  
シングルサインオンをするときに、ほかのログインモジュールを呼び出します。

## CTM

Cosminexus Component Transaction Monitor のことです。リクエストのスケジューリングをするための構成ソフトウェアです。

## CTM デーモン

クライアントからのリクエストを制御するスケジュールキューを管理するプロセスです。リクエストのスケジューリングを実現する、スケジューラとして機能します。

Management Server を利用している場合、CTM デーモンは論理サーバとして扱えます。

## CTM ドメイン

複数の CTM デーモンで構成される、情報共有と負荷分散の対象になる範囲です。CTM デーモンは CTM ドメイン内で、それぞれに登録された業務処理プログラムの情報やスケジュールキューの負荷情報を交換してリクエストを振り分けます。

## CTM ドメインマネージャ

CTM ドメインを管理するプロセスです。ホスト内の CTM デーモンの情報を、CTM ドメイン内のほかの CTM ドメインマネージャに配布します。

Management Server を利用している場合、CTM ドメインマネージャは論理サーバとして扱えます。

## CTM レギュレータ

EJB クライアントから CTM デーモンに集中するリクエストを、分散集約するためのプロセスです。TPBroker クライアント (CORBA クライアント) から J2EE サーバ内で動作している EJB アプリケーションを直接呼び出できるゲートウェイ機能も提供します。

## DB Connector

データベースに接続するためのリソースアダプタです。

## DD

アプリケーションを運用環境に配置するときの定義情報を記述したものです。EJB 用、Web アプリケーション用、Enterprise アプリケーション用などが Sun Microsystems, Inc. からの仕様で規定されています。

## Dependency Injection

EJB やインターセプタクラスのフィールドや setter メソッドに、@EJB や @Resource アノテーションを指定することで、EJB やリソースへの参照を EJB コンテナが自動的にセットする機能です。Dependency Injection を使用すると、EJB やリソースへの参照を JNDI を使用してルックアップする必要がなくなります。

## DI ターゲット

Dependency Injection 機能を使用して参照を注入する対象のことです。DI ターゲット名と名前が一致するフィールド、または DI ターゲット名をプロパティとして持つ setter メソッドに対して、Dependency Injection が実行されます。

## DTD

XML で使用するタグの属性や階層構造や出現順序などを定義したものです。

## EAR ファイル

J2EE アプリケーションを構成する複数の EJB-JAR ファイル、WAR ファイル、および DD を EAR ファイル形式でパッケージ化したものです。

## EIS

データベースやトランザクションサーバなど、企業内に構築されているバックエンドシステムです。

## EJB ( Enterprise JavaBeans )

業務ロジックをプログラムとして記述したビジネスロジック、および永続的データを格納するオブジェクトであるエンティティを Java コンポーネント化したものです。Sun Microsystems, Inc. から EJB 仕様が公開されています。

## EJB-JAR ファイル

Enterprise Bean、DDなどをJARファイル形式に圧縮したものです。

## EJB クライアント

J2EE サーバ上で開始されている Enterprise Bean を呼び出すクライアントプログラムです。次の3種類があります。

- EJB クライアントアプリケーション
- サブレットまたは JSP などの Web アプリケーション
- ほかの Enterprise Bean

## EJB クライアントアプリケーション

Enterprise Bean を呼び出す Java アプリケーションです。

## EJB コンテナ

Enterprise Bean を制御すると同時に、通信、トランザクション管理などのシステムレベルのサービスを提供する実行環境です。Enterprise Bean の実体は、EJB コンテナの中で実行されます。

## EJB タイマ

EJB コンテナの Timer Service の機能を利用して、タイムアウト時間を管理するタイマです。指定された時刻または間隔で、タイムアウトメソッドをコールバックし、イベントを実行します。

## Enterprise Bean

ビジネスロジックを EJB アーキテクチャに従って作成したものです。業務処理プログラムに該当します。

## Entity Bean

永続化されたデータそのものを表す Bean です。Entity Bean は、さらに CMP Entity Bean と

BMP Entity Bean に分類されます。

## HA モニタ

クラスタソフトウェアの一つです。

システムを監視し、障害発生時などにシステムの切り替えを実行するプログラムです。HA モニタと連携することで、Cosminexus システムの信頼性や稼働率を向上できます。AIX、HP-UX または Linux を使用しているときに連携できます。

HA モニタと連携することで、1:1 系切り替えシステム、相互系切り替えシステム、N:1 リカバリシステムが実現できます。

## Inbound

Connector 1.5 仕様で定められた J2EE サーバとリソースアダプタ間の通信モデルの一つです。リソースアダプタから J2EE サーバにアクセスする通信モデルです。

## J2EE

Java™ 2 Platform, Enterprise Edition の略です。Sun Microsystems, Inc. から J2EE 仕様が公開されています。

## J2EE アプリケーション

JSP、サーブレット、EnterpriseBean などで構成されるアプリケーションです。Cosminexus で扱う J2EE アプリケーションの形式には、EAR ファイル形式でパッケージ化されたアーカイブ形式のアプリケーションと、アーカイブ化しない展開ディレクトリ形式のアプリケーションがあります。EAR ファイル形式でパッケージ化されたアプリケーションの場合、複数の EJB-JAR ファイル、複数の WAR ファイル、および一つの DD から構成されます。

## J2EE アプリケーション実行時間の監視機能

EJB および Web アプリケーションのリクエスト実行時間を監視する機能です。実行時間監視中に障害を検知すると、メソッドタイムアウト機能およびメソッドキャンセル機能が動作します。

## J2EE アプリケーションの強制停止

J2EE アプリケーションの停止での強制停止とは、J2EE アプリケーションで処理中のリクエストがある場合でも、強制的に処理を中止する方法です。

## J2EE アプリケーションの通常停止

J2EE アプリケーションの停止での通常停止とは、Web アプリケーションや EJB の停止処理を順番に実施して、J2EE アプリケーションを安全に停止させる方法です。

## J2EE コンテナ

J2EE アプリケーションを実行するためのサーバ基盤です。

J2EE コンポーネントへ各種 API を提供する、Web コンテナ、EJB コンテナから構成されます。

## J2EE コンポーネント

サーブレット、JSP、Enterprise Bean などのユーザアプリケーションプログラムのことです。

## J2EE サーバ

J2EE コンテナを生成、実行する環境です。



## J2EE サーバモード

J2EE サーバの動作モードの一つです。このモードの場合、J2EE コンテナ上で動作するアプリケーションから、J2EE 関連の API を利用できます。なお、J2EE サーバモードには、1.4 モードと、互換の動作モードであるベーシックモードがあります。

## J2EE サービス

J2EE コンテナの部品機能として利用されます。J2EE コンポーネントに JNDI、JDBC、JTA、RMI-IIOP および JavaMail の API を提供します。

## J2EE リソース

サーバから利用できるリソースです。J2EE サーバの場合、データベース、OpenTP1、SMTP サーバなどを利用できます。また、バッチサーバの場合、データベースを利用できます。

## J2EE リソースアダプタ

J2EE サーバに共有スタンドアロンモジュールとして配備したリソースアダプタのことです。

## jaas.conf ( JAAS のコンフィグレーションファイル )

ユーザ認証ライブラリおよびシングルサインオンライブラリの機能を使用するために必要な JAAS 認証用の設定ファイルです。

## Java

1991 年に Sun Microsystems, Inc. 内の研究グループが開発したインタプリタ言語を基に、C++ をより移植性と信頼性が高くなるように改良した言語です。Web の普及によってブラウザに 응용されています。

## Java ロギング API

J2SE で提供されているコアロギング機能のクラスおよびインタフェースです。Application Server では、Java ロギング API を使用してユーザログを日立トレース共通ライブラリ形式で出力する機能を提供しています。

## JCA コンテナ

J2EE サーバやバッチサーバと、EIS を接続するためのシステムレベルのプラグイン機能を提供するコンテナです。

## JDBC ( Java Database Connectivity )

Java のプログラムでデータベースを利用するための API です。

JavaSoft が 1996 年前半に仕様を公開しました。Java プログラムから、JDBC API を経由して「JDBC ドライバ」を呼び出し、ドライバがデータベース・サーバへの接続機能を提供します。

## JNDI ( Java Naming and Directory Interface )

Java プラットフォーム用の標準拡張機能で、Java テクノロジに対応したアプリケーションに、企業内の複数のネーミングおよびディレクトリサービスへの統一したインタフェースを提供します。Application Server では、Cosminexus TPBroker の CORBA ネーミングサービスを利用して、JNDI の機能を実現しています。

## JNDI 名前空間

JNDI によってアクセスする名前情報が管理されている名前空間です。Application Server では、J2EE アプリケーションをデプロイした時に、EJB オブジェクトリファレンスが JNDI 名前空間の名前にバインドされます。

Application Server で使用する JNDI 名前空間の名前には、別名を付けることができます。これによって、Enterprise Bean や J2EE リソースを別名でルックアップできるようになります。

## JP1

日立の統合システム運用管理ソフトウェアです。

## JP1/AJS2

日々の業務の中から、定型的・定期的なものを自動化して、システム運用に掛かるコストを削減し、少ない人員で確実な運用を実現するためのプログラム群です。JP1/AJS2 - Manager, JP1/AJS2 - Agent, JP1/AJS2 - View の製品で構成されます。

## JP1/AJS2 - SO

一つ一つの運用手順を蓄積し、ライブラリとして統合する運用管理の基盤機能を提供するプログラムです。立案したシナリオに合う運用手順をライブラリから選択し、運用環境に合わせてカスタマイズすることで、業務運用の設計を容易に、かつ高品質で実現することができます。

## JP1/Cm2/NC - Manager

負荷分散機の設定をしたり、負荷分散機の制御をしたりするためのプログラムプロダクトです。

## JP1/Cm2/NC - Manager for Cosminexus

Cosminexus と連携して、負荷分散機を制御するためのプログラムプロダクトです。Management Server から負荷分散機を動的に制御できます。

## JP1/IM

企業情報システム全体を統合管理する基盤になるプログラム群です。JP1 Version 8 の場合は、JP1/IM - Manager, JP1/IM - View, および JP1/IM - CM の三つの製品で構成されます。JP1 Version 7i の場合は、JP1/IM - Central Console および JP1/IM - Central Scope, JP1/IM - View, および JP1/IM - CM の四つの製品で構成されます。

## JP1/IM - CM

業務システムを構成する各種のリソース（サーバ、ストレージなど）やシステム構成に関する情報を集約して管理するためのプログラムです。管理対象のリソースを任意にグルーピングできるため、システムの監視（性能、障害）や運用（設定配布、構成変更）の際に、グループ単位で管理したり操作したりできます。

## JP1/NETM/Audit

システム内の製品が出力した監査ログを自動収集して、一括管理するためのプログラムプロダクトです。JP1/Base と連携して、Cosminexus のアプリケーションサーバ、運用管理サーバ、連携する日立オープンモドルウェア製品などが出力した監査ログを監査ログ管理サーバで自動収集できます。また、収集した監査ログを、監査ログ管理サーバのデータベースで一元管理できます。

## JP1/SC/DPM

デプロイメント操作を実行するためのソフトウェアです。ディスクイメージをバックアップしたり、OS やアプリケーションのインストールを一括して実施したりできます。

## JP1 イベント

システム内で何らかの事象が発生した際に、その事象に関して JP1/Base に通知される情報です。

## JP1 ジョブ運用管理サーバ

JP1/AJS2 と連携してシステムの自動運転をする場合に、JP1/AJS2 - Manager を配置するホストのことです。

## JP1 統合運用管理サーバ

JP1/IM と連携してシステムの集中監視をする場合に、JP1/IM - Manager ( JP1 Version 8 の場合 ) を配置するホストのことです。JP1 Version 7i の場合は、JP1/IM - Central Console および JP1/IM - Central Scope を配置するホストのことです。

## JSP ( JavaServer Pages )

Web ページを作成する HTML の中に JSP 固有のタグを埋め込んで Enterprise Bean を呼び出すことや、サーブレットから値を受け取ることができます。また、簡単な Java 言語を直接記述することもできます。

## JSP EL

JSP 2.0 で定義された式言語です。JSP ファイルやタグファイル内に JavaBeans の属性へのアクセスなどを記述できます。また、API も提供されています。

## JSP コンパイル結果

JSP ファイルおよびタグファイルのコンパイルで生成される Java ソースファイルやクラスファイルです。

## JSP 事前コンパイル

Web アプリケーションに含まれる JSP ファイルおよびタグファイルをデプロイ前にコンパイルし、Java ソースファイルやクラスファイルを生成する機能です。JSP 事前コンパイル機能を使用してあらかじめ Java ソースファイルやクラスファイルを生成しておくと、Web アプリケーション実行時の初回リクエストでのレスポンスタイムを短縮できます。

## JSP タグライブラリ

JSP タグを作成するためのもので、JSP 仕様で規定されています。Cosminexus では、統合ユーザ管理をする場合に、Cosminexus 標準ログインモジュールを使用してユーザ認証をするための JSP タグライブラリを提供しています。

## Management Server

運用管理ドメインを構成するサーバプログラムです。運用管理ドメイン単位に一つ配置します。Management Server は運用管理ドメイン内の各ホストに配置した運用管理エージェントに指示を出して、運用管理ドメイン全体の運用管理を実行します。

## Management アクション

Management イベント発生時に自動的に実行される処理です。Management アクションとしてどのような処理をするかは、Management Server 側であらかじめ定義しておきます。

## Management イベント

Management Server を利用して運用している場合に、運用管理ドメイン内の論理サーバで発生した事象に応じて Management Server で発行するイベントです。論理サーバでメッセージが出力されたタイミングで発行できます。例えば、リソース監視中に設定したしきい値を超えた場合などに発行できます。

Management イベントに対して Management アクションを定義しておくことで、発生した事象に応じたアクションを自動的に実行できます。

## Message-driven Bean

JMS と連携するメッセージ駆動タイプの Bean です。EJB コンテナは JMS の Destination からの JMS メッセージ受信を契機に Bean を起動します。

## MIB オブジェクト

SNMP プロトコルで使用するオブジェクトです。

## MIB ファイル

SNMP プロトコルで使うオブジェクトの構造体を記述したテキストファイルです。すべてのオブジェクト（情報）は、オブジェクト ID で一意に決まります。

## Microsoft Cluster Service

クラスタソフトウェアの一つです。

システムを監視し、障害発生時などにシステムの切り替えを実行するプログラムです。Microsoft Cluster Service と連携することで、Cosminexus システムの信頼性や稼働率を向上できます。

Windows Server 2003 で使用できます。

Microsoft Cluster Service と連携したシステムでは、1 : 1 系切り替えシステム、相互系切り替えシステム、N : 1 リカバリシステムが実現できます。

## N:1 リカバリシステム

クラスタ構成の実行系のアプリケーションサーバに対して、1 台のリカバリ専用サーバを待機系として配置したシステムです。リカバリ専用サーバでは、障害が発生した実行系のアプリケーションサーバのトランザクションの決着をします。

## Oracle RAC

Oracle データベースのクラスタ機能です。Cosminexus では、Oracle RAC 機能を使用してクラスタ化された Oracle データベースに接続できます。

## ORB ゲートウェイ

CORBA クライアントから J2EE サーバ内で動作している EJB アプリケーションを直接呼び出しできるゲートウェイ機能です。CTM レギュレータプロセスによって提供されます。

## OTM ゲートウェイ

TPBroker OTM のクライアントから J2EE サーバ内で動作している EJB アプリケーションを直接呼び出しできるゲートウェイ機能です。OTM ゲートウェイプロセスによって提供されます。

## OTS ( Object Transaction Service )

1.4 モードで、TPBroker OTS による分散トランザクションを使用するために必要なサービスです。

## Outbound

Connector 1.5 仕様で定められた J2EE サーバとリソースアダプタ間の通信モデルの一つです。J2EE サーバからリソースアダプタにアクセスする通信モデルです。

## PRF デーモン

バッファに出力された PRF トレースを PRF トレースファイルに出力する IO プロセスです。Management Server では、論理サーバとして扱えます。Management Server で扱う場合は、パフォーマンストレーサといいます。

## PRF トレース

リクエストを処理する時に Cosminexus システムの各機能で出力されたトレース情報です。バッファに出力されて、一定量たまったら PRF トレースファイルに出力されます。

## PRF トレース出力ライブラリ

Cosminexus システムの各機能に組み込まれているライブラリです。機能レイヤごとの PRF トレースをバッファに出力する機能を提供しています。

## PRF トレース取得レベル

PRF トレースを取得するレベルです。「標準レベル」、「詳細レベル」、「保守レベル」があります。「詳細レベル」または「保守レベル」を指定した場合、「標準レベル」に比べてトレース取得ポイントが多くなります。

## PRF トレースファイル

PRF トレースを出力したバイナリ形式のファイルです。性能解析トレースファイル (CSV 形式) に変換する基になるファイルです。

## RMI-IIOP

JavaRMI と JavaIDL を統合した API です。

## SecurityManager

セキュリティポリシーに基づいた Web コンテナの実行時保護をする J2EE サーバの機能です。

## Server Plug-in

Eclipse のプラグインです。Server Plug-in を使用すると次の操作ができます。

- リソースアダプタのプロパティを設定したり、設定したリソースアダプタを J2EE サーバやバッチサーバにデプロイしたりする。
- J2EE アプリケーションのプロパティを設定したり、設定した J2EE アプリケーションを J2EE サーバにインポートしたりする。
- 論理サーバを起動・停止する。

## Session Bean

主に業務処理を実行するための Bean です。Session Bean は、クライアントを一つだけ持つことができます。クライアントが終了すると、対応する Session Bean も終了します。クライアントとのセッションの間だけ動作するため、一時的であり、非永続的な Bean です。Session Bean は、Stateful Session Bean と、Stateless Session Bean に分類されます。

## SFO サーバ

システム内の J2EE サーバの J2EE アプリケーション内で生成されたグローバルセッション情報を管理する J2EE サーバです。セッションフェイルオーバー機能を使用するときにシステム構成に含めます。

## SFO サーバアプリケーション

SFO サーバ上で稼働する J2EE アプリケーションを、SFO サーバアプリケーションといいます。SFO サーバアプリケーションは EJB で実装されています。なお、SFO サーバアプリケーションは Cosminexus で提供しています。

## Smart Composer 機能

Cosminexus を使用してアプリケーションサーバを構築する場合に、一般的な 3 階層モデルのアプリケーションサーバのシステムを、簡単に構築および運用できるよう支援する機能です。システム全体に対して、システムの設定、および J2EE アプリケーションやリソースアダプタのデプロイを一括で実行できるので、簡単、迅速にシステム構築ができます。

構築したシステムを一括で起動したり、一括でシステムの設定を変更したりなど、運用機能についてもサポートしています。

## snapshot ログ

ある時点での J2EE サーバやバッチサーバの状態が出力されたログファイルです。

トラブルシューティングに必要な情報が自動的に出力されます。

Management Server を利用している場合は、次のタイミングで出力されます。

- 論理サーバに異常が発生した時
- 運用管理ポータルまたは運用管理コマンドによって snapshot ログの収集を要求した時

Management Server を利用していない場合は、コマンド ( snapshotlog ) を実行した時に出力されます。このコマンドで出力できるのは、そのコマンドを実行したホストの情報だけです。

## SNMP 連携用形式ファイル

SNMP Manager 製品と連携するためのファイルです。MIB ファイルに変換する基になる形式のファイルです。Management Server の運用管理コマンド ( mngsvrutil ) で出力できます。

## Stateful Session Bean

内部的な状態を保持できる Bean です。Stateless Session Bean とは異なり、ホームインタフェースからの生成によってインスタンス化されたあとは、メソッドの呼び出しごとに生成、消滅することはありません。セッション単位で連続した処理をする場合に適した Bean です。

## Stateless Session Bean

状態を保持しない Bean です。Bean のインスタンスの生成・消滅は、すべて EJB コンテナが独自に管理しています。同じ Bean のメソッドを続けて呼び出した場合にも、それらが同じインスタンスである保証はありません。そのため、Stateless Session Bean では、内部状態を保持して、複数メソッドで連続した処理をすることはできません。1 メソッドで完結するような処理を記述する場合に適しています。

## Timer Service

EJB コンテナで提供される機能です。指定した時刻、経過時間、または間隔で、EJB コンテナがタイムアウトメソッドをコールバックします。

### ua.conf (統合ユーザ管理のコンフィグレーションファイル)

JAAS 対応ユーザ管理、およびシングルサインオンの機能を使用するための設定ファイルです。

### uCosminexus Batch Job Execution Server

オープン環境で基幹系のバッチ業務を実行・運用するためのバッチジョブ実行基盤を提供する製品です。JP1/AJS2 とあわせて使用します。

### uCosminexus Client

EJB クライアント環境を構築するための製品です。

### uCosminexus Operator

Application Server をインストールしたマシンとは別のマシンから、GUI (Eclipse プラグイン) 画面を使用して運用操作を実行するための製品です。例えば、実行環境が UNIX の場合に、別の Windows のマシンで Eclipse プラグインの GUI による運用管理操作を実行したいときに、この製品を使用します。

### usrconf.properties (ユーザプロパティファイル)

J2EE サーバ、バッチサーバ、Web コンテナサーバ、またはサーバ管理コマンドで使用する JavaVM のシステムプロパティを定義するファイルです。

### WAR ファイル

Web アプリケーションの構成要素を JAR ファイル形式に圧縮したファイルです。

Web アプリケーション実行に必要な HTML ファイル、JSP ページ、Java クラスファイル、JAR ファイル、および Web アプリケーション配置記述子が含まれます。

### Web アプリケーション

Web ブラウザを備えたクライアントを対象に作成されたアプリケーションです。具体的には、サーブレットプログラム、JSP ページ、HTML/XML ドキュメントなどの集合体です。

### Web コンテナ

J2EE アーキテクチャの Web コンポーネント規約を実装するコンテナです。セキュリティ、トランザクションなどの各種サービスを提供する実行環境です。Web アプリケーションは、Web コンテナ上で動作します。

Java Servlet2.4 仕様、および JavaServer Pages Specification v2.0 仕様に準拠した Web アプリケーションを実行できます。

### Web コンテナサーバ

サーブレットエンジンモードで動作するサーバを指します。

### Web サーバ

Web ブラウザからのリクエスト受信および Web ブラウザへのデータ送信に関連する処理を実行するプログラムです。Cosminexus では、Hitachi Web Server、Microsoft IIS、またはインプロセス HTTP サーバを使用できます。インプロセス HTTP サーバは、J2EE サーバプロセス内で動作する Web サーバです。

なお、Management Server を利用する場合、Hitachi Web Server は論理サーバとして扱えます。

## Web サーバ連携

アプリケーションサーバで使用する Web サーバとして、Hitachi Web Server または Microsoft IIS を使用方法です。

Hitachi Web Server または Microsoft IIS に Cosminexus Component Container が提供するリダイレクタモジュールを組み込んで使用します。

## Web システム

1 台の負荷分散機（ロードバランサ）と複数の Web サーバおよびアプリケーションサーバで構築される Web クラスタシステムで、単一の業務サービスを提供します。

## (ア行)

---

### アーカイブ形式

EJB やサーブレットなどのアプリケーションの実体を J2EE サーバの作業ディレクトリに持つ J2EE アプリケーションの形式です。

### アウトプロセス

プロセスの起動のしかたです。アウトプロセスで起動させると、J2EE サーバのプロセス外で実行します。アウトプロセスでネーミングサービスを使用する場合、CORBA ネーミングサービスはユーザが起動する必要があります。

### アSEMBL

単体では動作しない EJB-JAR をアプリケーションの 1 構成要素として位置づけるための組み立て作業のことです。

### アノテーション

ソースコードにクラスやメソッドの付加情報などを埋め込むための記述方式です。

### アプリケーションサーバ

情報システムの中間に位置し、ユーザの要求（プレゼンテーション層）とデータベースなどの業務システム（データ層）の処理を橋渡しするためのアプリケーション層を構築するためのミドルウェアです。

日立のアプリケーションサーバ Cosminexus は、業務の開発から運用まで一貫した環境を提供します。

### アプレット

Web 経由で読み手のマシンのブラウザ上にダウンロードされる Java プログラムです。

### イベントリスナ

イベントを受け取って処理するオブジェクトです。Application Server では、Web アプリケーション単位でサーブレットコンテキストオブジェクトまたはセッションオブジェクトの状態変化イベントを受け取ります。

### インプロセス

プロセスの起動のしかたです。インプロセスで起動させると、J2EE サーバのプロセス内で実行するように最適化されるので、パフォーマンスの高いシステムが実現できます。ネーミングサービス、



トランザクションサービスおよび HTTP サーバ機能をインプロセスで起動できます。このとき、CORBA ネーミングサービス、インプロセス OTS およびインプロセス HTTP サーバは、J2EE サーバ起動時に自動で起動されます。

### インプロセス HTTP サーバ

J2EE サーバのインプロセスで動作する Web サーバ機能です。Web コンテナの機能の一部として提供されます。

### 運用監視エージェント

ホスト上で動作する論理サーバの稼働状況を監視して、稼働情報を収集するエージェントプログラムです。

### 運用管理エージェント

運用管理者の代わりに、それぞれのホスト上の論理サーバを起動したり、設定ファイルを更新したりするエージェントプログラムです。

### 運用管理コマンド (mngsvrutil)

Management Server を操作するための CUI です。

### 運用管理サーバ

Smart Composer 機能または運用管理ポータルでシステムを構築している場合に、Management Server を配置したホストです。運用管理ドメイン内の各ホストに配置されている運用管理エージェントに対して、管理操作を指示します。

### 運用管理サーバモデル

Smart Composer 機能または運用管理ポータルでシステムを構築している場合に、アプリケーションサーバとは別の運用管理用のホストに Management Server を配置するシステム構成モデルです。

### 運用管理ドメイン

Management Server が管理する論理サーバの集合です。同じ運用管理ポリシーが適用されます。

### 運用管理ポータル

Management Server を操作するための GUI です。Web ブラウザで表示します。

### 運用管理ポリシー

運用、監視、セキュリティなど、運用管理に関するポリシーです。

### エラーページのカスタマイズ

Web クライアントから存在しないリソースへのアクセスなどがあった場合に、Web コンテナから返却されるエラーページをユーザが作成した任意のページに変更して表示させる機能です。

## (カ行)

---

### ガーベージコレクション制御機能

リソースの排他中にガーベージコレクションの実行を待ち状態にする機能です。リソースの排他中にフルガーベージコレクションが発生し、処理が長時間中断されるのを回避できます。ガーベージコレクション制御機能は、パッチサーバの場合に使用できます。

## カスタムジョブ

JP1/AJS2 以外のプログラムと JP1/AJS2 が連携するジョブを定義する場合に、目的のジョブを容易に作成するために使用できるジョブのテンプレートです。Windows の場合、Cosminexus では、「論理サーバ制御用カスタムジョブ」と「アプリケーション制御用カスタムジョブ」の 2 種類を提供しています。

## カスタムログインモジュール

Cosminexus 標準ログインモジュール以外を使用して、各アプリケーションのユーザ認証をしたい場合に作成する実装クラスです。

## 仮想サーバ

負荷分散機で複数の実サーバを管理するための仮想的なサーバです。負荷分散機では、クライアントから仮想サーバにアクセスされたトラフィックを、実サーバに自動分配します。

## 稼働情報収集機能

J2EE サーバやバッチサーバ内の各機能の稼働情報を定期的に収集し、ファイルに出力する機能です。出力されたファイルを稼働情報ファイルといいます。

## 簡易 Web サーバ

Web コンテナを HTTP サーバとして使用して、スタブなどのダウンロードに利用する機能です。インプロセス HTTP サーバとは異なり、実運用時にはスタブのダウンロード以外の目的では使用できません。

Management Server を利用する場合は、「管理用サーバ」という名称で扱います。

## 監査証跡

データベースの機能で出力する、データベースに対する操作記録のことです。データベースに対する権限の運用が適切に行われているかどうかをチェックするために使用します。Cosminexus では、監査証跡に Cosminexus システムの情報を出力するためのデータベース監査証跡連携機能を提供しています。

## 監査ログ

システム構築者やシステム運用者が Cosminexus のプログラムに対して実行した操作、およびその操作に伴うプログラムの動作の履歴が出力されるファイルです。監査者が監査ログを調査することで、「いつ」「だれが」「何をしたか」を知ることができて、システムの運用が法規制、セキュリティ評価基準、または業界ごとの各種の基準に準拠していることを証明できます。

## 監視ツリー

JP1/IM 連携をしている場合に、統合スコープに表示するツリーのことです。Cosminexus システムでは、監視ツリーとして、Cosminexus 業務指向ツリーとサーバ指向ツリーの 2 種類を作成する機能を提供しています。

## 管理対象オブジェクト (AdminObject)

Connector 1.5 仕様で定義された、キューやトピックなどを使用したメッセージの送受信で使用するオブジェクトです。メッセージ送信や、同期または非同期でのメッセージ受信でのメッセージ変換で使用します。

管理対象オブジェクトは、メッセージプロバイダごとに特有のオブジェクトです。リソースアダプタが使用する管理対象オブジェクトは、リソースアダプタの DD ( ra.xml ) の <adminobject> タグ

で指定します。

### 機能レイヤ

性能解析トレース収集をする場合に、PRFトレースを出力する機能の層です。EJBクライアント、リダイレクタ、Webコンテナ、CTM、EJBコンテナ、JNDI、JTA、JCA/JDBCなどがあります。

### 業務指向ツリー

JP1/IMと連携している場合に統合スコープに表示する監視ツリーの一つです。Cosminexusシステム上で稼働しているJ2EEアプリケーションを主体に構成されている監視ツリーです。どの業務で問題が発生しているか、業務の視点でトラブルを監視できます。

### 業務処理プログラム

ビジネスロジックを実装したEnterprise Beanのことです。

### 共有スレッド数

Webコンテナ上にデプロイされているWebアプリケーションで共有されるスレッド数です。

### クライアントアプリケーション情報

性能解析トレースに出力される情報です。次に示すEnterprise Beanを呼び出す単位で設定される情報です。

- WebコンテナからEJBコンテナの呼び出し
- EJBクライアントからEJBコンテナの呼び出し
- EJBコンテナからEJBコンテナの呼び出し

### クラスタ

ある共通の機能を提供するサーバの集合です。

Management Serverでは、J2EEサーバクラスタまたはWebサーバクラスタを論理サーバとして扱えます。

### クラスタソフトウェア

システムの信頼性向上、稼働率向上を目的として、システムの切り替えを実現するプログラムです。実行中のサーバシステムに障害が発生した場合、事前に待機しているサーバシステムに、直ちに自動で切り替えることができます。そのため、オペレータが特に意識することなく、システムの信頼性や稼働率を高められます。

Cosminexusでは、クラスタソフトウェアを使用して、1:1系切り替えシステム、N:1リカバリシステム、および相互系切り替えシステムを実現できます。

AIX、HP-UXまたはLinuxを使用している場合は、クラスタソフトウェアとしてHAモニタが使用できます。Windowsの場合は、クラスタソフトウェアとしてMicrosoft Cluster Serviceが使用できます。

### グローバルCORBAネーミングサービス

CTMによってリクエストをスケジューリングする場合に、CTMドメイン内に含まれる複数のJ2EEサーバに登録されている業務処理プログラム(Stateless Session Bean)の情報を共有管理するネーミングサービスです。

### グローバルセッション

セッションフェイルオーバー機能では、障害発生前のHTTPセッションと、障害発生後に引き継がれ

た HTTP セッションを一つのセッションとして扱います。これをグローバルセッションといいます。

### グローバルセッション ID

グローバルセッションを管理する ID です。システム内でグローバルセッション情報を一意に識別するために、グローバルセッションに付けられます。

### グローバルセッション情報

グローバルセッションが持つ情報をグローバルセッション情報といいます。グローバルセッション情報は SFO サーバで管理されます。

### グローバルトランザクション

J2EE サーバが提供するトランザクションマネージャによって管理されるトランザクションです。2 フェーズコミットプロトコルを使用できるので、トランザクションに複数のリソースを参加させることができます。

## 系

系とは、業務処理に必要なハードウェアのほか、実行するプログラムや通信機器を含めたシステム全体の総称です。系の種類には、実行系と待機系があります。

### 計画系切り替え

系切り替えの方法の一つで、オペレータが系を保守する際に、計画的に系を切り替える方法です。

### 系切り替え

システムに障害が発生した場合やシステムの保守のために、システムを切り替えることをいいます。系切り替えが行われると、待機系は業務処理を引き継いで実行系となります。一方、実行系は、さらに障害が発生した場合に備えて待機系となります。以降、系切り替えのたびに、実行と待機を交代して、どちらかの系で常に業務処理を実行するようにします。

### 系障害

サーバを除いた系に発生する障害を指します。系のハードウェア障害や HA モニタの障害などがあります。なお、サーバに発生する障害はサーバ障害といいます。

### ゲートウェイ指定機能

クライアントと Web サーバとの間に、SSL アクセラレータや負荷分散機などのゲートウェイを配置している場合に、Web コンテナにゲートウェイ情報を通知し、welcome ファイルや Form 認証画面に正しくリダイレクトできるようにするための機能です。

### 現用系

最初に実行系として動作させる系を現用系といいます。系切り替えによって、実行系が待機系になった場合でも、呼び方は変わりません。

### コネクションスイーパー

一定の間隔でコネクションプール内の未使用コネクションを破棄する機能です。

### コネクションプーリング

サブレット、JSP、Enterprise Bean などの J2EE コンポーネントや、バッチアプリケーションからリソースへのアクセス量に応じて、リソースコネクションをメモリ上にプーリングする機能です。

リソースコネクションには、JDBC コネクション、リソースアダプタのコネクションなどがあります。コネクションをプーリングしておくことで、アプリケーションからのリソース接続要求を高速に処理できます。

### コミットオプション

CMP フィールドのキャッシュ方法と Entity Bean の状態遷移のキャッシュモデルを指定するオプションです。

Full caching の場合、Entity Bean は前回のトランザクションコミット時と同じ状態で開始されます。

Caching の場合、Entity Bean はデータベースの最新状態と同じ状態で開始されます。

No caching の場合、Entity Bean はデータベースの最新状態と同じ状態で開始され、かつ、トランザクションコミット時に Entity Bean は非活性になります。

### コンテキスト

Web アプリケーションをまとめた管理単位です。

### コンテキストルート

コンテキストのルートパスです。コンテキスト内の Web アプリケーションにアクセスするときに URL 上に指定します。コンテキストパスとも呼びます。

### コンテナ拡張ライブラリ

ユーザ作成ライブラリのインタフェースおよび機能を、J2EE コンテナ、Web コンテナ、またはバッチサーバから利用できるように拡張したライブラリです。Enterprise Bean、サーブレットおよび JSP から共通して利用できます。

## (サ行)

---

### サーバ管理コマンド

サーバで管理しているアプリケーションおよびリソースの設定をするためのコマンド群です。

### サーバ起動・停止フック機能

J2EE サーバ、バッチサーバまたは Web サーバの起動・停止時に、コンテナ拡張ライブラリを使用するための初期化処理および停止処理を自動的に呼び出す機能です。J2EE サーバまたは Web サーバが異常終了した場合には機能しません。

### サーバ指向ツリー

JP1/IM と連携している場合に統合スコープに表示する監視ツリーの一つです。サーバマシン（ホスト）を主体に構成されている監視ツリーです。

### サービス閉塞

J2EE アプリケーションの運用中に特定の業務処理プログラムを入れ替えたい場合などに、システム全体は稼働させたまま、該当サービスだけを安全に閉塞させる機能です。

### サービスユニット

J2EE サーバや Web サーバで構成される業務サービスを提供する閉じた部分系を、サービスユニットといいます。Smart Composer 機能では、サービスユニット単位でシステムを構築したり、一括

起動や一括停止をしたりできます。

Smart Composer 機能を利用してシステムを構築するときの概念です。

## サーブレット

サーバ側で Java を実行させる方法の一つです。

サーブレットは、Web サーバに対して、単に HTML 文書や画像ファイルを送るだけではなく、Web サーバと連携して、アプリケーションを実行し、その結果を HTML 文書として送り返す機能を提供します。

## サーブレットエンジンモード

サーバの動作モードの一つです。Web コンテナ部分だけを使用して、サーブレットエンジンを単独で動作させます。Enterprise Bean は動作しません。

サーブレットエンジンとは、サーブレット実行機能および JSP 実行機能を持つサーバのことです。

なお、サーブレットエンジンモードは互換用の動作モードとなります。

## サーブレットフィルタ

サーブレット /JSP へのリクエストや、サーブレット /JSP からのレスポンスを、フィルタリングする機能です。サーブレットフィルタによって、サーブレット /JSP の実行前のリクエスト、または実行後のレスポンスをラップして、データの変更、リソースに対するトレースの取得などができます。Cosminexus の機能を実現するためのサーブレットフィルタとしては、セッションフェイルオーバーフィルタ、および HTTP レスポンス圧縮用フィルタが提供されています。

## 実行系

業務処理を実行させる系（実行中のサーバ）を指します。

## 実行待ちキュー

同時実行スレッドの最大値を設定している場合に、実行スレッド数が最大に達したときに、リクエストをためておくキューのことです。

## 実パラメタ

Cosminexus の定義ファイルに設定するパラメタです。Smart Composer 機能では、抽象パラメタを一つ設定することで、互いに関連のある複数の実パラメタに、一括で値を設定できます。

## 自動系切り替え

系切り替えの方法の一つで、クラスタソフトウェアが自動的に系を切り替える方法です。

## シナリオ

JP1/AJS2・SO で扱う、シナリオジョブに実行順序を付けてネットワーク化したオブジェクトです。

シナリオジョブとは、シナリオ中に定義されたコマンド、シェルスクリプト、Windows 実行ファイルなどを定義したオブジェクトです。

シナリオは JP1/AJS2 に登録して実行できます。

## ジョブ

JP1/AJS2 で扱うシステム運用の各作業のことです。

## ジョブネット

JP1/AJS2 で扱うジョブの実行順序を関連づけたものです。

## シングルサインオン

一度のログイン処理で、ユーザ ID が異なる複数のシステムをシームレスに利用できるようにする機能です。

## シングルサインオン情報リポジトリ

統合ユーザ管理フレームワークのユーザマッピングを実行して、シングルサインオンでユーザ認証するための各システムの認証情報とマッピング情報を格納するリポジトリです。

## シングルサインオンライブラリ

ユーザマッピング情報の格納されたシングルサインオン情報リポジトリの情報を基にユーザマッピングをして、シングルサインオンを実現するライブラリです。

## スケールアウト

システム全体の処理性能を向上させることを目的として、サーバの台数を増やすことをいいます。

## スケールイン

システムの規模を縮小する場合などに、サーバの台数を減らすことをいいます。

## スケジュールキュー

CTM によってリクエストをスケジューリングする場合に、リクエストをためておくキューです。J2EE アプリケーション単位で作成され、CTM デーモンによって管理されます。

## スケジュールポリシー

CTM によってリクエストをスケジューリングする場合に、振り分ける基準を決めるポリシーです。負荷が軽い CTM デーモンに振り分ける方法と、リクエストを受け付けた CTM デーモンに優先的に処理させる方法のどちらかを選択できます。

## スマートエージェント

1.4 モードでグローバルトランザクションを使用する場合、または CTM を使用する場合に、動的な分散ディレクトリサービスを提供するプロセスです。Management Server を利用する場合、スマートエージェントは論理サーバとして扱えます。CTM はスマートエージェントによって管理されます。なお、グローバルトランザクションの場合でも、インプロセストランザクションサービスを利用するときは、スマートエージェントは不要です。

## 静的コンテンツ

HTML ファイルや画像ファイルなど、クライアントからの要求に対する応答に使用するファイルのうち、リクエスト内容に影響されないで常に同じ内容になるコンテンツのことです。

## 性能解析トレース

Cosminexus システムの性能解析をするためのトレース情報です。

## 性能解析トレースファイル

Cosminexus システムの性能解析をするためのトレース情報を CSV 形式で編集出力したテキストファイルです。

## セキュリティアイデンティティ

J2EE サーバによるユーザ認証が成功した時に Web コンテナや EJB コンテナに伝達される認証情報

です。

## セッション

Web アプリケーションに対する一連の作業を示す単位です。セッションは通常、Web クライアントから Web サーバへの複数のリクエストの集合から構成されます。

## セッションフェイルオーバー機能

J2EE アプリケーション実行中の HttpSession オブジェクトに登録されたオブジェクトをセッション情報として管理し、J2EE サーバで障害が発生した場合には、管理しているセッション情報をほかの J2EE サーバに引き渡す機能です。J2EE サーバで障害が発生し、ほかの J2EE サーバにリクエストが転送された場合でも、障害発生前の状態で業務を続行できます。

## セッションフェイルオーバーサーバ

セッションフェイルオーバー機能を使用する場合に、J2EE サーバを配置しないで、SFO サーバを配置して使用するホストです。

## セットアップウィザード

対話形式のウィザードプログラムを使用してシステムを構築するための機能です。必要最低限の項目を設定するだけで、J2EE アプリケーションを実行するためのシステムを構築できます。できるだけ少ない手順で実行環境を構築したい場合にお勧めの機能です。ただし、セットアップウィザードで実行できるのは、1 台のマシンを対象にしたシステム構築だけです。

## 占有スレッド数

利用できるスレッド数のうち、同時実行スレッド数制御の対象となるリクエストを確実に実行できるスレッド数です。Web コンテナ上にデプロイされている Web アプリケーションごと、または Web アプリケーション内の業務処理ロジックごとに指定できます。

## 相互系切り替えシステム

クラスタソフトウェアを使用して実現できるシステムの一つです。アプリケーションサーバの実行系と待機系を 1:1 にする構成で、それぞれのアプリケーションサーバを実行系として稼働させながら、お互いの待機系にするシステムです。

## 相互スタンバイ

クラスタソフトウェアを使用した運用をする場合に、アプリケーションサーバの実行系と待機系を 1:1 にする構成で、それぞれのアプリケーションサーバを実行系として稼働させながら、お互いの待機系にすることです。

## ( 夕行 )

---

### 待機系

障害に備えて待機させている系（待機中のサーバ）を指します。

### 抽象パラメタ

互いに関連のあるパラメタを一つのパラメタとしてまとめたものです。Smart Composer 機能で使います。Smart Composer 機能のコマンドを使用してパラメタを展開すると、一度に複数の実パラメタに値が展開されます。



## 通常モード

J2EE アプリケーションの動作モードです。通常モードは、本番稼働する J2EE アプリケーションを動作させるためのモードです。

## データソース

JDBC を使用してデータベースに接続する機能です。

## テストモード

J2EE アプリケーションの動作モードです。テストモードは、テスト用の J2EE アプリケーションを本番環境と同じように動作させるためのモードです。

## デプロイ

J2EE アプリケーションの場合、J2EE サーバ内にインポートした J2EE アプリケーションを、クライアントから実行可能な状態にすることです。

J2EE リソースアダプタの場合、J2EE サーバ内にインポートした J2EE リソースアダプタを、その J2EE サーバ上で動作するすべての J2EE アプリケーションから使用可能な状態にすることです。

## 展開ディレクトリ形式

EJB やサーブレットなどのアプリケーションの実体を、J2EE サーバの外部にある一定のルールに従ったファイル/ディレクトリを持つ J2EE アプリケーションの形式です。

## 統合コンソール

JP1/IM と連携している場合に、システムで発生した事象に関するエラーメッセージなどを表示する画面です。

## 統合スコープ

JP1/IM と連携している場合に、システムの監視画面として管理者の目的に合った監視ツリーを表示する画面です。

## 統合ネーミングスケジューラサーバ

CTM によるリクエストのスケジューリングをする場合に、J2EE サーバを配置しないで、グローバル CORBA ネーミングサービスを配置して使用するホストです。統合ネーミングスケジューラサーバのレプリカを作成することで、可用性を向上できます。なお、統合ネーミングスケジューラサーバにも CTM デーモンは必要です。

## 統合ネーミングスケジューラサーバモデル

CTM によるリクエストのスケジューリングをする場合に、J2EE サーバを配置しないでグローバル CORBA ネーミングサービスを配置するホストを作成するシステム構成モデルです。

## 統合ユーザ管理

Cosminexus システムにログインするユーザを統合管理するための仕組みです。ユーザ認証を実現する統合ユーザ管理フレームワークと、リポジトリ管理およびリソース監視を実現する統合ユーザ管理 GUI が使用できます。

## 統合ユーザ管理 GUI

Management Server の運用管理ポータルから、統合ユーザ管理の対象になるユーザ情報を格納したリポジトリを管理したり、統合ユーザ管理に関連するリソースを監視したりするための GUI です。

## 統合ユーザ管理フレームワーク

統合ユーザ管理でのユーザ認証を実現するフレームワークです。Java 標準仕様 (JAAS) に従った API が提供されています。

## 動作モード

サーバの動作モードです。J2EE サーバモードとサーブレットエンジンモードがあります。J2EE サーバモードには、さらに、1.4 モード、ベーシックモードがあります。ただし、ベーシックモードおよびサーブレットエンジンモードは旧バージョンとの互換用の動作モードとなります。

## 動的コンテンツ

サーブレットや JSP のように、クライアントからの要求に応じて動的に生成されるコンテンツのことです。

## トランザクションサービス

グローバルトランザクションを使用する場合に、トランザクションを管理するサービスです。TPBroker OTS によって提供されるサービス全体を表します。トランザクションサービスは、J2EE サーバのインプロセスで起動されます。

## トランザクションマネージャ

トランザクションを管理する機能です。Application Server が提供しています。グローバルトランザクションの場合は、トランザクションマネージャのバックエンドで TPBroker OTS が使用されています。

## トレース取得ポイント

Cosminexus システムの各機能レイヤで PRF トレースを出力するポイントのことです。PRF トレース取得レベルの指定によって、トレース取得ポイントの数は異なります。

## (ナ行)

---

### ネーミング切り替え

JNDI を介して EJB ホームオブジェクトの登録、削除、検索をする時に、JNDI が接続する CORBA ネーミングサービスを切り替えることです。

### ネーミングサービス

オブジェクトに名前を付けて格納場所を管理しておくことで、格納先を知らなくても名前からそのオブジェクトを利用できるようにするサービスです。Application Server では CORBA ネーミングサービスを利用します。

Management Server では、ネーミングサービスを論理サーバとして扱えます。

## (ハ行)

---

### バインド先アドレス設定機能

Web サーバ連携および簡易 Web サーバで利用する IP アドレスを、Web コンテナで明示的に指定する機能をバインド先アドレス設定機能といいます。複数の物理ネットワークインタフェースを持つ

ホストまたは一つの物理ネットワークインタフェースに対して、ホストで実行する場合に、この機能を使用することで、任意の IP アドレスを設定できます。

### バッチアプリケーション

バッチ処理を実装した Java アプリケーションです。バッチ実行コマンドを使用して、バッチサーバ上で実行します。バッチ実行コマンドを JP1/AJS2 のジョブとして定義しておく、JP1/AJS2 からバッチアプリケーションを実行できます。

### バッチサーバ

バッチアプリケーションを実行するためのサーバです。バッチアプリケーションの実行機能のほかに、バッチアプリケーションからデータベースに接続したり、EJB にアクセスしたりする機能も提供しています。

### パフォーマンストレーサ

PRF デーモンのことです。

Management Server では、パフォーマンストレーサを論理サーバとして扱えます。

### ハンドラ

ロガーから受け取ったアプリケーションのログをファイルやコンソールに出力する、Java ログング API の Handler オブジェクトです。J2EE アプリケーション、バッチアプリケーション、EJB クライアントアプリケーションで使用できます。

Application Server では、日立トレース共通ライブラリ形式のログをファイルに出力するためのファイルハンドラである、CJMessageFileHandler を提供しています。

### ビジネスインタフェース

Enterprise Bean を呼び出すためのビジネスメソッドを定義するインタフェースです。

### 日立トレース共通ライブラリ

日立が提供するトレース採取用のライブラリです。複数の製品で共通のライブラリを利用するので、各製品のトレース形式が統一できます。

### 非保護区

メソッドキャンセルができる領域を指します。ユーザプログラムのコードが非保護区に該当します。メソッドキャンセルとは、J2EE アプリケーション実行時間の監視で提供される機能の一つです。また、バッチ強制停止コマンドでも使用されます。

### 負荷分散機

Web ブラウザなどからのリクエストを一元的に受け付けて、同等の機能を持つ複数のサーバに転送して各サーバの負荷を分散させるための装置です。

### 物理コネクション

接続先リソースのコネクションです。通常は J2EE コンテナが操作します。リソースアダプタの場合は、javax.resource.spi.ManagedConnection、JDBC の場合は javax.sql.XAConnection に該当します。

### 物理ティア

同一のデプロイメント（プロセスの配置、J2EE アプリケーションやリソースアダプタのデプロイ、定義）を適用するサーバマシンの集合を物理ティアといいます。

Smart Composer 機能を利用してシステムを構築するときの概念です。

## フロント EJB

リクエストの受け口となる Enterprise Bean をフロント EJB といいます。

## ベーシックモード

旧バージョンとの互換性を確保するためのサーバの動作モードの一つです。  
単一リソースとのトランザクション管理ができます。

## 別名

JNDI 名前空間に登録する EJB ホームオブジェクトリファレンスや、J2EE リソースに付けられる任意の名前です。ユーザ指定名前空間機能を使用している場合、別名を付けられます。Optional Name ともいいます。

## 保護区

メソッドキャンセルができない領域を指します。J2EE サービスのソースなどが保護区に該当します。メソッドキャンセルとは、J2EE アプリケーション実行時間の監視で提供される機能の一つです。また、バッチ強制停止コマンドでも使用されます。

## ホスト単位管理モデル

Smart Composer 機能または運用管理ポータルでシステムを構築している場合に、Management Server をアプリケーションサーバと同じホストに配置するシステム構成モデルです。

## (マ行)

---

### 未決着トランザクション

決着していない、仕掛かり中のトランザクションです。

### メソッドキャンセル機能

J2EE アプリケーション実行時間の監視機能の機能の一つです。メソッドタイムアウト機能でタイムアウトが通知されたあと、メソッドのキャンセルを実施する機能です。キャンセル時にデータベースにアクセスしている場合は、トランザクションの強制決着もします。  
また、メソッドキャンセル機能は、バッチ強制停止コマンド実行時にも使用されます。

### メソッドタイムアウト機能

J2EE アプリケーション実行時間の監視機能の機能の一つです。監視基盤にあるリクエストのうち、一定時間内に終了しなかったメソッド処理を、タイムアウトとしてユーザに通知します。

### メッセージインフロー

メッセージプロバイダがアプリケーションサーバ上のメッセージエンドポイントにメッセージを配送する場合の規約です。Connector 1.5 の Inbound の通信モデルを使用する場合に使用できる機能です。

メッセージプロバイダには EIS などが該当します。アプリケーションサーバ上のメッセージエンドポイントには Message-driven Bean が該当します。

## 面数

ログファイルを保持する数です。ログファイルは面数分作成され、循環して使用されます。

## メンバリソースアダプタ

コネクションプールをクラスタ構成にしている場合に、データベースノードに接続するリソースアダプタです。メンバリソースアダプタは、必ずルートリソースアダプタを経由してアクセスされます。

## (ヤ行)

---

### ユーザ指定名前空間機能

JNDI 名前空間に登録する EJB ホームオブジェクトリファレンスや J2EE リソースの名前に、ユーザが任意の別名を付けられる機能です。別名を付与することによって、CTM によってリクエストを振り分けたり、負荷分散機やロードバランサによって負荷分散したりする場合に、サーバ名やアプリケーション名に依存しない名前で見つけられるようになります。

### ユーザ情報

Cosminexus システムにユーザがログインするための情報です。

### ユーザ情報リポジトリ

統合ユーザ管理をする場合に、ログイン時のユーザ認証に使用する情報を格納するリポジトリです。LDAP ディレクトリサーバまたはデータベース (RDB) が使用できます。

### ユーザスレッド

ユーザがプログラムの中で明示して生成するスレッドのことです。

### ユーザ認証ライブラリ

LDAP ディレクトリサーバで構築されたユーザ情報リポジトリの情報を基にユーザ認証をして、認証したユーザの情報をアプリケーションに提供する、JAAS 対応ユーザ管理用ライブラリです。

### ユーザプロパティファイル

usrconf.properties のことです。J2EE サーバ、Web コンテナサーバまたはサーバ管理コマンドを実行する JavaVM のシステムプロパティを指定します。なお、J2EE サーバ、Web コンテナサーバおよびサーバ管理コマンドで使用するユーザプロパティファイルは、それぞれ格納先および指定できるキーが異なります。

### ユーザマッピング

異なるユーザ情報リポジトリに格納した同じユーザについての情報を対応づけることです。

### ユーザログ

J2EE アプリケーション、バッチアプリケーション、または EJB クライアントアプリケーションが出力するログのことです。

### 予備系

最初に待機系として起動する系を予備系といいます。系切り替えによって、待機系が実行系になった場合でも、呼び方は変わりません。

## (ラ行)

---

### ライトトランザクション

ローカルトランザクションを最適化した状態で使用できる機能です。

### ライブラリ JAR

JAR ファイル形式でパッケージされた、共通ライブラリです。J2EE アプリケーション内の J2EE コンポーネントから共通に使用できます。

### ラウンドロビン検索

複数の CORBA ネーミングサービス上にある同一名称の EJB ホームオブジェクトを、ラウンドロビンポリシーに従ってルックアップする検索機能のことです。

### ラウンドロビンポリシー

複数の CORBA ネーミングサービス上にある同一名称の EJB ホームオブジェクトをルックアップする場合に適用されるポリシーです。

EJB ホームオブジェクトリファレンスをラウンドロビン検索によって取得できるので、EJB クライアントでは意識しないで J2EE サーバに送信するリクエストの負荷分散を実現できます。

### リカバリ専用サーバ

クラスタソフトウェアを使用して N:1 リカバリシステムを構築している場合に待機系として機能するアプリケーションサーバです。

障害が発生した実行系のアプリケーションサーバに未決着のトランザクションがあった場合に、系が切り替えられ、該当するトランザクションを解決してリソースを解放します。

### リクエスト処理スレッド

Web ブラウザなどのクライアントからのリクエストを受け付けて処理するために Web サーバで生成されるスレッドです。インプロセス HTTP サーバの場合は、Web コンテナで生成されます。

### リクエストのスケジューリング

クライアントからのリクエストを実行するサーバを負荷に応じて的確に振り分けたり、サーバに送信するリクエストの数を制限したりする機能です。

### リソースアダプタ

J2EE Connector Architecture によって、J2EE サーバまたはバッチサーバと、EIS を接続するための接続機能です。

Cosminexus システムでは、データベースに接続するためのリソースアダプタである DB Connector および DB Connector for Cosminexus RM を提供しています。また、OpenTP1 の SPP と接続するためのリソースアダプタである uCosminexus TP1 Connector、TP1/Message Queue と接続するためのリソースアダプタである TP1/Message Queue - Access、データベース上に実現したキューに接続するためのリソースアダプタである Cosminexus RM も使用できます。

### リソースマネージャ

リソースを管理する機能です。DBMS などが該当します。

### リダイレクタ

Web サーバに登録し Web コンテナへの接続を可能にするプラグインコンポーネント (ライブラリ)

です。Web サーバに登録したリダイレクタによって、リクエストを URL パターンまたはラウンドロビン方式で複数の Web コンテナに振り分けて処理できます。

### リデプロイ機能

J2EE アプリケーションを入れ替えるときに使用する機能です。リデプロイ機能による入れ替えは、J2EE アプリケーションのテスト時などに、修正した J2EE アプリケーションと動作中の J2EE アプリケーションを入れ替えたいときに使用します。

### リバースプロキシサーバ

インターネットなどの外部ネットワークとアプリケーションサーバが配置されている内部ネットワークの間の DMZ に配置され、外部ネットワークからのリクエストを内部ネットワークに中継するための機能を持つ Web サーバです。

Cosminexus では、Hitachi Web Server をリバースプロキシサーバとして使用できます。リバースプロキシサーバには、リバースプロキシモジュールが組み込まれて動作します。

### リモートインタフェース

Java RMI のインタフェース規定に従い、RMI-IIOP 通信によって Enterprise Bean の呼び出しをするインタフェースです。

リモートホームインタフェースとリモートビジネスインタフェースが含まれます。

### リロード機能

開始状態にあるアプリケーションを停止させないで、アプリケーションを構成するクラスや JSP を、少ない手順で動的に入れ替える機能です。

### ルートアプリケーション情報

性能解析トレースに出力される情報です。イベントで一連の処理の先頭になるプロセスで取得した情報です。

### ルートコンテキスト

コンテキストルートが空文字（コンテキストルートに名称が指定されていない）の、WAR ファイル内のルートパスです。

### ルートルソースアダプタ

コネクションプールをクラスタ構成にしている場合に、メンバリソースアダプタを束ね、ユーザプログラムからのリクエストを受け付けるアダプタのことです。

### レルム

統合ユーザ管理をしているシステムで、同一の認証ポリシーを適用する範囲のことです。

### ローカルインタフェース

Enterprise Bean の呼び出しを Java のメソッド呼び出しで実行するインタフェースです。通信は発生しません。

ローカルホームインタフェースとローカルビジネスインタフェースが含まれます。

### ローカルトランザクション

接続先のリソースマネージャによって管理されるトランザクションです。単一のリソースだけがトランザクションに参加できます。

## ロードバランサ

リダイレクタを使用してリクエストを振り分ける場合に使用するワーカ定義のうち、クラスタ構成の場合に使用する特殊なワーカ定義です。振り分け先の Web コンテナのリストが定義されています。リダイレクタでは、この定義を基に、ラウンドロビン方式でリクエストの振り分け処理をします。

## ロール

コンテキストに対し、アクセス制御するときを使用される単位です。ロールはグループごとに定義されます。また、アクセス制御するコンテキストについては、そのコンテキストにアクセスするのに必要なロールが定義されます。アクセスしたユーザの持つロールがコンテキストに定義されたロールと一致した場合、そのコンテキストへのアクセスは成功します。

## ロガー

J2EE アプリケーション、バッチアプリケーション、または EJB クライアントアプリケーションのログを取得する、Java ロギング API の Logger オブジェクトです。

## ログインモジュール

JAAS に準拠した統合ユーザ管理のユーザ認証に使用する、JAAS API の LoginModule インタフェースの実装クラスです。

## 論理コネクション

サーブレットや Enterprise Bean などの J2EE コンポーネントが直接操作するコネクションです。リソースアダプタの場合は `javax.resource.cci.Connection`、JDBC の場合は `java.sql.Connection` に該当します。

## 論理サーバ

Management Server の運用管理の対象になる、サーバまたはクラスタです。サーバには、Web サーバ、J2EE サーバなどがあります。クラスタとは、ある共通の機能を提供するサーバの集合です。

## 論理ネーミングサービス

JNDI がラウンドロビン検索の対象にする複数の CORBA ネーミングサービスのことです。

## 論理ユーザサーバ

任意のサービスやプロセスを Management Server の管理対象として定義した論理サーバです。Management Server で起動・停止したり、ステータス監視したりできるようになります。

## (ワ行)

---

## ワーカプロセス

Web サーバの背後で動作する Web コンテナのプロセスです。リダイレクタから見た場合の Web コンテナ実行プロセスになります。具体的には、J2EE サーバまたは Web コンテナサーバに当たりません。



---

# 索引

## 記号

---

-XX:+HitachiJavaClassLibTrace 306  
-XX:+HitachiLocalsInThrowable オプション  
が指定されている場合 322  
-XX:+HitachiOutOfMemoryStackTrace 306  
-XX:+HitachiVerboseGC 306

## 数字

---

1.4 モード 664  
1:1 系切り替えシステム 664  
1:1 系切り替えシステムで計画的に系を切り  
替える場合の起動と停止 438  
1:1 系切り替えシステムで実行できる運用操  
作 434  
1:1 系切り替えシステムでトラブルが発生し  
た場合 339  
1:1 系切り替えシステムの起動 435  
1:1 系切り替えシステムの起動と停止 434  
1:1 系切り替えシステムの停止 436

## A

---

Application Server 664

## B

---

Bean-Managed Persistence 664  
Bean-Managed Transaction 664  
BJEX 664  
BMP 664  
BMT 664

## C

---

cjgetsysinfo コマンドの実行によって実行さ  
れる OS のコマンド 252  
cjjspe コマンド 149  
cjreloadapp コマンド 148  
CMP 664  
CMT 664  
Component Container 管理者 29, 632, 664

Component Container 管理者が実施できる  
システム運用時の操作 29  
Component Container 管理者による運用に  
関する留意事項 29  
Component Container 管理者を設定してい  
る場合にパーミッションが変更されるディレ  
クトリ 30  
Container-Managed Persistence 664  
Container-Managed Transaction 664  
Cookie スイッチング機能 665  
CORBA ネーミングサービス 665  
CORBA ネーミングサービスのスレッドダン  
プの取得 243  
core ダンプの生成を示すメッセージの出力内  
容 319  
Cosminexus 665  
Cosminexus Component Container 665  
Cosminexus Component Transaction  
Monitor 665  
Cosminexus Component Transaction  
Monitor のログ 265  
Cosminexus Component Transaction  
Monitor のログの取得 215  
Cosminexus DABroker Library のイベント  
トレースの出力項目 298  
Cosminexus DABroker Library のスプール  
情報の調査 273  
Cosminexus DABroker Library のトレース  
ファイルの運用時の留意事項 347  
Cosminexus DABroker Library のログ 265  
Cosminexus DABroker Library のログの取  
得 216  
Cosminexus Operator Plug-in 665  
Cosminexus Performance Tracer 665  
Cosminexus Performance Tracer のログ  
265  
Cosminexus Performance Tracer のログの出  
力先 215  
Cosminexus Performance Tracer のログの取  
得 214

Cosminexus Performance Tracer のログの種類 214  
Cosminexus Reliable Messaging 665  
Cosminexus システム 665  
Cosminexus の作業ディレクトリの内容の取得 258  
Cosminexus のシステムの運用サイクル 2  
Cosminexus の定義情報の取得 257  
Cosminexus のリソース設定情報の取得 258  
Cosminexus のログの調査 260  
Cosminexus 標準ログインモジュール 665  
CTM 666  
CTM デーモン 666  
CTM ドメイン 666  
CTM ドメインマネージャ 666  
CTM の稼働情報を監視する 22  
CTM の稼働統計情報の確認 81  
CTM の稼働統計情報の収集方法 81  
CTM の稼働統計情報の出力先と出力情報 82  
CTM の稼働統計情報の出力例 85  
CTM のコマンド 19  
CTM のスケジュールキューの稼働状況の確認 123  
CTM のスケジュールキューの稼働状況の確認と最大同時実行数の変更 123  
CTM のスケジュールキューの稼働状況の確認と同時実行数の変更 114  
CTM のスケジュールキューの同時実行数の変更 123  
CTM のトレース取得ポイント 466  
CTM レギュレータ 666  
CTM を使用したオンライン状態での J2EE アプリケーションの入れ替え 153  
CTM を使用したサービス閉塞 135  
C ヒープが不足した場合 316  
C ヒープが不足した場合のメッセージログの出力項目 318  
C ヒープ不足を示すメッセージログの出力内容 316

---

## D

DB Connector 666

DB Connector , JCA コンテナのトレース取得ポイント 526  
DB Connector の稼働情報ファイルに出力される情報 54  
DD 666  
Dependency Injection 666  
DI ターゲット 667  
DTD 667

---

## E

EAR ファイル 667  
EIS 667  
EJB 667  
EJB-JAR ファイル 667  
EJB クライアント 667  
EJB クライアントアプリケーション 667  
EJB クライアントアプリケーションのシステムログに関する留意事項 344  
EJB クライアントアプリケーションのシステムログの出力先 234  
EJB クライアントアプリケーションのシステムログの取得 232  
EJB クライアントアプリケーションのシステムログの種類 233  
EJB クライアントアプリケーションのユーザログの取得 218  
EJB クライアントアプリケーションのログの調査 272  
EJB クライアントでトラブルが発生した場合 342  
EJB コンテナ 667  
EJB コンテナのトレース取得ポイント 504  
EJB タイマ 667  
Enterprise Bean 667  
Enterprise JavaBeans 667  
Entity Bean 667  
Exception トレースログ 284  
Exception トレースログとして出力される情報 284  
Exception トレースログの出力項目 288  
Exception トレースログのファイル 348

## H

HA モニタ 668  
 HA モニタによる 1:1 系切り替えシステム  
 434  
 HTTP リクエスト実行待ちキュー枯渇監視情  
 報 78

## I

Inbound 668  
 Internal Error が発生した場合 319

## J

J2EE 668  
 J2EE アプリケーション 668  
 J2EE アプリケーション実行時間監視で出力  
 されるログ情報 622  
 J2EE アプリケーション実行時間の監視機能  
 668  
 J2EE アプリケーションとリソースの管理  
 113  
 J2EE アプリケーションとリソースの管理の  
 概要 114  
 J2EE アプリケーションとリソースを保守す  
 る 25  
 J2EE アプリケーションの入れ替え 144  
 J2EE アプリケーションの入れ替えと保守  
 115, 144  
 J2EE アプリケーションの稼働状態に応じた  
 パフォーマンスチューニングを実行する 24  
 J2EE アプリケーションの強制停止 668  
 J2EE アプリケーションの強制停止に失敗し  
 た場合 333  
 J2EE アプリケーションの実行時間の監視と  
 キャンセル 114, 126  
 J2EE アプリケーションの実行状態の確認  
 127  
 J2EE アプリケーションの状態 139  
 J2EE アプリケーションの通常停止 668  
 J2EE アプリケーションの停止 115, 137  
 J2EE アプリケーションの停止方法の種類  
 130  
 J2EE アプリケーションの閉塞 115  
 J2EE アプリケーションの閉塞と停止 130  
 J2EE アプリケーションの名称変更 151  
 J2EE アプリケーションのユーザログ 217  
 J2EE アプリケーションを入れ替える 25  
 J2EE アプリケーションを入れ替える場合に  
 テストモードを使用する手順 155  
 J2EE アプリケーションを運用する 23  
 J2EE アプリケーションを開始するときの  
 JSP の事前コンパイル 149  
 J2EE アプリケーションを強制停止する 25  
 J2EE アプリケーションを停止する 24  
 J2EE コンテナ 668  
 J2EE コンポーネント 668  
 J2EE サーバ 668  
 J2EE サーバ・リダイレクタ・サーバ管理コ  
 マンドのログの取得 189  
 J2EE サーバおよびリダイレクタのログ 192  
 J2EE サーバおよびリダイレクタのログ種別  
 261  
 J2EE サーバ起動時に出力されるログ情報  
 622  
 J2EE サーバのスレッドダンプの取得 242  
 J2EE サーバのメモリダンプの取得 246  
 J2EE サーバモード 669  
 J2EE サーバを操作するコマンド 19  
 J2EE サービス 669  
 J2EE リソース 669  
 J2EE リソースアダプタ 669  
 jaas.conf 669  
 JAAS のコンフィグレーションファイル 669  
 Java 669  
 JavaBeans リソースの入れ替え 116, 160  
 Java Database Connectivity 669  
 javagc コマンド 245, 651  
 Java Naming and Directory Interface 669  
 JavaServer Pages 671  
 javatrace 起動コマンドのコマンドライン  
 316  
 JavaVM が異常終了した場合 336  
 JavaVM が出力するメッセージログ 308  
 JavaVM が出力するメッセージログの調査  
 (標準出力およびエラーレポートファイル)  
 308

- JavaVM のガーベージコレクションのログ 174
  - JavaVM のガーベージコレクションログの取得 245
  - JavaVM のガーベージコレクションログの調査 305
  - JavaVM の稼働情報ファイルに出力される情報 49
  - JavaVM のスタックトレース 175
  - JavaVM のスタックトレース情報の取得 259
  - JavaVM のスレッドダンプ 174
  - JavaVM のスレッドダンプの取得 240
  - JavaVM のスレッドダンプの調査 303
  - Java ヒープの使用状況 313
  - Java ロギング API 669
  - JCA コンテナ 669
  - JCA リソースの稼働情報ファイルに出力される情報 56
  - JDBC 669
  - JDBC インタフェースメソッドのトレースの出力項目 301
  - JDBC ドライバのトレース (Cosminexus DABroker Library とのイベントトレース) 301
  - JDBC ドライバのトレース (JDBC インタフェースメソッドトレース) 300
  - JDBC ドライバのトレースファイル 349
  - JNDI 669
  - JNDI 名前空間 670
  - JNDI のトレース取得ポイント 517
  - JP1 670
  - JP1/AJS2 670
  - JP1/AJS2 - SO 670
  - JP1/AJS2 と連携したシステムでのトラブルへの対処 339
  - JP1/Cm2/NC - Manager 670
  - JP1/Cm2/NC - Manager for Cosminexus 670
  - JP1/IM 670
  - JP1/IM - CM 670
  - JP1/IM と連携したシステムでのトラブルへの対処 338
  - JP1/NETM/Audit 670
  - JP1/SC/DPM 670
  - JP1 イベント 671
  - JP1 ジョブ運用管理サーバ 671
  - JP1 統合運用管理サーバ 671
  - JP1 と連携したシステムでトラブルが発生した場合 338
  - JSP 671
  - JSP EL 671
  - JSP コンパイル結果 671
  - JSP 事前コンパイル 671
  - JSP 事前コンパイル機能 149
  - JSP タグライブラリ 671
  - JTA のトレース取得ポイント 521
- ## M
- 
- Management Server 5, 671
  - Management Server による運用管理の概要 5
  - Management アクション 36, 671
  - Management イベント 22, 36, 672
  - Management イベント発行ログ 211, 212
  - Message-driven Bean 672
  - Message-driven Bean の稼働情報ファイルに出力される情報 53
  - MIB オブジェクト 672
  - MIB ファイル 672
  - Microsoft Cluster Service 672
  - mngsvrutil 677
- ## N
- 
- N:1 リカバリシステム 672
  - N:1 リカバリシステムでトラブルが発生した場合 340
  - N:1 リカバリシステムの起動 430, 457
  - N:1 リカバリシステムの起動と停止 430, 457
  - N:1 リカバリシステムの停止 431, 459
  - N 秒 72
- ## O
- 
- Object Transaction Service 672

Oracle RAC 672  
 ORB ゲートウェイ 672  
 OS の状態・ログ 174  
 OS の状態情報および OS のログの調査 320  
 OS の状態情報と OS のログの取得 250  
 OS の統計情報 174  
 OS の統計情報の取得 254  
 OS のログの取得 254  
 OTM ゲートウェイ 672  
 OTS 672  
 OTS のトレース取得ポイント 551  
 Outbound 673

## P

---

PRF デモン 673  
 PRF トレース 673  
 PRF トレース出力ライブラリ 673  
 PRF トレース取得レベル 92, 462, 637, 673  
 PRF トレース取得レベル〔BMT〕523  
 PRF トレース取得レベル〔CMT〕521  
 PRF トレース取得レベル〔CTM〕466  
 PRF トレース取得レベル〔DB Connector〕  
 526, 535  
 PRF トレース取得レベル〔DB Connector for  
 Cosminexus RM〕540  
 PRF トレース取得レベル〔J2EE サーバ〕  
 566  
 PRF トレース取得レベル〔JNDI〕518  
 PRF トレース取得レベル〔Message-driven  
 Bean (EJB2.0)〕507  
 PRF トレース取得レベル〔Message-driven  
 Bean (EJB2.1)〕509  
 PRF トレース取得レベル〔OTS〕551  
 PRF トレース取得レベル〔RMI〕550  
 PRF トレース取得レベル〔Session Bean・  
 Entity Bean〕504  
 PRF トレース取得レベル〔Web コンテナ〕  
 476, 483  
 PRF トレース取得レベル〔インプロセス  
 HTTP サーバ〕479  
 PRF トレース取得レベル〔コネクションア  
 ソシエーション〕537

PRF トレース取得レベル〔コネクション自  
 動クローズ時〕538  
 PRF トレース取得レベル〔セッションフェ  
 イルオーバー機能〕498  
 PRF トレース取得レベル〔トランザクシ  
 ョンタイムアウト〕524  
 PRF トレース取得レベル〔バッチアプリ  
 ケーション実行機能〕565  
 PRF トレース取得レベル〔フィルタのト  
 レース(正常に処理が終了した場合)〕490  
 PRF トレース取得レベル〔フィルタのト  
 レース(例外が発生した場合)〕494  
 PRF トレース取得レベル〔リダイレクタ〕  
 472  
 PRF トレース取得レベル〔ワーク管理〕544  
 PRF トレースファイル 23, 92, 637, 673  
 PRF のコマンド 19

## R

---

RMI-IIOP 673  
 RMI のトレース取得ポイント 549

## S

---

SecurityManager 673  
 Server Plug-in 20, 673  
 Session Bean 673  
 SFO サーバ 674  
 SFO サーバアプリケーション 674  
 siginfo 情報 (UNIX の場合) 311  
 Smart Composer 機能 6, 19, 674  
 snapshotlog.2.conf 182  
 snapshotlog.conf 182  
 snapshot ログ 26, 178, 674  
 snapshot ログとして収集できる資料 182  
 snapshot ログの収集 186  
 snapshot ログの収集対象一覧 568  
 SNMP 連携用形式ファイル 674  
 Stateful Session Bean 674  
 Stateful Session Bean の稼働情報ファイル  
 に出力される情報 51  
 Stateless Session Bean 674

Stateless Session Bean の稼働情報ファイル  
に出力される情報 52  
syslog 191, 221  
syslog の出力形式と出力項目 (UNIX の場  
合) 270

---

## T

Timer Service 674

---

## U

ua.conf 675  
uCosminexus Batch Job Execution Server  
675  
uCosminexus Client 675  
uCosminexus Operator 675  
URL グループの稼働情報ファイルに出力さ  
れる情報 60  
usrconf.properties 675

---

## V

VM の状態 313

---

## W

WAR ファイル 675  
Web アプリケーション 675  
Web アプリケーション (WebAPI) の最大同  
時実行スレッド数の動的変更の設定例 121  
Web アプリケーション単位の稼働状態でのパ  
フォーマンスチューニング 117  
Web アプリケーションの稼働状況の確認  
118  
Web アプリケーションの稼働情報ファイルに  
出力される情報 58  
Web アプリケーションの最大同時実行スレ  
ッド数の設定変更 120  
Web アプリケーションの最大同時実行スレ  
ッド数の動的変更 117  
Web アプリケーションの最大同時実行スレ  
ッド数を変更する場合に参考になる情報 119  
Web クライアント構成の例 100  
Web コンテナ 675

Web コンテナサーバ 675  
Web コンテナサーバの起動 634  
Web コンテナサーバの停止 635  
Web コンテナの稼働情報ファイルに出力され  
る情報 59  
Web コンテナのトレース取得ポイント  
476, 483  
Web コンテナのトレース取得ポイント (セッ  
ションフェイルオーバー機能のトレース) 498  
Web サーバ 675  
Web サーバで収集した性能解析トレースファ  
イルの例 102  
Web サーバのレスポンスタイムの解析 100  
Web サーバのログ 175  
Web サーバ連携 676  
Web サーバログの取得 258  
Web システム 676  
Web フロントシステムで実行できるサービス  
閉塞 132  
Web フロントシステムでのサービス閉塞  
132

---

## X

XA トレースの取得形式 298

---

## あ

アーカイブ形式 676  
アウトプロセス 676  
アクセス状況に応じた Web アプリケーショ  
ンの最大同時実行数の動的変更 114  
アクセス状況に応じた一時的な Web アプリ  
ケーション単位の最大同時実行スレッド数の  
変更 117  
アクセスログの出力項目 269  
アセンブル 676  
アノテーション 676  
アプリケーションサーバ 676  
アプリケーションサーバで収集した性能解析  
トレースファイルをフィルタリングした例  
102  
アプリケーションサーバ内でのリクエストの  
処理状況の調査 101

アプリケーションサーバの 1:1 系切り替えシステムをメンテナンスする場合の起動と停止 438

アプリケーションサーバを対象にした 1:1 系切り替えシステム 418

アプリケーションサーバを対象にした 1:1 系切り替えシステムで計画的に系を切り替えるときの起動と停止 420

アプリケーションサーバを対象にした 1:1 系切り替えシステムの起動 419

アプリケーションサーバを対象にした 1:1 系切り替えシステムの起動と停止 418

アプリケーションサーバを対象にした 1:1 系切り替えシステムの停止 419

アプリケーションサーバを対象にした 1:1 系切り替えシステムをメンテナンスする場合の起動と停止 420

アプリケーションのユーザログの取得 217, 232

アプレット 676

## い

移行コマンド (cjenvupdate) のログ 205, 229, 263

異常終了とシグナル種別 310

異常停止 40

一次送付資料 172, 183

イベントトレース 297

イベントリスナ 676

イベントログ 190, 220

イベントログの出力形式と出力項目

(Windows の場合) 270

入れ替える J2EE アプリケーション内の JSP の事前コンパイル 149

インプロセス 676

インプロセス HTTP サーバ 677

インプロセス HTTP サーバのアクセスログの出力形式と出力項目 268

インプロセス HTTP サーバのログ 191, 203, 263

インプロセストランザクションサービスを使用している場合 214

## う

運用監視エージェント 677

運用監視作業の概要 36

運用管理エージェント 677

運用管理エージェント, 運用監視エージェント, および Management Server のログの出力先 209

運用管理エージェント・運用監視エージェント・Management Server のログ 264

運用管理エージェント・運用監視エージェント・Management Server のログの取得 208, 231

運用管理コマンド 677

運用管理コマンド (mngsvrutil) 6, 19

運用管理コマンド (mngsvrutil) を利用する場合の留意事項 31

運用管理サーバ 677

運用管理サーバの 1:1 系切り替えシステムをメンテナンスする場合の起動と停止 442

運用管理サーバモデル 677

運用管理サーバを対象にした 1:1 系切り替えシステム 424

運用管理サーバを対象にした 1:1 系切り替えシステムで計画的に系を切り替えるときの起動と停止 425

運用管理サーバを対象にした 1:1 系切り替えシステムの起動 424

運用管理サーバを対象にした 1:1 系切り替えシステムの起動と停止 424

運用管理サーバを対象にした 1:1 系切り替えシステムの停止 425

運用管理ドメイン 5, 677

運用管理ポータル 19, 677

運用管理ポリシー 677

## え

エラートレース 279

エラートレースのエラー事象を示す情報の出力項目 280

エラートレースのデータベース接続情報の出力項目 280

エラートレースの履歴情報の出力項目 281

エラートレースファイル 347  
 エラーページのカスタマイズ 677  
 エラーメッセージ出力 172  
 エラーレポートファイル 174, 249  
 エラーレポートファイルの出力先 250

## か

---

ガーベージコレクション制御機能 677  
 回復中 40  
 拡張 verbosegc 情報の取得 306  
 拡張 verbosegc 情報の取得を指定するオプション 307  
 拡張スレッドダンプ 242, 650  
 拡張データベースアクセストレース 274  
 拡張データベースアクセストレースの出力項目 275  
 拡張データベースアクセストレースファイル 347  
 拡張データベーストレースのファイル名称の規則 278  
 カスタムジョブ 678  
 カスタムログインモジュール 678  
 仮想サーバ 678  
 稼働情報監視で確認できる項目 62  
 稼働情報監視で表示できる項目 (J2EE サーバの場合) 63  
 稼働情報監視で表示できる項目 (SFO サーバの場合) 72  
 稼働情報監視で表示できる項目 (バッチサーバの場合) 70  
 稼働情報収集機能 678  
 稼働情報として監視できる項目 62  
 稼働情報の監視 (運用管理コマンド) 62  
 稼働情報の監視 (稼働情報ファイル) 41  
 稼働情報ファイル 41  
 稼働情報ファイルで収集できる情報 41  
 稼働情報ファイルの出力 43  
 稼働情報ファイルの出力間隔とファイル面数 43  
 稼働情報ファイルの出力先 43  
 稼働情報ファイルの調査 44  
 稼働情報ファイルのファイル名 44  
 稼働中 40

稼働中のトランザクションの確認 88  
 稼働統計出力ファイル 82  
 稼働統計出力ファイルに出力される情報 83  
 稼働統計出力ファイルの出力先 82  
 稼働統計出力ファイル名 82, 83  
 稼働統計情報の出力情報 83  
 カレントスレッド情報 310  
 簡易 Web サーバ 678  
 環境変数 314  
 監査証跡 678  
 監査ログ 678  
 監査ログで出力するログ 266  
 監視ツリー 678  
 管理対象オブジェクト (AdminObject) 678

## き

---

起動順序の依存関係 331, 641  
 起動中 39  
 機能ごとに取得できる稼働情報の種類 42  
 機能レイヤ 92, 636, 679  
 強制停止 130  
 強制停止中 40  
 強制閉塞 136  
 業務指向ツリー 679  
 業務処理プログラム 679  
 共有スレッド数 679

## く

---

クライアントアプリケーション情報 679  
 クラスタ 679  
 クラスタ構成のデータベースのメンテナンス 116  
 クラスタコネクションプールの運用 163  
 クラスタソフトウェア 679  
 クラスタソフトウェアを使用したシステムの起動と停止 (UNIX の場合) 433  
 クラスタソフトウェアを使用したシステムの起動と停止 (Windows の場合) 417  
 クラスまたは配列型の変数の実際の型名を出力する場合の出力例 327  
 クラスまたは配列型の変数を文字列として出力する場合の出力例 325



グローバル CORBA ネーミングサービス  
679  
 グローバルセッション 679  
 グローバルセッション ID 107, 680  
 グローバルセッション情報 680  
 グローバルセッションの有効期間の確認 107  
 グローバルトランザクション 680

## け

---

系 680  
 計画系切り替え 680  
 計画停止中 40  
 系切り替え 680  
 系障害 680  
 ゲートウェイ指定機能 680  
 現用系 418, 435, 446, 680

## こ

---

構成ソフトウェアのプロセス（論理サーバ）  
が異常終了した場合 330  
 構築済み実行環境の切り替えで出力するログ  
266  
 構築済み実行環境の切り替えで出力するログ  
の出力形式と出力項目（UNIX の場合）271  
 構築済み実行環境の切り替えで出力するログ  
の取得 218  
 コネクション ID 603  
 コネクションスイーパー 680  
 コネクションプーリング 680  
 コネクションプール枯渇監視情報 79  
 コネクションプールの一時停止 163  
 コネクションプールの再開 164  
 コネクションプールの状態によるコマンド実  
行の可否 167  
 コネクションプールの状態の確認 163  
 コマンドおよび VM パラメタ 314  
 コマンドによってメソッドキャンセルを実行  
した場合に出力されるログ情報 626  
 コミットオプション 681  
 コンソールログ 214  
 コンテキスト 681  
 コンテキストルート 681

コンテナ拡張ライブラリ 681

## さ

---

サーバ管理コマンド 19, 681  
 サーバ管理コマンドのログ 200, 225, 262  
 サーバ管理コマンドのログの出力先（互換  
モード）202, 227  
 サーバ管理コマンドを利用する場合の留意事  
項 31  
 サーバ起動・停止フック機能 681  
 サーバ指向ツリー 681  
 サーバの稼働情報の監視方法（J2EE サー  
バ, SFO サーバ, バッチサーバ）62  
 サーバプロセスを操作する機能 19  
 サービス部分閉塞 134  
 サービス閉塞 130, 681  
 サービスユニット 681  
 サブレット 682  
 サブレットエンジンモード 682  
 サブレットフィルタ 682  
 最大同時実行スレッド数を動的に変更したと  
きの Web アプリケーションの動作 121  
 作業ディレクトリ 174, 258  
 サンプリング時間 72

## し

---

時間情報 315  
 時間帯に応じた計画的な Web アプリケー  
ション単位の最大同時実行スレッド数の変更  
117  
 シグナル情報 311  
 シグナル情報の格納先アドレス 311  
 システム運用の概要 1  
 システム運用の作業 21  
 システム運用の特長 5  
 システムごとに実行できるサービスの閉塞方  
法および J2EE アプリケーションの停止方法  
131  
 システムダウン 172  
 システムで提供されている障害検知時コマン  
ドで出力できる資料 181

- システムの運用サイクル (J2EE サーバモードの場合) 4
- システムの運用サイクルを構成する作業 3
- システムの運用状況を監視する 21
- システムの運用で使用する機能 19
- システムの処理性能の解析 91
- システムの処理性能を監視する 23
- システム保守の作業 25
- システム名, CPU, 実メモリ, および VM 情報 315
- システムモニタの設定 255
- システムを起動/停止する 21
- 実行系 418, 430, 435, 446, 457, 682
- 実行待ちキュー 682
- 実パラメタ 682
- 自動系切り替え 682
- 自動再起動中 40
- 自動停止中 40
- 自動で資料を収集する場合の流れ 179
- シナリオ 682
- 出力される情報の種類 73
- 取得が必要な資料の一覧 174
- 取得が必要な資料の取得方法と調査方法の参照先 176
- 取得が必要な資料の種類 172
- 取得できるイベント ID と機能レイヤ 462
- 取得できるトレース情報 [BMT] 523
- 取得できるトレース情報 [CMT] 522
- 取得できるトレース情報 [CTM] 471
- 取得できるトレース情報 [DB Connector] 533, 536
- 取得できるトレース情報 [DB Connector for Cosminexus RM] 542
- 取得できるトレース情報 [J2EE サーバ] 567
- 取得できるトレース情報 [JNDI] 519
- 取得できるトレース情報 [Message-driven Bean (EJB2.0)] 508
- 取得できるトレース情報 [Message-driven Bean (EJB2.1)] 510
- 取得できるトレース情報 [OTS] 554
- 取得できるトレース情報 [RMI] 550
- 取得できるトレース情報 [Session Bean・Entity Bean] 506
- 取得できるトレース情報 [Web コンテナ] 478, 487
- 取得できるトレース情報 [インプロセス HTTP サーバ] 482
- 取得できるトレース情報 [コネクションアソシエーション] 538
- 取得できるトレース情報 [コネクション自動クローズ時] 539
- 取得できるトレース情報 [セッションフェイルオーバー機能] 501
- 取得できるトレース情報 [トランザクションタイムアウト] 525
- 取得できるトレース情報 [バッチアプリケーション実行機能] 566
- 取得できるトレース情報 [フィルタのトレース (正常に処理が終了した場合)] 492
- 取得できるトレース情報 [フィルタのトレース (例外が発生した場合)] 496
- 取得できるトレース情報 [リダイレクタ] 475
- 取得できるトレース情報 [ワーク管理] 548
- 障害検知時コマンド 180
- 障害検知時コマンドで出力できる資料 180
- 障害発生時の CMR 用の表の回復 628
- ジョブ 682
- ジョブネット 682
- 処理性能の解析作業の概要 92
- 資料の個別取得 188
- 資料の取得方法 178
- 資料の調査 260
- シングルサインオン 683
- シングルサインオン情報リポジトリ 683
- シングルサインオンライブラリ 683

## す

---

- スケールアウト 683
- スケールイン 683
- スケジュールキュー 683
- スケジュールポリシー 683
- スタックトレース 312
- スタックトレース情報 259

スタックトレースにローカル変数情報を出力  
 するためのオプション 321  
 スタックトレースの調査 321  
 スタックの先頭から格納されている情報 312  
 ステータス監視で確認できる項目 39  
 ステータスの監視方法 38  
 ステートメントキャンセル 131  
 スマートエージェント 683  
 スレッド枯渇監視情報 78  
 スレッド作成に失敗した場合 320  
 スレッド情報 313  
 スレッドダンプ枯渇監視情報 78  
 スレッドダンプ情報の構成 303  
 スレッドの状態とメソッドキャンセルの実行  
 可否 128  
 スローダウン 172

## せ

静的コンテンツ 683  
 性能解析トレース 174, 683  
 性能解析トレースの取得 240  
 性能解析トレースの調査 303  
 性能解析トレースファイル 23, 93, 637, 683  
 性能解析トレースファイルとスレッドダンプ  
 を対応づけた問題個所の調査 111  
 性能解析トレースファイルに出力する情報  
 96  
 性能解析トレースファイルの収集 93, 95  
 性能解析トレースファイルの収集方法 95  
 性能解析トレースファイルの出力先 96  
 性能解析トレースファイルの出力先と出力情  
 報 96  
 性能解析トレースファイルの出力情報 96  
 性能解析トレースファイルのファイル 96  
 性能解析トレースファイルをフィルタリング  
 した例 101  
 性能解析トレースファイルを利用したシステ  
 ムの処理性能の解析 94  
 セキュリティアイデンティティ 683  
 セッション 684  
 セッション数枯渇監視情報 79  
 セッショントレース 103

セッショントレース情報が出力されている性  
 能解析トレースファイルの例（セッションを  
 生成するリクエスト部分）104  
 セッショントレース情報が出力されている性  
 能解析トレースファイルの例（セッションを  
 破棄するリクエスト部分）106  
 セッショントレース情報が出力されている性  
 能解析トレースファイルの例（セッションを  
 利用するリクエスト部分）105  
 セッションの有効期間の確認 106  
 セッションのライフサイクルの調査 103  
 セッションフェイルオーバー機能 684  
 セッションフェイルオーバー機能でトラブルが  
 発生した場合 333  
 セッションフェイルオーバーサーバ 684  
 セットアップウィザード 684  
 占有スレッド数 684

## そ

相互系切り替えシステム 684  
 相互系切り替えシステムで計画的に系を切り  
 替える場合の起動と停止 429, 451  
 相互系切り替えシステムの起動 427, 446  
 相互系切り替えシステムの起動と停止  
 427, 444  
 相互系切り替えシステムの起動の流れ 447  
 相互系切り替えシステムのシステム構成例  
 445  
 相互系切り替えシステムの停止 428, 448  
 相互系切り替えシステムの停止の流れ 449  
 相互スタンバイ 684

## た

待機系 418, 430, 435, 446, 457, 684  
 タイムアウトが発生したトランザクションの  
 特定 107  
 タイムアウトが発生したリクエストのキャン  
 セル 129  
 タイムアウトが発生したリクエストの特定  
 108  
 タイムアウトしたリクエストを確認するた  
 めに使用できる性能解析トレース 109

タイムアウトしたリクエストをキャンセルする 25  
 タイムアウトの延長でメソッドキャンセルを実行した場合に出力されるログ情報 623

## ち

---

抽象パラメタ 684

## つ

---

通常停止 130  
 通常のサービス閉塞 134  
 通常の閉塞 135  
 通常モード 155, 685  
 通信障害 40

## て

---

定義情報 174  
 停止 39  
 停止中 40  
 停止中のトランザクションの確認 88  
 データソース 685  
 データベースと接続中にトラブルが発生したコネクションの特定 603  
 データベースと接続中にトラブルが発生した場合 335  
 テストモード 154, 685  
 テストモードで実行できるサーバ管理コマンド 159  
 テストモードでの J2EE アプリケーションの運用 154  
 デフォルトのタイムアウト時間で通常停止を実行して、停止しなかった場合に強制停止を実行する 140  
 デフォルトのタイムアウト時間で通常停止を実行する 138  
 デプロイ 685  
 展開ディレクトリ形式 685

## と

---

統合コンソール 685  
 統合スコープ 685

統合ネーミングスケジューラサーバ 685  
 統合ネーミングスケジューラサーバモデル 685  
 統合ユーザ管理 685  
 統合ユーザ管理 GUI 685  
 統合ユーザ管理のコンフィグレーションファイル 675  
 統合ユーザ管理のログの取得 214  
 統合ユーザ管理フレームワーク 686  
 統合ログ 208  
 統合ログの出力先 208  
 動作モード 686  
 動的コンテンツ 686  
 登録済みシグナルハンドラ 314  
 トラブルが発生したコネクションの特定 110  
 トラブルシューティング 169  
 トラブルシューティングに関連する留意事項 344  
 トラブル種別ごとの取得資料一覧 173  
 トラブルに対処する 26  
 トラブル発生時に自動的に取得できる資料 178  
 トラブル発生時に資料を収集するための準備 30  
 トラブル発生時の対処の流れ 170  
 トラブルへの対処と回復 330  
 トランザクションサービス 686  
 トランザクションサービスの稼働情報ファイルに出力される情報 57  
 トランザクション情報の確認 88  
 トランザクションタイムアウト時に出力される性能解析トレース 108  
 トランザクションの状態を確認する 23  
 トランザクションマネージャ 686  
 トレース取得ポイント 92, 462, 637, 686  
 トレース取得ポイント〔BMT〕 523  
 トレース取得ポイント〔CMT〕 521  
 トレース取得ポイント〔CTM〕 466  
 トレース取得ポイント〔DB Connector〕 526, 535  
 トレース取得ポイント〔DB Connector for Cosminexus RM〕 540  
 トレース取得ポイント〔J2EE サーバ〕 566

トレース取得ポイント〔JNDI〕 518  
 トレース取得ポイント〔Message-driven Bean (EJB2.0)〕 507  
 トレース取得ポイント〔Message-driven Bean (EJB2.1)〕 509  
 トレース取得ポイント〔OTS〕 551  
 トレース取得ポイント〔RMI〕 550  
 トレース取得ポイント〔Session Bean・Entity Bean〕 504  
 トレース取得ポイント〔Web コンテナ〕 476, 483  
 トレース取得ポイント〔インプロセス HTTP サーバ〕 479  
 トレース取得ポイント〔コネクションアソシエーション〕 537  
 トレース取得ポイント〔コネクション自動クローズ時〕 538  
 トレース取得ポイント〔セッションフェイルオーバー機能〕 498  
 トレース取得ポイント〔トランザクションタイムアウト〕 524  
 トレース取得ポイント〔フィルタのトレース (正常に処理が終了した場合)〕 490  
 トレース取得ポイント〔フィルタのトレース (例外が発生した場合)〕 494  
 トレース取得ポイント〔リダイレクタ〕 472  
 トレース取得ポイント〔ワーク管理〕 544

## に

---

二次送付資料 172, 183  
 任意のタイムアウト時間を設定して通常停止を実行して、停止しなかった場合に強制停止を実行する 141  
 任意のタイムアウト時間を設定して通常停止を実行して、停止しなかった場合に自動的に強制停止を実行する 143  
 任意のタイムアウト時間を設定して通常停止を実行する 139

## ね

---

ネーミング切り替え 686  
 ネーミングサービス 686

## は

---

バインド先アドレス設定機能 686  
 バックシステム (CTM を使用している場合) で実行できるサービス閉塞 133  
 バックシステムでのサービス閉塞 (CTM を使用しているシステムの場合) 132  
 バッチアプリケーション 687  
 バッチアプリケーションのユーザログの出力先 232  
 バッチサーバ 687  
 バッチサーバのログ 222  
 パフォーマンストレーサ 23, 91, 687  
 ハングアップ (無応答) 172  
 ハンドラ 687

## ひ

---

ビジネスインタフェース 687  
 日立 JavaVM 固有のログの調査 (日立 JavaVM ログファイル) 306  
 日立 JavaVM ログファイル 174, 249  
 日立 JavaVM ログファイルを出力するオプション 306  
 日立トレース共通ライブラリ 687  
 日立トレース共通ライブラリ形式 266  
 日立トレース共通ライブラリ形式 (シングルプロセス) 261  
 日立トレース共通ライブラリ形式 (マルチプロセス) 261  
 日立トレース共通ライブラリ形式のログの出力形式と出力項目 266  
 日立トレース共通ライブラリ形式のログの出力項目 266  
 日立トレース共通ライブラリ形式のログを参照する場合の注意 268  
 非保護区 687  
 標準出力 / 標準エラー出力 / ユーザログのトレース取得ポイント 561  
 標準の形式および簡易出力フォーマットでの出力例 322

## ふ

---

ファイルディスクリプタ枯渇監視情報 77

フィルタリングに使用するイベント ID が示すトレース取得ポイント 101  
負荷分散機 687  
負荷分散機を利用したサービス部分閉塞 134  
負荷分散機を利用したサービス閉塞 133  
物理コネクション 687  
物理ティア 687  
プロセスの再起動方法 642  
プロセスリソースの稼働情報ファイルに出力される情報 50  
フロント 131  
フロント EJB 133, 688

## へ

---

ベーシックモード 688  
ヘッダファイル 44  
ヘッダファイルと稼働情報ファイルの対応 45  
別名 688

## ほ

---

保護区 128, 688  
保守員 172  
保守員に連絡が必要なトラブル 172  
保守用ログ 174, 190, 220  
保守レベル 93  
ホスト単位管理モデル 688

## ま

---

マシン情報 315

## み

---

未決着トランザクション 688

## め

---

命令コード情報 312  
メソッドキャンセル機能 688  
メソッドタイムアウト機能 688  
メッセージインフロー 688  
メッセージログ 174, 190, 220, 274  
メモリ枯渇監視情報 75

メモリ情報 313  
メモリダンプ 174  
メモリダンプの取得 246  
メモリ不足を示すメッセージの出力内容 318  
面数 689  
メンバコネクションプールの状態 166  
メンバコネクションプールの状態遷移（自動で一時停止が実行される場合）166  
メンバコネクションプールの状態遷移（手動で一時停止を実行する場合）165  
メンバリソースアダプタ 689

## ゆ

---

ユーザが作成した障害検知時コマンドで出力できる情報 182  
ユーザ指定名前空間機能 689  
ユーザ情報 689  
ユーザ情報リポジトリ 689  
ユーザスレッド 689  
ユーザダンプの取得 246  
ユーザ認証ライブラリ 689  
ユーザプロパティファイル 675, 689  
ユーザマッピング 689  
ユーザログ 174, 190, 220, 689

## よ

---

予備系 418, 435, 446, 689

## ら

---

ライトトランザクション 690  
ライブラリ 314  
ライブラリ JAR 690  
ラウンドロビン検索 690  
ラウンドロビンポリシー 690

## り

---

リカバリ専用サーバ 690  
リクエスト処理スレッド 690  
リクエストのスケジューリング 690  
リソースアダプタ 690

リソースアダプタバージョンアップコマンド  
(cjrupdate) のログ 202, 228, 263  
 リソース枯渇監視情報 73  
 リソース枯渇監視情報に出力される情報 73  
 リソース枯渇監視情報の出力形式 75  
 リソース枯渇監視のログ 206, 229, 263  
 リソース枯渇監視ログ 191, 221  
 リソース枯渇監視ログファイル 73  
 リソース枯渇の監視 73  
 リソースの使用率または使用数を基にリソースの枯渇を監視する 22  
 リソースの設定 175  
 リソースマネージャ 690  
 リソースをメンテナンスする 26  
 リダイレクタ 690  
 リダイレクタのトレース取得ポイント 472  
 リデプロイ 146  
 リデプロイ機能 691  
 リデプロイによる J2EE アプリケーションの入れ替え 146  
 リバースプロキシサーバ 691  
 リモートインタフェース 691  
 リロード 148  
 リロード機能 691  
 リロードによって J2EE アプリケーションを入れ替えるときの JSP の事前コンパイル 150  
 リロードによる J2EE アプリケーションの入れ替え 148

---

## る

ルートアプリケーション情報 691  
 ルートアプリケーション情報を利用したログ調査 109  
 ルートコンテキスト 691  
 ルートリソースアダプタ 691

---

## れ

例外ログ 174, 190, 220  
 レジスタ情報 312  
 レルム 691

---

## ろ

ローカルインタフェース 691  
 ローカルトランザクション 691  
 ローカル変数 321  
 ローカル変数情報 321  
 ロードバランサ 692  
 ロール 692  
 ロガー 692  
 ログ以外に取得が必要な情報 214, 232  
 ログインモジュール 692  
 ログの出力先 191, 217, 218, 221, 647  
 ログの出力先を設定するユーザ定義ファイル 207, 231  
 ログの出力先を設定するユーザ定義ファイル (サブレットエンジンモードの場合) 646  
 論理コネクション 692  
 論理サーバ 5, 692  
 論理サーバのステータスの監視 38  
 論理ネーミングサービス 692  
 論理ユーザサーバ 692

---

## わ

ワーカプロセス 692  
 ワトソンログが正しく出力されたかどうかを示すメッセージの出力内容 319





# ソフトウェアマニュアルのサービス ご案内

## 1. マニュアル情報ホームページ

ソフトウェアマニュアルの情報をインターネットで公開しています。

URL <http://www.hitachi.co.jp/soft/manual/>

ホームページのメニューは次のとおりです。

マニュアル一覧	日立コンピュータ製品マニュアルを製品カテゴリ、マニュアル名称、資料番号のいずれかから検索できます。
CD-ROMマニュアル	日立ソフトウェアマニュアルと製品群別CD-ROMマニュアルの仕様について記載しています。
マニュアルのご購入	マニュアルご購入時のお申し込み方法を記載しています。
オンラインマニュアル	一部製品のマニュアルをインターネットで公開しています。
サポートサービス	ソフトウェアサポートサービスお客様向けページでのマニュアル公開サービスを記載しています。
ご意見・お問い合わせ	マニュアルに関するご意見、ご要望をお寄せください。

## 2. インターネットでのマニュアル公開

2種類のマニュアル公開サービスを実施しています。

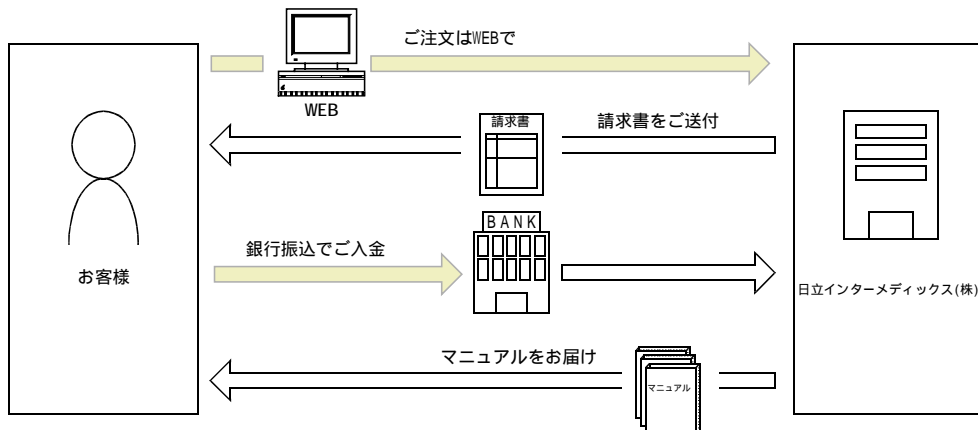
### (1) マニュアル情報ホームページ「オンラインマニュアル」での公開

製品をよりご理解いただくためのご参考として、一部製品のマニュアルを公開しています。

### (2) ソフトウェアサポートサービスお客様向けページでのマニュアル公開

ソフトウェアサポートサービスご契約のお客様向けにマニュアルを公開しています。公開しているマニュアルの一覧、本サービスの対象となる契約の種別などはマニュアル情報ホームページの「サポートサービス」をご参照ください。

## 3. マニュアルのご注文



マニュアル情報ホームページの「マニュアルのご購入」にアクセスし、お申し込み方法をご確認のうえWEBからご注文ください。ご注文先は日立インターメディアックス(株)となります。

ご注文いただいたマニュアルについて請求書をお送りします。

請求書の金額を指定銀行へ振り込んでください。

入金確認後7日以内にお届けします。在庫切れの場合は、納期を別途ご案内いたします。