

Hitachi Application Server

トラブルシューティング手順書 [Windows]

システム運用時のトラブルシューティング手順

2016. **10**
October

はじめに

本書は、マニュアル「Hitachi Application Server V10 ユーザーズガイド (Windows(R)用)」の「1.1 マニュアルの読み方について」で定義している「Web フロントシステム」の構築・運用フェーズで使用するドキュメントとして次の基準に基づいて記述しています。

1. 対象とする読者

Web フロントシステムを構築・運用する立場にあるシステム構築者を対象としています。本書によって、「Web フロントシステム」を運用できます。

2. 対象とする製品

Hitachi Application Server 10-11

■商標類

- HITACHI、HiRDB は、株式会社 日立製作所の商標または登録商標です。
- Microsoft、SQL Server、Windows、および Windows Server は、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。
- Oracle と Java は、Oracle Corporation 及びその子会社、関連会社の米国及びその他の国における登録商標です。
- RSA および BSAFE は、米国 EMC コーポレーション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。
- This product includes software developed by Andy Clark.
- This product includes software developed by Daisuke Okajima and Kohsuke Kawaguchi (<http://relaxngcc.sf.net/>).
- This product includes software developed by IAIK of Graz University of Technology.
- 本製品は、米国 EMC コーポレーションの RSA BSAFE(R)ソフトウェアを搭載しています。
- その他記載の会社名、製品名などは、それぞれの会社の商標もしくは登録商標です。

■製品名の表記

本書では、製品名を次のように表記しています。

表記		製品名
Application Server		Hitachi Application Server
Oracle	Oracle 11g	Oracle® Database 11g
	Oracle 11g R2	Oracle® Database 11g Release 2
	Oracle 12c	Oracle® Database 12c
Windows	Windows Server 2008 R2	Microsoft® Windows Server® 2008 R2 Standard 日本語版
		Microsoft® Windows Server® 2008 R2 Enterprise 日本語版
		Microsoft® Windows Server® 2008 R2 Datacenter 日本語版
	Windows Server 2012	Microsoft® Windows Server® 2012 Standard 日本語版
		Microsoft® Windows Server® 2012 Datacenter 日本語版

表記		製品名
	Windows Server	Microsoft® Windows Server® 2012 R2 Standard 日本語版
	2012 R2	Microsoft® Windows Server® 2012 R2 Datacenter 日本語版

■ 英略語の表記

本書では、英略語を次のように表記しています。

表記	英字での表記
EAR	Enterprise Archive
EJB	Enterprise JavaBeans™
Java	Java™
Java EE	Java™ Platform, Enterprise Edition
JDBC	Java™ Database Connectivity
	JDBC™
Servlet	Java™ Servlet

■ 発行元

株式会社日立製作所 ICT 事業統括本部 サービスプラットフォーム事業本部

All Rights Reserved. Copyright (C) 2014, 2016, Hitachi, Ltd.

トラブルシュート手順書

システム運用時のトラブルシュート手順

目次

1 概要	1
1.1 システム構成と運用の流れ.....	2
1.2 本書の表記および前提について.....	4
2 障害運用	5
2.1 障害運用の流れ.....	6
2.2 障害回復方法.....	7
3 トラブルシュート	10
3.1 想定されるトラブルの種類.....	11
3.2 システム構築時のトラブル.....	12
3.2.1 サーバ構築時に発生するトラブル.....	12
3.2.2 システム起動時に発生するトラブル.....	15
3.2.3 アプリケーションデプロイ時発生するトラブル.....	20
3.2.4 データベース接続時に発生するトラブル.....	22
3.3 システム運用時のトラブル.....	25

1 概要

この章ではシステム構成と運用の流れについて説明します。

本章の構成

- 1.1 システム構成と運用の流れ
- 1.2 本書の表記および前提について

1.1 システム構成と運用の流れ

システム構成と運用の流れを次に示します。

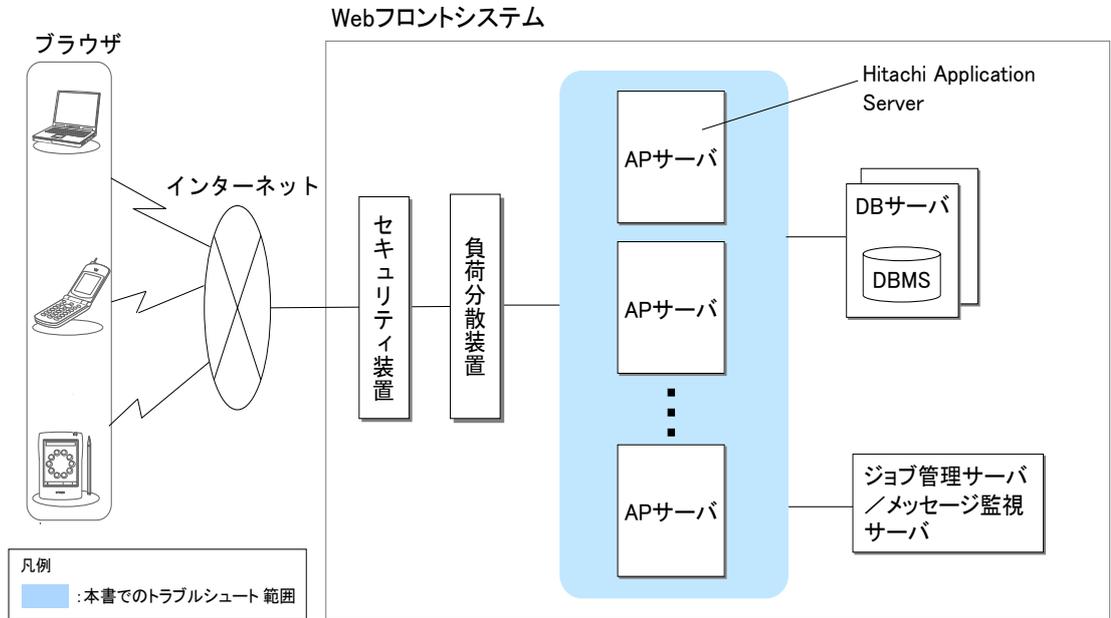


図 1.1-1 システム構成

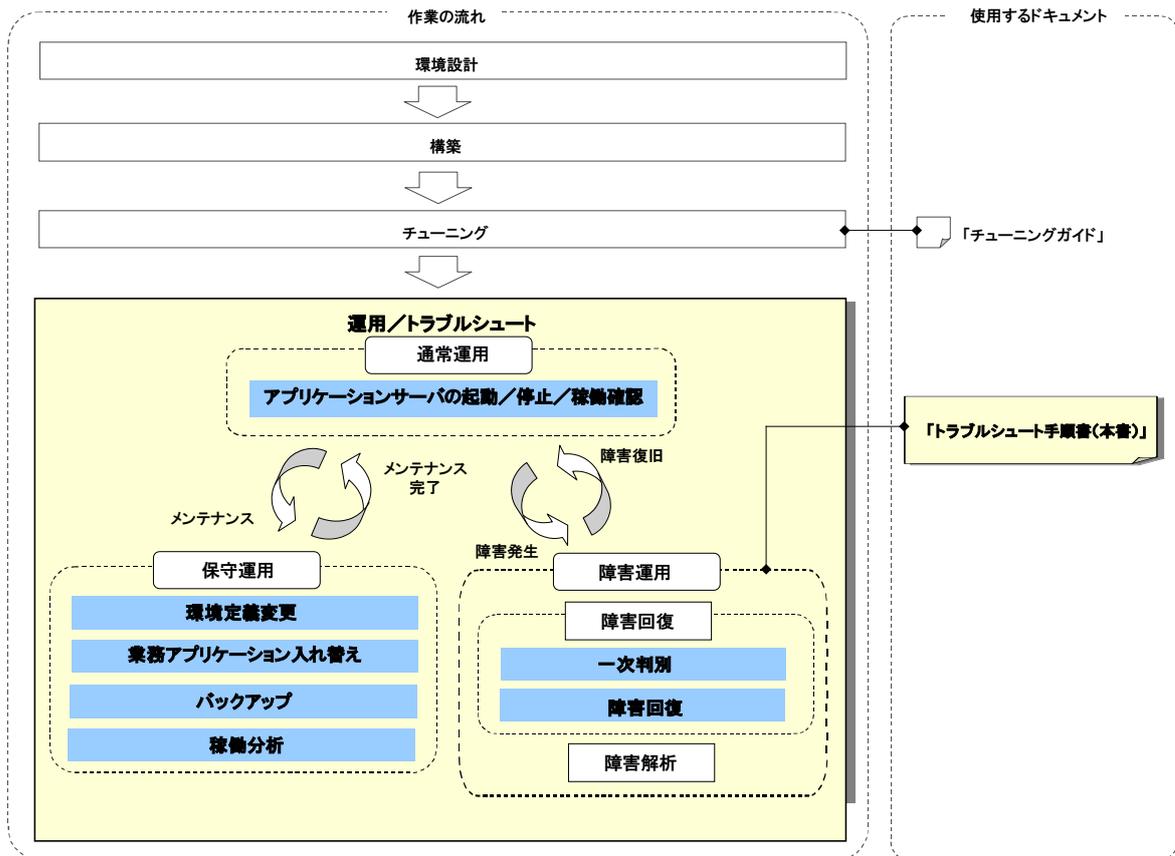


図 1.1-2 運用の流れ

1.2 本書の表記および前提について

本書は、次の前提に基づいて記載しています。

- Application Server インストール先ディレクトリーは C:\¥Hitachi¥APServer です。
異なるディレクトリーを使用している場合は、C:\¥Hitachi¥APServer を適宜読み替えてください。
- ドメイン数は 1、ドメインの名称は domain1、ノード数は 1、ノードの名称は localhost-domain1、Web サーバの名称は Web1、Java EE サーバの名称は JavaEE1、パフォーマンストレーサーの名称は PRF1 とします。
ノード数が複数の場合は、localhost-domain1、Web1、JavaEE1、PRF1 を読み替え、それぞれの構成要素に対してコマンドを実行してください。

本書では、次の表記を使用しています。

- <snapshot ログ（初期調査用システム情報）>
対応する snapshot ログ「snapshot-initinfo-<収集対象名>-<日時>-<プロセス ID>-<スレッド ID>.zip」を展開したディレクトリーを示します。
- <snapshot ログ（すべてのシステム情報）>
対応する snapshot ログ「snapshot-sysinfo-<収集対象名>-<日時>-<プロセス ID>-<スレッド ID>.zip」を展開したディレクトリーを示します。

2 障害運用

この章では障害運用の方法について説明します。

本章の構成

- 2.1 障害運用の流れ
- 2.2 障害回復方法

2.1 障害運用の流れ

障害運用の流れを次に示します。本書では障害回復の手順を示します。

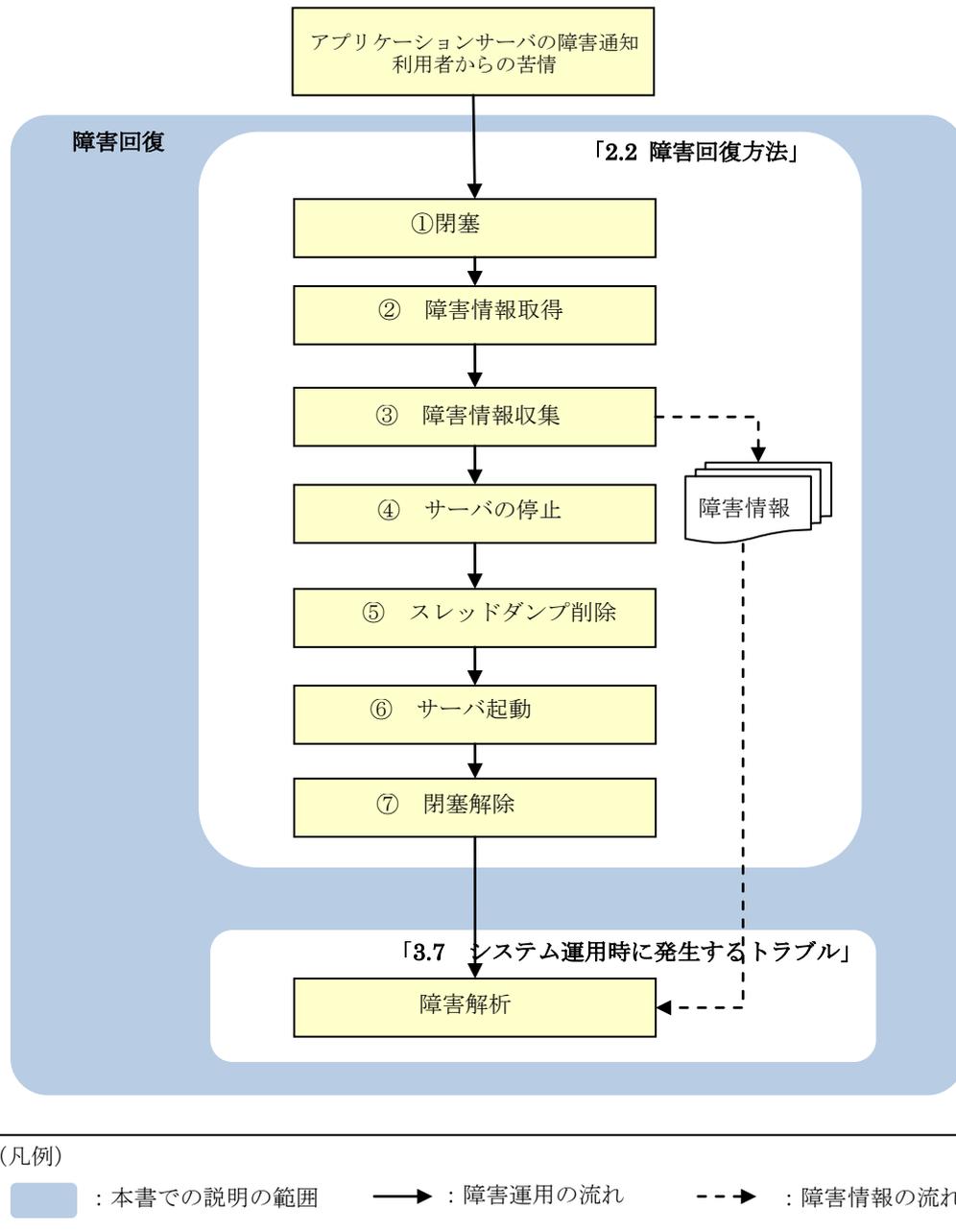


図 2.1-1 障害運用の流れ

2.2 障害回復方法

障害が発生した場合は、必要に応じて障害の発生したアプリケーションサーバに対して手動で障害を回復します。

<手順>

① 閉塞する

ヘルスチェックアプリケーションを停止し、負荷分散装置から障害の発生したアプリケーションサーバへの振り分けを停止します。次のコマンドを実行します。

コマンド	<code>C:\Hitachi\APServer\javaee\glassfish\bin\asadmin disable --target JavaEE1 healthcheck*</code>
実行結果	Command disable executed successfully.

注※ healthcheck はヘルスチェックアプリケーションの EAR 名 (application.xml の display-name 指定値) です。

② 障害情報を取得する

Java EE サーバが起動している場合は、次のコマンドを 3 秒ごとに 10 回実行し障害情報を取得します。

コマンド	<code>C:\Hitachi\APServer\jdk\jre\bin\jheapprof -f -garbage -p <Java EE サーバのプロセス ID></code>
	<code>C:\Hitachi\APServer\jdk\jre\bin\jheapprof -f -p <Java EE サーバのプロセス ID></code>

次に示す出力先に Java EE サーバのスレッドダンプが出力されます。

スレッドダンプ出力先	<code>C:\Hitachi\APServer\javaee\glassfish\nodes ¥localhost-domain1¥JavaEE1¥config¥javacoren.txt</code>
------------	---

③ 障害情報を収集する

次のコマンドを実行し障害情報を収集します。ドメイン管理サーバが起動している必要があります。

なお、KDKD10112-E を出力してプロセス障害(プロセスダウンまたはハングアップ)を検知した場合は、本手順は自動的に行われているため不要となります。

コマンド	C:\¥Hitachi¥APServer¥javaee¥glassfish¥bin¥asadmin collect-snapshot --targettype domain
実行結果	KDKD10200-I Execution of the collect-snapshot subcommand started. KDKD10217-I The collect-local-snapshot subcommand was invoked. (host name = localhost) KDKD10219-I Execution of the invoked collect-local-snapshot subcommand ended successfully. (host name = localhost) KDKD10201-I Execution of the collect-snapshot subcommand ended successfully. Command collect-snapshot executed successfully.

次に示すファイルに定義ファイル、ログファイル、およびメモリ情報が収集されます。

snapshot ログ (初期調査用システム情報)	C:\¥Hitachi¥APServer¥javaee¥snapshot¥snapshot-initinfo-<収集対象名>-<日時>-<プロセス ID>-<スレッド ID>.zip
snapshot ログ (すべてのシステム情報)	C:\¥Hitachi¥APServer¥javaee¥snapshot¥snapshot-sysinfo-<収集対象名>-<日時>-<プロセス ID>-<スレッド ID>.zip

上記は、障害発生箇所が特定できていない場合にドメイン全体の障害情報を収集している例です。障害が発生しているノード、またはサーバの特定が出来ている場合は、対象のノードまたはサーバの障害情報を収集してください。

④ Web サーバ、Java EE サーバ、およびパフォーマンストレーサーを停止する

Web サーバ、Java EE サーバ、およびパフォーマンストレーサーが起動している場合は、Web サーバ、Java EE サーバ、およびパフォーマンストレーサーを停止します。次のコマンドを実行します。

コマンド	asadmin stop-servers
実行結果	KDKD10094-I The server was stopped successfully. (server type = webserv, server name = Web1, stop method = graceful) The instance, JavaEE1, is stopped. KDKD10094-I The server was stopped successfully. (server type = prf, server name = PRF1, stop method = normal) KDKD10107-I The servers were successfully stopped in a batch. Command stop-servers executed successfully.

⑤ スレッドダンプを削除する

スレッドダンプは自動的に削除されないため、手動で削除する必要があります。
すべてのスレッドダンプを削除します。次のファイルを削除します。

削除するスレッドダンプ	C:¥Hitachi¥APServer¥javaee¥glassfish¥nodes ¥localhost-domain1¥JavaEE1¥config¥javacore.n.txt
-------------	--

⑥ Web サーバ、Java EE サーバ、およびパフォーマンストレーサーを起動する

Web サーバ、Java EE サーバ、およびパフォーマンストレーサーを起動します。次のコマンドを実行します。

コマンド	asadmin start-servers
実行結果	KDKD10083-I The server was started successfully. (server type = prf, server name = PRF1) Attempting to start JavaEE1.... Please look at the server log for more details..... The instance, JavaEE1, was started on host localhost KDKD10083-I The server was started successfully. (server type = webserver, server name = Web1) KDKD10103-I The servers were successfully started in a batch. Command start-servers executed successfully.

⑦ 閉塞を解除する

ヘルスチェックアプリケーションを開始することで、負荷分散装置からアプリケーションサーバへの振り分けを開始します。次のコマンドを実行します。

コマンド	C:¥Hitachi¥APServer¥javaee¥glassfish¥bin¥asadmin enable --target JavaEE1 healthcheck*
実行結果	Command enable executed successfully.

注※ healthcheck はヘルスチェックアプリケーションの EAR 名(application.xml の display-name 指定値)です。

3 トラブルシュート

この章ではトラブルシュートの流れについて説明します。

本章の構成

- 3.1 想定されるトラブルの種類
- 3.2 システム構築時のトラブル
- 3.3 システム運用時のトラブル

3.1 想定されるトラブルの種類

想定されるトラブルの種類を次に示します。

フェーズ	現象	影響	主な原因
システム構築	<ul style="list-style-type: none"> サーバ構築中にエラーが発生 システム起動時にエラーが発生 アプリケーションのデプロイ時にエラーが発生 データベース接続時にエラーが発生 	<ul style="list-style-type: none"> システム構築ができず、作業が止まる システムが起動できず、サービスを開始できない アプリケーションがデプロイできず、サービスを開始できない データベースに接続できず、サービスを開始できない 	<ul style="list-style-type: none"> 環境の問題 (OS、ネットワーク、メモリ、ディスクなど) 操作ミス、設定ミス アプリケーションの作りの問題 データベース側の問題
システム運用	サービス運用中にエラー、応答遅延、応答なし、などが発生	<ul style="list-style-type: none"> 業務処理が滞る 一部のユーザ、もしくは一部の業務処理ができない サービスが提供できない 	<ul style="list-style-type: none"> CPU、メモリなどのリソース不足、ネットワークやディスク I/O ネック データベースなどの連携システムの問題 アプリケーションの問題

ログ、トレースなどのトラブル資料についての詳細は、マニュアル ユーザーズガイド(V10.1)「10. トラブルシュート資料の活用」を参照してください。



お問い合わせの前に

日立ソリューションサポートサービスのサイトから、発生している現象が既知の不良かどうかを確認します。既知の不良の場合は、パッチの適用を検討します。

URL : http://www.hitachi-support.com/open_new/index.html

日立ソリューションサポートサービスのサイトの閲覧には、サポートサービス契約が必要です (ユーザー名・パスワードが要求されます)。

3.2 システム構築時のトラブル

3.2.1 サーバ構築時に発生するトラブル

以下に、サーバ構築の流れと主なログの出力ポイントを示します。

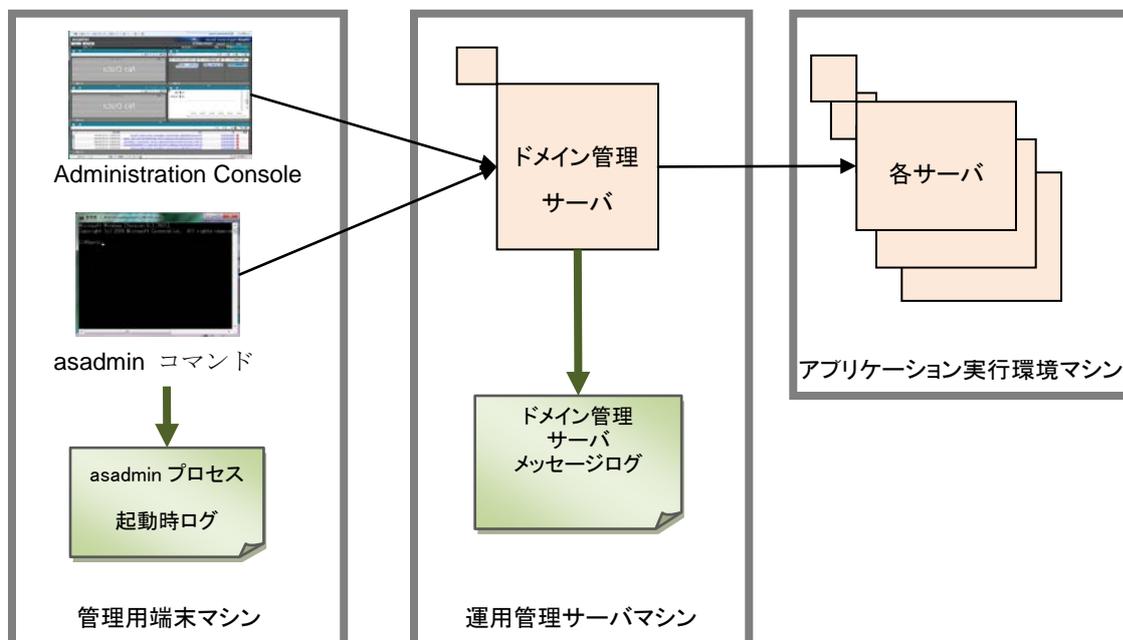


図 3.2-1 サーバ構築の流れと主なログ

サーバ構築時のトラブルは、主に次の箇所を確認できます。

- Administration Console 上に表示されるメッセージ
- asadmin コマンドを実行したコンソールに表示されるメッセージ
- ドメイン管理サーバメッセージログ

ドメイン管理サーバメッセージログは、以下に出力されます。

```
C:\¥Hitachi¥APServer¥javaee¥logs¥domains¥domain1¥das_message.n.log
```

それでも原因が判明しない場合は、ドメイン管理サーバから起動されたサブコマンドのログを追うことで、エラーや警告の原因がわかることがあります。サブコマンドのログは asadmin プロセス起動時ログに出力されます。

asadmin プロセス起動時ログは、以下に出力されます。

```
C:\¥Hitachi¥APServer¥javaee¥logs¥commands¥asadmin¥asadmin_launch.n.log
```

(1) サーバセットアップ時のトラブル例

Web サーバ名「MyWeb」の Web サーバを新規作成しようとしたところ、エラーが出力され構築に失敗した場合を例に挙げます。

■トラブルシュート資料を利用した障害解析の流れ

この例では、`asadmin` コマンドを実行したコンソールに表示されるメッセージに従い、トラブルシュートを行います。

コマンド	<code>asadmin create-webserver --node localhost-domain1 --prf PRF1 MyWeb</code>
実行結果	<pre>remote failure: KDKD10140-E An error occurred during the execution of the subcommand on the node. (node name = localhost-domain1, message = KDKD10138-E An error occurred in the execution of the file.(file name = D:\Hitachi\APServer\javaee\glassfish\nodes\localhost-domain1\MyWeb\root\bin\create-webserver.bat, return code = 721073, details = create-webserver.bat: Command failed. "D:\Hitachi\APServer\httpsd\httpsd.exe" -n "MyWeb" -f "D:\Hitachi\APServer\javaee\glassfish\nodes\localhost-domain1\MyWeb\root\conf\httpsd.conf" -k install, ret=721073 Syntax OK Installing the MyWeb service (OS 1073)The specified service already exists. : Failed to create WinNT Service Profile) KDKD10036-E An error occurred in the processing to build a server. (server type = webserver, server name = MyWeb) Command_call-server-adapter-construct executed successfully.) KDKD10034-E The server build failed. (server type = webserver, server name = MyWeb) Command create-webserver failed.</pre>

(1)**KDKD10140-E An error occurred during the execution of the subcommand on the node.**

ドメイン管理サーバから起動したサブコマンドでエラーが発生したことがわかります。

(2)**the MyWeb service (OS 1073)The specified service already exists.**

同名のサービス「MyWeb」が既に存在することにより、Web サーバの構築に失敗したことがわかります。

(3)**KDKD10034-E The server build failed. (server type = webserver, server name = MyWeb)**

Command create-webserver failed.

上記により、最終的に Web サーバの構築に失敗したことがわかります。

■ 想定される原因と対処

構築しようとした Web サーバ名と同一名称の Windows サービスが存在するため、Web サーバの構築に失敗しています。Web サーバ名を変更して再構築することで、問題が解決できます。

(2) オプション変更時のトラブル例

Java EE サーバ起動時の VM オプションの変更失敗した場合を例に挙げます。オプション変更時には、まず `asadmin list-jvm-options` でサーバインスタンスのオプション一覧から、変更したいオプションを確認します。

コマンド	<code>asadmin list-jvm-options --target JavaEE1</code>
実行結果	<code>-XX:MetaspaceSize=256m</code> <code>-XX:MaxMetaspaceSize=256m</code> <code>-XX:CompressedClassSpaceSize=256m</code> <code>-server</code> <code>-Djava.awt.headless=true</code> <code>-Djdk.corba.allowOutputStreamSubclass=true</code> ～以下省略～

`-XX:MaxMetaspaceSize` の値を変更します。設定済みのオプションを変更する場合は、`delete-jvm-options` サブコマンドで、変更するオプションを削除します。

コマンド	<code>asadmin delete-jvm-options --target JavaEE1</code> <code>-XX:MaxMetaspaceSize=256m</code>
実行結果	JavaEE1: <code>No jvm-options were deleted (perhaps they never existed in the configuration)</code> <code>No jvm-options were deleted (perhaps they never existed in the configuration)</code> Command delete-jvm-options executed successfully.

オプションが存在せず、削除されなかったというメッセージが表示されています。

オプション入力時、記号に対するエスケープ処理が必要になります。

`-XX:MaxMetaspaceSize` を指定する場合は、「:」に対して「\」を付けてエスケープ処理を行い、`-XX\:MetaspaceSize` と入力します。

以下のように、オプションを複数指定する場合は、オプション名中に使われる「:」と複数指定するために「:」が混在するため、特に注意が必要です。

コマン	asadmin create-jvm-options
ド	-XX*:MaxMetaspaceSize=192m:-XX*:MetaspaceSize=192m:-XX*:CompressedClassSpaceSize=192m

3.2.2 システム起動時に発生するトラブル

以下に、システム起動の流れと主なログの出力ポイントを示します。

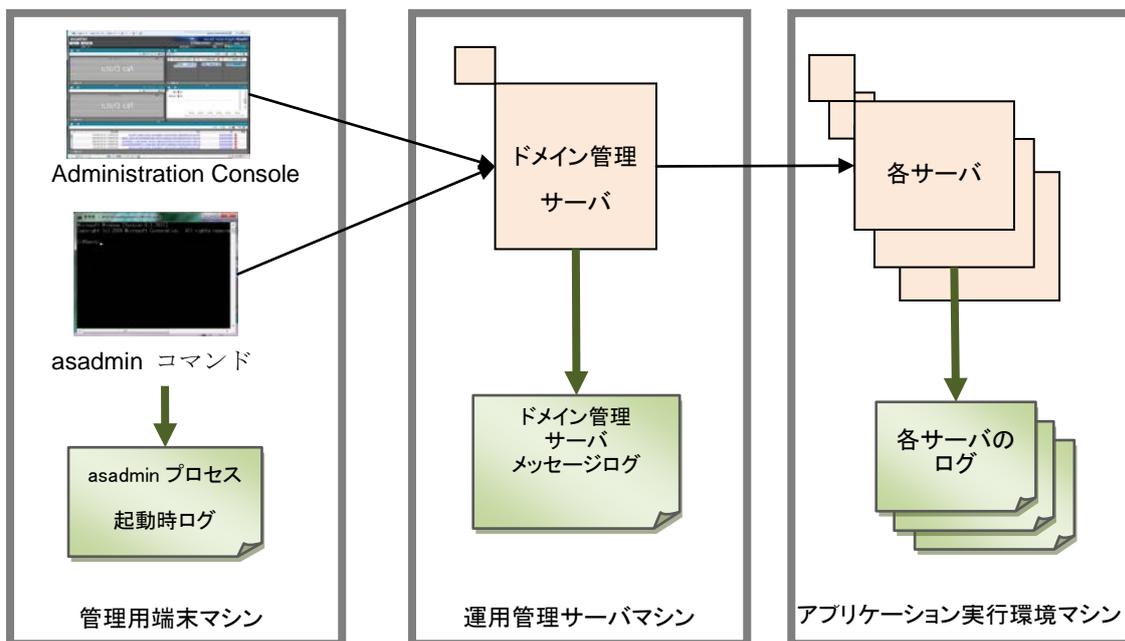


図 3.2-2 システム起動の流れと主なログ

システム起動時のトラブルは、主に次の箇所で原因を確認できます。

- Administration Console 上に表示されるメッセージ
- asadmin コマンドを実行したコンソールに表示されるメッセージ
- ドメイン管理サーバメッセージログ

ドメイン管理サーバメッセージログは、以下に出力されます。

```
C:\¥Hitachi¥APServer¥javaee¥logs¥domains¥domain1¥das_message.n.log
```

それでも原因が判明しない場合は、ドメイン管理サーバから起動されたサブコマンドのログを追うことで、エラーや警告の原因がわかることがあります。サブコマンドのログは asadmin プロセス起動時ログに出力されます。

asadmin プロセス起動時ログは、以下に出力されます。

```
C:\Hitachi\APServer\javaee\logs\commands\asadmin\asadmin_launchn.log
```

各サーバの起動まで処理が正常に実行さえしていれば、各サーバからのログが出力されます。各サーバのログは以下のディレクトリーに格納されています。

```
C:\Hitachi\APServer\javaee\logs\nodes\localhost-domain1\<サーバ名>
```

(1) Web サーバ起動時のトラブル例

構築済みのサーバ(PRF1、JavaEE1、Web1)を一括起動しようとした際に、Web サーバの起動に失敗した例を挙げます。

■ トラブルシュート資料を利用した障害解析の流れ

この例では、asadmin コマンドを実行したコンソールおよびドメイン管理サーバのメッセージログを参照してトラブルシュートを行います。

コマンド	asadmin start-servers
実行結果	<pre>KDKD10083-I The server was started successfully. (server type = prf, server name = PRF1) CLI801 Instance is already synchronized Attempting to start JavaEE1.... Please look at the server log for more details..... The instance, JavaEE1, was started on host localhost KDKD10140-E An error occurred during the execution of the subcommand on the node. (node name = localhost-domain1, message = KDKD10138-E An error occurred in the execution of the file. (file name = D:\Hitachi\APServer\javaee\glassfish\nodes\localhost-domain1\Web1\root\bin\start-webserver.bat, return code = 1, details = start-webserver.bat: Command failed. "D:\Hitachi\APServer\httpsd\httpsd.exe" -n "Web1" -k start, ret=1 (OS 10048)Only one usage of each socket address (protocol/network address/port) is normally permitted. : AH00072: make_sock: could not bind to address 0.0.0.0:80 AH00451: no listening sockets available, shutting down AH00015: Unable to open logs) KDKD10089-E An error occurred in the processing to start the server. (server type = webserver, server name = Web1) Command _call-server-adapter-start executed successfully.) KDKD10084-E Processing to start the server failed. (server type = webserver, server name = Web1) KDKD10104-E Processing to start the servers in a batch failed. Command start-servers failed.</pre>

(1) **KDKD10084-E Processing to start the server failed. (server type = webserver, server name = Web1)**

Web サーバ Web1 の起動に失敗しています。

(2) **KDKD10104-E Processing to start the servers in a batch failed.**

Web サーバの起動に失敗したため、サーバの一括起動が失敗となっています。

(3) ドメイン管理サーバのメッセージログを参照して、詳細を確認します。

0055	2015/06/18 11:28:11.681	JavaEEServer	00000F00 000009DC	KDKD10083-I	The server was started successfully. (server type = prf, server name = PRF1)
0056	2015/06/18 11:28:24.554	JavaEEServer	00000F00 00001C78	KDKD10083-I	The server was started successfully. (server type = javaee, server name = JavaEE1)
0057	2015/06/18 11:28:24.558	JavaEEServer	00000F00 00001C78	KDKD10162-I	The command was executed successfully. (command = start-instance, input = {instance_name=JavaEE1})
0058	2015/06/18 11:28:27.914	JavaEEServer	00000F00 00000124	KDKD10138-E	An error occurred in the execution of the file. (file name = D:\Hitachi\APServer\javaee\glassfish\nodes\localhost-domain1\Web1\root\bin\start-webserver.bat, return code = 1, details = start-webserver.bat: Command failed. "D:\Hitachi\APServer\httpsd\httpsd.exe" -n "Web1" -k start, ret=1 (OS 10048)Only one usage of each socket address (protocol/network address/port) is normally permitted. : AH00072: make_sock: could not bind to address 0.0.0.0:80 AH00451: no listening sockets available, shutting down AH00015: Unable to open logs)
0059	2015/06/18 11:28:27.914	JavaEEServer	00000F00 00000124	KDKD10089-E	An error occurred in the processing to start the server. (server type = webserver, server name = Web1)
0060	2015/06/18 11:28:27.915	JavaEEServer	00000F00 00000124	KDKD10140-E	An error occurred during the execution of the subcommand on the node. (node name = localhost-domain1, message = KDKD10138-E An error occurred in the execution of the file. (file name = D:\Hitachi\APServer\javaee\glassfish\nodes\localhost-domain1\Web1\root\bin\start-webserver.bat, return code = 1, details = start-webserver.bat: Command failed. "D:\Hitachi\APServer\httpsd\httpsd.exe" -n "Web1" -k start, ret=1 (OS 10048)Only one usage of each socket address (protocol/network address/port) is normally permitted. : AH00072: make_sock: could not bind to address 0.0.0.0:80 AH00451: no listening sockets available, shutting down AH00015: Unable to open logs) KDKD10089-E An error occurred in the processing to start the server. (server type = webserver, server name = Web1) Command _call-server-adapter-start executed successfully.)
0061	2015/06/18 11:28:27.916	JavaEEServer	00000F00 00000F88	KDKD10084-E	Processing to start the server failed. (server type = webserver, server name = Web1)
0062	2015/06/18 11:28:27.918	JavaEEServer	00000F00 0000235C	KDKD10104-E	Processing to start the servers in a batch failed.

(4) **KDKD10083-I The server was started successfully. (server type = prf, server name = PRF1)**

KDKD10083-I The server was started successfully. (server type = javaee, server name = JavaEE1)

パフォーマンストレーサ(PRF1)と JavaEE サーバ(JavaEE1)は起動に成功していることが確認できます。

(5) **AH00072: make_sock: could not bind to address 0.0.0.0:80 AH00451: no listening sockets available, shutting down AH00015: Unable to open logs) KDKD10089-E An error occurred in the processing to start the server. (server type = webserver, server name = Web1)**

80 番ポートを開くのに失敗したため、Web サーバ(Web1)の起動に失敗しています。

(6) (OS 10048)Only one usage of each socket address (protocol/network address/port) is normally permitted.

他のプログラムによって既に 80 番ポートが開かれていることが推測できます。

■想定される原因と対処

起動しようとした Web サーバの HTTP ポート番号として設定した 80 番ポートが、他のプログラムによって使用中のために、Web サーバの起動に失敗したと考えられます。80 番ポートを使用しているプログラムを終了するか、Web サーバの HTTP ポート番号を変更してください。

(2) Java EE サーバ起動時のトラブル例

構築済みの Java EE サーバの起動に失敗した例を挙げます。

■トラブルシュート資料を利用した障害解析の流れ

コマンド	<code>asadmin start-instance JavaEE1</code>
実行結果	<p>remote failure: Could not start instance JavaEE1 on node localhost-domain1 (localhost).</p> <p>Command failed on node localhost-domain1 (localhost): Attempting to start JavaEE1.... Please look at the server log for more details.....</p> <p>To complete this operation run the following command locally on host localhost from the GlassFish install location C:\Hitachi\APServer\javaee:</p> <pre>lib/nadmin start-local-instance --node localhost-domain1 --sync normal --wait-timeout 60000 JavaEE1</pre> <p>Command start-instance failed.</p>

(1) **remote failure: Could not start instance JavaEE1 on node localhost-domain1 (localhost).**

Java EE サーバの起動に失敗したことがわかります。

(2)asadmin プロセス起動時ログを参照します。

asadmin プロセス起動時ログから複数の原因が考えられるため、事例ごとに想定される原因と対処を説明します。

<不正オプションが指定された場合>

asadmin プロセス起動時ログに以下のメッセージが出力されます。

0000	2015/06/23 15:00:56.678	asadmin	0000239C 00001D10	(name =)NCLS-GFLAUNCHER-00005
0001	2015/06/23 15:00:56.812	asadmin	0000239C 00000ED8	(name =)Unrecognized VM option 'ForTest'
0002	2015/06/23 15:00:56.812	asadmin	0000239C 00000ED8	(name =)Error: Could not create the Java Virtual Machine.
0003	2015/06/23 15:00:56.812	asadmin	0000239C 00000ED8	(name =)Error: A fatal exception has occurred. Program will exit.

■想定される原因と対処

(name =)Unrecognized VM option 'ForTest'

から、不正なオプションが指定されていることがわかります。正しいオプションを指定して Java EE サーバを起動してください。

<メモリ不足で起動に失敗した場合>

asadmin プロセス起動時ログに以下のメッセージが出力されます。

0000	2015/06/23 14:06:13.707	asadmin	000016FC 00002160	(name =)NCLS-GFLAUNCHER-00005
0001	2015/06/23 14:06:13.987	asadmin	000016FC 000015FC	(name =)Error occurred during initialization of VM
0002	2015/06/23 14:06:13.987	asadmin	000016FC 000015FC	(name =)Could not reserve enough space for object heap
0003	2015/06/23 14:06:13.987	asadmin	000016FC 000015FC	(name =)requested commit size: 1073741824 bytes.
0004	2015/06/23 14:06:13.987	asadmin	000016FC 000015FC	(name =) Memory in use : 67%
0005	2015/06/23 14:06:13.987	asadmin	000016FC 000015FC	(name =) Physical memory : 790495232 / 2415452160 free

■想定される原因と対処

(name =)Could not reserve enough space for object heap

から、指定された Java ヒープの大きさが確保できなかったということがわかります。Java ヒープ領域の初期サイズを見直してください。

3.2.3 アプリケーションデプロイ時発生するトラブル

以下に、アプリケーションデプロイの流れと主なログの出力ポイントを示します。

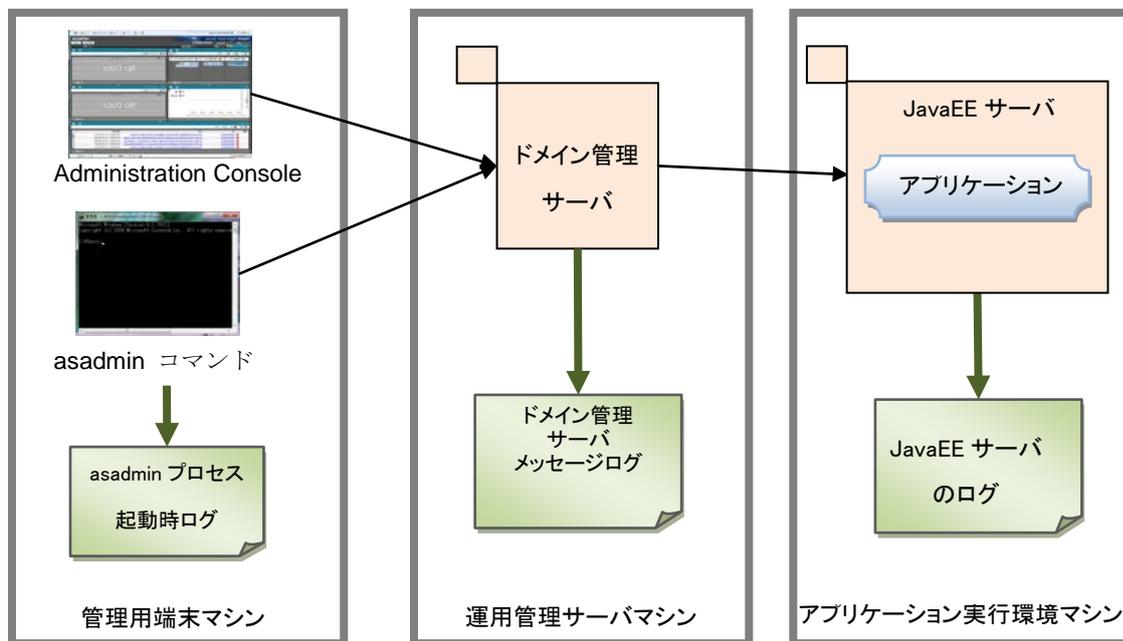


図 3.2-3 アプリケーションデプロイの流れと主なログ

アプリケーションデプロイ時のトラブルは、主に次の箇所で原因を確認できます。

- Administration Console 上に表示されるメッセージ
- `asadmin` コマンドを実行したコンソールに表示されるメッセージ
- ドメイン管理サーバメッセージログ

ドメイン管理サーバメッセージログは、以下に出力されます。

```
C:\Hitachi\APServer\javaee\logs\domains\domain1\das_message.n.log
```

それでも原因が判明しない場合は、ドメイン管理サーバから起動されたサブコマンドのログを追うことで、エラーや警告の原因がわかることがあります。サブコマンドのログは `asadmin` プロセス起動時ログに出力されます。

`asadmin` プロセス起動時ログは、以下に出力されます。

```
C:\Hitachi\APServer\javaee\logs\commands\asadmin\asadmin_launch.n.log
```

上記でも解決しない場合、デプロイ対象の JavaEE サーバのログを参照します。

C:\Hitachi\APServer\javaee\logs\nodes\localhost-domain1\JavaEE1\je_message.n.log

(1) アプリケーションデプロイ時のトラブル例

アプリケーションを JavaEE サーバにデプロイしようとした際に失敗した例を挙げます。

■ トラブルシュート資料を利用した障害解析の流れ

この例では、`asadmin` コマンドを実行したコンソールに表示されるメッセージに従い、トラブルシュートを行います。

コマンド	<code>asadmin deploy --target JavaEE1 D:\sample.ear</code>
実行結果	remote failure: Error occurred during deployment: Exception while deploying the app [sample] : org.xml.sax.SAXParseException ; lineNumber: 28; columnNumber: 12; Deployment descriptor file WEB-INF/web.xml in archive [sample_war] . 要素タイプ"filter"を宣言する必要があります。 . Please see server.log for more details. Command deploy failed.

(1) **org.xml.sax.SAXParseException**

SAX パーサで例外が発生しています。XML ファイルの読み込み時にエラーが発生したと考えられます。

(2) **WEB-INF/web.xml in archive [sample_war]. 要素タイプ"filter"を宣言する必要があります。**

web.xml 内に”filter”の宣言がないため、失敗しています。

(3) web.xml を確認します。

```
<!DOCTYPE web-app PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application  
2.2//EN" "http://java.sun.com/j2ee/dtds/web-app_2.2.dtd">  
  
<web-app>  
  <display-name>sample_war</display-name>  
  :  
  :  
<filter>
```

```
<filter-name>com.hitachi.software.was.sfo.web.SFOFilter</filter-name>
<filter-class>com.hitachi.software.was.sfo.web.SFOFilter</filter-class>
</filter>

</web-app>
```

(4) "http://java.sun.com/j2ee/dtds/web-app_2.2.dtd"

DTD 宣言の Servlet バージョンが 2.2 になっています。

<filter>タグは Servlet バージョン 2.3 以降でサポートされています。

■ 想定される原因と対処

web.xml 内で、Servlet バージョン 2.3 以降でサポートされている<filter>タグを用いているが、DTD 宣言の Servlet バージョンが 2.2 であったため、XML の読み込みでエラーが発生したと考えられます。DTD 宣言の Servlet バージョンを 2.3 以降に修正してください。

3.2.4 データベース接続時に発生するトラブル

以下に、データベース接続の流れと主なログの出力ポイントを示します。

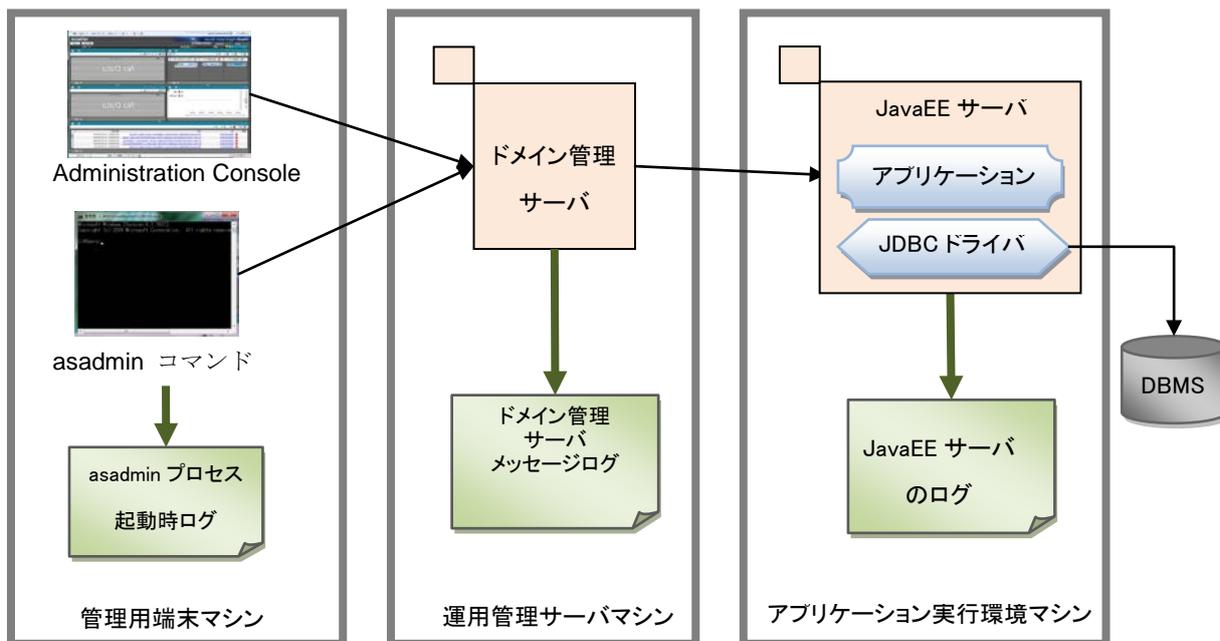


図 3.2-4 データベース接続の流れと主なログ

(1) データベースサーバへの接続時のトラブル例

データベースサーバへの接続テストで失敗した例を挙げます。

■ トラブルシュート資料を利用した障害解析の流れ

この例では、`asadmin` コマンドを実行したコンソールに表示されるメッセージに従い、トラブルシュートを行います。

コマンド	<code>asadmin create-jdbc-connection-pool --datasourceclassname JP.co.Hitachi.soft.HiRDB.JDBC.PrdbDataSource --restype javax.sql.DataSource --property User=user1:Password=pass1:DBHostName=192.0.2.0:description=22200 MyDBPool</code>
実行結果	JDBC connection pool MyDBPool created successfully. JavaEE1: JDBC connection pool MyDBPool created successfully. Command create-jdbc-connection-pool executed successfully.
コマンド	<code>asadmin ping-connection-pool MyDBPool</code>
実行結果	remote failure: Ping Connection Pool failed for MyDBPool. Class name is wrong or classpath is not set for : JP.co.Hitachi.soft.HiRDB.JDBC.PrdbDataSource Please check the server.log for more details. Command ping-connection-pool failed.

(1) **Command ping-connection-pool failed.**

データベースへの接続に失敗しています。

(2) **Class name is wrong or classpath is not set for : JP.co.Hitachi.soft.HiRDB.JDBC.PrdbDataSource**

クラス名が間違っている、もしくはクラスパスが設定されていないために失敗しています。

(3) JDBC ドライバの配置を確認します。

```
C:\¥Hitachi¥APServer¥javaee¥glassfish¥domains¥domain1¥lib¥databases¥pdjdbc4.jar
```

JDBC ドライバの配置場所が誤っています。

■ 想定される原因と対処

JDBC ドライバの配置位置が誤っているためクラスパスが通っておらず、データベースへの接続に失敗しています。以下の場所に JDBC ドライバを配置してください。databases

という名称のディレクトリーが存在して間違えやすいため、注意してください。

C:\Hitachi\APServer\javaee\glassfish\domains\domain1\lib\pdjdbc4.jar

3.3 システム運用時のトラブル

システム運用時に発生するトラブルでは、事例ごとに調査の流れおよびログなどを説明します。

(1) Java EE サーバプロセスダウンの場合

Java EE サーバプロセス障害で想定される原因を次に示します。障害解析方法では、図中に示す原因の切り分け方法と、調査を依頼するための調査情報の取得方法を説明します。

注意事項

Application Server の障害解析の前に、システム（ハードウェア、OS、およびネットワーク）に問題がないかどうか確認してください。

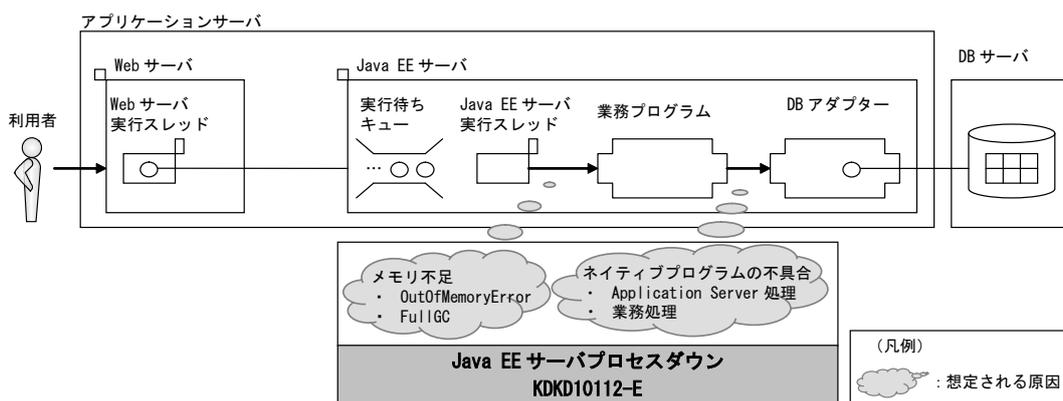


図 3.3-1 想定される原因

■ トラブルシュート資料を利用した障害解析の流れ

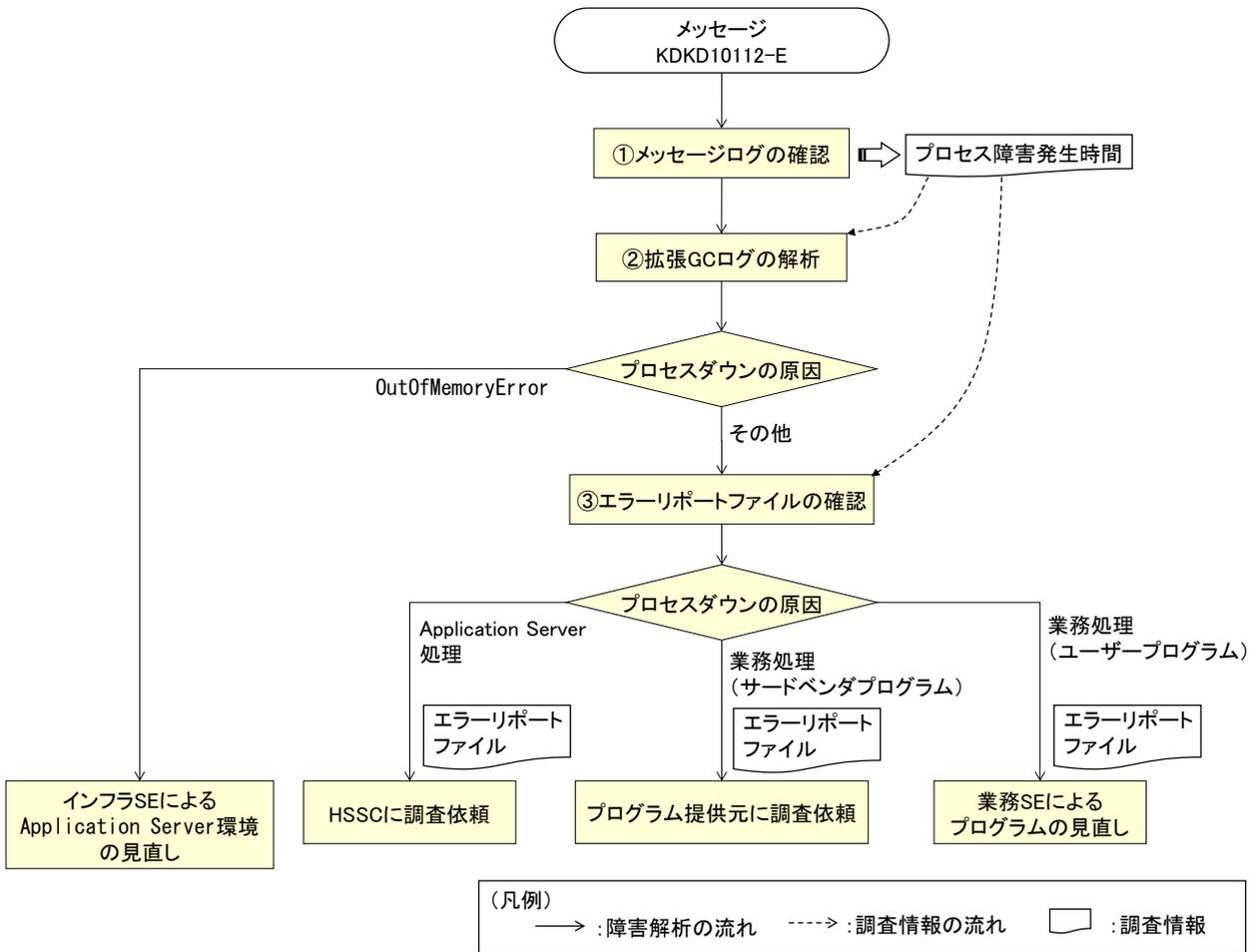


図 3.3-2 障害解析の流れ

① メッセージログを確認する

ドメイン管理サーバのメッセージログで、プロセス障害発生時間を取得します。次に示すファイルにメッセージログが格納されています。

snapshot ログ (初期調査用システム情報)	C:\¥Hitachi¥APServer¥javaee¥snapshot¥snapshot-initinfo-<収集対象名>-<日時>-<プロセス ID>-<スレッド ID>.zip
メッセージログ	<snapshot ログ (初期調査用システム情報)>¥C:\¥Hitachi¥APServer¥javaee¥logs¥domains¥domain1¥das_message.n.log
メッセージ例	0819 2014/09/25 15:26:55.827* JavaEEServer 00000398 000004A8 KD KD10112-E A server failure was detected. (server type = javaee, server name = JavaEE1, server status = not running)

注※ この例では、プロセス障害発生時間は「2014/09/25 15:26:55.827」となります。

② 拡張 GC ログを解析する

拡張 GC ログで、プロセス障害発生時間帯に `OutOfMemoryError` が発生していないかどうか確認します。次に示すファイルに拡張 GC ログが格納されています。

snapshot ログ (初期調査用システム情報)	C:\Hitachi\APServer\javaee\snapshot\snapshot-initinfo-<収集対象名>-<日時>-<プロセス ID>-<スレッド ID>.zip
拡張 GC ログ	<snapshot ログ (初期調査用システム情報) >\C:\Hitachi\APServer\javaee\logs\nodes\localhost-domain1\JavaEE1\je_javavm*.log

下記に拡張 GC ログの例を示します。

`java.lang.OutOfMemoryError` の表記が `OutOfMemoryError` の発生を示しています。

[OOM][Thread: 0x00000000015b000]<Tue Jul 07 17:22:13.719 2015>[java.lang.OutOfMemoryError : requested 32818160 bytes. (Java Heap) : 12 threads exist]

③ エラーレポートファイルを確認する

ネイティブプログラムの不具合によるプロセス障害発生時に出力されるエラーレポートファイルで、問題のライブラリー名を確認します。次に示すファイルにエラーレポートファイルが格納されています。

snapshot ログ (初期調査用システム情報)	C:\Hitachi\APServer\javaee\snapshot\snapshot-initinfo-<収集対象名>-<日時>-<プロセス ID>-<スレッド ID>.zip
エラーレポートファイル	<snapshot ログ (初期調査用システム情報) >\C:\Hitachi\APServer\javaee\glassfish\nodes\localhost-domain1\JavaEE1\config\hs_err_pid*.log

(2) スローダウンの場合

スローダウンで想定される原因を次に示します。障害解析方法では、図中に示す原因の切り分け方法と、調査を依頼するための調査情報の取得方法を説明します。

注意事項

Application Server の障害解析の前に、システム（ハードウェア、OS、およびネットワーク）に問題がないかどうか確認してください。

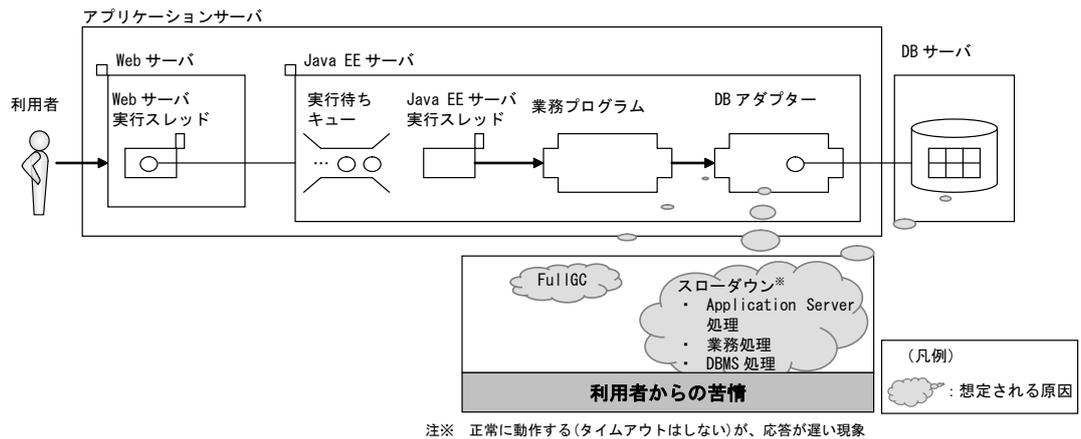


図 3.3-3 想定される原因

■ トラブルシュート資料を利用した障害解析の流れ

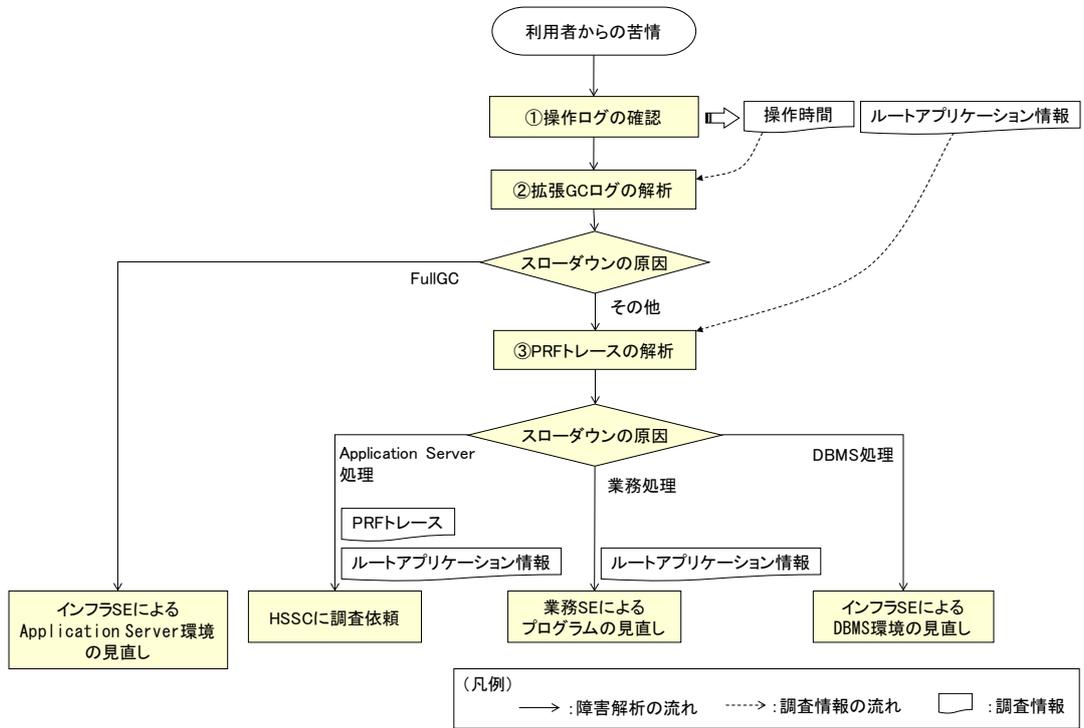


図 3.3-4 障害解析の流れ

<手順>

① 操作ログを確認する

業務プログラムが出力する操作ログで、操作時間、ルートアプリケーション情報を取得します。

② 拡張 GC ログを解析する

拡張 GC ログで、プロセス障害発生時間帯の FullGC 状況（時間、頻度）に問題がないかどうか確認します。次に示すファイルに拡張 GC ログが格納されています。

snapshot ログ (初期調査用システム情報)	C:\¥Hitachi¥APServer¥javaee¥snapshot¥snapshot-initinfo-<収集対象名>-<日時>-<プロセス ID>-<スレッド ID>.zip
拡張 GC ログ	<snapshot ログ (初期調査用システム情報) >¥C:\¥Hitachi¥APServer¥javaee¥logs¥nodes¥localhost-domain1¥JavaEE1¥je_javavm.n.log

下記に拡張 GC ログの例を示します。1回の FullGC ごとに、ログが 1 行出力されます。

この例では、ある 1 回の FullGC に約 0.145 秒を要していることが確認できます。

[VGC]<Tue Jul 07 17:22:13.413 2015>[Full GC 65284K->64964K(83200K), 0.1450083 secs][DefNew::Eden: 12889K->7620K(23040K)][DefNew::Survivor: 0K->0K(2816K)][Tenured: 52394K->57344K(57344K)][Metaspace: 3784K(4104K, 4352K)->3784K(4104K, 4352K)][cause:ObjAllocFail][User: 0.1404009 secs][Sys: 0.0000000 secs][IM: 3130K, 3488K, 0K][TC: 12][DOE: 0K, 0][CCI: 493K, 49152K, 2496K]
--

③ PRF トレースを確認する

PRF トレースで、スローダウンしたリクエストのどの処理に問題があるのかを確認します。次に示すファイルに PRF トレースが格納されています。

snapshot ログ (初期調査用システム情報)	C:\¥Hitachi¥APServer¥javaee¥snapshot¥snapshot-initinfo-<収集対象名>-<日時>-<プロセス ID>-<スレッド ID>.zip
PRF トレース	<snapshot ログ (初期調査用システム情報)>¥C:\¥Hitachi¥APServer¥javaee¥logs¥nodes¥localhost-domain1¥PRF1¥log¥PRF1

スローダウンしたリクエストのトレース (①で取得したルートアプリケーション情報 (RootAP IP/RootAP PID/RootAP CommNo.)を使って"RootAP IP"、"RootAP PID"、"RootAP CommNo."をフィルタリング) で、問題がある処理を確認します。

業務プログラムがスローダウンした場合、ユーザー拡張性能解析トレース機能によって、どの業務処理に時間が掛かっているか確認できます。

ユーザー拡張性能解析トレースは、ユーザー拡張性能解析トレース設定ファイルで設定したメソッドがトレース取得対象となり、指定メソッドの入口と出口でトレースが出力されます。

詳細については、マニュアル「Hitachi Application Server V10 定義リファレンス (Windows(R)用)」の「ユーザー拡張性能解析トレースで使用するプロパティ」および「userprf.cfg」を参照してください。