

**WebOTX 運用トラブル時の
早期解決方法
(対象バージョン:V8)**

NEC

第二システムソフトウェア事業部

改版履歴

2009 年 11 月 初版

目次

1. はじめに	3
2. トラブル事例とその解決方法	4
2.1. 起動処理失敗	4
2.1.1. ドメインまたはサービスの起動エラー	4
2.1.2. プロセス残存に伴う起動エラー	10
2.1.3. タイムアウトエラー	10
2.1.4. ドメイン設定ファイル上の問題(不正定義、ファイルデータ破損)	11
2.1.5. 仮想マシンエラー (VirtualMachineError)	13
2.2. 異常終了	14
2.2.1. 予期せぬシャットダウンイベント発生	15
2.2.2. hs_err_pid****.logファイル出力(JVMクラッシュ)を伴う異常終了	16
2.3. 実行環境のリソース不足に伴うエラー	23
2.3.1. メモリ不足	23
2.3.2. ディスク不足	26
2.4. ストール現象	27
2.5. 環境不正に伴う問題	29
2.6. トラブル解決作業流れ図	29
3. トラブル回避に有用な機能のご紹介	32
3.1. 統計情報採取(モニタリング)、イベント通知	32
3.2. ログ、統計情報採取設定	38
3.3. JVM監視・管理ツール	38
3.3.1. JVM動作情報出力コマンド	39
3.3.2. jconsole接続	39
3.4. WebOTXプロファイラ(性能分析ツール)	39
3.5. 実行環境リカバリコマンド	40
3.6. 設定項目の比較	40
3.7. 診断サービス(実行情報、設定ファイル、ログの一括採取)	41

1. はじめに

本資料では、WebOTX Application Server(以降、WebOTX と記載)を使用して業務システム構築作業を行う方、およびそのシステムにおける WebOTX の運用・保守を行われる方を対象に、WebOTX が正常に動作しなくなったり、異常終了したりするような場面に遭遇した際にもできるだけ早急な問題解決が行えるよう、そのヒントやノウハウについてわかりやすく記載しています。

関連資料

- WebOTX マニュアル 運用編 運用管理コマンド、統合運用管理ツール、Web 版統合運用管理コンソール、プロファイラツール
- WebOTX マニュアル 運用編 運用と操作(ドメインの運用)
- WebOTX マニュアル 運用編 運用と操作(ログイン)
- WebOTX マニュアル 運用編 運用と操作(モニタリング)
- WebOTX マニュアル 運用編 運用と操作(診断サービス)
- WebOTX マニュアル 運用編 障害解析
- WebOTX マニュアル 運用編 リファレンス(運用管理コマンドリファレンス)

これらのマニュアルは、以下のサイトからダウンロードすることが可能です。

<http://www.nec.co.jp/WebOTX/download/develop.html>

補足

本資料中で使用している<...>形式(斜体表記)の文字列は、利用する環境によって値が変化しますので、環境に合わせて読み替えてください。

(例) <INSTALL_ROOT> : WebOTX のインストールディレクトリへのパスを表します。

<INSTANCE_ROOT> : WebOTX 上のドメインに対するルートディレクトリへのパスを表します。

2. トラブル事例とその解決方法

本章では、WebOTX を使用した業務システム構築時、またはそのシステム運用時に遭遇しやすいトラブルについて取り上げ、その解決方法について紹介します。

なお、本章の最後には、実際に発生したトラブルが以降で紹介する事例に該当するかどうかを効率良く見極めるための流れ図を記載しています。併せてご覧ください。

2.1. 起動処理失敗

直前まで正常に動作していた WebOTX が設定変更後にエラーが発生したり、OS 再起動後に正常動作しなくなったりする場合は、次のような事例に該当していることが考えられます。

- ドメインまたはサービスの起動エラー
- プロセス残存に伴う起動エラー
- タイムアウトエラー
- ドメイン設定ファイル上の問題(不正定義、ファイルデータ破損)
- 仮想マシンエラー (VirtualMachineError)







以下では、これらの事例に対する問題の解決方法について紹介します。

2.1.1. ドメインまたはサービスの起動エラー








まず、WebOTX が現在正常に起動しているかどうかを確認するために、WebOTX 上のドメインおよびドメイン内で起動するサービスの状態を以下の方法で確認します。

- ・「統合運用管理ツール」もしくは「Web版統合運用管理コンソール」による確認方法

画面左の管理対象ツリーにおいて、「WebOTX管理ドメイン」ツリーノード配下の「<任意のドメイン名>」ノードのアイコンを確認します。より正確な動作状況を把握するには、さらに「<任意のドメイン名>」→「アプリケーションサーバ」→「内部ライフサイクルモジュール」と辿り、その配下にある各システムサービスノードのアイコンも確認します。確認したそれぞれのアイコンから、以下の表を参考に動作状況を把握します。¹

ドメイン	ライフサイクルモジュール	判定
	全て 	正常起動(システム移動中)
	   が1つでも存在する	ドメインは起動したが一部のサービスに異常あり

¹ 事前設定により起動されないドメインもしくはサービスのアイコンは、 にはなりません。

		(そのサービスを使用する業務アプリケーションがある場合には正常に動作しない可能性大)
	全て 	ドメイン起動処理に異常あり (起動処理は継続中かまたは停滞している可能性大、しばらく経ってドメインのアイコンが  になるかどうか要確認)
	   が1つでも存在する	ドメインおよびサービスの起動処理に異常あり (途中で起動処理が停滞している可能性大、システムは使用不可)

・ 「**運用管理コマンド**」による確認方法

プロンプト画面上でlist-domainsコマンドを実行します。

```
otxadmin> list-domains
list of domains:
domain1 running
WebOTXAdmin running
```

より正確な動作状況を把握するには、続けて各ドメインにログインして各サービスの状態を確認します(上記の実行結果がnot runningでない場合)。

```
otxadmin> get server.internal-lifecycle-module.*.state
server.internal-lifecycle-module.AdminService.state=1
server.internal-lifecycle-module.ApplicationService.state=1
:
```

確認したそれぞれの結果から、以下の表を参考に動作状況を把握します。²

ドメイン	ライフサイクルモジュール	判定
running	全て 1 または ready-complete	正常起動(システム稼動中)
running	上記以外の値が1つでも存在する	ドメインは起動したが一部のサービスに異常あり (そのサービスを使用する業務アプリケーションがある場合には正常に動作しない可能性大)
starting	全て 1 または ready-complete	ドメイン起動処理に異常あり

² 事前設定により起動されないドメインはrunningにはなりません。同様に、サービスは1(ready-complete)にはなりません。

		(起動処理は継続中かまたは停滞している可能性大、しばらく経ってドメインの状態が 1 または ready-complete になるかどうか要確認)
starting	上記以外の値が1つでも存在する	ドメインおよびサービスの起動処理に異常あり (途中で起動処理が停滞している可能性大、システムは使用不可)

これらの確認作業により、ドメインもしくはサービスが正常に起動していない場合は、まず異常が発生したドメインに対するログを調査し、エラー情報の出力がないかをチェックします。<INSTANCE_ROOT>/logs ディレクトリ配下にある webotx_agent.log を参照し、以下のようなログが出力されているかを確認してください。

サービス "<SERVICE_NAME>" を起動することができません! <SERVICE_NAME>は起動していないか、もしくは起動に失敗しています!!

Service [<SERVICE_NAME>] cannot be started! : <SERVICE_NAME> is not started or fails to start!!

アプリケーションサーバ [<DOMAIN_NAME>:<SERVICE_NAME>] は起動しましたが、サービス [<SERVICE_NAME1>(, <SERVICE_NAME2>,...)] は正常に起動していない可能性があります。

Application server [<DOMAIN_NAME>:<SERVICE_NAME>] is started, but the service(s) [<SERVICE_NAME1>(, <SERVICE_NAME2>,...)] may not start normally.

これらのログが出力されている場合、ドメイン内の<SERVICE_NAME>で示されたサービス内で何らかの異常が発生したためにサービス(ライフサイクルモジュール)の起動に失敗したことを示しています。そのため、<SERVICE_NAME>で示されたサービスについて調査する必要があります。ただし、このログが複数出力されている場合、ログファイル中のより先頭に(起動開始時刻以降での最も初めに)出力されたログに示されているサービスがエラーの発生源となりますので、まずはそのサービスについて調査します。

なお、「WebOTX マニュアル運用編 障害解析」では、各サービスのエラー発生パターンとその解決方法がまとめられていますので、それらのケースに該当するかどうかを確認しながら解析を進めるのが効率的です。

以下に、ドメイン内部で動作する各サービスと、それに対応するログファイル、およびマニュアルの参照先を示します。

サービス(ライフサイクルモジュール)名	対象ログファイル	マニュアル参照先
全サービス共通	server.log server_err.log webotx_agent.log	「障害の種類から」 →「サービスの起動失敗・異常終了」
AdminService (管理サービス)	webotx_admin.log、 webotx_deploy.log、	「機能別」 →「JMX」 (JMX の障害解析： WebOTX の起動失敗について)
DBControllerService (データベースコントローラ)	(webotx_agent.log に出力)	
JMSProvider (JMS プロバイダ)	wojms.log、 wojmsadmin.log、 wojmserror.log、 wojmsexec.log、 wojmsmessage.log、 wojmsocket.log、 wojmserver.log、 std.log	「機能別」 →「JMS」 (JMS の障害解析)
ObjectBrokerService (Object Broker サービス)	webotx_ospi.log、 Oad.log、 ObLog.log、 InterfaceRepository.log、 cnamesv.log、 colbaloc.log、 message.log、 namesv.log、 oadj.log、 oadjinit.log、 objava.log	「機能別」 →「Object Broker」 (Object Broker の障害解析)
JNDIService (JNDI サービス)	webotx_jndisp.log、 webotx_naming.log	「機能別」 →「JNDI」 (JNDI の障害解析)
TransactionService (Transaction サービス)	webotx_ts.log、 rcsjava.log、 rcsproxy.log、	「機能別」 →「Transaction サービス」

	rcd.trc、 rcs.trc、 lrs.trc、	(Transaction サービス の障害解析)
J2EEServer (J2EE サーバ)	webotx_ejbcont.log、 wojdbc.log、 wojta.log	「機能別」 →「JDBC データソー ス」 (JDBC データソースの 障害解析)
PersistenceManagerService (永続マネージャサービス)	(webotx_agent.log に出力)	「機能別」 →「EJB コンテナ」 (EJB コンテナの障害解 析)
SystemApplicationService (システムアプリケーションサービス)	(webotx_agent.log に出力)	
ApplicationService (アプリケーションサービス)	(webotx_agent.log に出力)	
WebContainerService (Web コンテナサービス)	webotx_catalina.log、 webotx_container.log、 webotx_struts.log、 webotx_webc_message.log、 webotx_webc_stat_vm.log、 webotx_webc_stat_webap.log、 webotx_webcont.log、 mod_jk.log、 mod_jk-20.log、 isapi.log、 nsapi.log、	「機能別」 →「JMX」 (JMX の 障 害 解 析： WebOTX の起動失敗 について)
WebServerService (Web サーバサービス)	webotx_websv.log、 access.log、 error.log、(UNIX 系 OS 上では access_.log、error_.log)	「機能別」 →「Web サーバ (Apache HTTP Server ベース)」 (WebOTX Web サーバ の障害解析)
DownloaderManagerService(ダウンローダ管理サービス)	(webotx_agent.log に出力)	「機能別」 →「TP モニタ (Foundation/Standard/ Enterprise)」 (TP モニタ機能の障害
TPMonitorManagerService (TP モニタマネージャサービス)	webotx_tpmmgr.log を 含 む tpsystem ディレクトリ配下のログ	

		解析:起動・停止関連)
RMSERVICE (WSRM サービス) ※Web サービスの高信頼メッセージング機能を使用する場合	wsrcm.log	
SecurityService (セキュリティサービス)	webotx_security.log	「機能別」 →「JMX」 (JMX の障害解析: WebOTX の起動失敗 について)
WSMgmtService (Web サービス管理)	(webotx_agent.log に出力)	
RemoteJMXConnector (リモート JMX コネクタ)	(webotx_agent.log に出力)	「機能別」 →「JMX」 (JMX の障害解析: WebOTX の起動失敗 について)
LifecycleModuleService (ライフサイクルモジュールサービス)	webotx_esb.log(WebOTX Enterprise Service Bus)を使用する場合	

解析の例として、次のようなログが出力された場合について解説します。

```

2009-07-13 17:11:42,921 WARN ... - OTX01205102: サービス "リモート JMX コネクタ(RemoteJmxConnector)" を初期化することができません！ 起動する際は、各種運用管理ツールから再度実行して下さい : java.rmi.server.ExportException: Port already in use: 6212; nested exception is:
    java.net.BindException: Address already in use: JVM_Bind [main]
    :
2009-07-13 17:12:41,750 WARN ... - OTX01205104: サービス "リモート JMX コネクタ(RemoteJmxConnector)" を起動することができません！ : サービスは起動していないか、もしくは起動に失敗しています!! [main]
2009-07-13 17:12:42,703 WARN ... - OTX01205166: アプリケーションサーバ [domain1:server] は起動しましたが、サービス [RemoteJmxConnector] は正常に動作していない可能性があります。[main]

```

この場合、「リモート JMX コネクタ(RemoteJmxConnector)」というサービスの起動に失敗していますので、上述の表より、「WebOTX マニュアル運用編 障害解析」における「機能別」→「JMX」(JMX の障害解析:WebOTX の起動失敗について) を参照する、という流れになります。

なお、この例では、最初のログメッセージからも、指定されたポート番号(6212)が既に他のアプリケーションによって使用されていたことが起動エラーの原因であるとわかります。

2.1.2. プロセス残存に伴う起動エラー

前回起動していた WebOTX のプロセスが何らかの原因で停止処理後も残り続けていた場合、その状態で新たに WebOTX の起動処理を行うと、残存プロセスが影響して正常に起動できなくなることがあります。そのため、起動に失敗した際には必ずその時点で残存プロセスがないかを確認し、プロセスが残っている場合にはそれらを削除します。

対象のプロセスについての確認は、Windows 系 OS の場合、タスクマネージャや Process Explorer などのツールを使用します。UNIX 系 OS の場合は、`ps` コマンドを使用します。

また、対象のプロセスについての詳細は、「WebOTX マニュアル 運用と操作(ドメインの運用)」本文中の「動作プロセスについて」で記載しています。

残存プロセスがある場合、まずは運用管理コマンド上で、対象となるドメインに対して次のコマンドを実行します。

```
otxadmin> stop-domain --force <ドメイン名>
```

コマンドの実行完了後、残存プロセスが削除されているかを確認します。このコマンドを実行してもプロセスが削除されない場合は、それらのプロセスを一つずつ強制終了します。

2.1.3. タイムアウトエラー

Windows のサービス(WebOTX Agent Service)による WebOTX 起動処理に時間が掛かり過ぎると、サービスのタイムアウトが発生します。この場合、WebOTX はタイムアウト経過後も処理を続けるために起動処理が完了し、システムが使用可能な状態になっていても、サービス上の状態は「**停止**」になっていることがあります。

タイムアウトが発生する原因には、主に次のことが挙げられます。

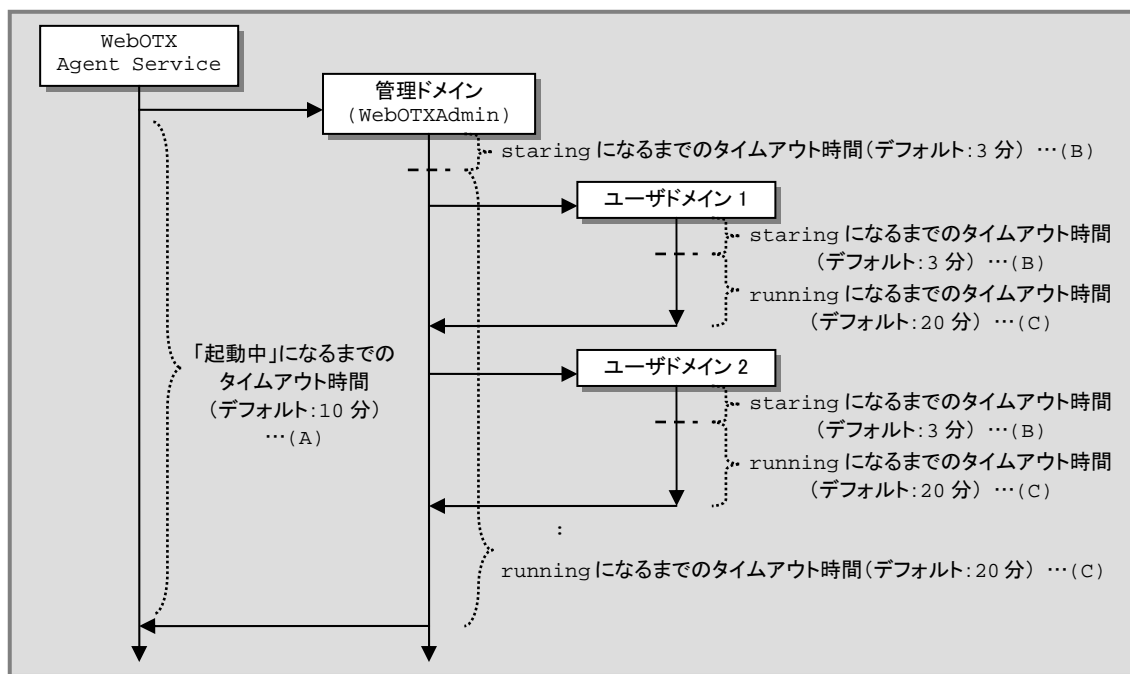
- ・ マシンが高負荷な状況であるために、通常より WebOTX 全体の起動処理が遅延している。
- ・ WebOTX 内の特定のドメインに対する起動処理に時間がかかっている。
- ・ WebOTX 全体の起動処理時間に対してタイムアウト設定時間に余裕がない。

また、`start-domain` コマンドによるドメインの起動においても、ドメイン内の特定の処理に時間が掛かり過ぎると、同様にタイムアウトエラーが発生します。この場合、ドメインの起動処理は中断されてしまいます。

このような場合、タイムアウト時間の設定を見直すことでタイムアウトの発生を回避す

ることができます。その際、具体的にタイムアウトが発生しているポイントを見つけ、その部分に対するタイムアウト時間を延ばします。

以下に、サービスおよびドメインとタイムアウトの関係について示します。



例えば、管理ドメイン(WebOTXAdmin)から全てのユーザドメインの起動完了を終えるまでに(A)で設定したタイムアウト時間を越える場合には、(A)のタイムアウト時間を延ばします。しかし、(B)や(C)の箇所で設定したタイムアウト時間を越える場合には、(B)や(C)のタイムアウト時間を延ばした結果、(A)のタイムアウト時間を越えることのないように(A)の値も調整する必要があります。

「WebOTX マニュアル運用編 障害解析」→「障害の種類から」→「サービス起動失敗・異常終了」→「ドメイン起動失敗(ドメインの起動に失敗する。ドメインが異常終了する。)」には、起動処理中のタイムアウトに関する記載がありますので、こちらを参考に対象となるタイムアウト時間設定の見直しを行います。

なお、タイムアウトが発生する際にマシンが高負荷の状況にある場合には、WebOTX の起動タイミングを変更することや、他の不要なアプリケーションを起動しないように設定してマシン全体の負荷を抑えることで、WebOTX の設定を変更することなくタイムアウトの発生を抑えることが期待できます。

2.1.4. ドメイン設定ファイル上の問題(不正定義、ファイルデータ破損)

webotx_agent.log を参照し、以下のようなログが出力されているかを確認します。

ドメイン構成ファイル <INSTANCE_ROOT>/config/domain.xml の読み込みに失敗しました：

Failed to load the configuration file <INSTANCE_ROOT>/config/domain.xml：

このログが出力されている場合、WebOTX 上のドメインに対する設定ファイルである domain.xml に問題があり、起動に失敗したことが考えられます。このログの続きにはさらに詳細なエラーメッセージが出力されていますので、以下で示すエラーメッセージの種類に応じて対処してください。

<INSTANCE_ROOT>/config/domain.xml（指定されたファイルが見つかりません。）

このメッセージが出力されている場合、設定ファイルが誤って削除されたか、ファイル名が変更されている可能性があります。もし、ファイルが存在しない場合は、<INSTANCE_ROOT>/backup/files1/config ディレクトリ配下にある(バックアップされた)domain.xml ファイルをコピーし、再度ドメインを起動します。

XML-DOM ドキュメントの作成に失敗しました：

このメッセージが出力されている場合、さらに続きのメッセージによって設定ファイル上の問題点が出力されていますので、下記の表を参考に修正を行ってください。

メッセージ	対処方法
Premature end of file.	設定ファイルの破損により、中身が空になっています。同一ディレクトリ内に domain.xml.backup というファイルが存在する場合、そのファイル名を domain.xml に変更します。存在しない場合は、<INSTANCE_ROOT>/backup/files1/config ディレクトリ配下にある domain.xml ファイルをコピーします。その後、ドメインを起動します。
XML document structures must start and end within the same entity.	ファイル中のタグ要素が整形形式でない箇所があるため、修正します。
The markup in the document preceding the root element must be well-formed.	ルートの要素に先行するドキュメントが整形形式でない箇所があるため、修正します。
The content of element type	<ELEMENT> タグの個要素タグが不正であり、

"<ELEMENT>" must match "(<CHILD_ELEMENT_1> <CHILD_ELEMENT_2> ...)*".	<CHILD_ELEMENT_1>、<CHILD_ELEMENT_2>、...のいずれかに修正します。
Element type "<ILLEGAL_ELEMENT>" must be declared.	不正なタグ<ILLEGAL_ELEMENT>が含まれているため、修正します。
Attribute "<ATTR_NAME>" is required and must be specified for element type "<ELEMENT>" .	<ELEMENT> タグにおいて必須項目である<ATTR_NAME>属性が定義されていないため、追加します。
Attribute "<ATTR_NAME>" with value "<ATTR_VALUE>" must have a value from the list "<LEGAL_ATTR_ELEMENTS>" .	<ATTR_NAME>属性の指定値<ATTR_VALUE>が不正であるため、<LEGAL_ATTR_ELEMENTS>の属性リストから選択し直します。
Open quote is expected for attribute "<ATTR_NAME>" associated with an element type "<ELEMENT>" .	<ELEMENT>タグにおける<ATTR_NAME>属性に対して不正なキャラクタ'<'が検出されたため、修正します。
Element type "<ELEMENT>" must be followed by either attribute specifications, ">" or ">".	<ELEMENT>タグ内は属性もしくは文字列">"、"/>"で続くように修正します。
Attribute "<ILLEGAL_ATTR_NAME>" must be declared for element type "<ELEMENT>" .	<ELEMENT>タグにて<ILLEGAL_ATTR_NAME>という不正な属性が定義されているため、修正します。

2.1.5. 仮想マシンエラー (VirtualMachineError)

server.log、加えて Foundation/Standard/Enterprise 環境である場合は、<INSTANCE_ROOT>/logs/tpsystem 配下のログファイルにおいて、以下のようなメッセージが出力されていないかを確認します。

```
Error occurred during initialization of VM
Could not reserve enough space for object heap
```

このメッセージが出力されている場合、起動時に要求されている Java ヒープメモリをマシン上で確保できず、JVM を立ち上げられなかったことを示しています。その結果、WebOTX の起動に失敗しています。このメッセージが出力されたログを基に、設定の見直しを行います。

■server.log に出力されている場合の対処方法

<INSTANCE_ROOT>/config ディレクトリ配下にある domain.xml ファイル内の次の設

定箇所を編集します。

```
<java-config ... max-heap-size="<指定サイズ>" ...>
:
```

<java-config>タグ内で指定している max-heap-size 属性(-Xmx オプションと同義)の値を指定サイズよりも小さくします。この時、<java-config>タグ内で同様に heap-size 属性(-Xms オプションと同義)についても同じサイズを指定している場合は、その値も同様に小さくします。設定変更後、ファイルを保存します。

■<INSTANCE_ROOT>/logs/tpsystem に出力されている場合の対処方法

- ・ 「統合運用管理ツール」もしくは「Web 版統合運用管理コンソール」による変更方法
画面左の管理対象ツリーにおいて、「WebOTX 管理ドメイン」→「<対象のドメイン名>」→「アプリケーショングループ」→「<対象のアプリケーショングループ名>」→「プロセスグループ」→「<対象のプロセスグループ名>」と各ノードを辿り、画面右側の「JVM オプション」タブを選択後、表示される項目「最大ヒープサイズ」にて指定されている値よりも小さくします。その後、「更新」ボタンを押下します。この時、同様に「初期ヒープサイズ」についても同じサイズを指定している場合は、その値も同様に小さくします。
- ・ 「運用管理コマンド」による変更方法
対象のドメインにログインし、次のコマンドを実行します。

```
otxadmin> set tpsystem.applicationGroups.<アプリケーショングループ名>.  
.processGroups.<プロセスグループ名>.maxHeapSize=<指定サイズ>
```

この時、同様に「initialHeapSize」についても同じサイズを指定している場合は、その値も同様に小さくします。設定後、アプリケーショングループを再起動します。

2.2. 異常終了

WebOTX を起動したにも関わらずアプリケーションが実行できなかつたり、あるいは直前まで正常に動作していたアプリケーションが突如実行できなくなったりする場合、次のような事例に該当していることが考えられます。

➤ 予期せぬシャットダウンイベント発生

➤ `hs_err_pid****.log` ファイル出力(JVM クラッシュ)を伴う異常終了

以下では、これらの事例に対する問題の解決方法について紹介します。

2.2.1. 予期せぬシャットダウンイベント発生

`webotx_agent.log` またはイベントログに以下のようなメッセージが出力されることがあります。

予期せぬイベントにより、システム内部からアプリケーションサーバのシャットダウン要求が行われました。

```
The shutdown request of the application server was done from the internal system  
by an unexpected event.
```

このメッセージが出力されている場合、WebOTX 内部で発生した何らかのシャットダウン要求によって強制的に WebOTX が停止されたことを示しています。ここで言うシャットダウン要求を受けるのは次のような場合です。

1. WebOTX 起動処理の開始直後にログオフ操作を行った場合(Windows 系)
2. WebOTX プロセスに対して kill コマンドを実行した場合(UNIX 系)
(注意) -SIGQUIT などのオプションを付与した場合は除きます。
3. 運用管理コマンド上で Ctrl+C 操作を行った場合(UNIX 系)
4. ターミナル画面から運用管理コマンドを使用している状態でターミナルを終了した場合(UNIX 系)
5. OutOfMemoryError などの実行環境に起因する致命的なエラーが発生した場合
6. OS レベルで致命的なエラーが発生し、それによって発行されたシグナルを WebOTX が受信した場合
7. OS シャットダウン処理に合わせて WebOTX のシャットダウンを行っている場合

ここで、1.~4. については主に運用者による操作が原因で発生します。これらは OS の仕様に基づいた動作となるため、今後同様の操作を行わないよう再発防止のための対策を取ってください。なお、次の項目については、マニュアルにて対策などを記載しています。

3.、4. : 「WebOTX マニュアル 注意制限事項」

JMX → otxadmin コマンドの「Ctrl+C」とターミナルウィンドウからの強制終了によるドメイン停止について

しかし、5. に該当する場合は、設定上の問題やマシン環境そのものの影響を受けている可能性がありますので、**2.3.節**をご参考に問題解決を行ってください。

また、7. に該当する場合は、Windows OS シャットダウン処理の影響を受けるために大抵の場合出力されます。これは、OS が各プロセスに対して発行した停止要求シグナルを WebOTX のプロセスが受け取るためですが、この場合の出力による業務アプリケーションへの影響はないため、無視することが可能です。もし、このメッセージの出力が運用上好ましくない場合には、Windows OS が提供するシャットダウンスクリプトを使用し、OS のシャットダウンイベントを受け取る前に WebOTX を停止させることで出力を抑止することが可能です。そのための手順は以下の通りです。

1. 次に示す 1 行を含むスクリプト woShutdown.bat を作成し、
C:\¥WINDOWS¥system32¥GroupPolicy¥Machine¥Scripts¥Shutdown、または任意の場所に保存します。
net stop WebOTXAgentService
2. [ファイル名を指定して実行]から「gpedit.msc」を起動します。
3. 「グループ ポリシー」画面より、左側ツリーの[ローカルコンピュータ ポリシー]-[コンピュータの構成]-[Windows の設定]-[スクリプト]を辿り、画面右側に表示される「シャットダウン」を右クリックし、メニューより「プロパティ」を選択します。
4. 「シャットダウンのプロパティ」の「追加」ボタンより、作成したシャットダウンスクリプトを登録します。

2.2.2. hs_err_pid****.log ファイル出力(JVM クラッシュ)を伴う異常終了

WebOTX が異常終了した際、<INSTANCE_ROOT>/config ディレクトリ配下に hs_err_pid<PROCESS_ID>.log という JVM がクラッシュしたことを示すログファイルが生成されることがあります。また、Foundation/Standard/Enterprise の場合は、<INSTALL_ROOT>/Trnsv/bin 配下にも同様のファイルが生成されることがあります。これは大抵の場合、次のような原因が挙げられます。

- ・ JVM が使用するライブラリやネイティブライブラリの不具合(JVM の不具合)
- ・ アプリケーションのネイティブコード上の不具合
- ・ OS のリソース不足、あるいは OS 上の不具合による外的要因

この時、WebOTX が異常終了した時刻とほぼ同一のタイムスタンプを持つクラッシュファイルが存在する場合は、そのファイルの内容を基にクラッシュ発生の原因を特定し、解決手段を見つける必要があります。以下に、本ファイルの出力内容における確認ポイント、および幾つかの代表的なケースでの対処例について示します。

2.2.2.1 異常終了(クラッシュ)の原因

ファイル内の先頭行には、次のようなメッセージが出力されています。

```
#
# An unexpected error has been detected by HotSpot Virtual Machine:
#
# EXCEPTION_ACCESS_VIOLATION (0xc0000005) at pc=0x080451c7, pid=708, tid=3744
#           (1)                               (2)           (3)
# Java VM: Java HotSpot(TM) Server VM (1.4.2_14-b06 mixed mode)
# Problematic frame:
#           (4)
# V [jvm.dll+0x451e7]
#           (5)
```

※利用する JVM バージョンやベンダーにより多少出力内容は異なります。

主に注目すべき内容を下線で示しています。それぞれの意味は次の通りです。

- (1) 異常終了の原因となったシグナル名、あるいはエラー名
- (2) プロセス ID
- (3) スレッド ID
- (4) JVM バージョン
- (5) 異常終了の原因となった JVM 内のプログラム(関数フレーム実行箇所)

ここで、(1)においては、主に次のような内容が出力されます。

EXCEPTION_ACCESS_VIOLATION	: アクセス(読み込み/書き込み)違反(Windows 系)
SIGSEGV	: メモリへの不正アクセス(UNIX 系)
SIGILL	: 不正なマシン命令の呼び出し(UNIX 系)
EXCEPTION_STACK_OVERFLOW	: 実行スレッド内でのスタックオーバーフローの発生
Internal Error	: JVM に関わる内部的なエラーの発生

これらのエラーを発生させたのは、(2)のプロセス内の(3)で示されたスレッドであり、さらにそのスレッドにおいて当時実行されていたプログラムは(5)であったことを示しています。(3)の具体的なスレッドについての情報は、上記出力例のすぐ下に、次のような形式で出力されています。

```
----- T H R E A D -----
```

```
Current thread (0x00f20400):  JavaThread "main" [_thread_in_vm, id=3744]
```

この場合、(3)はmain という名前を持つ **Java** スレッドであったことを示しています。

すなわち、この main スレッドの実行中に(5)で示された `jvm.dll` という **JVM** が使用するライブラリが参照されており、その中で致命的なエラーが発生してクラッシュしたことになります。

以上のようにして本ファイルの内容を読み解くことで、クラッシュ発生当時の状況の概要を理解することができます。

2.2.2.2 回避方法

クラッシュ発生当時の状況を把握した上で、次にクラッシュ発生の原因を突き止める必要があります。この時、先に示したクラッシュ発生当時の情報を基に調査を進めることになりますが、**UNIX** 系 **OS** の場合は通常クラッシュファイルの出力先に **core** ファイルも同時に出力されていますので、**gdb** ツールによって **core** ファイルを読み込ませて解析することで、さらにエラー発生当時のプロセス内部の具体的な動作状況を知ることができます。

これらの解析結果を基に、まずは以下に示す **JVM** ベンダーの公式サイトにて、同様のクラッシュ発生要因が **JVM** の不具合として掲載されていないか、あるいはその不具合が改修されていないかどうかを確認します。

<http://bugs.sun.com/bugdatabase/index.jsp> : Javaバグ情報データベース

<http://java.sun.com/javase/ja/6/webnotes/ReleaseNotes.html> : Java SE 6 リリースノート

<http://java.sun.com/j2se/1.5.0/ja/ReleaseNotes.html> : JDK 5.0 リリースノート

<http://h50146.www5.hp.com/products/software/development/java/index.html> : HP製Java情報

ここで、クラッシュが発生した **JVM** のバージョン(先述の(4)で示された部分)よりも新しいバージョンにて、同様のクラッシュ発生要因が不具合として改修されている場合、その新しいバージョンの **JVM** に切り替えることで、クラッシュの問題は解決します。

なお、異常終了の原因となったシグナル名、あるいはエラー名(先述の(1)で示された部分)に `EXCEPTION_STACK_OVERFLOW` が出力されている場合は、クラッシュの対象となったスレッドにおいて実行されていたネイティブコードによりスタックオーバーフローが発生しています。このネイティブコードによるプログラム箇所(前述の出力例での(5)で示されたプログラム)を確認し、それが業務アプリケーションによるものであれば、そのプログラム上の問題点を確認してください。

2.2.2.3 暫定処置について

前節において、同様のクラッシュ要因がサイト内の情報から見つからない場合や、クラッシュの発生が最新の JVM バージョンで発生している場合、あるいは、システムの運用上の都合で直ちに JVM のバージョンを切り替えられない場合には、独自に何らかの対策を講じなければなりません。その際、クラッシュファイルの出力内容によっては、幾つかの設定変更を行うだけで暫定的にクラッシュの発生を回避できる場合がありますので、その主な対処法について示します。

ただし、いずれの対処法においても、少なからずシステムの実行性能に影響することがあります。アプリケーションのパフォーマンス低下が致命的となる環境においては、本対処は行わないでください。

まず、クラッシュファイルにおいて、次のような出力がないかどうか確認してください。

[ケース 1: JVM 内の特殊スレッドによるクラッシュ発生]

```
----- T H R E A D -----  
  
Current thread (0x00c3c400): VMThread [id=2968]
```

※数値部分は発生の度に値が異なります。

[ケース 2: JVM 内のコンパイルスレッドによるクラッシュ発生]

```
----- T H R E A D -----  
  
Current thread (0x001d8250): JavaThread "CompilerThread0" daemon [_thread_in_vm, id=2208]
```

※数値部分は発生の度に値が異なります。

※"CompilerThread0"の部分は"CompilerThread1"や"AdapterCompiler"になることがあります。

これらに該当する場合、JVM が内部で生成する専用スレッド上で実行されていたプログラムが原因でクラッシュしたことになります。これらのスレッドには、幾つかの実行タイプが存在し、そのタイプの切り替え(=JVM オプションの設定変更)により実行動作が変化します。そのため、クラッシュが発生した実行タイプでない設定に変更した場合、JVM 内部に含まれている不具合が(実行タイプに依存するなどの理由で)抵触しなくなる可能性が高くなり、一時的に現状の実行環境のままシステムの運用を継続できる場合があります。

■ケース 1(JVM 内の特殊スレッドによるクラッシュ発生)の一時的対処方法

ーGC 実行タイプの切り替えー

クラッシュログをさらに追っていくと、次のような情報が記録されている場合があります

す。

```
----- T H R E A D -----  
  
VM_Operation (0x48d6fb44): parallel gc system gc, ...
```

※数値部分は発生の度に値が異なります。

これは、VM_Operation という JVM が GC 処理中に行っている操作に関する情報を示しています。例のように、parallel gc という文字列を含む場合、JVM はパラレル(並列)タイプの GC 処理を行っていることになります。そこで、GC の実行タイプをパラレルからシリアル(直列)に変更することでクラッシュを回避できる場合があります。

- ・ <INSTANCE_ROOT>/config ディレクトリ配下にクラッシュログが出力されていた場合
<INSTANCE_ROOT>/config ディレクトリ配下にある domain.xml ファイル内の次の設定箇所を編集します。

```
<java-config ...>  
  :  
  <jvm-options>-XX:+UseSerialGC</jvm-options>  
  :
```

<jvm-options> タグ内に、他の設定と同様の定義方法で <jvm-options>-XX:+UseSerialGC</jvm-options>という設定情報を追加し、ファイルを保存します。その後、対象となるドメインを起動します。

- ・ <INSTALL_ROOT>/Trnsv/bin ディレクトリ配下にクラッシュログが出力されていた場合
対象となるプロセスグループの設定を変更します。

- 「統合運用管理ツール」もしくは「Web 版統合運用管理コンソール」による変更方法
画面左の管理対象ツリーにおいて、「WebOTX 管理ドメイン」→「<対象のドメイン名>」→「アプリケーショングループ」→「<対象のアプリケーショングループ名>」→「プロセスグループ」→「<対象のプロセスグループ名>」と各ノードを辿り、画面右側の「JVM オプション」タブを選択後、表示される項目「その他の引数」にて -XX:+UseSerialGC を追加します。その後、「更新」ボタンを押下します。
- 「運用管理コマンド」による変更方法

対象のドメインにログインし、次のコマンドを実行します。

```
otxadmin> set tpsystem.applicationGroups.<アプリケーショングループ名>.  
processGroups.<プロセスグループ名>.otherArguments="<設定済みの引数>  
-XX¥:+UseSerialGC"
```

設定後、アプリケーショングループを再起動します。

■ケース 2(JVM 内のコンパイルスレッドによるクラッシュ発生)の一時的対処方法

ーJVM 実行タイプの切り替えー

デフォルトの状態では、HotSpot Server VM というサーバアプリケーションの実行に適したタイプの JVM を使用して動作しています。これを、HotSpot Client VM というタイプの JVM を使用して動作するように設定を変更します。

- ・ <INSTANCE_ROOT>/config ディレクトリ配下にクラッシュログが出力されていた場合
<INSTANCE_ROOT>/config ディレクトリ配下にある domain.xml ファイル内の次の設定箇所を編集します。

```
<java-config ...>  
  :  
  <jvm-options>-server</jvm-options>  
  :
```

既に設定済みの -server の部分を -client に変更し、ファイルを保存します。その後、対象となるドメインを起動します。

- ・ <INSTALL_ROOT>/Trnsv/bin ディレクトリ配下にクラッシュログが出力されていた場合
対象となるプロセスグループの設定を変更します。

➤ 「統合運用管理ツール」もしくは「Web 版統合運用管理コンソール」による変更方法

画面左の管理対象ツリーにおいて、「WebOTX 管理ドメイン」→「<対象のドメイン名>」→「アプリケーショングループ」→「<対象のアプリケーショングループ名>」→「プロセスグループ」→「<対象のプロセスグループ名>」と各ノードを辿り、画面右側の「JVM オプション」タブを選択後、表示される項目「その他の引数」にて -server を追加します。その後、「更新」ボタンを押下します。

➤ 「運用管理コマンド」による変更方法

対象のドメインにログインし、次のコマンドを実行します。

```
otxadmin> set tpsystem.applicationGroups.<アプリケーショングループ名>.  
processGroups.<プロセスグループ名>.otherArguments="<設定済みの引数>  
-client"
```

設定後、アプリケーショングループを再起動します。

ーコンパイル対象からの除外設定ー

クラッシュログをさらに追っていくと、次のような情報が記録されている場合があります。

```
----- T H R E A D -----  
:  
Current CompilerTask:  
C2:758    org.apache.catalina.core.StandardWrapperValve.createFilterChain(...)... (292 bytes)
```

※最後の行は発生の度に表示内容が異なります。

この場合、`org.apache.catalina.core.StandardWrapperValue` というクラスの `createFilterChain` メソッドに対する JIT(Just-In-Time)コンパイル処理が原因でクラッシュが発生したことになります。同様のクラッシュが発生し続ける場合に、同一クラスのメソッドがその原因となっている場合には、それを JIT コンパイル処理の対象外に指定します。

クラッシュログが出力されていたディレクトリ配下に `.hotspot_compiler` というファイルを作成し、そのファイル内に次のような形式で記述します。

```
exclude <完全修飾クラス名(" ." は "/" に変換)> <メソッド名>
```

具体的には、次のように記述します。

```
exclude org/apache/catalina/core/StandardWrapperValve createFilterChain
```

- `<INSTANCE_ROOT>/config` ディレクトリ配下にクラッシュログが出力されていた場合
 `.hotspot_compiler` ファイルを保存後、対象となるドメインを起動します。
- `<INSTALL_ROOT>/Trnsv/bin` ディレクトリ配下にクラッシュログが出力されていた場合
 `.hotspot_compiler` ファイルを保存後、対象となるプロセスグループを起動します。

2.3. 実行環境のリソース不足に伴うエラー

WebOTX が動作中のマシンにおけるリソースが不足すると、WebOTX の動作が不安定になることがあります。特に、メモリやディスク領域が不足するとそれが顕著になり、その状況で WebOTX の運用を継続しても、止むを得ず停止してしまうことがあります。リソース不足となっている原因を追究し、設定の見直しや実行環境の見直しを図ります。

2.3.1. メモリ不足

メモリの不足はシステム評価中や運用中に発生しやすい問題ですが、メモリ不足となっている原因を正しく理解しなければ、安易に物理メモリの増設などの対策を施しても効果がありません。ここでは、特に発生しやすいメモリ不足のケースを取り上げ、その対策について解説します。

2.3.1.1. OutOfMemoryError の発生

現状指定されている JVM のヒープサイズではシステムを安定して動作させるためには不十分である場合、あるいはアプリケーションロジックが原因でメモリ不足を引き起こす場合、それを示すメッセージがログとして出力されます。その場合、設定の見直しや、アプリケーションロジックの見直しが必要です。<INSTANCE_ROOT>/logs 配下の server.log と webotx_agent.log、加えて Foundation/Standard/Enterprise 環境である場合は、<INSTANCE_ROOT>/logs/tpssystem 配下のログファイルにおいて、システムの動作が不安定になった時間帯や、WebOTX が異常停止した時間の直前に OutOfMemoryError というキーワードが出力されていないかを確認します。もし出力されている場合は、単純にメモリの割り当て領域が不足しているのか、あるいはアプリケーションによるメモリの使い方に問題があるのかを調べ、メモリ不足を解消しなければなりません。

メモリ不足に関する問題解決には、次のような順で作業に取り掛かります。

1. メモリ不足となっている原因の特定

まずは、ログに出力されている OutOfMemoryError というキーワードの続きのメッセージを確認します。そこには、メモリ不足に関する具体的な内容が表示されていますので、そのメッセージを基に、次の表を参考にして対処します。

メッセージ	原因	対処方法
Java heap space	インスタンス領域が不足しています。	Java ヒープ領域を拡張します。
PermGen space	パーマネント領域が不足	パーマネント領域を拡張します。

	しています。	
Out of swap space?	スワップ領域が不足しています。	マシンのスワップ領域を増やすか、マシンのメモリを増設します。
unable to create new native thread	仮想メモリが不足しているため、新しいスレッドを生成できません。	仮想メモリを増やします。もしくは、実行中のスレッドの生成を少なくするようアプリケーションを見直します。 マシン OS が HP-UX の場合は、カーネルパラメータ max thread proc を変更します。

ここで、Java ヒープ領域もしくはパーマネント領域を拡張する場合、OutOfMemoryError ログの出力先により設定変更箇所が異なりますので注意が必要です。

➤ **server.log または webotx_agent.log に OutOfMemoryError が出力されていた場合**

<INSTANCE_ROOT>/configディレクトリ配下にあるdomain.xmlファイル内の次の設定箇所を編集します。

・ Javaヒープ領域の拡張

```
<java-config ... max-heap-size="512m" ...>
    :
```

・ パーマネント領域の拡張

```
<java-config ... max-perm-size="192m" ...>
    :
```

既に設定済みのmax-heap-sizeまたはmax-perm-sizeの属性を現在値よりも大きな値で設定を変更し、ファイルを保存します。その後、対象となるドメインを起動します。

なお、対象となるドメインが稼働中である場合には、次の箇所での設定項目を同様にして変更します。

■「**統合運用管理ツール**」もしくは「**Web版統合運用管理コンソール**」による変更方法

「WebOTX管理ドメイン」→「<対象のドメイン名>」→「アプリケーションサーバ」→「JVM構成」と各ノードを辿り、画面右側の「一般」タブを選択します。

・ Javaヒープ領域の拡張

画面に表示されている項目「**最大Javaヒープサイズ**」の設定値を変更します。

・ パーマネント領域の拡張

画面に表示されている項目「**最大Javaパーマネントサイズ**」の設定値を変更します。
その後、「**更新**」ボタンを押下します。設定の反映には、ドメインの再起動が必要です。

■「運用管理コマンド」による変更方法

対象のドメインにログインし、次のコマンドを実行します。

- ・ Javaヒープ領域の拡張

```
otxadmin> set server.java-config.max-heap-size=<変更後の値>
```

- ・ パーマネント領域の拡張

```
otxadmin> set server.java-config.max-perm-size=<変更後の値>
```

設定の反映には、ドメインの再起動が必要です。

➤ <INSTANCE_ROOT>/logs/tpsystem 配下のログに OutOfMemoryError が出力されていた場合

対象となるアプリケーショングループに対する次の箇所の設定項目を変更します。

■「統合運用管理ツール」もしくは「Web版統合運用管理コンソール」による変更方法

「WebOTX管理ドメイン」→「<対象のドメイン名>」→「アプリケーショングループ」→「<対象のアプリケーショングループ名>」→「プロセスグループ」→「<対象のプロセスグループ名>」と各ノードを辿り、画面右側の「JVMオプション」タブを選択します。

- ・ Javaヒープ領域の拡張

画面に表示されている項目「**最大ヒープサイズ**」の設定値を変更します。必要に応じて「**最大ヒープサイズの単位**」の設定値も変更します。

- ・ パーマネント領域の拡張

画面に表示されている項目「**その他の引数**」に、-XX:MaxPermSize=<変更後の値>を追加します。

設定後、「**更新**」ボタンを押下します。設定の反映には、アプリケーショングループの再起動が必要です。

■「運用管理コマンド」による変更方法

対象のドメインにログインし、次のコマンドを実行します。

- ・ Javaヒープ領域の拡張

```
otxadmin> set tpsystem.applicationGroups.<アプリケーショングループ名>.processGroups.<プロセスグループ名>.maxHeapSize=<変更後の値>
```

※必要に応じて、maxHeapSizeScaleの設定値も変更します。

- ・ パーマネント領域の拡張

```
otxadmin> set tpsystem.applicationGroups.<アプリケーショングループ名>.processGroups.<プロセスグループ名>.otherArguments="<設定済みの引数>
```

```
-XX¥:MaxPermSize=<設定値>"
```

設定の反映には、アプリケーショングループの再起動が必要です。

2. GC ログ解析

Java ヒープ領域やパーマネント領域の拡張を行っても **OutOfMemoryError** が再発する場合には、その背後にメモリリークの問題が潜んでいることが考えられます。あるいは、特定の処理で突如巨大な(あるいは大量の)オブジェクトが生成されることで、一時的なメモリ不足に陥っている可能性があります。それらを確認するためには、GC ログを出力するための設定を行い、その出力データを解析するのが有効な手段です。ここで、GC ログ設定を行う場合、**OutOfMemoryError** ログの出力先により設定変更箇所が異なりますので注意が必要です。設定方法などの詳細は、「WebOTX マニュアル運用編 運用と操作(ドメインの運用)」内の「GC ログ出力」を参照してください。

3. プロファイラツールの使用

2. の作業でメモリ上の問題点が見つかった場合には、プロファイラツールを使用してその原因となっているアプリケーションの実装箇所を特定します。WebOTX では、プロファイラツールとして「WebOTX プロファイラ」を提供していますので、こちらを利用して分析することが可能です。ツールの詳しい使い方は、「WebOTX マニュアル運用編 プロファイラツール」を参照してください。

2.3.1.2. ネイティブ領域によるメモリの圧迫

JVM が確保するメモリ領域のうち、ネイティブ領域(C ヒープ領域)については、ガベージコレクションの対象にはなりません。そのため、アプリケーションで JNI を経由したネイティブ呼び出しを行っている場合に、その部分でのメモリの解放漏れが潜在していると、メモリは次第に増加していきます。すると、Java プロセスは指定したヒープサイズよりかはるかに大きなメモリサイズに達することがあります。これがマシン全体のメモリ領域を圧迫する要因となり、結果的にシステムの動作が不安定になるなどの事象を引き起こします。

このようなケースでは、**OutOfMemoryError** が発生しないこともあるために直接異常に気付くことが困難です。評価段階でネイティブ呼び出し部分でのメモリの解放漏れを発見し、問題解決を行うのは勿論のこと、タスクマネージャや **ps** コマンドによってプロセスのメモリ状況を定期的に確認するか、プロセス監視ツールなどによってメモリサイズを監視の対象に含めるのが効果的です。

2.3.2. ディスク不足

WebOTX では、ドメインの起動完了直後に自動でドメイン環境(ファイル)のバックアッ

プを取る処理が行われます。しかし、徐々にシステムの開発規模が大きくなると、対象となるバックアップファイルも多くなるため、ディスク容量を圧迫する可能性があります。また、配備操作を実行中に、以下のようなメッセージがログに出力される場合があります。

ディスク使用率は<USED RATE>%です。ディスクの空き容量を増やすことを推奨します。

Disk used <USED RATE>%. It is recommended to free up some space.

このように、マシン上のディスク空き容量が不足してきた場合は、以下に示す対策を行います。

まず、<INSTANCE_ROOT>/backup ディレクトリ配下にある files1～files3 ディレクトリ内には、最大過去 3 回分のドメイン正常起動時の環境(構成ファイル)をバックアップしたものが含まれています(files1→files2→files3 の順で新しくなります)。ディスク容量が不足してきた場合は、これらのディレクトリを優先的に削除するようにしてください(files3→files2→files1 の順で削除してください)。

次に、バックアップ処理自体を抑制、あるいはバックアップ保存回数を減らすことを検討します。設定方法などの詳細は、「WebOTX **マニュアル運用編 運用と操作(ドメインの運用)**」内の「**バックアップ/リストア ドメイン環境の自動バックアップ**」を参照してください。

さらに、<INSTANCE_ROOT>/logs ディレクトリ配下にあるログファイルを整理します。特に、Foundation/Standard/Enterprise の場合、<INSTANCE_ROOT>/logs/tpssystem/<アプリケーショングループ名>/<プロセスグループ名>/save ディレクトリ配下には過去のプロセスグループの起動ログが蓄積されていますので、不要な場合は削除するなどの対策を行います。

これらの対策を行ってもディスクの空き容量が十分でない場合は、WebOTX 以外のディレクトリの整理を行ってください。

2.4. ストール現象

WebOTX の稼働中あるいは起動/停止処理中に何らかの影響を受けて処理が進まなくなり、ストール状態に陥ることがあります。そのような場合には、対象の Java プロセスに対してスレッドダンプの取得を試みます。スレッドダンプの取得方法については、「WebOTX **マニュアル運用編 運用と操作(ドメインの運用)**」内の JavaVM のスレッドダンプ取得」を参照してください。

取得したスレッドダンプの出力内容に関して、次の事象の有無を確認します。

■デッドロック

プロセス内でデッドロックが発生していると、出力内容に次のようなメッセージが表記されています。

```
Found one Java-level deadlock:
=====
"Thread-21":
  waiting to lock monitor 0x0c3ce154 (object 0x04487e78, a java.lang.Object),
  which is held by "Thread-20"
"Thread-20":
  waiting to lock monitor 0x0c3cdf94 (object 0x04487e80, a java.lang.Object),
  which is held by "Thread-21"
```

これらは、特定のスレッド間でデッドロックが発生していることを示しています。上記の例では、スレッド名 Thread-20、Thread-21 を持つスレッドがその対象です。上記メッセージの後にはさらにそれぞれのスレッドスタックが表記されていますので、それがアプリケーションコード呼び出しのスレッドである場合、スレッドスタックの内容を参考にアプリケーションロジックを修正します。

■無限ループ

出力されたスレッドのうち、特定のスレッドの状態が実行中(RUNNABLE)であるにも関わらず、特定のメソッドの呼び出しから状況が変化しない場合には、そのメソッド内のロジックに無限ループの問題を含んでいる可能性があります。あるいは、そのメソッドがネイティブメソッド(JNI 呼び出し)である場合、そのネイティブメソッド内部で同様の問題を含んでいる可能性があります。対象のメソッドがアプリケーションロジックである場合には、システム開発者がその部分のソースコードを参照してロジックの問題解決を進めます。

なお、デッドロックや無限ループの問題が発見された場合、対象となるプロセスを終了させます。それがプロセスグループの場合は、プロセスグループの強制停止を行います。強制停止は、プロセスグループのツリーノードを右クリックして「**プロセスグループの強制停止**」を行うか、次のコマンドを実行します。

```
otxadmin> stop-pg --force --apgroup <アプリケーショングループ名> <プロセスグループ名>
```

対象のプロセスがエージェントプロセスの場合は、ドメインの強制停止を実行します。

ドメインを強制停止するためには次のコマンドを実行します。

```
otxadmin> stop-domain --force <ドメイン名>
```

このコマンドを実行してもプロセスが削除されない場合は、関連する WebOTX のプロセスを一つずつ強制終了します。

2.5. 環境不正に伴う問題

システム構築中や運用中に OS レベルの異常が発生すると、その影響を受けて WebOTX の構成ファイル内のデータが破損してしまうことがあります。また、システム構築中には、運用者のミスにより WebOTX の構成ファイルを誤って削除してしまうことがあります。これらの事象は直接発見することが難しいため、次のような事象によって問題が浮上します。

- ・ 起動処理中のログに(普段は編集などを行わない)設定ファイルの読み込みエラーメッセージが多数出力される。
- ・ WebOTX の起動が完了しても、作成したアプリケーションの定義情報が統合運用管理ツールや運用管理コマンド上で表示されない。
- ・ 設定変更したはずの項目値がデフォルト値に戻っている。

このような場合、環境を正常な状態に復旧しなければせっかく作成した環境の使用を継続することができなくなりますので、早急に復旧作業を行います。

復旧のためのポイントは、ログから問題になっている設定ファイルを特定し、それらをバックアップファイルによって正常な状態に戻すことです。最新のバックアップファイルは、通常<INSTANCE_ROOT>/backupディレクトリ配下に格納されています。³

ただし、不正ファイルを全て特定することは難しい場合があります。WebOTX ではそのような場合に備え、前回正常に起動した状態に環境を自動的に復旧させるためのコマンドを提供しています。システム開発中に生じたこのような問題は、コマンドを使用することで素早い復旧が行えます。コマンドについては、**3.5.節**を参照してください。

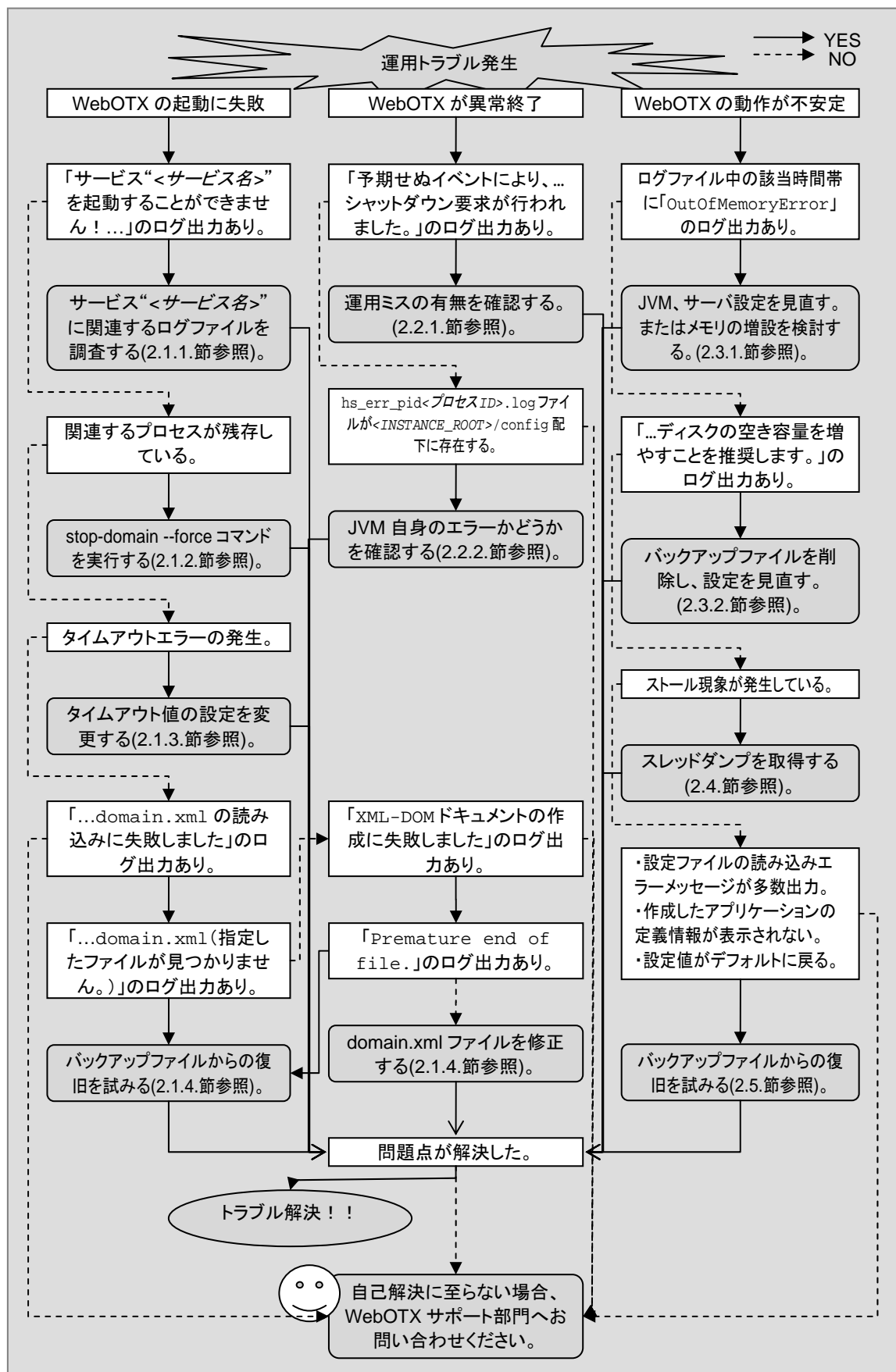
2.6. トラブル解決作業流れ図

これまでに紹介したトラブル事例に対して、運用者自ら問題解決が行えるよう、行うことを目的に、トラブル解決作業の流れ図を作成しました。運用トラブル時にはこの図に従

³ 自動バックアップ設定を行わないようにしている場合は、最新のバックアップファイルは存在しません。

って解析ポイントを押さえ、作業を進めることで、トラブルの早期解決につながります。

なお、この図はこれまでに当サポート部門に問い合わせがあった内容を基に作成しているため、必ずこの方法で問題が解決することではない点に注意してください。



3. トラブル回避に有用な機能のご紹介

前章では、トラブル発生時の解決方法について取り上げましたが、本章では、それらの問題を事前に検知、あるいは防止することが期待できる、WebOTX が提供している様々な機能やツールについてご紹介します。

3.1. 統計情報採取(モニタリング)、イベント通知

WebOTXではJMX (Java™ Management Extensions) およびJava™ 2 Platform, Enterprise Edition Management Specificationという仕様に準拠しており、WebOTX上で動作する全てのアプリケーションやリソース等を共通の手法で統合的に管理しています。また、それらの管理対象に対し、連続的なデータの採取や監視、監視データに基づくイベント通知が行えます。そして、それらの機能をGUIでより容易に活用できるよう提供しているのが「統合運用管理ツール」です。本ツールによって、前章で取り上げたようなトラブルが発生する前に検知することが可能になり、トラブル防止にも大きな効果を発揮します。そこで、本ツールを使用した幾つかの具体例を基に紹介します。

開発・運用時には、次のような項目を評価の対象あるいは監視の対象とすることが想定されます。

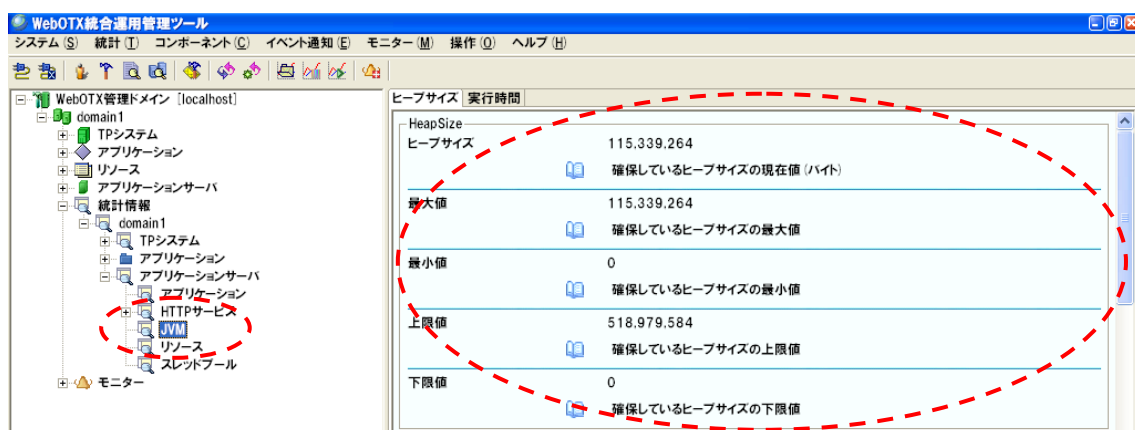
- ・ 開発中のアプリケーションを動作させた際の JVM ヒープ使用量
- ・ JDBC データソースにおける JDBC コネクション数
- ・ リクエストのキューイング状況
- ・ 接続クライアント数

これらの項目は、「統合運用管理ツール」や「運用管理コマンド」によって容易に情報を採取することが可能です。詳細については、「WebOTX マニュアル運用編 運用と操作(モニタリング)」を参照してください。


以下では、JVM ヒープ使用量を基に具体的なツールの活用例を示します。

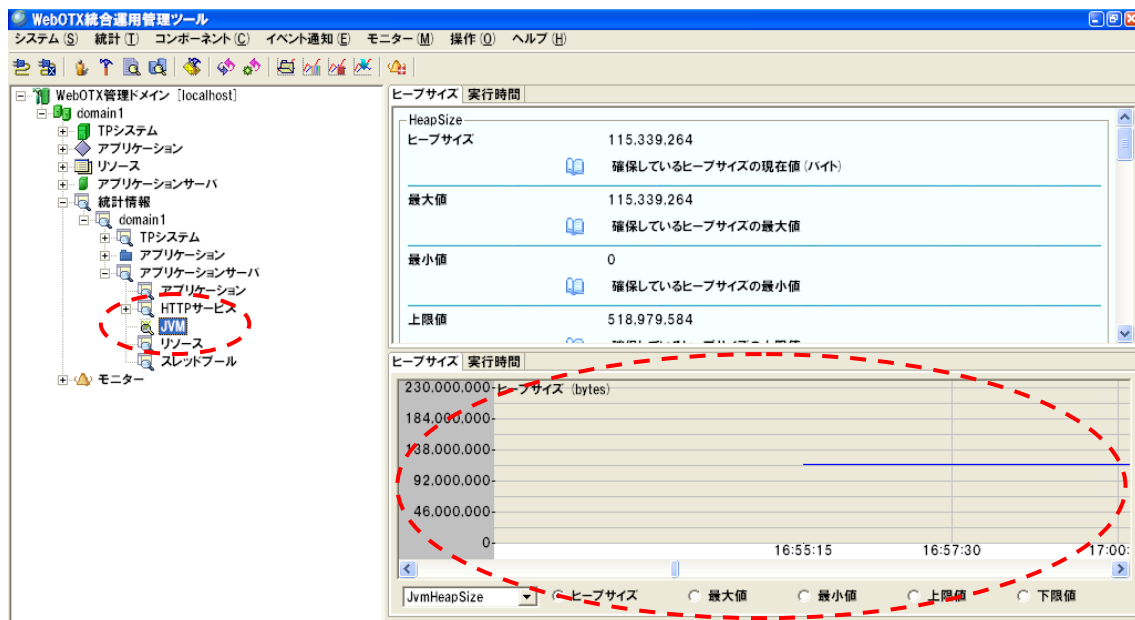
■統合運用管理ツールによる情報採取

画面左の管理対象ツリーにおいて、「WebOTX 管理ドメイン」→「<任意のドメイン名>」→「統計情報」→「<任意のドメイン名>」→「アプリケーションサーバ」と辿り、その配下の「JVM」ノードをクリックします。すると、画面右側に JVM に関する統計データが表示されます。



上記画面では、WebOTX のドメイン domain1 において動作しているアプリケーションサーバの JVM ヒープサイズに関する統計データを示しています。この場合、現在使用中のヒープサイズ値は 115,339,264 バイト (約 110M バイト)、最大 518,979,584 バイト (約 495M バイト) まで使用可能となります。

さらに、メニューバーの「統計(T)」→「統計情報採取開始(A)」(または「JVM」ツリーノードを右クリックして「統計情報採取開始(A)」)を選択、あるいはツールバーの  ボタンを押すと、画面右の中央部に統計データに基づくグラフが表示されます。



上記グラフは、統計を開始した時点から一定間隔で採取したデータを基に作成されたものです。このグラフは、ツール利用者が任意の時点で終了させるまで継続的に更新されますので、リアルタイムなデータの推移を確認することができます。


なお、統合運用管理ツールでは、採取したデータを規定のファイルに保存する機能が備

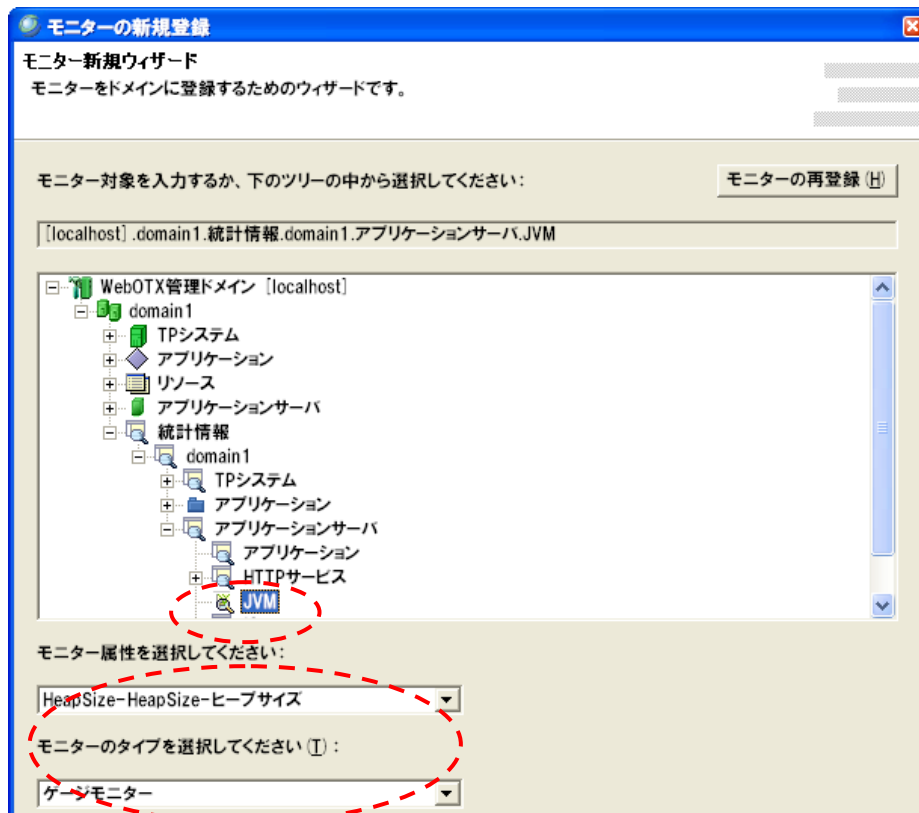
わっています。保存したファイルは後に同ツールに読み込ませることで、採取データやグラフを再表示することが可能です。詳しい手順については「WebOTX マニュアル運用編 統合運用管理ツール」を参照してください。

さらに、統合運用管理ツールではこれらの採取データに対して一定条件を予め指定しておき、その条件を満たした場合に画面上にイベントを通知させることが可能です。

ここでは、エージェントプロセスの JVM におけるヒープサイズが 300M バイトを超えた際にイベントを通知するようにするための方法について簡単に紹介します。

<イベント通知のための作業手順>

1. メニューバーより、「モニター(M)」→「モニター対象追加(R)」(または「WebOTX管理ドメイン」→「<ドメイン名>」と辿り、配下の「モニター」ツリーノードを右クリックして「モニター対象追加(R)」を選択)、あるいはツールバーの  ボタンを押下。
2. モニター登録ウィザード画面の管理対象ツリーにおいて、「WebOTX 管理ドメイン」→「<ドメイン名>」→「統計情報」→「<ドメイン名>」→「アプリケーションサーバ」と辿り、その配下の「JVM」ノードを選択。
3. 画面下の「モニター属性を選択して下さい(I):」において、モニタリングの種類の中から「ゲージモニター」を選択。その後、「次へ(N)」ボタンを押下。



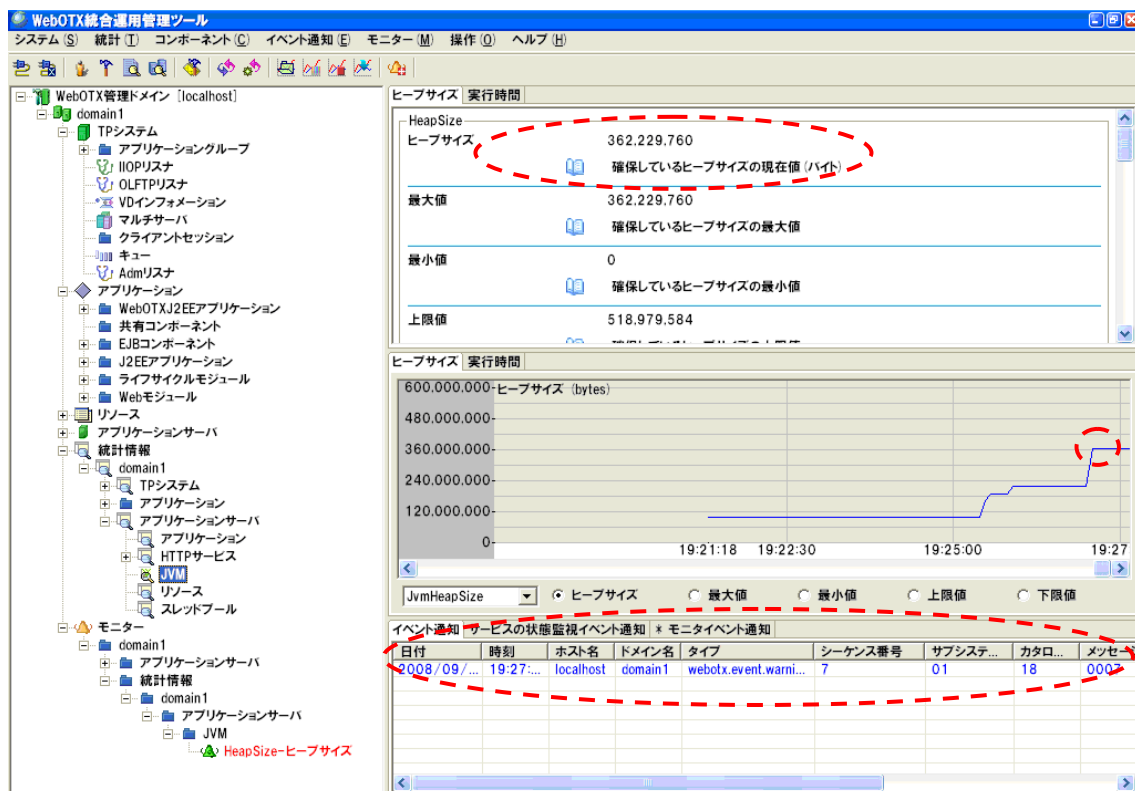
4. 詳細設定画面の「highThreshold(上限値)」項目において、イベントを通知するための

対象データ（ヒープサイズ）の上限値（ここでは 300M(=314572800)バイト）を入力。
「lowThreshold(下限値)」項目についても同様にして入力（ここでは 0 バイト）。また、
「notifyHigh(上 限 値 に 対 す る 通 知 の 有 無)」 項 目 を チェック。

5. 「登録(R)」ボタンを押下。「操作の実行に成功しました」というダイアログが表示されたら「OK」ボタンを押下。その後、「別のモニターを登録しますか?」というダイアログが表示されたら「いいえ」ボタンを押下。

作業完了後、「モニター」ツリーノード配下の「<ドメイン名>」→「統計情報」→「<ドメイン名>」→「アプリケーションサーバ」→「JVM」と辿り、その配下の「Heap-size(ヒープサイズ)」ノードをクリックすると、画面右側には先程設定したモニタリング設定項目が表示されます。

これで設定は完了です。この後、例えばアプリケーション評価中において、ヒープサイズが 300M(314572800)バイトを超えた際に、画面右側のイベント通知画面にメッセージが表示されます。



通知されたイベントには、イベントの発生時刻や対象属性などの詳細情報が表示されます。もし、イベントの発生時刻時間帯がアプリケーションの評価中であれば、そのアプリケーションが大量のメモリを使用したことになります。そのような状況を素早く検知するのに、本機能が効果を発揮します。

■運用管理コマンドによる情報採取(その1)

プロンプト画面上で次のコマンドを実行します。

```
otxadmin> get --monitor server.jvm.*
```

コマンドの実行結果は次のようになります。

```
server.jvm.HeapSize-Current = 65011712
server.jvm.HeapSize-HighWaterMark = 65011712
server.jvm.HeapSize-LowWaterMark = 0
```

```
server.jvm.HeapSize-LowerBound = 0
server.jvm.HeapSize-UpperBound = 518979584
:
```

ここで、文字列“server.jvm.HeapSize”は統合運用管理ツールでの例と同じ JVM ヒープサイズを表しており、その後のハイフン(-)に続く文字列は、JVM ヒープサイズに対する各統計項目を表します。以下にその対応関係を示します。

Current	: 現在値
HighWaterMark	: 最大値
LowWaterMark	: 最小値
UpperBound	: 上限値
LowerBound	: 下限値

例えば、一定の間隔で次のようなコマンドを実行するスクリプトを作成し、それを実行することで、実行期間中の JVM ヒープサイズの現在値の推移をファイルに出力することができます。

```
otxadmin get --user <ユーザ名> --password <パスワード> --host <ホスト名> --port <管理ポート番号> --monitor server.jvm.HeapSize-Current >> jvm_heapsize.log
```

■運用管理コマンドによる情報採取(その2)

プロンプト画面上で運用管理コマンドを実行し、対象のドメインにログイン後、次のコマンドを実行します。

```
monitor --type jvm --filename <ファイルパス> --interval <採取間隔(秒)> server
```

コマンドの実行結果は次のようになります。

```
JVM Monitoring
UpTime(ms)
current      min      max      low      high      count
1341562      0        518979584 0        57044992 57044992
1344383      0        518979584 0        57044992 57044992
1347383      0        518979584 0        57044992 57044992
:
```

実行結果は、`--interval` オプションで指定した間隔で定期的にプロンプト画面上に表形式で出力されます。さらに、この実行結果は`--filename` オプションで指定したファイルに、CSV 形式で出力されますので、出力内容をそのまま評価資料として活用することも可能です。

3.2. ログ、統計情報採取設定

システム開発・評価中には、以下に示すような WebOTX における特定のコンポーネントの動作状況を把握したい場合があります。

• Web/EJB コンテナ	
アプリケーションのロードや初期化状況	→ ログによる確認
アプリケーションの呼び出し回数、処理実行時間	→ 統計情報による確認
• JDBC データソース	
データソースのロードやコネクションプールの初期化状況	→ ログによる確認
コネクション数	→ 統計情報による確認
• トランザクションサービス	
サービスの初期化状況	→ ログによる確認
トランザクションのコミット/ロールバック数の把握	→ 統計情報による確認

WebOTX ではこのようなケースに対し、そのコンポーネント単位でのログ出力や、前節で紹介した統計情報の採取設定を行うことが可能です。設定は特定のコンポーネントのみに反映されるため、他のコンポーネントへの影響はありません。かつ、設定の反映には WebOTX の再起動が不要なため、採取したい情報をすぐに得られるようになります。

これらの設定についての詳細は、それぞれ以下のマニュアルを参照してください。

ログに関する設定	: 「WebOTX マニュアル運用編 運用と操作(ログ)」
統計情報に関する設定	: 「WebOTX マニュアル運用編 運用と操作(モニタリング)」

3.3. JVM監視・管理ツール

WebOTXには、WebOTXの実行環境内で動作するJVMの実行情報をリアルタイムに監視・管理するための機能や、外部監視・管理ツールと連携するための機能を提供しています。

3.3.1. JVM動作情報出力コマンド

システム評価時または運用中におけるJVMのパフォーマンス状況を把握できるよう、WebOTX内で動作するJVMに対する詳細な動作情報をレポート形式で出力するためのコマンドを提供しています。

このコマンド実行により、JVMに対する次のような情報をテキストベースで得ることができます。

- JVMが動作するOS情報、クラスパスやシステムプロパティの設定状況
- Javaヒープメモリの使用状況、GC実行状況
- クラスのロード/アンロード状況、コンパイル実行状況
- スレッドダンプ、デッドロック発生の有無

このコマンドは、業務アプリケーションに対して殆ど影響を与えることなく、任意のタイミングで実行することができます。そのため、稼働中のシステムに対するJavaヒープメモリの使用状況や、GCの実行状況を収集するのに役立ちます。また、それらのレポート結果を、JVMに対するチューニングの参考資料として使用することも可能です。

本コマンドに関する詳細は、「WebOTX マニュアル運用編 運用と操作(ドメインの運用)」内の「Javaプロセスの監視・管理」を参照してください。

3.3.2. jconsole接続

WebOTXの運用管理基盤は、JDKに付属のJVM監視ツールであるjconsoleをWebOTX内で動作するJVMに対して接続することができるように構築されています。そのため、jconsoleが持つGUIや監視機能をそのまま使用して実行データを監視・管理することができます。

jconsole接続に関する詳細は、「WebOTX マニュアル運用編 運用と操作(ドメインの運用)」内の「Javaプロセスの監視・管理」を参照してください。

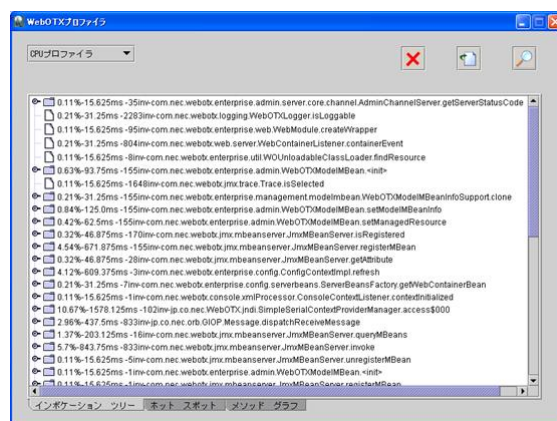


3.4. WebOTXプロファイラ(性能分析ツール)

業務システムにおいてメモリーリークの疑いがある場合や、アプリケーションのパフォーマンスに問題がある場合、その特定のために「WebOTX プロファイラ」が役立ちます。本ツ

ールは GUI であるため、分析が容易に行えます。また、測定情報をファイルに出力することもできるため、性能分析資料の作成にも役立ちます。

ツールについての詳細は、「WebOTX マニュアル運用編 プロファイラツール」を参照してください。



3.5. 実行環境リカバリコマンド

システム開発中の運用ミスなどによって WebOTX のドメイン環境が不正になり、その復旧が困難になった場合に備え、WebOTX が自動生成するバックアップファイルから前回ドメインが正常に起動した状態に環境を復旧するためのコマンドを提供しています。

復旧対象のファイルや復旧時の注意事項など、詳しくは、「WebOTX マニュアル運用編 運用と操作(ドメインの運用)」内の「バックアップ/リストア 自動バックアップからの復元」を参照してください。

3.6. 設定項目の比較

同一環境のサーバ構成する際や、評価環境から本番環境への移行の際に、各サーバ間での WebOTX に関する設定値に差異が生じていたことに作業者が気付かなかったなどの理由で、新たな環境において予期せぬエラーが発生し、その解決に思わぬ工数を費やしてしまうことがあります。そのような事態を避けるために、各サーバ間の設定情報を比較するための方法を紹介します。

対象となるサーバに対して、コマンドのプロンプト画面より、WebOTX が提供する運用管理コマンドを利用して次のように実行します(以下は事前に運用管理コマンドの実行ディレクトリである<INSTALL_ROOT>/bin ディレクトリに移動した状態での実行結果です)。

```
otxadmin get --user <ユーザ名> --password <パスワード> --host <ホスト名> --port <管理ポート番号> "*" > <任意ファイル名>
```

上記コマンドを実行後、それぞれの出力ファイル内のデータを、比較ツールなどを使用して比較します。この時、サーバ間での設定情報に差分が生じている場合は、その差分が妥当なものであるかを確認します。ホスト名などの環境依存項目については大抵の場合問題にはなりませんが、JVM ヒープサイズの設定や Web サーバ同時接続数などの設定に差分が生じていた場合には、この設定の差異がサーバ間での動作の違いに影響していた可能性

が高くなります。

例えば、次のような設定差異があったとします。

[比較元] server.java-config.max-heap-size=1024m

[比較先] server.java-config.max-heap-size=512m

※それぞれ、ドメインプロセスに対する「最大 Java ヒープサイズ」の設定です。

この場合、「比較元」が評価環境、「比較先」が本番環境であったとすると、本番環境ではシステムが想定している Java ヒープサイズよりも小さいために、WebOTX 起動中に OutOfMemoryError が発生するなどのトラブルが想定されます。

別の例として、次のような設定差異があったとします。

[比較元] server.WebServer.MaxClients=200

[比較先] server.WebServer.MaxClients=100

※それぞれ、Web サーバに対する「最大同時接続数」の設定です。

この場合、「比較元」が評価環境、「比較先」が本番環境であったとすると、本番環境ではシステムが想定している最大同時接続数よりも小さいために、評価時に比べて多数のクライアントからの処理要求に対する実行結果が異なってしまうなどのトラブルが想定されます。

なお、設定情報を比較して差分が生じた場合、その項目が具体的に何を示すものかわからない場合には、「WebOTX マニュアル運用編 リファレンス(MO 定義リファレンス)」を参照してください。

このような比較作業を事前に行うことで、思わぬトラブルを未然に防ぐことができますので、是非活用してください。

3.7. 診断サービス(実行情報、設定ファイル、ログの一括採取)

WebOTX では、アプリケーションサーバ内の動作情報や実行ログ情報などを、単一の操作で自動収集する機能を提供しています。これは、次のような場面で大いに役立ちます。

- 評価中に性能測定やエラー発生の有無といった検証などの目的で、定期的にこれらの

情報を取得したい。

- 運用中のログ採取や実行情報採取作業を効率的に行いたい。
- 障害発生時の調査用情報一式を特定の場所に保存しておきたい。

取得する情報の種類などは統合運用管理ツールや運用管理コマンド上で細かくカスタマイズできるため、運用・保守担当者は調査に必要な情報を容易に、かつ確実に収集することができるようになり、作業の効率化が図れます。

機能の詳細は、「WebOTX **マニュアル運用編 運用と操作(診断サービス)**」を参照してください。

4. おわりに

WebOTX の運用管理基盤は、WebOTX で動作する様々な管理対象に対する情報の管理や監視、イベント処理の統一化を実現しています。また、それによって、運用者が遭遇しがちなトラブルにも的確に対処、あるいは事前に防止できるような利便性の高い運用機能を提供しています。

そして、本資料では、それらの機能が実際にどのようなケースで活用することができるのか、さまざまな運用トラブルの事例を取り上げながらわかりやすくまとめました。

常に最新の技術動向を調査し、市場のニーズを満たす機能を製品に順次導入している WebOTX ですが、このように製品開発だけでなく、製品を購入して頂いたお客様のことを第一に考え、トラブル解決方法や事前防止策についても力を入れて取り組んでいます。本資料は、その一環として少しでもお客様の作業に（WebOTX サポート部門への問い合わせ等の）時間的ロスを発生させないよう検討し、作成したものです。本資料により、お客様のシステム運用に対する作業効率が少しでも向上すれば幸いです。