



WebOTX AS Web コンテナ チューニングとトラブルシューティング

WebOTX V6、V7 編

2007.7.4
第 3.1 版

NEC
第二システムソフトウェア事業部

前書き

【本書の位置付け】

本書は、WebOTX V6、V7 を利用して構築したシステムにおける Web コンテナの性能チューニングとトラブルシューティングに関するガイドブックです。

改版履歴

版数	年月日	改訂内容	備考
1.0	2006/06/14	第 1.0 版	
2.0	2006/07/10	資料の適用範囲を V6 全体に変更	
		2.Web コンテナのチューニング Web サーバとの組み合わせの説明を追加	
		2.1.2 メモリ使用量 Permanent 領域の拡大方法を追加	
		3.2.1. OutOfMemory への対応 復旧方法で HP-UX のプロセス毎のスレッド数、Linux でのスレッド数に関する概要を追加 -Xrunhprof オプション指定時にはパフォーマンスが低下することを追加 プロファイル情報の解析に HPjmeter が利用できることを追加 Permanent 領域の拡大に関する記述を追加	
		3.2.2. 異常終了への対応 HotSpot コンパイルのエラー調査で、HotSpot コンパイルのエラー以外が発生した場合には、HotSpot コンパイルのエラー対応では解決しないことを追加	
		3.2.3. 無応答障害（デッドロック）への対応 スレッドダンプでのデッドロックの表示例を追加	
3.0	2006/11/28	3.障害対策 を 3.トラブルシューティングに変更	
		3.1. Web コンテナでの障害と調査 Web アプリケーションで取得したパラメータで文字化けが発生する場合の対応を追加	
		3.1. Web コンテナでの障害と調査 Web アプリケーション配備、配備解除でエラーが発生する場合の対応を追加	
3.1	2007/7/4	対象に V7.1 を追加	
		3.1. Web コンテナでの障害と調査 Web アプリケーションで ClassNotFoundException, NoClassDefFoundError が発生する場合の対処を追加	

目次

1. はじめに	1
2. チューニング	2
2.1. Java VMへの割り当てメモリ	2
2.1.1. メモリ使用量の確認	3
2.1.2. メモリ使用量	3
2.2. プロセッサ数	4
2.2.1. 最大プロセッサ数	4
2.3. 接続バックログ数	4
2.3.1. 接続バックログ数	5
2.4. プラグインのリクエスト処理数	5
2.4.1. 最大同時リクエスト処理数	5
3. トラブルシューティング	7
3.1. Webコンテナでの障害と調査	7
3.1.1. WebサーバにはアクセスできるがWebコンテナにアクセスできない	7
3.1.2. クライアントのWebブラウザに応答が返らなくなった	8
3.1.3. レスポンスが低下する	8
3.1.4. セキュリティ例外が発生する	8
3.1.5. Webアプリケーションの表示で文字化けが発生する	8
3.1.6. Webアプリケーションで取得したパラメータで文字化けが発生する	9
3.1.7. WebアプリケーションでClassNotFoundException、NoClassDefFoundErrorが発生する	10
3.1.8. 配備でエラーが発生する	11
3.1.9. 配備解除でエラーが発生する	12
3.2. Webコンテナでの障害対策	12
3.2.1. OutOfMemoryErrorへの対応	12
3.2.2. 異常終了への対応	13
3.2.3. 無応答障害(デッドロック)への対応	14
3.2.4. セキュリティ例外への対応	15
3.3. JavaVMのスレッドダンプ取得	16
3.3.1. UNIXのスレッドダンプ	16
3.3.2. Windowsのスレッドダンプ	16

1.はじめに

Web システム構築において、Web コンテナの性能を発揮させるためのチューニングは重要です。本書では Web コンテナの性能をチューニングするための具体的な設定項目や設定方法について記載しています。

また、Web コンテナのトラブルシューティングについて、障害対応のための具体的な方法について記載しています。

WebOTX を利用した Web システム構築のためにぜひご活用ください。

本ガイドは皆様のご意見を反映させ、随時改版していく予定です。

本書での表記について

とくに記述していない場合、ディレクトリの表記は、UNIX OS での表記となっております。Windows をお使いの場合、Windows でのディレクトリの表記に読み替えをお願い致します。

2. チューニング

Web コンテナのチューニングについて説明します。

Web システムで Web コンテナを利用するときに、システムのリソースを有効に利用し、できる限り良いレスポンスを得るためには、そのシステムに合わせたチューニングが必要です。デフォルトの状態では、大量のメモリを必要とするような Web アプリケーションを同時に複数実行するような状況で、メモリが不足することがあります。また、同時にアクセスするクライアントの数が非常に多いシステムでは、リクエストが同時に処理されずに処理の待ち時間が長くなり、Web ブラウザにエラーが返ることがあります。

Web コンテナのチューニングでは、Web アプリケーションが必要とするメモリが確保できること、システムで想定される同時処理数を処理できることがポイントとなります。

Web コンテナのチューニング項目として次の内容を説明しています。

WebOTX Web Server や Apache などの外部 Web サーバを利用している場合には、「プロセッサ数」、「プラグインのリクエスト処理数」の設定を行います。WebOTX の内蔵 Web サーバを利用している場合は、「プロセッサ数」、「接続バックログ数」の設定を行います。

- Java VM への割り当てメモリ
Web アプリケーションが正常に動作するためには、適切なメモリ量が必要です。システム毎に必要なメモリ量を確認し、割り当てを行います。
- プロセッサ数
プロセッサ数は、Web コンテナでのリクエスト同時処理数です。システムで想定される同時リクエスト数を元に設定します。
WebOTX Web Server や Apache などの外部 Web サーバを利用している場合、WebOTX の内蔵 Web サーバを利用している場合ともに設定を行います。
- 接続バックログ数
接続バックログ数は、Web コンテナに対する接続の待機キューの数になります。Web コンテナで他のリクエストが処理されているときに、処理を待ち合わせるリクエストの数です。
WebOTX の内蔵 Web サーバを利用している場合のみ有効です。
- プラグインのリクエスト処理数
Sun Java System Web Server などの外部 Web サーバと連携した場合に、Web コンテナと連携するために、Web サーバ上で動作するプラグインの同時リクエスト処理数になります。WebOTX Web Server や Apache などの外部 Web サーバの同時リクエスト処理数を元に設定します。
WebOTX Web Server や Apache などの外部 Web サーバを利用している場合に設定を行います。

2.1. Java VM への割り当てメモリ

Web コンテナが動作する Java VM には、Web アプリケーションが動作するためのメモリも含めて、必要なメモリを割り当てる必要があります。

メモリ使用量を確認して、必要なメモリを割り当てます。

システムで想定される負荷をかけた上で Java VM の使用メモリ量の確認を行って、使用する可能性のあるメモリ量を Java VM へ割り当てます。Java VM への割り当てメモリ量の変更後、再度負荷をかけてメモリの使用量を確認しながら調整を行います。

WebOTX Standard Edition / Enterprise Edition Ver6.5 以降で、Web コンテナをマルチプロセス動作させている場合には、「Java VM への割り当てメモリ量 × Web コンテナを動作させるプロセス数」分のメモリがオペレーティング・システムに必要です。

2.1.1. メモリ使用量の確認

必要なメモリ量を知るには、Web アプリケーションを配備し、必要な負荷を与えた上で、メモリの使用状況を確認します。メモリの使用状況は、運用管理コンソール (<http://Webサーバ名:ポート番号/admin/> ポート番号はデフォルトでインストールされた domain1 の場合は 4848 です) の「統計情報」-「JVM の統計情報」で確認できます。メモリの使用状況は、「JVM の統計情報」の「使用メモリ量」で確認します。使用状況によって、メモリの使用量を確認し、利用する可能性のあるメモリ量を Java VM へ割り当てます。

Java VM のヒープメモリに空きがあるにもかかわらず、“OutOfMemory”が発生する場合には、Permanent 領域の不足が考えられますので、Permanent 領域を拡大します。Permanent 領域は、Java のクラスやメソッドの情報が格納される領域で、アプリケーションで多くのオブジェクトを生成すると、不足することがあります。

評価のために負荷を与えた結果、“OutOfMemory”が発生する場合には、割り当てメモリ量が不足していますので、適宜、例えば 128MB くらいメモリ割当量を増やして、再評価してメモリ使用状況を確認してください。

2.1.2. メモリ使用量

Web コンテナの使用メモリ量を変更するには、Java VM のオプション指定を変更します。メモリ使用量の確認をおこなって、利用するメモリを算出して設定します。

otxadmin コマンドを使用して次のコマンドを実行します。

デフォルトのメモリ指定を削除します。

```
otxadmin> delete-jvm-options --user <ユーザ名> --password <パスワード>
--host <ホスト名> --port <管理ポート>
-Xmx512m:
```

新しいメモリ指定を設定します。次のコマンドは、割り当てメモリの最大値を 640 MB にする例です。

```
otxadmin> create-jvm-options --user <ユーザ名> --password <パスワード>
--host <ホスト名> --port <管理ポート>
-Xmx640m:
```

デフォルトのメモリ指定を削除します。

```
otxadmin> delete-jvm-options --user <ユーザ名> --password <パスワード>
--host <ホスト名> --port <管理ポート>
-Xmx512m:
```

Permanent 領域を拡大する場合には、次のようにコマンドを実行します。次のコマンドは、Permanent 領域の最大値を 128 MB にする例です。

```
otxadmin> create-jvm-options --user <ユーザ名> --password <パスワード>
--host <ホスト名> --port <管理ポート>
create-jvm-options -XX¥¥:MaxPermSize=128m:
```

2.2. プロセッサ数

Web コンテナには、Web ブラウザからのリクエストを処理するプロセッサが存在し、リクエストが来るとスレッドが割り当てられ、プロセッサがリクエスト処理を実行します。

WebOTX Web Server や Apache などの外部 Web サーバを利用している場合、WebOTX の内蔵 Web サーバを利用している場合ともに設定を行います。

同時に複数のリクエストが来た場合それだけプロセッサを消費しますが、その数を調整することで、同時に処理できるリクエストの数を制限したり、拡大したりすることができます。

システムで想定される同時リクエストの発生数を最大プロセッサ数として設定することができますが、同時に処理できるリクエストの数が増える分、CPU リソースを消費します。

CPU 使用率を抑えたい場合には、プロセッサ数を減らす調整をしてください。

WebOTX Web サーバなどの外部 Web サーバを利用する場合には、外部 Web サーバ側の同時リクエスト処理数 (MaxClients 等) より、大きい値を設定してください。

同時処理リクエスト数を調整したい場合には、WebOTX Web サーバなどの外部 Web サーバの同時リクエスト処理数もあわせて調整してください。

2.2.1. 最大プロセッサ数

同時に処理できるリクエストの数を拡大するには、最大プロセッサ数の値を変更します。

otxadmin コマンドを使用して次のコマンドを実行します。

```
otxadmin> set --user <ユーザ名> --password <パスワード>
--host <ホスト名> --port <管理ポート>
server.http-service.http-listener.<リスナ ID>.max-processors=<最大数>
```

get コマンドを利用すれば、現在の値を確認できます。

```
otxadmin> get --user <ユーザ名> --password <パスワード>
--host <ホスト名> --port <管理ポート>
server.http-service.http-listener.<リスナ ID>.*
```

なお、標準では以下のリスナ (リスナ ID) が定義されています。

- 外部 Web サーバと連携している場合
ajp-listener-1
- 内蔵 Web サーバを利用している場合
http-listener-1, http-listener-2

2.3. 接続バックログ数

Web コンテナで、Web ブラウザからリクエストを受け付けるときに、受け付けられないリクエストは、ソケットで待ち合わせます。この待ち合わせのキューの数が、接続バックログ数になります。

WebOTX の内蔵 Web サーバを利用している場合のみ有効です。

「connection refused」で、接続ができなくなる現象が発生した場合に、待ち合わせを行ってリクエストを実行したい場合には、接続バックログ数を拡大することで対処できます。接続バックログを増やす場合には、現状の値より 25% 程度ずつくらい増やしながら、「connection refused」の発生状況を確認して、値の調整を行ってください。

接続バックログ数の最大値は、システムの設定によります。

この接続バックログ数は、WebOTX Web Server や Apache などの Web サーバを利用しているときには意味を持ちません。WebOTX Web Server や Apache などの Web サーバを利用しているときには、Web サーバの接続バックログに相当する値を設定してください。

2.3.1. 接続バックログ数

接続バックログ数を拡大するには、otxadmin コマンドを使用して次のコマンドを実行します。

```
otxadmin> set --user <ユーザ名> --password <パスワード>
               --host <ホスト名> --port <管理ポート>
               server.http-service.http-listener.<リスナ ID>.accept-count=<バックログ数>
```

get コマンドを利用すれば、現在の値を確認できます。

```
otxadmin> get --user <ユーザ名> --password <パスワード>
               --host <ホスト名> --port <管理ポート>
               --visibility=2 server.http-service.http-listener.<リスナ ID>.*
```

なお、標準では以下のリスナ(リスナ ID)が定義されています。

- 内蔵 Web サーバを利用している場合
http-listener-1, http-listener-2

2.4. プラグインのリクエスト処理数

ドメインを Apache などの外部 Web サーバと連携した場合、Web サーバとドメインを接続するプラグインというモジュールが存在します。

Web サーバやドメインでリクエストを処理するプロセス(スレッド)が調整可能であるように、プラグインでもリクエスト処理数を調整することができます。これにより、Web サーバと Web コンテナ間の通信処理を効率化し、処理性能を向上させることができます。

プラグインの処理リクエスト数には、WebOTX Web サーバなどの外部 Web サーバ側の同時リクエスト処理数(MaxClients 等)を設定してください。

WebOTX がサポートする以下の Web サーバに対して、設定してください。

- WebOTX Web Server 1.3(Apache 1.3)
(UNIX 上で利用する場合、スレッドは使われないため、設定は不要です)
- WebOTX Web Server 2.0(Apache 2.0)
(設定しない状態では、自動設定になります)
- Internet Information Services(IIS) 5.0、6.0
- Sun Java System Web Server 6.1、Sun ONE Web Server 6.0 以上

2.4.1. 最大同時リクエスト処理数

同時に処理できるリクエストの数を拡大するには、cachesize の値を変更します。

cachesize は、ドメインディレクトリの config/WebCont/workers.properties に定義します。このファイルをエディタで開いて編集してください。

具体的には、次のようになります。“worker.apache”については通常固定と考えてください(プラグインの負荷分散機能を利用した場合は可変となります)。Windows 版では、デフォルト値を 150 で設定しています。UNIX 版では、デフォルトでの設定はありません。

```
worker.apache.cacheSize=150
```

値を反映するには Web サーバを再起動する必要があります。

3.トラブルシューティング

3.1. Web コンテナでの障害と調査

Web コンテナに関連する障害とその原因の調査の概略について説明します。

障害は、その現象から大きく次のように分けることができます。

- Web サーバにはアクセスできるが Web コンテナにアクセスできない
- クライアントの Web ブラウザに応答が返らなくなった
- レスポンスが低下する
- Web アプリケーションの実行結果異常
 - セキュリティ例外が発生する
 - Web アプリケーションの表示で文字化けが発生する
 - Web アプリケーションで取得したパラメータで文字化けが発生する
 - Web アプリケーションで `ClassNotFoundException` が発生する
- Web アプリケーションの配備、配備解除のエラー
 - 配備でエラーが発生する
 - 配備解除でエラーが発生する

WebOTX でよく発生する障害としては、「クライアントの Web ブラウザに応答が返らなくなった」現象として、Java VM が異常終了していた、というのがあります。これは、Java VM の HotSpot コンパイルの問題か、メモリなどの資源不足が多くの場合原因となっています。

また、他によく起こる障害としては、「セキュリティ例外が発生する」です。これは、WebOTX では、セキュリティ強化のために、Security Manager がデフォルトで有効になっているためで、セキュリティのポリシー定義の追加が必要になります。

これらの対処方法についても、本書に記述していますので、参考にしてください。

それぞれの調査、対処方法について、次に説明します。

3.1.1. Web サーバにはアクセスできるが Web コンテナにアクセスできない

Web サーバにはアクセスできるが Web コンテナにアクセスできない場合、次のステップで原因の調査をしてください。

- (1) Web コンテナの環境設定の実施を確認します。
WebOTX Web Server や Apache などと連携するには、環境設定が必要です。Windows では、WebOTX のメニューから「環境設定ツール」を実行してください。UNIX では、環境設定ツールとして `setconf.sh` を実行してください。
- (2) Web サーバと Web コンテナの連携ポートを確認します。
環境設定ツールで設定した連携ポートで通信が可能かを確認します。`netstat` コマンドで、該当するポートを他のプロセスが使用していないかを確認してください。

3.1.2. クライアントの Web ブラウザに応答が返らなくなった

応答が返らなくなった場合、次のステップで原因の調査をしてください。

- (1) Webコンテナ(WebOTX のドメイン)が動作するJava VM のプロセスが動作しているかを確認します。
Java VM のプロセスが動作していない場合には、なんらかの原因により Java VM のプロセスが異常終了していることが考えられます。coreファイルなどのプロセスのダンプファイル、hs_err_pidxxx.logファイルが出力されていれば、これらを調査します。
原因としては、メモリ不足や Java VM のHotSpotコンパイルのエラーが考えられます。ログに”OutOfMemory”が出力されている場合には、メモリ不足ですので Java VM へのメモリの割り当てを増やすことを検討してください。詳細は、「3.2.1OutOfMemoryErrorへの対応」、「3.2.2異常終了への対応」を参照してください。
- (2) Webコンテナ(WebOTX のドメイン)が動作するJava VM のプロセスが動作している場合は、Webコンテナの動作状況をスレッドダンプにより確認します。
UNIXであれば、WebOTXのドメインのプロセスに対して、「kill -3 <pid>」を実行することにより、スレッドダンプがログに出力できます。スレッドダンプの取得の詳細は、「3.3JavaVMのスレッドダンプ取得」を参照してください。
スレッドダンプを複数回取得し、その結果から、資源に対するデッドロックが起こっていないか、同じ処理がループしていないかを確認します。「3.2.3無応答障害(デッドロック)への対応」を参考にしてください。
解析が困難であれば、開発元へお問い合わせください。

3.1.3. レスポンスが低下する

レスポンスが低下する場合、次のステップで原因の調査をしてください。

- (1) Webコンテナ(WebOTX のドメイン)が動作するJava VM のプロセスが動作している場合は、Webコンテナの動作状況をスレッドダンプにより確認します。
スレッドダンプの取り方は、「3.3JavaVMのスレッドダンプ取得」を参照してください。
スレッドダンプを複数回取得し、その結果から、スレッドの動作状態を確認します。
例えば、JDBCで処理中のスレッドが多ければ、DBからのレスポンスに問題があると考えられます。
解析が困難であれば、開発元へお問い合わせください。

3.1.4. セキュリティ例外が発生する

セキュリティ例外が発生する原因としては、WebOTXでは、セキュリティ強化のために、Security Managerがデフォルトで有効になっているためです。対処するには、セキュリティのポリシー定義の追加が必要です。ログなどに出力されているセキュリティエラーを元に、セキュリティポリシーを追加します。詳細は、「3.2.4セキュリティ例外への対応」を参照して対処をしてください。

3.1.5. Web アプリケーションの表示で文字化けが発生する

Web アプリケーションの表示で文字化けが発生する場合、次のステップで原因の調査と対処をしてください。

- (1) Servlet で、Content-Type による charset を正しく指定(出力)しているか確認します。
Web ブラウザが表示する文字コードを判断するのに Content-Type を使用します。Servlet が出力する文字のコードを Content-Type による charset で正しく指定する必要があります。
機種依存文字などを出力する際には、charset に”Windows-31J”を指定します。
- (2) JSP で、page ディレクティブで、contentType を正しく指定しているか確認します。
Servlet と同様に、出力する文字コードを contentType で指定します。

Servlet と同様、機種依存文字などを出力する際には、charset に"Windows-31J"を指定します。

- (3) JSP で、page ディレクティブで pageEncoding を正しく指定しているかを確認します。
pageEncoding は、JSP 自体がどのコードで記述されているかを示します。
個々の JSP では pageEncoding を指定せずに、まとめて指定することもできます。web.xml で次のように<page-encoding> を指定します。

```
———— web.xml ————

<web-app ....

  <jsp-config>

    <jsp-property-group>

      ....

      <page-encoding>Windows-31J</page-encoding>

    </jsp-property-group>

  </jsp-config>

  ....

  <servlet>

    ....

  </servlet>

</web-app>
```

3.1.6. Web アプリケーションで取得したパラメータで文字化けが発生する

Web アプリケーションの GET メソッド中で取得したパラメータで文字化けが発生する場合、次のいずれかの対策が必要です。

- (1) WebOTX でエンコードの指定を行う。
WebOTX にエンコード指定の設定を行うことで、コードを変換します。パラメータを一律同じ文字コードに変換する場合に利用できます。WebOTX V6.31 より利用可能です。
エンコードの指定は、otxadmin コマンドを用いて次のように設定することができます。

```
otxadmin> set --user <ユーザ名> --password <パスワード>
               --host <ホスト名> --port <管理ポート>
               server.http-service.http-listener.<リスナ ID>.property.uri-encoding
               =<文字エンコード>
```

文字エンコードには、"Windows-31J" などが指定できます。

- (2) SetCharacterEncoding() で文字エンコードを指定する。
Web アプリケーションで、SetCharacterEncoding() で文字エンコードを指定して、パラメータのコードを変換します。WebOTX V6.31 より利用可能です。
SetCharacterEncoding() で文字エンコードを指定するには、otxadmin コマンドを用いて次のように設定する必要があります。

```
otxadmin> set --user <ユーザ名> --password <パスワード>
--host <ホスト名> --port <管理ポート>
server.http-service.http-listener.<リスナ ID>.property.use-body-encoding-for-uri
=true
```

(3) フィルターを利用する。

Web アプリケーションのフィルターで、パラメータをエンコードする方式です。フィルターの処理によって、パラメータのエンコードを行います。
フィルターのサンプルとしては、Tomcat に付属している SetCharacterEncodingFilter.java がありますので、これが利用できます。

(4) Web アプリケーションでエンコードする。

取得したパラメータ文字列を、Web アプリケーションの処理で、文字コードを指定して、エンコードします。
例として、strParam に設定したパラメータを、“Windows-31J”に変換するには次のように処理します。

```
String strEncoded = new String(strParam.getBytes("8859_1"), "Windows-31J");
```

3.1.7. Web アプリケーションで ClassNotFoundException、NoClassDefFoundError が発生する

対象となるクラスが、そのクラスを参照しているクラスから参照できなくなっている可能性があります。WebOTX では、Jakarta Commons のライブラリを“WebOTX/lib”に格納しています。それら WebOTX で格納しているクラスは Web アプリケーションのクラスローダより上位のクラスローダでロードされており、Web アプリケーションの WEB-INF/lib のクラスがクラスローダの上下関係で参照できなくなっている可能性があります。

もしくは、対象のクラスが、javax のパッケージの場合には、Servlet の仕様により、WEB-INF 配下から読み込めないことがあります。これは、Servlet 仕様での勧告(javax などの J2SE,J2EE クラスを WAR 内のライブラリでオーバライドさせるべきでない、という仕様のためです。

WebOTXで同梱しているライブラリとクラスのロードについては、「Tomcat から WebOTX AS への移行ガイド」資料の「1.1.4. 同梱するライブラリ」と「1.1.6. クラスのロードについて」を参照してください。

(1) クラスのロード処理を変更する

nec-web.xml の delegate 指定を変更することにより、クラスのロード処理を変えて、アプリケーションの動作を確認してください。

もしくは、WEB-INF/lib にある WebOTX で同梱されているライブラリ(例えば commons-logging*.jar など)を削除して、アプリケーションの動作を確認してください。
nec-web.xml の delegate 指定の変更は、次章の「3.1.8 配備でエラーが発生する」を参照してください。

(2) javax パッケージを WEB-INF 配下から読み込み利用可能にする

javax パッケージは、WEB-INF 配下から読み込みません。

javax パッケージを“WebOTX/lib”配下に移動するか、JavaVM のオプションで利用可能にする必要があります。JavaVM のオプションで利用可能にするには、次のようにコマンドを実行します。

```
otxadmin> create-jvm-options --user <ユーザ名> --password <パスワード>
--host <ホスト名> --port <管理ポート>
-Dcom.nec.webotx.enterprise.overrideablejavaxpackages=クラス名
```

javax.xml.ws を指定すれば、サブパッケージの javax.xml.ws.handler や javax.xml.ws.handler.soap 等も対象となります。

3.1.8. 配備でエラーが発生する

Web アプリケーションの配備でエラーが発生する場合、Web アプリケーションの WEB-INF/lib 配下のライブラリ(jar)が、WebOTX のライブラリと重複していることが原因と考えられますので、(1)~(5)に示す順番で対策を実施してください。

(1) nec-web.xml での delegate 指定

nec-web.xml で delegate を true とする。delegate を true とすることにより、上位のクラスローダーでロードされるクラスが優先され、ライブラリの重複による問題が回避できます。nec-web.xml は、配備時に自動生成されますが、配備前に Web アプリケーションの WEB-INF ディレクトリ配下に作成してください。作成時には、配備された Web アプリケーションの nec-web.xml を参考にしてください。nec-web.xml の定義例は次のようになります。

```
----- nec-web.xml -----  
<?xml version="1.0" encoding="UTF-8"?>  
  
<nec-web-app xmlns="http://java.sun.com/xml/ns/j2ee">  
    <context-root>context_name</context-root>  
    <class-loader delegate="true"/>  
</nec-web-app>
```

(2) commons-logging*.jar を Web アプリケーションから取り出す

nec-web.xml の定義変更だけでは問題が解決できない場合、commons-logging*.jar が Web アプリケーションの WEB-INF/lib に格納されている場合には、commons-logging*.jar を取り除いてください。commons-logging*.jar は WebOTX のライブラリに含まれています。

(3) xml-apis.jar を Web アプリケーションから取り出す

nec-web.xml の定義変更だけでは問題が解決できない場合、xml-apis.jar が Web アプリケーションの WEB-INF/lib に格納されている場合には、xml-apis.jar を取り除いてください。xml-apis.jar は WebOTX のライブラリに含まれています。

(4) javax のクラスでエラーが発生した場合

Web アプリケーション内の javax クラスは、Web アプリケーションレベルではロードされません。次のいずれかの対処を行ってください。

- Java VM のオプションでオーバーライドを指定する

次の Java VM オプションを指定します。

-Dcom.nec.webotx.enterprise.overrideablejavaxpackages=javax クラス名
コマンドによる設定は次のように行います。

```
otxadmin> create-jvm-options --user <ユーザ名> --password <パスワード>  
--host <ホスト名> --port <管理ポート>  
-Dcom.nec.webotx.enterprise.overrideablejavaxpackages=クラス名
```

- javax のクラスを含む jar ファイルを WebOTX インストールディレクトリの lib 配下に格納する

- (5) jar ファイルを取り除いた、もしくは移動した後にクラスの参照に失敗するようになった場合
上位のクラスローダーでロードされているクラスから、Web アプリケーション内のクラスを参照しようとしてエラーになっていると考えられます。
参照が失敗するようになったクラスを含む jar ファイルを WebOTX インストールディレクトリの lib 配下に格納してください。

3.1.9. 配備解除でエラーが発生する

Web アプリケーションの配備解除でエラーが発生する場合として、Web アプリケーションの WEB-INF 配下などに格納したファイル(*.properties など)を JDBC や XML パーサなどで処理していると、これらのファイルが参照状態になり配備解除に失敗することがあります。ドメインを停止後、domain のディレクトリの applications ディレクトリ配下にある Web アプリケーションのディレクトリの WEB-INF 配下の “lib”、 “classes”、 “nec-web.xml”、 “web.xml” 以外のファイルを手動で削除して、ドメインを起動後配備解除してください。

例えば、Apache Portals Project で公開されている Jetspeed がこの Web アプリケーションに該当します。

3.2. Web コンテナでの障害対策

Web コンテナに関連する障害に対する対策について説明します。

3.2.1. OutOfMemoryError への対応

メモリの割り当て量の不足、システムの資源不足の場合、“OutOfMemoryError”が発生し、アプリケーションの実行に失敗することがあります。

状況の確認方法

WebOTX インストールディレクトリ/domains/domain_name/logs/serer.log で、“OutOfMemoryError”が記録されているかを確認します。

もしくは、運用管理コンソール(<http://Webサーバ名:ポート番号/admin/>)の「統計情報」-「JavaVM の統計情報」によって、Java VM の空きメモリの情報が確認できます。ただ、メモリ不足の状態が継続している場合には、運用管理コンソールが正しく動作できない場合もあります。

Java VM のヒープメモリに空きがあるにもかかわらず、“OutOfMemoryError”となっている場合には、Permanent 領域の不足が考えられます。

復旧方法

1 メモリ割り当て量の増加

WebOTXの動作する Java VM に割り当てているメモリ量を増やすことが考えられます。640MBのメモリを割り当てる場合には、Java VM のオプションで“-Xmx640m”のように指定します。メモリ割り当て量の変更の詳細については、「2.1Java VMへの割り当てメモリ」を参照してください。

2 Permanent 領域の増加

Java VM のヒープメモリに空きがあるにもかかわらず、“OutOfMemoryError”が発生している場合には、Permanent領域を増やすことが考えられます。Permanent領域を最大 128MBにする場合には、Java VM のオプションで“-XX:MaxPermSize=128m”のように指定します。メモリ割り当て量の変更については、「2.1Java VMへの割り当てメモリ」を参照してください。

3 システム資源

WebOTX の動作するシステムの資源(メモリ、スレッド、プロセス毎のスレッド)が不足している場合には、システムの資源の見直しを行ってください。“OutOfMemoryError”の後に“unable to create new native thread”が表示されている場合には、オペレーティングシステムでスレッドを生成することができない状態と考えられますので、システムの状態を確認してください。

HP-UX の場合には、システムのプロセス毎のスレッド数の上限値も確認してください。HP-UX11i の場合

には、プロセス毎のスレッド数の上限値は、カーネルパラメータの `max_thread_proc` にあたりますので、これを変更します。変更する手順については、HP-UX のマニュアルを参照してください。
Red Hat Enterprise Linux AS 4.0 の場合は、`/proc/sys/kernel/threads-max` でシステム全体のスレッド数の最大値が設定されますので、必要であれば変更してください。変更する手順については、OS のマニュアルを参照してください。

予防のための対策

アプリケーションでメモリリークを起こしているために、この問題が発生している場合があります。この場合には、WebOTX を起動している時間に応じて、空きメモリが減っていき、Java のガベージコレクションが動作しても、空きメモリが増えなくなってしまう。運用管理コンソール (<http://Webサーバ名:ポート番号/admin/>) の「統計情報」→「JavaVM の統計情報」によって、Java VM の空きメモリの情報が確認できます。通常であれば、空きメモリが少なくなったところで、Java のガベージコレクションが動作し空きメモリが増えます。メモリリークが発生している場合には、時間の経過とともに空きメモリが減少していきます。

アプリケーションのメモリリークが疑われるときには、アプリケーションのコードを見直す必要があります。メモリリークをさらに詳しく調べるには、Java の `-Xrunhprof` オプションを使ってプロファイル情報を採取します。`-Xrunhprof` オプションを使うとパフォーマンスが低下しますので、評価環境で実施されることをお勧めします。`-Xrunhprof` オプションは、例えば、Java VM の次のようなオプションになります。

```
-Xrunhprof:cutoff=0,heap=all,file=/tmp/java.prof
```

Java VM オプションを追加するには、`otxadmin` コマンドを使用して次のコマンドを実行します。

```
otxadmin> create-jvm-options --user <ユーザ名> --password <パスワード>
--host <ホスト名> --port <管理ポート>
-Xrunhprof¥¥:cutoff=0,heap=all,file=/tmp/java.prof
```

オプションを追加したらドメインを停止します。

ドメインを起動後、必要な操作 (AP の実行など) が終わったら、ドメインを停止する前に `-Xrunhprof` オプションを削除します (削除も、統合運用管理ツールや `otxadmin` コマンドを利用します)。

ドメインの停止により、上記のオプションであれば、`/tmp/java.prof` にメモリ状況のプロファイル情報が出力されますので、これを解析します。解析には、HPjmeter などが利用できます。HPjmeter は、Hewlett-Packard の Web サイトよりダウンロードできます。

なお、プロファイルデータを取得するには、WebOTX が利用するメモリに加えて、プロファイルデータ採用途のメモリが必要になります。標準の状態ではメモリ不足になる事が考えられますので適宜増やしてください。

3.2.2. 異常終了への対応

Web コンテナ (WebOTX ドメイン) が動作する Java VM が異常終了する場合の原因の一つとして、HotSpot コンパイルの問題があります。

ここでは、HotSpot コンパイルの問題の確認と対処について説明します。

状況の確認方法

Java VM のオプションに `-XX:+PrintCompilation` を追加して起動し、ログを採取します。Java VM オプションは、`otxadmin` コマンドを使用して次のコマンドを実行します。

```
otxadmin> create-jvm-options --user <ユーザ名> --password <パスワード>
--host <ホスト名> --port <管理ポート>
-XX¥¥:+PrintCompilation
```

Windows の場合には、次の Java VM オプションも追加します。

```
otxadmin> create-jvm-options --user <ユーザ名> --password <パスワード>
--host <ホスト名> --port <管理ポート>
-Xrunwojconsole:
```

オプションを追加したらドメインを停止します。

ドメインを起動後、アプリケーションを実行して、現象を再現させます。現象が再現した時点で、WebOTX インストールディレクトリ/domains/domain_name/logs/server.log (Windows の場合、\$[INSTALL_ROOT]¥domains¥ドメイン名¥logs¥jvm_stdout.log) に出力されている最後のメソッド名が HotSpot コンパイルでエラーになっているメソッドです。次の復旧方法に従って、対処してください。対処後には、HotSpot コンパイルのログが出力されるままになりますので、追加したオプションは元に戻してください。

HotSpot コンパイルの調査中に、他の原因により Java VM が終了した場合には、上記の対処を行っても改善しないことがあります。hs_err_pidxxx.log ファイルや、コアダンプで原因を調査する必要があります。開発元へお問い合わせください。

復旧方法

該当メソッドを HotSpot コンパイルの対象外にするために、“hotspot_compiler”ファイルを作成します。“hotspot_compiler”ファイルには、対象となるメソッドを記述します。記述例を次に示します。

com/test/testap の testMethod が問題になっている場合

```
exclude com/test/testap testMethod
```

WebOTX インストールディレクトリ/domains/domain_name/config に“hotspot_compiler”ファイルを格納してください。

3.2.3. 無応答障害(デッドロック)への対応

アプリケーションに対するアクセスに対する応答がなく、無応答状態になっている場合には、Web コンテナ (WebOTX ドメイン) 上の処理で資源確保待ちなどのデッドロックが発生しているか、もしくは、Web コンテナ (WebOTX ドメイン) の動作する Java VM が異常終了している場合があります。

ここでは、処理がデッドロックしている場合について説明します。

状況の確認方法

Java VM のスレッドダンプを取得します。スレッドダンプを取得する方法については、「3.3JavaVMのスレッドダンプ取得」を参照してください。

復旧方法

スレッドダンプを参照して、デッドロックの原因を推定します。J2SE 1.4.1 以上では、スレッドダンプにデッドロックになっていることが出力される場合もあります。

スレッドダンプでのデッドロックの出力例は次のとおりです。

```
Full thread dump Java HotSpot(TM) Server VM (1.4.2_06-b03 mixed mode):
```

```
....
```

```
Found one Java-level deadlock:
```

```
=====
```

```
“TP-Processor27”:
```

```
  waiting to lock monitor 0x8a276f8c (object 0x9b5d0570, a test.testClass),
```

```

which is held by "TP-Processor25"
"TP-Processor25":
    waiting to lock monitor 0x8a276fc4 (object 0x9b5cf7e0, a test.testClass),
    which is held by "TP-Processor27"

Java stack information for the threads listed above:
=====
"TP-Processor27":
    at test.testClass.close (test.Class.java:127)
    - waiting to lock <0x9b5d0570> (a test.testClass.)
    . . . . .

```

特定のオブジェクトの待ち合わせでスレッドの処理が止まっている場合、そのオブジェクトに関する処理に問題があると思われます。

DB へのアクセス処理で処理が止まっている場合、DB アクセスに問題があることが考えられます。

アプリケーションの処理中でなく、WebOTX 内の処理で止まっている場合には、お手数ですが、開発元へお問い合わせ願います。その際、スレッドダンプを含むログファイル等の送付もお願い致します。

3.2.4. セキュリティ例外への対応

WebOTX では、デフォルトでセキュリティマネージャが動作しています。このためアプリケーションが OS のリソースへアクセスした際に、セキュリティ例外が発生することがあります。対象となるリソースへのアクセスを許可する場合には、次の手順で許可してください。

状況の確認方法

次の様な例外が、クライアントの画面、もしくはログファイルに出力されます。

```
access denied (java.io.FilePermission /temp read)
```

復旧方法

WebOTX インストールディレクトリ/domains/domain_name/config/server.policy に read 権限を追加します。次は、"/temp"への read 権限を追加する例です。

server.policy 47行目付近

```
// Basic set of required permissions granted to all remaining code
grant {
    permission java.io.FilePermission "${}/temp", "read";
    ...
}
```

ポリシーファイルに記述する構文のフォーマットについては、お使いの Java VM のマニュアルを参照ください。

server.policy ファイルを更新した後は、WebOTX のドメインを再起動してください。

3.3. JavaVMのスレッドダンプ取得

WebOTX が起動中にストール状態になる、または、コマンドからの応答がなく停止できない状態など、WebOTX が無応答状態に陥った場合には以下の手順で Java VM のスレッドダンプを取得し開発元に問い合わせを行ってください。

3.3.1. UNIX のスレッドダンプ

以下の手順でスレッドダンプを採取します。

取得方法

1. ドメインの Java VM を特定します。
ps コマンドによりドメインの Java VM を特定します。ps コマンドの結果では、他の java プロセスとの区別が付きにくい Java VM に指定された-X オプションを手がかりにプロセスの特定を行ってください。

```
ps -exf |grep java |grep ドメイン名 |grep '-X'
```

ドメインの Java VM にはデフォルトインストールで以下のオプションが引数に指定されています。

```
-server -Xms64m -Xmx512m -XX:NewRatio=2
```

オプションを変更している場合には、変更後のオプションに読み替えてください。

2. 1で特定したプロセス番号に対して kill -SIGQUIT を実行します。
無限ループによる障害を検出するために、数秒間隔で複数回実行するようにしてください。

```
kill -3 <pid>
```

3. 以下のファイルにスレッドダンプの結果が出力されます。

```
/opt/WebOTX/domains/ドメイン名/logs/server.log
```

3.3.2. Windows のスレッドダンプ

Windows では、Java プロセスに対して Ctrl+Break キーを送信することでスレッドダンプが出力されます。しかし、WebOTX はサービスとして起動されているために通常ではスレッドダンプの取得は行えません。スレッドダンプを出力させるようにするには以下の設定を行うことで取得できます。

ドメインを再起動する必要があるため再現性のある問題に対して有効です。障害調査以外では設定を元にもどすようにしてください。

設定方法

1. WebOTX を停止した状態で設定を行います。
起動中にストールしコマンドを受け付けられない状態になる場合は、一旦 “WebOTX Agent Service” のサービスを自動から手動に切り替えてサーバを再起動します。

スタート→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→サービス
“WebOTX Agent Service” を自動から手動に変更

2. ドメインの設定に Java VM オプションを追加します。
各ドメインの設定ファイル\${INSTANCE_ROOT}¥config¥domain.xml に-Xrunwojconsole の記述を追加します。-Xrunwojconsole は WebOTX V6.22 よりサポートしています。

～

```
<jvm-options>-Xms64m</jvm-options>
```

```
<jvm-options>-Xmx512m</jvm-options>
```

```
<jvm-options>-Xrunwojconsole</jvm-options> ←この行を追加
```

```
<jvm-options>-XX:NewRatio=2</jvm-options>
```

～

3. サービスのデスクトップとの会話を変更
サービスのコントロールパネルを開き、WebOTX Agent Service の“デスクトップとの対話をサービスに許可”にチェックを入れます。

スタート→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→サービス
WebOTX Agent Service→ログオン→デスクトップとの対話をサービスに許可

4. WebOTX Agent Service サービスを起動します。
WebOTX が起動すると、デスクトップ画面上にコンソール画面が表示されるようになります。
5. 表示されたコンソール画面に、Ctrl+Break キーを入力します。
無限ループによる障害を検出するために、数秒間隔で複数回実行するようにしてください。
6. 以下のファイルにスレッドダンプの結果が出力されます。

```
${INSTALL_ROOT}¥domains¥ドメイン名¥logs¥jvm_stdout.log
```

設定解除方法

1. WebOTX を停止した状態で設定を行います。
起動中にストールしコマンドを受け付けられない状態になる場合は、一旦 “WebOTX Agent Service” のサービスを自動から手動に切り替えてサーバを再起動します。

スタート→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→サービス
“WebOTX Agent Service” を自動から手動に変更

2. ドメインの設定から追加した Java VM オプションを削除します。
各ドメインの設定ファイル \${INSTANCE_ROOT}¥config¥domain.xml から -Xrunwojconsole の記述を削除します。

～

```
<jvm-options>-Xms64m</jvm-options>  
<jvm-options>-Xmx512m</jvm-options>  
<jvm-options>-Xrunwojconsole</jvm-options> ←この行を削除  
<jvm-options>-XX:NewRatio=2</jvm-options>
```

～

3. サービスのデスクトップとの会話を変更
サービスのコントロールパネルを開き、WebOTX Agent Service の“デスクトップとの対話をサービスに許可”からチェックをはずします。

スタート→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→サービス
“WebOTX Agent Service”→ログオン→デスクトップとの対話をサービスに許可

4. WebOTX Agent Service サービスを起動します。

信頼性、柔軟性、サポート
3つの安心でお客様のシステムを支えます。



をどうぞよろしくお願いします。

- お問い合わせ先
NEC 第二システムソフトウェア事業部
mailto:info-webotx@isd.jp.nec.com
- 製品ホームページURL
<http://www.nec.co.jp/WebOTX/>