



Web アプリケーションでの log4j 利用ガイド

WebOTX V6.4,6.5 編



NEC

第二システムソフトウェア事業部

改版履歴

2007 年 5 月 初版

目次

1. はじめに	1
1.1. 対象読者	1
1.2. 表記について	1
2. WebOTXのクラスローダの仕組み	1
3. WebAPからlog4j利用手順	3
3.1. WebAPにlog4jを含める場合	3
3.2. WebOTXが提供するlog4jを利用する場合	5
3.3. マルチプロセスモードでWebAPのログ出力を行う場合	5
3.4. WebAPのログ出力にcommons-logging経由でlog4jを使用する場合	7
3.5. NTEventLogAppenderを利用する場合	7
4. おわりに	8

1. はじめに

WebOTX V6 では、WebOTX 自身のログ出力に log4j を使用しており、log4j のログ一定義ファイルである log4j.xml を利用しています。Web アプリケーション(以下 WebAP)で log4j を利用する場合、AP 側で利用する定義を、WebOTX のログ定義が記述してある log4j.xml ファイルに追記する必要がありました。WebOTX は V6.4 から、WebAP との定義の混同を避けるために、WebOTX のログ出力を行うために独自の logger を利用するようになりました。この変更により、WebAP で log4j を利用する場合は、log4j 定義ファイルにはユーザ AP で使用する定義のみを記述することが可能になりました。

1.1. 対象読者

本ガイドは WebOTX V6.4、V6.5 の各 Edition 上で動作させる WebAP から log4j を利用したログ出力を行いたい、アプリケーション開発者、SE を対象としています。

1.2. 表記について

本ガイドでは以下の環境変数を用います。なお、OS を限定しない限り、セパレータは/(スラッシュ)を利用します。

`${INSTALLROOT}` : インストールディレクトリ
`${INSTANCEROOT}` : ドメインルートディレクトリ
`${APGNAME}` : アプリケーショングループ名
`${PGNAME}` : プロセスグループ名
`${PID}` : プロセス ID
`${WEBAPNAME}` : WebAP 名

2. WebOTX のクラスローダの仕組み

WebAP のログ出力方法の説明を行う前に、WebOTX のクラスローダについて説明します。WebOTX のクラスローダの仕組みを理解しておくことで、本ガイドの理解が深まると思います。

クラスローダには階層があり、上位のクラスローダでロードされたクラスから、下位のクラスローダでロードされたクラスを参照することはできません。逆に、下位のクラスローダでロードされたクラスから、上位のクラスローダでロードされたクラスを参照することはできます。WebOTX のクラスローダがロードする先のパスは以下のようになります。

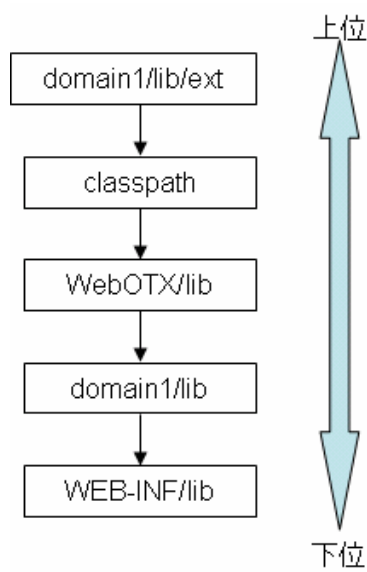


図 1. WebOTX のクラスローダの階層

また、`nec-web.xml` の `delegate` の設定により、ロードする優先順位を決定することができます。同じ名前のライブラリがあると、競合が発生し、優先順位の高いパスに含まれるのライブラリが利用されます。ロードされるライブラリの優先順位は `delegate=true` の場合は以下ようになります。`delegate` の設定は WebAP の WEB-INF/nec-web.xml で設定します。

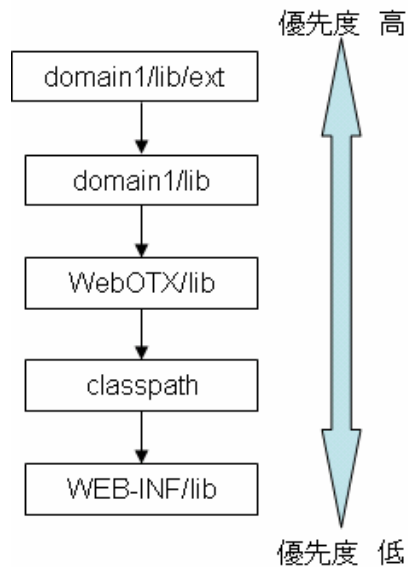


図 2. `delegate=true` の場合のロード優先順位

また、`delegate=false` の場合は以下のような優先順位になります。

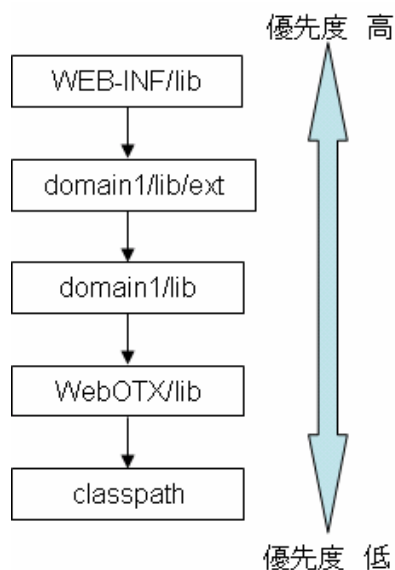


図 2. delegate=false の場合のロード優先順位

例えば、`${INSTALLDIR}/lib` と `WEB-INF/lib` に同じライブラリがある場合、`delegate=true` の場合は、`${INSTALLDIR}/lib` 側のライブラリがロードされ、`delegate=false` の場合は、`WEB-INF/lib` が優先してロードされます。

3. WebAP から log4j 利用手順

本章では、WebOTX 上で動作させるアプリケーションから、log4j を利用してログ出力を行う手順について説明します。

WebOTXでは、V6.4 からV6.50.01 では、log4jのバージョンは 1.2.13、V6.50.02 以降では 1.2.14 をバンドルしています。WebOTXがバンドルしているlog4jでログ出力を行う場合は、WebAPにlog4jを含める必要はありません。バンドルしているlog4jの利用方法は [3.2 節「WebOTXが提供するlog4jを利用する場合」](#)を参照してください。違うバージョンのlog4jを利用する場合は、以下の [3.1 節「WebAPにlog4jを含める場合」](#)を参照してください。

3.1. WebAP に log4j を含める場合

WebOTX がバンドルしているバージョンと違う log4j を利用する場合は、war ファイルに log4j を含める必要があります。また、WebAP で利用するログ定義ファイルは、`WEB-INF/classes` 直下に配置してください。WebAP のログ出力を行うには以下の手順を行ってください。

1. WebOTX に設定されている log4j.configuration の削除

WebOTX で設定されている log4j.configuration 定義を削除します。Windows 環境では以下のコマンドを実行してください。

```
otxadmin
otxadmin > login --user admin --password adminadmin --port 6212 --host localhost
otxadmin > delete-jvm-options
-Dlog4j.configuration=file¥¥:///{com.nec.webotx.instanceRoot}{file.separator}
config{file.separator}log4j.xml
```

UNIX 環境では、login コマンド後に以下を実行してください。

```
otxadmin > delete-jvm-options
-Dlog4j.configuration=file¥¥:///{com.nec.webotx.instanceRoot}{file.separator}
config{file.separator}log4j.xml
```

上記コマンドのユーザ名、パスワード、ポート番号、ホスト名は環境にあわせてください。コマンドは 1 行で実行してください。

2. war ファイルの配備

war ファイルを WebOTX に配備してください。

3. nec-web.xml の確認

war ファイルを配備すると、nec-web.xml が自動作成されます。nec-web.xml は以下にあります。

```
{INSTANCEROOT}/applications/j2ee-modules/{WEBAPNAME}/WEB-INF/nec-web.xml
```

nec-web.xml で delegate=false となっていることを確認してください。Servlet 2.4 仕様の WebAP を配備した場合は delegate=true となっているので、true の場合は、nec-web.xml を以下のように変更します。

```
<class-loader delegate=false/>
```

変更後、配備前の WebAP の WEB-INF 以下に nec-web.xml をコピーし、war ファイルを再作成してください。

4. war ファイルの再配備

3 で war ファイルを再作成した場合は、WebOTX に再配備してください。

5. ドメインの再起動

設定変更を有効にするために、ドメインを再起動してください。

V6.50.02 パッチ適用済み環境ではドメインを停止し、以下も実行してください。

6. log4j.jar の削除

V6.50.02 パッチを適用済み環境では、上記に手順に加えて `${INSTALLDIR}/lib/log4j.jar` を削除してください。V6.50.02 より前の環境では、log4j.jar を削除する必要はありません。削除した場合、Web コンテナが起動しなくなります。なお、V6.50.02 以降の標準修正パッチを適用すると、`${INSTALLDIR}/lib/log4j.jar` が再度配置されます。忘れずに削除してください。

注意事項

- WebAPにlog4jを含んでいる場合、複数のWebAPからNTEventLogAppenderを利用することはできません(単一のWebAPからの利用ならば可能)。詳しくは、[3.5 節「NTEventLogAppenderを利用する場合」](#)を参照してください。

3.2. WebOTX が提供する log4j を利用する場合

war ファイルに log4j を含めず、`${INSTALLDIR}/lib/log4j.jar` を利用する方法について説明します。WebAP で利用する定義は、`${INSTANCEROOT}/config/log4j.xml` に追記してください。このとき、log4j.xml にデフォルトで定義されている内容は削除しないようにしてください。

注意事項

- V6.50.01 以前の環境で、WebAPに含めた独自に作成したAppenderクラスを利用する場合は、warファイルにlog4jを含めてください。手順については[3.1 節「WebAPにlog4jを含める場合」](#)を参照してください。
なお、V6.50.02 以降の環境では問題ありません。

3.3. マルチプロセスモードで WebAP のログ出力を行う場合

V6.5 Standard または Enterprise Edition をマルチプロセスモードで実行する場合の WebAP のログ出力の手順について説明します。手順について説明する前に、プロセスグループ上で動かす WebAP で使用する Appender について説明します。

- 使用する Appender について
マルチプロセスモードで実行する場合、WebAP はプロセスグループ上で動作します。プロセスグループは、多重度を設定することができるため、ファイルにログ出力を行う場合、複数のプロセスが同じファイルに書き込みを行ってしまいます。

そのため、ログの解析を行うことが困難になってしまう場合があります。それを回避するために、プロセスグループ上のアプリケーションでは、ログ出力に標準出力(stdout)を使用しなければなりません。プロセスグループ上のアプリケーションで標準出力を行うと、プロセスグループのログに出力されます。ログは以下のファイルに出力されます。

```
${INSTANCEROOT}/logs/tpssystem/${APGNAME}/${PGNAME}/${PGNAME}.${PID}.log
```

ログ出力の手順について説明します。

1. log4j.configuration の設定

log4j のシステムプロパティ log4j.configuration をプロセスグループの JavaVM オプションに追加してください。以下のコマンドを実行してください。ただし、プロセスグループには、log4j.configuration の値にデフォルトで \${INSTANCEROOT}/config/tpssystem/logconf/\${APGNAME}-\${PGNAME}/log4j.xml が設定してあります。この設定ファイルに定義を追加する場合は、以下のコマンドを実行する必要はありません。

```
otxadmin
```

```
otxadmin > login --user admin --password adminadmin --port 6212 --host localhost
```

```
otxadmin > set
```

```
tpssystem.applicationGroups.<apgname>.processGroups.<pgname>.otherArguments="-Dlog4j.configuration=file¥¥://${com.nec.webotx.instanceRoot}${file.separator}config${file.separator}<log4j 定義ファイル名>
```

UNIX 環境では、login コマンド後に以下を実行してください。

```
otxadmin > set
```

```
tpssystem.applicationGroups.<apgname>.processGroups.<pgname>.otherArguments="-Dlog4j.configuration=file¥¥://${com.nec.webotx.instanceRoot}${file.separator}config${file.separator}<log4j 定義ファイル名>
```

上記コマンドのユーザ名、パスワード、ポート番号、ホスト名、log4j.configuration のパスは実際の環境に合わせて変更してください。

2. war ファイルの配備

war ファイルをプロセスグループに配備してください。

3. プロセスグループの再起動

ログ定義ファイルを有効にするために、プロセスグループを再起動してください。

注意事項

- ログ定義ファイルについて

プロセスグループ上で WebAP を動作させる場合、WebAP が利用するログ定義ファイルを AP ごとに分けて利用することはできません。AP で利用するログ定義は `log4j.configuration` で指定したファイルにすべて記述してください。

3.4. WebAP のログ出力に commons-logging 経由で log4j を使用する場合

V6.50.01 以前では、WebAPにcommons-loggingを含めると、WebOTX側で提供しているcommons-loggingとの競合が発生します。この問題を解決するには、`nec-web.xml`で`delegate=true`にする、またはWebAPからcommons-loggingとlog4jを取り除き、WebOTXがバンドルしているlog4jとcommons-loggingを利用してください。commons-loggingは`${INSTALLROOT}/lib/commons-pack.jar`に含まれています。WebOTXがバンドルしているlog4jを利用したログ出力方法は、[3.2 節「WebOTXが提供するlog4jを利用する場合」](#)を参照してください。

なお、上記の問題は V6.50.02 以降の環境では発生しません。

3.5. NTEventLogAppender を利用する場合

Windows 環境において、NTEventLogAppender を利用する場合の注意点について説明します。NTEventLogAppender を使用するには、log4j が提供するNTEventLogAppender.dll を利用します。同一プロセス内において、dll を複数のクラスローダから同時にロードしようとすると、以下のエラーが発生します。

```
java.lang.UnsatisfiedLinkError: Native Library <dll 名> already loaded in another classloader
```

複数のWebAPからNTEventLogAppenderを利用する場合は、WebOTXが提供しているlog4jを利用する必要があります。WebOTXが提供しているlog4jの利用方法については、[3.2 節「WebOTXが提供するlog4jを利用する場合」](#)を参照してください。ただし、NTEventLogAppenderを利用するWebAPが1つの場合は、WebAPに含めたlog4jから利用

することができます。

4. おわりに

WebOTX V6 上で、WebAP から log4j を利用する方法に関して、現在も WebOTX サポート部門に問い合わせの中からよくある質問と、その解決方法について説明しました。

本ガイドは、ログ出力に関するトラブル解決のために、WebOTX サポート部門への問い合わせによる時間的ロスが発生させないように作成したものです。本ガイドによって、少しでもお客様が効率的に作業を進めることができたならば幸いです。