TERASOLUNA®

個別修正モジュール(PI-SJW-261-1)
適用手順書

2016 年 6 月 7 日

NTT DATA

株式会社 NTTデータ

目次

1. はじめに

- 本書では、TERASOLUNA Server Framework for Java WEB 2.0.x.x を利用したシステム向けに、個別修正モジュール（PI-SJW-261-1）（以下「品質強化パッチ」と呼ぶ）を適用する手順を説明します。
- 本ドキュメントに記載してある内容をそのままコピー&ペーストする場合は、文字コードの違いにご注意ください。具体的には Word よりコピー＆ペーストすることで web.xml に BOM（byte order mark）がつきエラーとなる事象が確認されております。設定後のファイルが正しい文字コードになっていることを確認するようしてください。

2. 品質強化パッチ

品質強化パッチ(ServletFilter)を以下の名称の jar ファイルの形態で提供します。

- terasoluna-thin-patch-ExtensionFilter-1.0.0.jar

上記の ServletFilter の適用にあたり、追加/修正が必要な資材を表1に示します。

表 1 追加/修正が必要な資材一覧

資材	追加/修正	備考
terasoluna-thin-patch-ExtensionFilter-1.0.0.jar	追加	品質強化パッチ
web.xml	修正	デプロイメント記述子
xxxxx.properties（ファイル名は任意）	修正	プロパティファイル ※1

※1. ファイル名は実装方法で異なります。実際のファイル名に関しては「3.1 品質強化パッチ適用対象のバックアップを取得する」を参照してください。

3. 適用手順

本手順では、ローカルの開発環境で資材の追加・変更を行い、本番サーバへリリースする方式を想定しています。直接本番サーバの資材を変更する場合は、すべての手順を終えた後に AP サーバを再起動してください。

3.1. 品質強化パッチ適用対象のバックアップを取得する

品質強化パッチによる意図しない動作や、回帰試験時の障害検出時のリカバリ作業に備え、開発環境のアプリケーションから「4 リカバリ手順」で必要になる「表 2 バックアップ対象一覧」のバックアップを取得します。

表 2 バックアップ対象一覧

フォルダ	バックアップ対象ファイル	説明
WEB-INF/	web.xml	Web デプロイメントデスクリプタ
ソースフォルダ配下	xxxxx.properties（ファイル名は任意）	プロパティファイル※2

※2. ファイル名は実装方法により異なります。ソースフォルダ配下の全 properties ファイルから「access.control.prohibited.extension.1」が設定されているファイルを検索してください。

（機能説明書（「TERASOLUNA Server Framework for Java（Web版） 機能説明書.pdf」-「WA-05 拡張子直接アクセス禁止機能」）に記載の使用方法に従って設定していれば、[ソースフォルダ]/system.properties に設定されています。）

3.2. 拡張子直接アクセス禁止機能の旧設定を無効にする

「拡張子直接アクセス禁止旧機能設定（プロパティファイル）」の (1)、(2)の各設定値を退避したうえで、「access.control.prohibited.extension」および「restrictionEscape」プロパティを行頭に"#"を付与し、すべてコメントアウトしてください。

拡張子直接アクセス禁止旧機能設定（プロパティファイル）

```
#-----------------------------------------------------------------------
# (1) 直接アクセス禁止対象の拡張子を指定する。
#access.control.prohibited.extension.1=.jsp
#access.control.prohibited.extension.2=.csv


#-----------------------------------------------------------------------
# (2)「拡張子直接アクセス禁止機能」を適用しないパスを指定する。
#restrictionEscape.1=/sample/logon/index.jsp
#restrictionEscape.2=/sample/error/error.jsp
```

すべてコメントアウト

3.3. Jar ファイルを追加する

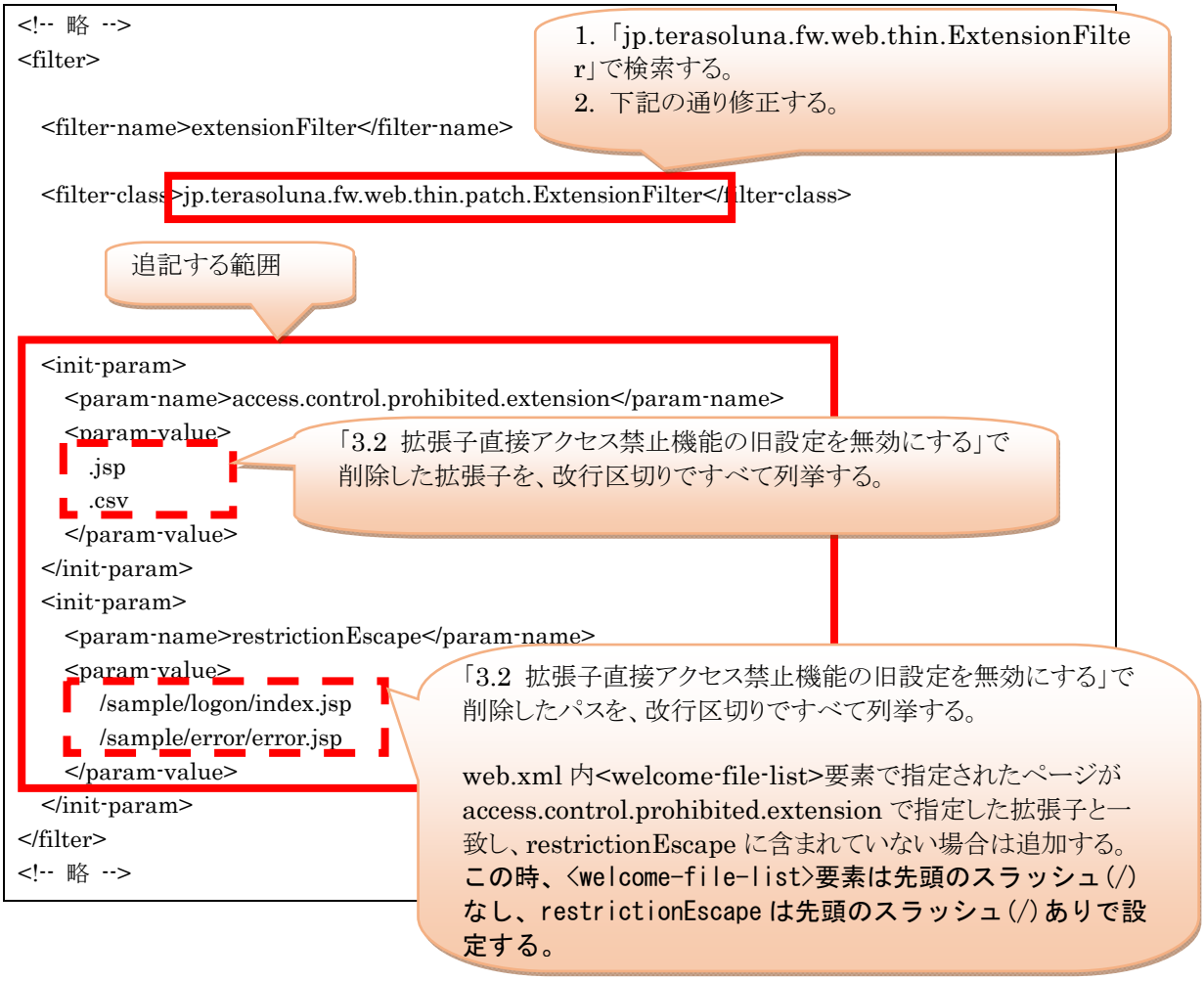
WEB-INF/lib フォルダに品質強化パッチの jar ファイル(terasoluna-thin-patch-ExtensionFilter-1.0.0.jar)を追加してください。

3.4. デプロイメントデスクリプタ（web.xml）を修正する

対象アプリケーションの WEB-INF/web.xml に、品質強化パッチが提供する ServletFilter の定義を追加します。

web.xml の「<filter-class>」要素を、「jp.terasoluna.fw.web.thin.ExtensionFilter」で検索し、ServletFilter 定義に「修正例（web.xml）」の通り修正／追記してください。

修正例(web.xml)



3.5. アクセス禁止を検知した時のログを設定する

本品質強化パッチが提供する ExtensionFilter では、アクセスを禁止した事を通知するログを commons-logging のロガークラスを使用して出力します。

アクセス禁止を検知したメッセージをログ出力するため、ログレベルを確認する必要があります。commons-logging API の実装として log4j を利用している際は、log4j.properties(あるいは log4j.xml)の設定にて以下を追記してください。(既存の設定に合わせ、log4j 1.1 以前、1.2 以降どちらか一方の書式で追記してください。)

ログ出力定義(log4j.properties)

```
# log4j 1.2 以降の書式
log4j.logger.jp.terasoluna.fw.web.thin.patch.ExtensionFilter=DEBUG

# log4j 1.1 以前の書式(1.2 以降でも動作)
log4j.category.jp.terasoluna.fw.web.thin.patch.ExtensionFilter=DEBUG
```

ログ出力定義(log4j.xml)

```
<!-- log4j 1.2 以降の書式 -->
<logger name="jp.terasoluna.fw.web.thin.patch.ExtensionFilter">
  <level value="debug" />
</logger>

<!-- log4j 1.1 以前の書式（1.2 以降でも動作） -->
<category name="jp.terasoluna.fw.web.thin.patch.ExtensionFilter">
  <priority value="debug" />
</category>
```

また web.xml と同様に、ロガーのカテゴリ(クラス)に旧フィルタの FQCN が指定されていた場合は、「jp.terasoluna.fw.web.thin.patch.ExtensionFilter」に差し替えてください。

3.6. Web アプリケーションをビルド、デプロイする

ビルド、デプロイ手順は、プロジェクト毎に異なりますので、プロジェクトのビルド、デプロイ手順やリリース手順に従い、Web アプリケーションをビルド、デプロイしてください。ただし、jar ファイルの追加と web.xml の更新を反映させることが目的なので、業務で開発した Java ソースコードをコンパイルし直す必要はありません。

4. リカバリ手順

適用に失敗した場合、以下の手順でリカバリを行ってください。

本手順では、ローカル開発環境で資材の追加・変更を行い、本番サーバへリリースする方式を想定しています。直接本番サーバの資材を変更する場合は、以下の手順を実施後、AP サーバを再起動してください。

1. 「3.3 Jar ファイルを追加する」で追加した jar ファイルを WEB-INF/lib フォルダから削除する。
2. WEB-INF フォルダの web.xml を、「3.1 品質強化パッチ適用対象のバックアップを取得する」でバックアップした web.xml に戻す。
3. プロパティファイルが格納されたフォルダの xxxxx.properties(ファイル名は任意)を「3.1 品質強化パッチ適用対象のバックアップを取得する」でバックアップしたプロパティファイルに戻す。
4. Web アプリケーションをビルド、デプロイする