

TERASOLUNA 2.0.6.2 移行ガイド(Web 版)

変更履歴

バージョン	日付	改訂箇所	改訂内容
2.0.6.2	2016/08/31	-	新規作成

■ 概要

本ドキュメントは、TERASOLUNA Server Framework for Java (Web 版) 2.0.6.1 で作成したアプリケーションを、2.0.6.2 へ移行する際の手順を示すドキュメントである。

本リリースにおける拡張子直接アクセス禁止フィルタの改善に伴い、アクセス制御用のパスを利用している場合、手順に沿った修正を行う。

影響を受ける箇所および対応方法に関しては、本ガイド「**■移行手順 ③拡張子直接アクセス禁止フィルタの改善に伴う影響調査および対応**」を参照のこと。

TERASOLUNA Server Framework for Java (Web 版) 2.0.6.1 に対して、2016/06/07 リリースのパッチを適用している場合においても、本ガイドに沿って移行を行なう。その際、以下の箇所の記載に従い、対応を行うこと。

・「Apache Struts 1.2.9 with SP3 by TERASOLUNA」を適用している場合
「**■移行手順 ②依存ライブラリの変更 2.0.6.1 から 2.0.6.2 へ移行する場合**」の注記を参照。

・「個別修正モジュール(PI-SJW-261-1)」を適用している場合
「**■移行手順 ②依存ライブラリの変更 2.0.6.1 に個別修正モジュール(PI-SJW-261-1)を適用済みであり、2.0.6.2 へ移行する場合**」の手順を参照。

◆ 2.0.6.2 の変更点概要

- アクセス制御用のパス(RequestUtil#getPathInfo の戻り値)の導出方法の変更
- 依存ライブラリのバージョンアップ
- commons-fileupload の脆弱性回避用のフィルタ設定削除
- build.xml のクラスパス設定変更

■ 移行手順

2.0.6.1 から 2.0.6.2 への移行手順を説明する。

前提条件

TERASOLUNA Server Framework for Java (Web 版) 2.0.6.1 を使用したアプリケーションが正常に動作していること。

以後、本書ではこれをアプリケーションと呼称する。

① blank プロジェクトのダウンロード

TERASOLUNA Server Framework for Java (Web 版) 2.0.6.2 の blank プロジェクト (terasoluna-server4jweb-blank_2.0.6.2.zip) を以下の URL よりダウンロードし、任意のフォルダに展開する。

ダウンロードサイトの URL :

➤ OSDN <https://osdn.jp/projects/terasoluna/releases/>

② 依存ライブラリの変更

TERASOLUNA フレームワークや Struts のライブラリの差し替えを行う。

2.0.6.1 から 2.0.6.2 へ移行する場合

差し替えるライブラリは以下の通りである。

2.0.6.1のライブラリ	2.0.6.2のライブラリ	種別
terasoluna-ContentTypeLengthLimitFilter-1.0.0.jar	-	削除
commons-fileupload-1.3.1.jar	commons-fileupload-1.3.2.jar	差し替え
struts-1.2.9-sp2.jar	struts-1.2.9-sp3.jar	差し替え(※)
terasoluna-thin-2.0.6.1.jar	terasoluna-thin-2.0.6.2.jar	差し替え

※2016/06/07 リリースのパッチ(sp3)を適用済の場合は差し替え不要。

「種別」に従って下記の修正を行う。

(「2.0.6.2 のライブラリ」は terasoluna-server4jweb-blank_2.0.6.2.zip を展開したフォルダの lib フォルダ直下に格納されている)

削除：

アプリケーションの lib フォルダ直下から「2.0.6.1 のライブラリ」に記載した jar ファイルを削除する。

差し替え：

アプリケーションの lib フォルダ直下から「2.0.6.1 のライブラリ」に記載した jar ファイルを削除し、「2.0.6.2 のライブラリ」に記載した jar ファイルを追加する。

同様に、Javadoc やソースコードが格納されている jar ファイルを差し替える。これらの jar ファイルは terasoluna-server4jweb-blank_2.0.6.2.zip を展開したフォルダの terasoluna フォルダ配下に格納されている。

2.0.6.1のJavaDoc格納jarファイル	2.0.6.2のJavaDoc格納jarファイル	種別
terasoluna-ContentTypeLengthLimitFilter-1.0.0-javadoc.jar	-	削除
terasoluna-thin-2.0.6.1-javadoc.jar	terasoluna-thin-2.0.6.2-javadoc.jar	差し替え

2.0.6.1のソースコード格納jarファイル	2.0.6.2のソースコード格納jarファイル	種別
terasoluna-ContentTypeLengthLimitFilter-1.0.0-sources.jar	-	削除
terasoluna-thin-2.0.6.1-sources.jar	terasoluna-thin-2.0.6.2-sources.jar	差し替え

2.0.6.1 に個別修正モジュール(PI-SJW-261-1)を適用済みであり、

2.0.6.2 へ移行する場合

「2.0.6.1 から 2.0.6.2 へ移行する場合」の対応に加え、以下の対応を行う。

- アプリケーションの lib フォルダ直下から
「terasoluna-thin-patch-ExtensionFilter-1.0.0.jar」を削除する。
- 「個別修正モジュール(PI-SJW-261-1)適用手順書」の設定値変更内容を元に戻す。
⇒ 適用手順書の 3.1 で取得したバックアップを反映する
 ※バックアップが無い場合、適用手順書の 3.2～3.4 の手順を手動で戻す
⇒ 適用手順書の 3.5 の手順を手動で戻す

③ 拡張子直接アクセス禁止フィルタの改善に伴う影響調査および対応

TERASOLUNA Server Framework for Java (Web 版) は、ブラウザからのリクエストに対し、任意の拡張子をもつコンテンツへのアクセスを制限する機能を備えているが、2.0.6.1 以前では、細工されたパスを与えられた場合、当該機能が機能しない。また、細工されたパスの解釈方法や解釈結果は、AP サーバによっても異なるため、2.0.6.2 にて、アクセス制御に使用するパスの導出方法を変更している。それに伴い、移行にあたり、②の対応に加えて、アプリケーション側で以下の対応を実施する。

対象：

A) アクセス制御系の以下のインタフェースを実装しているクラス

インタフェース名	機能
jp.terasoluna.fw.web.thin.AuthenticationController	ログオン済みチェック
jp.terasoluna.fw.web.thin.AuthorizationController	アクセス権限チェック
jp.terasoluna.fw.web.thin.BlockageController	業務閉塞チェック
jp.terasoluna.fw.web.thin.ServerBlockageController	サーバ閉塞チェック

B) jp.terasoluna.fw.web.RequestUtil#getPathInfo を利用しているクラス¹

対応：

各種 Controller インタフェースのメソッドの引数、もしくは RequestUtil#getPathInfo の戻り値で取得したパス情報を編集しているか確認し、以下の対応を行う。

「;」以降(「;jsessionid=xxx」等のパスパラメータ)を削除している場合
⇒AP サーバによってパスパラメータが除去されたパスを使用するよう、フレームワーク側の処理を変更したため、アプリケーション側の該当コードを削除

URL デコードしている場合

⇒AP サーバによって URL デコードされたパスを使用するよう、フレームワーク側の処理を変更したため、アプリケーション側の該当コードを削除

¹通常、A) で挙げたインタフェースを実装しているクラスと重複する。

④ commons-fileupload の脆弱性回避用のフィルタ設定削除

web.xml に記載された commons-fileupload の脆弱性回避用のフィルタ設定（以下）を削除する。

【web.xml】

```
*** (中略) ***
<!-- Content-Type サイズ制限(commons-fileupload 脆弱性回避用) -->
<filter>
  <filter-name>contentTypeLengthLimitFilter</filter-name>
  <filter-class>jp.terasoluna.fw.web.common.filter.ContentTypeLengthLimitFilter</filter-class>
  <!--
  <init-param>
    <param-name>limitBytes</param-name>
    <param-value>2048</param-value>
  </init-param>
  -->
</filter>
*** (中略) ***
<filter-mapping>
  <filter-name>contentTypeLengthLimitFilter</filter-name>
  <url-pattern>*.do</url-pattern>
</filter-mapping>
*** (中略) ***
```

⑤ build.xml のクラスパス設定変更

②依存ライブラリの変更に従って、build.xml のクラスパスの設定を修正する。
アプリケーション直下の/ant/build.xml を②依存ライブラリの変更で使用した依存ライブラリ表を参照し、「種別」に従って下記の修正を行う。

削除：

「2.0.6.1 のライブラリ」に記載した jar ファイルのクラスパス設定を削除する。

差し替え：

「2.0.6.1 のライブラリ」に記載した jar ファイルから「2.0.6.2 のライブラリ」に記載した jar ファイルにクラスパス設定の jar ファイル名を変更する。

●修正例●

【2.0.6.1 の build.xml】

```
<!-- クラスパスの設定 -->
<property name="classpath.lib" value="
    * * * (中略) * * *
    ${lib.dir}/commons-fileupload-1.3.1.jar;
    * * * (中略) * * *
    ${lib.dir}/struts-1.2.9-sp2.jar;
    * * * (中略) * * *
    ${lib.dir}/terasoluna-ContentTypeLengthLimitFilter-1.0.0.jar;
    ${lib.dir}/terasoluna-dao-3.3.1.jar;
    * * * (中略) * * *
    ${lib.dir}/terasoluna-thin-2.0.6.1.jar;
    * * * (中略) * * *
"/>
```

【2.0.6.2 の build.xml】

```
<!-- クラスパスの設定 -->
<property name="classpath.lib" value="
    * * * (中略) * * *
    ${lib.dir}/commons-fileupload-1.3.2.jar;
    * * * (中略) * * *
    ${lib.dir}/struts-1.2.9-sp3.jar;
    * * * (中略) * * *
    ${lib.dir}/terasoluna-dao-3.3.1.jar;
    * * * (中略) * * *
    ${lib.dir}/terasoluna-thin-2.0.6.2.jar;
    * * * (中略) * * *
"/>
```


⑥ 動作確認

アプリケーションを起動させて、問題なく動作することを確認する。