富士通 Java ブログ

Eclipse Transformer のご紹介と、javax パッケージ名前空間から jakarta パッケージ名前空間への変換例（Part 2）～Maven プラグインを利用～

Part1 | Part2

2024 年 6 月 19 日 初版

岡田 崚佑

Eclipse Transformer のご紹介（Part1）では、Jakarta EE 8 から Jakarta EE 10 への移行の課題と、その解決策となる Eclipse Transformer の紹介、および CLI での変換例を説明しました。CLI による変換は、ソースをまとめて変換できるため、Jakarta EE 10 移行後も継続的に機能追加や修正を加えていく場合に有効です。
一方、今後アプリケーションの修正や機能追加の予定がなく、Jakarta EE 8 以前の規約に準拠したアプリケーションでも Jakarta EE 10 準拠のアプリケーションサーバー上で動くのかを確認したい場合、 ビルドツールで変換をできたほうが都合がいいこともあります。 そこで今回は、Java のビルドツールのひとつである Maven のプラグインを利用した変換例をご紹介します。
Maven プラグインを利用することで、ソースコードは変えずに、生成物だけ Jakarta EE 10 に対応させることができます。
本記事では、Eclipse Transformer(Eclipse Foundation のページへ)のプラグインを使った生成物の変換例をご紹介します。
なお、本記事の内容は、Fujitsu Software Enterprise Application Platform V1.2.0 に適用することができます。

Maven と Maven-プラグイン

まずはじめに、Maven について簡単にご説明します。 Maven は Apache Foundation のプロジェクトのひとつであり、Java のビルドツールとして知られています。 Project Object Model(POM)の概念に基づいて、プロジェクトのビルドやテスト、ドキュメンテーション、成果物の配備などを管理できます。

また Maven には、ビルドライフサイクルの特定のフェーズに結びつけて実行できるプラグインがあります。 今回は、Transformer が提供している名前空間を変換するプラグインの使い方をご紹介します。

Eclipse Transformer Maven プラグインの使用方法

Maven で Eclipse Transformer 使用したい場合は、プロジェクトの pom.xml の plugins タグ配下に以下を記述してください。

```
<plugin>
    <groupId>org.eclipse.transformer</groupId>
    <artifactId>transformer-maven-plugin</artifactId>
    <version>0.5.0</version>
    <configuration>
        <rules>
            <jakartaDefaults>true</jakartaDefaults>
        </rules>
    </configuration>
    <executions>
        <execution>
            <id>default-transform</id>
            <goals>
                <goal>transform</goal>
            </goals>
        </execution>
    </executions>
</plugin>
```

Eclipse Transformer Maven plugin の使用例

Jakarta EE 8 以前のアプリケーションの POM ファイルに、使用方法で示したプラグインの記述を追加し、Jakarta EE 10 準拠の GlassFish 7.0.11 上で動作させる例をご紹介します。 なお、ここで紹介するプログラムの完全なソースコードや手順は以下で参照できます。

- https://github.com/fujitsu/app_blog/tree/master/maven-transformer-202406/（GitHub のページ）

ここでは、Jakarta EE 8 の実装として GlassFish 5.1 を、Jakarta EE 10 の実装として GlassFish 7.0.11 を使います。
それぞれの実装を以下のリンクからダウンロードし、任意のディレクトリに展開してください。

- GlassFish 5.1
- GlassFish 7.0.11

また、GlassFish 5.1 では JDK8 が、GlassFish 7.0.11 では JDK11 以降がそれぞれ必要になるため、適宜用意してください。 用意した JDK のパスを、glassfish5/glassfish/config/asenv.conf と glassfish7/glassfish/config/asenv.conf の AS_JAVA に設定してください。以下は GlassFish 5.1 の設定例です。

```
...
AS_JAVA="/work/jdk8u302-b08"
AS_IMQ_LIB="../../mq/lib"
...
```

GlassFish 5.1 にアプリケーションを配備するには、以下のコマンドを実行してください。

```
$ $PATH_TO_GF5/glassfish5/glassfish/bin/asadmin start-domain
$ $PATH_TO_GF5/glassfish5/glassfish/bin/asadmin deploy --contextroot=ee8 --name=ee8app sample-app-ee8-1.0.war
```

GlassFish はデフォルトで 8080 ポートを使用しますので、この Web アプリケーションにアクセスするには、curl コマンドなどで以下のように実行します。

```
$ curl http://localhost:8080/ee8/hello
Hello !
java.version : 1.8.0_302
glassfish.version : GlassFish Server Open Source Edition  5.1.0  (build default-private)
```

GlassFish 5.1 上に配備されたアプリケーションにアクセスすると、JDK のバージョンと、GlassFish のバージョンが出力されることが確認できました。

続いて、同じアプリケーションを GlassFish 7.0.11 で実行してみます。GlassFish 5.1 の停止、および GlassFish 7.0.11 にアプリケーションを配備するには、以下のコマンドを実行してください。

```
$ $PATH_TO_GF5/glassfish5/glassfish/bin/asadmin stop-domain
$ $PATH_TO_GF7/glassfish7/glassfish/bin/asadmin start-domain
$ $PATH_TO_GF7/glassfish7/glassfish/bin/asadmin deploy --contextroot=ee8 --name=ee8app sample-app-ee8-1.0.war
```


GlassFish 5.1 の時と同様に、curl コマンドなどで以下のようにアクセスします。

```
$ curl http://localhost:8080/ee8app/hello
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
strict.dtd"><html xmlns="http://www.w3.org/1999/xhtml"><head><title>Eclipse GlassFish 7.0.11 - Error
report</title><style type="text/css"><!--H1 {font-family:Tahoma,Arial,sans-serif;color:white;background-
color:#525D76;font-size:22px;} H2 {font-family:Tahoma,Arial,sans-serif;color:white;background-
color:#525D76;font-size:16px;} H3 {font-family:Tahoma,Arial,sans-serif;color:white;background-
color:#525D76;font-size:14px;} BODY {font-family:Tahoma,Arial,sans-serif;color:black;background-color:white;} B
{font-family:Tahoma,Arial,sans-serif;color:white;background-color:#525D76;} P {font-family:Tahoma,Arial,sans-
serif;background:white;color:black;font-size:12px;}A {color : black;}HR {color : #525D76;}--></style>
</head><body><h1>HTTP Status 500 - Internal Server Error</h1><hr/><p><b>type</b> Exception
report</p><p><b>message</b>Internal Server Error</p><p><b>description</b>The server encountered an internal
error that prevented it from fulfilling this request.</p><p><b>exception</b>
<pre>jakarta.servlet.ServletException: Error allocating a servlet instance</pre></p><p><b>root cause</b>
<pre>java.lang.NoClassDefFoundError: javax/servlet/http/HttpServlet</pre></p><p><b>root cause</b>
<pre>java.lang.ClassNotFoundException: javax.servlet.http.HttpServlet</pre></p><p><b>note</b> <u>The full stack
traces of the exception and its root causes are available in the Eclipse GlassFish 7.0.11
logs.</u></p><hr/><h3>Eclipse GlassFish 7.0.11</h3></body></html>
```

GlassFish 7.0.11 では、ClassNotFoundException が出力され、エラーページが返ってきます。
これは、Jakarta EE 10 では、javax パッケージ名前空間のクラスを使用できないため発生するエラーです。これを解決するために、Eclipse Transformer の Maven プラグインによって、アプリケーション内の javax パッケージ名前空間を jakarta パッケージ名前空間に書き換えます。 https://github.com/fujitsu/app_blog/tree/master/maven-plugin-202406 の sample-app ディレクトリ配下の pom.xml に以下を書き足すか、 ビルド時に [-f] オプションで pom-transformer.xml を指定してください。

```
<plugins>
      <plugin>
          <groupId>org.apache.maven.plugins</groupId>
          <artifactId>maven-war-plugin</artifactId>
          <version>3.3.2</version>
          <configuration>
              <failOnMissingWebXml>false</failOnMissingWebXml>
          </configuration>
      </plugin>
+     <plugin>
+         <groupId>org.eclipse.transformer</groupId>
+         <artifactId>transformer-maven-plugin</artifactId>
+         <version>0.5.0</version>
+         <configuration>
+             <rules>
+                 <jakartaDefaults>true</jakartaDefaults>
+             </rules>
+         </configuration>
+         <executions>
+             <execution>
+                 <id>default-transform</id>
+                 <goals>
+                     <goal>transform</goal>
+                 </goals>
+             </execution>
+         </executions>
+     </plugin>
  </plugins>
```


再度ビルドしたアプリケーションを用いて、GlassFish 5.1、GlassFish 7.0.11 で動作確認します。 まずは、GlassFish 5.1 にアプリケーションを配備し、curl コマンドでアクセスします。

```
$ $PATH_TO_GF5/glassfish7/glassfish/bin/asadmin stop-domain
$ $PATH_TO_GF5/glassfish5/glassfish/bin/asadmin start-domain
$ $PATH_TO_GF5/glassfish5/glassfish/bin/asadmin undeploy ee8app
$ $PATH_TO_GF5/glassfish5/glassfish/bin/asadmin deploy --contextroot=ee10 --name=ee10app sample-app-ee10-1.0.war
$ curl http://localhost:8080/ee10/hello
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
strict.dtd"><html xmlns="http://www.w3.org/1999/xhtml"><head><title>GlassFish Server Open Source Edition  5.1.0
- Error report</title><style type="text/css"><!--H1 {font-family:Tahoma,Arial,sans-serif;color:white;background-
color:#525D76;font-size:22px;} H2 {font-family:Tahoma,Arial,sans-serif;color:white;background-
color:#525D76;font-size:16px;} H3 {font-family:Tahoma,Arial,sans-serif;color:white;background-
color:#525D76;font-size:14px;} BODY {font-family:Tahoma,Arial,sans-serif;color:black;background-color:white;} B
{font-family:Tahoma,Arial,sans-serif;color:white;background-color:#525D76;} P {font-family:Tahoma,Arial,sans-
serif;background:white;color:black;font-size:12px;}A {color : black;}HR {color : #525D76;}--></style>
</head><body><h1>HTTP Status 500 - Internal Server Error</h1><hr/><p><b>type</b> Exception
report</p><p><b>message</b>Internal Server Error</p><p><b>description</b>The server encountered an internal
error that prevented it from fulfilling this request.</p><p><b>exception</b>
<pre>javax.servlet.ServletException: Error allocating a servlet instance</pre></p><p><b>root cause</b>
<pre>java.lang.NoClassDefFoundError: jakarta/servlet/http/HttpServlet</pre></p><p><b>root cause</b>
<pre>java.lang.ClassNotFoundException: jakarta.servlet.http.HttpServlet</pre></p><p><b>note</b> <u>The full
stack traces of the exception and its root causes are available in the GlassFish Server Open Source Edition
5.1.0  logs.</u></p><hr/><h3>GlassFish Server Open Source Edition  5.1.0 </h3></body></html>
```

先ほどとは違い、GlassFish 5.1 で ClassNotFoundException が出力され、エラーページが返ってきました。今度は GlassFish 7.0.11 で同様の操作をします。

```
$ $PATH_TO_GF5/glassfish5/glassfish/bin/asadmin stop-domain
$ $PATH_TO_GF7/glassfish7/glassfish/bin/asadmin start-domain
$ $PATH_TO_GF7/glassfish7/glassfish/bin/asadmin undeploy ee8app
$ $PATH_TO_GF7/glassfish7/glassfish/bin/asadmin deploy --contextroot=ee10 --name=ee10app sample-app-ee10-1.0.war
$ curl http://localhost:8080/ee10/hello
Hello !
java.version : 17.0.2
glassfish.version : Eclipse GlassFish 7.0.11 (commit: ad98c44e896c7a78cb4eb2a84ca9fac450fb3a10)
```

今度は、GlassFish 7.0.11 に配備したアプリケーションで JDK のバージョンと、GlassFish のバージョンが出力されることが確認できました。 このように、ソースファイルを書き換えずにアプリケーションの javax パッケージ名前空間を jakarta パッケージ名前空間に変換することができました。

最後に

本投稿では、Jakarta EE 8 以前で使用されていた javax パッケージ名前空間を含むアプリを、Jakarta EE 10 以降で使用される jakarta パッケージ名前空間に変換する、Eclipse Transformer の Maven プラグインについて、 その使い方を説明しました。 Eclipse Transformer プロジェクトには、富士通のエンジニアもコミッターとして参加しており、品質向上に努めています。既存のアプリケーションを jakarta パッケージ名前空間に移行する際には、ぜひお試しください。