MyListener -

package com.javatpoint;

**import** jakarta.servlet.\*;

**import** java.sql.\*;

public class MyListener implements ServletContextListener{

public void contextInitialized(ServletContextEvent arg0) {

Connection con=null;

try{

ResultSet rs;

Class.forName("com.mysql.jdbc.Driver");

con=DriverManager.getConnection("jdbc:mysql://localhost:3306/ecommerce", "root", "$DunH7vz4M");

PreparedStatement ps1=con.prepareStatement("Select \* from QUIZCONTACT");

rs=ps1.executeQuery();

if(rs.next())

{System.out.println("your table name already exist");}

else

{System.out.println("else if part table does not exist new table has created");

PreparedStatement ps=con.prepareStatement("CREATE TABLE QUIZCONTACT (NAME VARCHAR(4000),EMAIL VARCHAR(4000),PHONE VARCHAR(4000),MESSAGE VARCHAR(4000))");

ps.executeUpdate();

PreparedStatement ps4=con.prepareStatement("CREATE TABLE QUIZINFO (SUBJECT VARCHAR(4000),QUIZNAME VARCHAR(4000))");

ps4.executeUpdate();

PreparedStatement ps5=con.prepareStatement("CREATE TABLE QUIZQ(NAME VARCHAR(4000),EMAIL VARCHAR(4000),PHONE VARCHAR(4000),QUESTION VARCHAR(4000))");

ps5.executeUpdate();

ps5= con.prepareStatement("CREATE TABLE QUIZQUES(QUESTION VARCHAR(4000),OPTION1 VARCHAR(4000),OPTION2 VARCHAR(4000),OPTION3 VARCHAR(4000),OPTION4 VARCHAR(4000),ANSWER VARCHAR(4000),QUIZNAME VARCHAR(4000),QID VARCHAR(4000),DESCRIPTION VARCHAR(4000),CONSTRAINT QUIZQUES\_PK PRIMARY KEY (QID) ENABLE)");

ps5.executeUpdate();

ps5= con.prepareStatement("CREATE TABLE QUIZREGISTER (USERNAME VARCHAR(4000),USERPASS VARCHAR(4000),CATEGORY VARCHAR(4000),EMAIL VARCHAR(4000))");

ps5.executeUpdate();

Statement stmt=con.createStatement();

}

}

catch(Exception e){

e.printStackTrace();

}

}

public void contextDestroyed(ServletContextEvent arg0) {

System.out.println("project undeployed");

}

}

Pom –

<!--

~ Copyright (c) 2012-2017 The ANTLR Project. All rights reserved.

~ Use of this file is governed by the BSD 3-clause license that

~ can be found in the LICENSE.txt file in the project root.

-->

<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4\_0\_0.xsd">

<modelVersion>4.0.0</modelVersion>

<parent>

<groupId>org.antlr</groupId>

<artifactId>antlr4-master</artifactId>

<version>4.12.0</version>

<relativePath>../../pom.xml</relativePath>

</parent>

<artifactId>antlr4-runtime</artifactId>

<name>ANTLR 4 Runtime</name>

<description>The ANTLR 4 Runtime</description>

<prerequisites>

<maven>3.8</maven>

</prerequisites>

<properties>

<!-- Assumes dot is in the system path, or specified for the build. -->

<dot.path>dot</dot.path>

</properties>

<build>

<sourceDirectory>src</sourceDirectory>

<plugins>

<plugin> <!-- create src jar -->

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-source-plugin</artifactId>

<version>3.2.1</version>

<executions>

<execution>

<goals>

<goal>jar</goal>

</goals>

</execution>

</executions>

</plugin>

<plugin> <!-- run dot on the images used by PredictionContext.java -->

<groupId>us.bryon</groupId>

<artifactId>graphviz-maven-plugin</artifactId>

<version>1.0</version>

<executions>

<execution>

<goals>

<goal>dot</goal>

</goals>

<configuration>

<dot>${dot.path}</dot>

<destdir>${project.build.directory}/apidocs</destdir>

<output>svg</output>

</configuration>

</execution>

</executions>

</plugin>

<plugin> <!-- create javadoc jar -->

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-javadoc-plugin</artifactId>

<version>3.3.1</version>

<configuration>

<javadocVersion>1.8</javadocVersion>

<failOnError>false</failOnError>

</configuration>

<executions>

<execution>

<goals>

<goal>javadoc</goal>

</goals>

</execution>

</executions>

</plugin>

<plugin>

<groupId>org.apache.felix</groupId>

<artifactId>maven-bundle-plugin</artifactId>

<version>5.1.7</version>

<executions>

<execution>

<id>bundle-manifest</id>

<phase>process-classes</phase>

<configuration>

<instructions>

<Automatic-Module-Name>org.antlr.antlr4.runtime</Automatic-Module-Name>

<Bundle-SymbolicName>org.antlr.antlr4-runtime</Bundle-SymbolicName>

<Import-Package>org.antlr.v4.gui;resolution:=optional, \*</Import-Package>

</instructions>

</configuration>

<goals>

<goal>manifest</goal>

</goals>

</execution>

</executions>

</plugin>

<plugin>

<artifactId>maven-jar-plugin</artifactId>

<version>3.2.0</version>

<configuration>

<archive>

<manifest>

<addDefaultImplementationEntries>true</addDefaultImplementationEntries>

</manifest>

<manifestFile>${project.build.outputDirectory}/META-INF/MANIFEST.MF</manifestFile>

</archive>

</configuration>

</plugin>

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-compiler-plugin</artifactId>

<version>3.8.1</version>

<configuration>

<release>8</release>

<source>1.8</source>

<target>1.8</target>

</configuration>

</plugin>

</plugins>

</build>

</project>

Pom –

<?xml version="1.0" encoding="UTF-8"?>

<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<parent>

<groupId>net.bytebuddy</groupId>

<artifactId>byte-buddy-parent</artifactId>

<version>1.14.2</version>

</parent>

<!--

About precompilation:

This **module** contains compiled class files which require a given Java version **to** meet all properties. To avoid

that the build target Java version conflicts **with** this required version, precompiled files are used unless

those files are supposed **to** be built from source which can be activated by a given profile. This functionality

is mainly required for tests but is also used for adding some Advice templates.

Unfortunately, it is not possible in Maven **to** disable a profile from another profile. Therefore, profiles must be

activated by setting properties **with** the same name as the profiles for precompilation. This **requires** a minimal

change in syntax by substituting a 'P' **with** a 'D'.

It is also possible **to** compile the native method sample by using the 'native-compile' profile on Linux.

-->

<artifactId>byte-buddy-dep</artifactId>

<packaging>jar</packaging>

<name>Byte Buddy (**with** dependencies)</name>

<description>

Byte Buddy is a Java library for creating Java classes at run time.

This artifact is a build of Byte Buddy **with** a remaining dependency onto ASM.

You should never depend on this **module** without repackaging Byte Buddy and ASM into your own namespace.

</description>

<dependencies>

<dependency>

<groupId>org.ow2.asm</groupId>

<artifactId>asm</artifactId>

<version>${version.asm}</version>

</dependency>

<dependency>

<groupId>org.ow2.asm</groupId>

<artifactId>asm-commons</artifactId>

<version>${version.asm}</version>

<exclusions>

<exclusion>

<groupId>org.ow2.asm</groupId>

<artifactId>asm-analysis</artifactId>

</exclusion>

<exclusion>

<groupId>org.ow2.asm</groupId>

<artifactId>asm-tree</artifactId>

</exclusion>

</exclusions>

</dependency>

<dependency>

<groupId>net.java.dev.jna</groupId>

<artifactId>jna</artifactId>

<version>${version.jna}</version>

<scope>provided</scope>

</dependency>

<dependency>

<groupId>net.java.dev.jna</groupId>

<artifactId>jna-platform</artifactId>

<version>${version.jna}</version>

<scope>provided</scope>

</dependency>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>${version.junit}</version>

<scope>test</scope>

</dependency>

<dependency>

<groupId>org.mockito</groupId>

<artifactId>mockito-core</artifactId>

<version>${version.mockito}</version>

<scope>test</scope>

<exclusions>

<exclusion>

<groupId>net.bytebuddy</groupId>

<artifactId>byte-buddy</artifactId>

</exclusion>

<exclusion>

<groupId>net.bytebuddy</groupId>

<artifactId>byte-buddy-agent</artifactId>

</exclusion>

</exclusions>

</dependency>

<dependency>

<groupId>org.ow2.asm</groupId>

<artifactId>asm-util</artifactId>

<version>${version.asm}</version>

<scope>test</scope>

</dependency>

<dependency>

<groupId>org.ow2.asm</groupId>

<artifactId>asm-analysis</artifactId>

<version>${version.asm}</version>

<scope>test</scope>

</dependency>

<dependency>

<groupId>${project.groupId}</groupId>

<artifactId>byte-buddy-agent</artifactId>

<scope>test</scope>

</dependency>

</dependencies>

<build>

<resources>

<resource>

<directory>src/main/resources</directory>

</resource>

<resource>

<directory>..</directory>

<targetPath>META-INF</targetPath>

<filtering>true</filtering>

<includes>

<include>LICENSE</include>

<include>NOTICE</include>

</includes>

</resource>

</resources>

<plugins>

<!-- Create hashCode/equals method and cache some expensive properties. -->

<plugin>

<groupId>net.bytebuddy</groupId>

<artifactId>byte-buddy-maven-plugin</artifactId>

<version>1.14.1</version>

<executions>

<execution>

<phase>compile</phase>

<goals>

<goal>transform</goal>

</goals>

<configuration>

<initialization>

<entryPoint>REDEFINE</entryPoint>

</initialization>

<transformations>

<transformation>

<groupId>net.bytebuddy</groupId>

<artifactId>byte-buddy</artifactId>

<version>1.14.1</version>

<plugin>net.bytebuddy.build.HashCodeAndEqualsPlugin$WithNonNullableFields</plugin>

<arguments>

<argument>

<index>0</index>

<value>net.bytebuddy.utility.nullability.MaybeNull</value>

</argument>

</arguments>

</transformation>

<transformation>

<groupId>net.bytebuddy</groupId>

<artifactId>byte-buddy</artifactId>

<version>1.14.1</version>

<plugin>net.bytebuddy.build.CachedReturnPlugin</plugin>

</transformation>

<transformation>

<groupId>net.bytebuddy</groupId>

<artifactId>byte-buddy</artifactId>

<version>1.14.1</version>

<plugin>net.bytebuddy.build.AccessControllerPlugin</plugin>

<arguments>

<argument>

<index>0</index>

<value>net.bytebuddy.securitymanager</value>

</argument>

</arguments>

</transformation>

<transformation>

<groupId>net.bytebuddy</groupId>

<artifactId>byte-buddy</artifactId>

<version>1.14.1</version>

<plugin>net.bytebuddy.build.DispatcherAnnotationPlugin</plugin>

</transformation>

<transformation>

<groupId>net.bytebuddy</groupId>

<artifactId>byte-buddy</artifactId>

<version>1.14.1</version>

<plugin>net.bytebuddy.build.RepeatedAnnotationPlugin</plugin>

</transformation>

</transformations>

</configuration>

</execution>

</executions>

</plugin>

</plugins>

<pluginManagement>

<plugins>

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-javadoc-plugin</artifactId>

<version>${version.plugin.javadoc}</version>

<configuration>

<offlineLinks>

<!-- Use previously downloaded 'package-list' file as substitute for published 'element-list' file. -->

<link>

<url>https://javadoc.io/doc/org.ow2.asm/asm/${version.asm}</url>

<location>${project.basedir}/target/javadoc-lists/asm-${version.asm}</location>

</link>

</offlineLinks>

<links>

<link>https://javadoc.io/doc/net.java.dev.jna/jna/${version.jna}</link>

</links>

</configuration>

</plugin>

</plugins>

</pluginManagement>

</build>

<profiles>

<profile>

<id>build-java-7</id>

<activation>

<activeByDefault>false</activeByDefault>

<jdk>1.7</jdk>

</activation>

<build>

<plugins>

<!-- Avoids occasional segfault. -->

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-surefire-plugin</artifactId>

<version>${version.plugin.surefire}</version>

<configuration>

<excludes>

<exclude>net.bytebuddy.agent.builder.AgentBuilderDefaultApplicationTest</exclude>

<exclude>net.bytebuddy.pool.TypePoolDefaultAnnotationDescriptionTest</exclude>

</excludes>

</configuration>

</plugin>

</plugins>

</build>

</profile>

<profile>

<id>build-java-11</id>

<activation>

<activeByDefault>false</activeByDefault>

<jdk>11</jdk>

</activation>

<build>

<plugins>

<!-- Avoids occasional segfault. -->

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-surefire-plugin</artifactId>

<version>${version.plugin.surefire}</version>

<configuration>

<excludes>

<exclude>net.bytebuddy.description.type.TypeDescriptionArrayProjectionTest</exclude>

<exclude>net.bytebuddy.description.annotation.AnnotationDescriptionForLoadedAnnotationTest</exclude>

<exclude>net.bytebuddy.description.annotation.AnnotationDescriptionForLoadedAnnotationDifferentClassLoaderTest</exclude>

<exclude>net.bytebuddy.description.annotation.AnnotationDescriptionLatentTest</exclude>

<exclude>net.bytebuddy.pool.TypePoolDefaultAnnotationDescriptionTest</exclude>

</excludes>

</configuration>

</plugin>

</plugins>

</build>

</profile>

<profile>

<id>extras</id>

<activation>

<activeByDefault>false</activeByDefault>

</activation>

<build>

<plugins>

<!-- Download 'element-list' file which is published by newer javadoc versions, store as 'package-list' file. -->

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-antrun-plugin</artifactId>

<version>${version.plugin.antrun}</version>

<executions>

<execution>

<id>download-javadoc-lists</id>

<phase>process-resources</phase>

<goals>

<goal>run</goal>

</goals>

<configuration>

<skip>${javadoc.download.skip}</skip>

<target>

<mkdir dir="${project.basedir}/target/javadoc-lists/asm-${version.asm}" />

<get src="https://javadoc.io/doc/org.ow2.asm/asm/${version.asm}/element-list" dest="${project.basedir}/target/javadoc-lists/asm-${version.asm}/package-list" skipexisting="true" />

</target>

</configuration>

</execution>

</executions>

</plugin>

</plugins>

</build>

</profile>

<profile>

<id>native-compile</id>

<activation>

<activeByDefault>false</activeByDefault>

</activation>

<build>

<plugins>

<plugin>

<groupId>org.codehaus.mojo</groupId>

<artifactId>exec-maven-plugin</artifactId>

<version>${version.plugin.exec}</version>

<executions>

<execution>

<id>compile-test</id>

<phase>compile</phase>

<goals>

<goal>exec</goal>

</goals>

<configuration>

<executable>gcc</executable>

<arguments>

<argument>-fPIC</argument>

<argument>-I${java.home}/../include</argument>

<argument>-I${java.home}/../include/linux</argument>

<argument>-I${project.basedir}/src/test/c</argument>

<argument>-shared</argument>

<argument>-o</argument>

<argument>${project.basedir}/src/test/resources/net\_bytebuddy\_test\_c\_NativeSample.so</argument>

<argument>${project.basedir}/src/test/c/net\_bytebuddy\_test\_c\_NativeSample.c</argument>

</arguments>

</configuration>

</execution>

</executions>

</plugin>

</plugins>

</build>

</profile>

<!-- Precompilation configuration for Java 1.4 class files. -->

<profile>

<id>java-4-precompile</id>

<activation>

<activeByDefault>false</activeByDefault>

<property>

<name>java-4-precompile</name>

</property>

</activation>

<properties>

<sourcecode.test.version>1.4</sourcecode.test.version>

<bytecode.test.version>1.4</bytecode.test.version>

</properties>

<build>

<plugins>

<plugin>

<groupId>org.codehaus.mojo</groupId>

<artifactId>build-helper-maven-plugin</artifactId>

<version>${version.plugin.buildhelper}</version>

<executions>

<execution>

<id>java-4-precompile-test</id>

<phase>generate-sources</phase>

<goals>

<goal>add-test-source</goal>

</goals>

<configuration>

<sources>

<source>src/test/java-4</source>

</sources>

</configuration>

</execution>

</executions>

</plugin>

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-compiler-plugin</artifactId>

<version>${version.plugin.compiler}</version>

<executions>

<execution>

<id>default-testCompile</id>

<phase>test-compile</phase>

<goals>

<goal>testCompile</goal>

</goals>

<configuration>

<testIncludes>

<include>net/bytebuddy/test/precompiled/v4/\*\*</include>

</testIncludes>

</configuration>

</execution>

</executions>

</plugin>

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-antrun-plugin</artifactId>

<version>${version.plugin.antrun}</version>

<executions>

<execution>

<id>java-4-precompile-test-copy</id>

<phase>process-test-classes</phase>

<goals>

<goal>run</goal>

</goals>

<configuration>

<target>

<copy todir="${project.basedir}/src/test/precompiled-4">

<fileset dir="${project.build.testOutputDirectory}" includes="net/bytebuddy/test/precompiled/v4/\*.class" />

</copy>

</target>

</configuration>

</execution>

</executions>

</plugin>

</plugins>

</build>

</profile>

<profile>

<id>java-4-no-precompile</id>

<activation>

<activeByDefault>true</activeByDefault>

<property>

<name>!java-4-precompile</name>

</property>

</activation>

<build>

<testResources>

<testResource>

<directory>src/test/precompiled-4</directory>

</testResource>

</testResources>

</build>

</profile>

<!-- Precompilation configuration for Java 1.4 (JSR 14) class files. -->

<profile>

<id>java-4-jsr14-precompile</id>

<activation>

<activeByDefault>false</activeByDefault>

<property>

<name>java-4-jsr14-precompile</name>

</property>

</activation>

<properties>

<version.plugin.compiler>2.0.1</version.plugin.compiler>

<sourcecode.test.version>1.4</sourcecode.test.version>

<bytecode.test.version>jsr14</bytecode.test.version>

</properties>

<build>

<plugins>

<plugin>

<groupId>org.codehaus.mojo</groupId>

<artifactId>build-helper-maven-plugin</artifactId>

<version>${version.plugin.buildhelper}</version>

<executions>

<execution>

<id>java-4-jsr14-precompile-test</id>

<phase>generate-sources</phase>

<goals>

<goal>add-test-source</goal>

</goals>

<configuration>

<sources>

<source>src/test/java-4-jsr14</source>

</sources>

</configuration>

</execution>

</executions>

</plugin>

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-compiler-plugin</artifactId>

<version>${version.plugin.compiler}</version>

<executions>

<execution>

<id>default-testCompile</id>

<phase>test-compile</phase>

<goals>

<goal>testCompile</goal>

</goals>

<configuration>

<target>${bytecode.test.version}</target>

<testIncludes>

<include>net/bytebuddy/test/precompiled/v4jsr14/\*\*</include>

</testIncludes>

</configuration>

</execution>

</executions>

</plugin>

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-antrun-plugin</artifactId>

<version>${version.plugin.antrun}</version>

<executions>

<execution>

<id>java-4-precompile-test-copy</id>

<phase>process-test-classes</phase>

<goals>

<goal>run</goal>

</goals>

<configuration>

<target>

<copy todir="${project.basedir}/src/test/precompiled-4-jsr14">

<fileset dir="${project.build.testOutputDirectory}" includes="net/bytebuddy/test/precompiled/v4jsr14/\*.class" />

</copy>

</target>

</configuration>

</execution>

</executions>

</plugin>

</plugins>

</build>

</profile>

<profile>

<id>java-4-jsr14-no-precompile</id>

<activation>

<activeByDefault>true</activeByDefault>

<property>

<name>!java-4-jsr14-precompile</name>

</property>

</activation>

<build>

<testResources>

<testResource>

<directory>src/test/precompiled-4-jsr14</directory>

</testResource>

</testResources>

</build>

</profile>

<!-- Precompilation configuration for Java 1.6 class files. -->

<profile>

<id>java-6-precompile</id>

<activation>

<activeByDefault>false</activeByDefault>

<property>

<name>java-6-precompile</name>

</property>

</activation>

<properties>

<bytecode.main.version>1.6</bytecode.main.version>

<bytecode.test.version>1.6</bytecode.test.version>

</properties>

<build>

<plugins>

<plugin>

<groupId>org.codehaus.mojo</groupId>

<artifactId>build-helper-maven-plugin</artifactId>

<version>${version.plugin.buildhelper}</version>

<executions>

<execution>

<id>java-6-precompile</id>

<phase>generate-sources</phase>

<goals>

<goal>add-source</goal>

</goals>

<configuration>

<sources>

<source>src/main/java-6</source>

</sources>

</configuration>

</execution>

<execution>

<id>java-6-precompile-test</id>

<phase>generate-sources</phase>

<goals>

<goal>add-test-source</goal>

</goals>

<configuration>

<sources>

<source>src/test/java-6</source>

</sources>

</configuration>

</execution>

</executions>

</plugin>

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-antrun-plugin</artifactId>

<version>${version.plugin.antrun}</version>

<executions>

<execution>

<id>java-6-precompile-copy</id>

<phase>process-classes</phase>

<goals>

<goal>run</goal>

</goals>

<configuration>

<target>

<copy todir="${project.basedir}/src/main/precompiled-6">

<fileset dir="${project.build.outputDirectory}" includes="net/bytebuddy/build/CachedReturnPlugin$Advice$\*.class" />

</copy>

</target>

</configuration>

</execution>

<execution>

<id>java-6-precompile-test-copy</id>

<phase>process-test-classes</phase>

<goals>

<goal>run</goal>

</goals>

<configuration>

<target>

<copy todir="${project.basedir}/src/test/precompiled-6">

<fileset dir="${project.build.testOutputDirectory}" includes="net/bytebuddy/test/precompiled/v6/\*.class" />

</copy>

</target>

</configuration>

</execution>

</executions>

</plugin>

</plugins>

</build>

</profile>

<profile>

<id>java-6-no-precompile</id>

<activation>

<activeByDefault>true</activeByDefault>

<property>

<name>!java-6-precompile</name>

</property>

</activation>

<build>

<resources>

<resource>

<directory>src/main/precompiled-6</directory>

</resource>

</resources>

<testResources>

<testResource>

<directory>src/test/precompiled-6</directory>

</testResource>

</testResources>

</build>

</profile>

<!-- Precompilation configuration for Java 1.7 class files. -->

<profile>

<id>java-7-precompile</id>

<activation>

<activeByDefault>false</activeByDefault>

<property>

<name>java-7-precompile</name>

</property>

</activation>

<properties>

<bytecode.main.version>1.7</bytecode.main.version>

<bytecode.test.version>1.7</bytecode.test.version>

</properties>

<build>

<plugins>

<plugin>

<groupId>org.codehaus.mojo</groupId>

<artifactId>build-helper-maven-plugin</artifactId>

<version>${version.plugin.buildhelper}</version>

<executions>

<execution>

<id>java-7-precompile-test</id>

<phase>generate-sources</phase>

<goals>

<goal>add-test-source</goal>

</goals>

<configuration>

<sources>

<source>src/test/java-7</source>

</sources>

</configuration>

</execution>

</executions>

</plugin>

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-antrun-plugin</artifactId>

<version>${version.plugin.antrun}</version>

<executions>

<execution>

<id>java-7-precompile-test-copy</id>

<phase>process-test-classes</phase>

<goals>

<goal>run</goal>

</goals>

<configuration>

<target>

<copy todir="${project.basedir}/src/test/precompiled-7">

<fileset dir="${project.build.testOutputDirectory}" includes="net/bytebuddy/test/precompiled/v7/\*.class" />

</copy>

</target>

</configuration>

</execution>

</executions>

</plugin>

</plugins>

</build>

</profile>

<profile>

<id>java-7-no-precompile</id>

<activation>

<activeByDefault>true</activeByDefault>

<property>

<name>!java-7-precompile</name>

</property>

</activation>

<build>

<testResources>

<testResource>

<directory>src/test/precompiled-7</directory>

</testResource>

</testResources>

</build>

</profile>

<!-- Precompilation configuration for Java 1.8 class files. -->

<profile>

<id>java-8-precompile</id>

<activation>

<activeByDefault>false</activeByDefault>

<property>

<name>java-8-precompile</name>

</property>

</activation>

<properties>

<sourcecode.test.version>1.8</sourcecode.test.version>

<bytecode.test.version>1.8</bytecode.test.version>

</properties>

<build>

<plugins>

<plugin>

<groupId>org.codehaus.mojo</groupId>

<artifactId>build-helper-maven-plugin</artifactId>

<version>${version.plugin.buildhelper}</version>

<executions>

<execution>

<id>java-8-precompile-test</id>

<phase>generate-sources</phase>

<goals>

<goal>add-test-source</goal>

</goals>

<configuration>

<sources>

<source>src/test/java-8</source>

</sources>

</configuration>

</execution>

</executions>

</plugin>

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-antrun-plugin</artifactId>

<version>${version.plugin.antrun}</version>

<executions>

<execution>

<id>java-8-precompile-test-copy</id>

<phase>process-test-classes</phase>

<goals>

<goal>run</goal>

</goals>

<configuration>

<target>

<copy todir="${project.basedir}/src/test/precompiled-8">

<fileset dir="${project.build.testOutputDirectory}" includes="net/bytebuddy/test/precompiled/v8/\*.class" />

</copy>

</target>

</configuration>

</execution>

</executions>

</plugin>

</plugins>

</build>

</profile>

<profile>

<id>java-8-no-precompile</id>

<activation>

<activeByDefault>true</activeByDefault>

<property>

<name>!java-8-precompile</name>

</property>

</activation>

<build>

<testResources>

<testResource>

<directory>src/test/precompiled-8</directory>

</testResource>

</testResources>

</build>

</profile>

<!-- Precompilation configuration for Java 1.8 (with parameter names) class files. -->

<profile>

<id>java-8-parameters-precompile</id>

<activation>

<activeByDefault>false</activeByDefault>

<property>

<name>java-8-parameters-precompile</name>

</property>

</activation>

<properties>

<sourcecode.test.version>1.8</sourcecode.test.version>

<bytecode.test.version>1.8</bytecode.test.version>

</properties>

<build>

<plugins>

<plugin>

<groupId>org.codehaus.mojo</groupId>

<artifactId>build-helper-maven-plugin</artifactId>

<version>${version.plugin.buildhelper}</version>

<executions>

<execution>

<id>java-8-parameters-precompile-test</id>

<phase>generate-sources</phase>

<goals>

<goal>add-test-source</goal>

</goals>

<configuration>

<sources>

<source>src/test/java-8-parameters</source>

</sources>

</configuration>

</execution>

</executions>

</plugin>

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-compiler-plugin</artifactId>

<version>${version.plugin.compiler}</version>

<executions>

<execution>

<id>default-testCompile</id>

<phase>test-compile</phase>

<goals>

<goal>testCompile</goal>

</goals>

<configuration>

<parameters>true</parameters>

<testIncludes>

<include>net/bytebuddy/test/precompiled/v8parameters/\*\*</include>

</testIncludes>

</configuration>

</execution>

</executions>

</plugin>

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-antrun-plugin</artifactId>

<version>${version.plugin.antrun}</version>

<executions>

<execution>

<id>java-8-parameters-precompile-test-copy</id>

<phase>process-test-classes</phase>

<goals>

<goal>run</goal>

</goals>

<configuration>

<target>

<copy todir="${project.basedir}/src/test/precompiled-8-parameters">

<fileset dir="${project.build.testOutputDirectory}" includes="net/bytebuddy/test/precompiled/v8parameters/\*.class" />

</copy>

</target>

</configuration>

</execution>

</executions>

</plugin>

</plugins>

</build>

</profile>

<profile>

<id>java-8-parameters-no-precompile</id>

<activation>

<activeByDefault>true</activeByDefault>

<property>

<name>!java-8-parameters-precompile</name>

</property>

</activation>

<build>

<testResources>

<testResource>

<directory>src/test/precompiled-8-parameters</directory>

</testResource>

</testResources>

</build>

</profile>

<!-- Precompilation configuration for Java 11 class files. -->

<profile>

<id>java-11-precompile</id>

<activation>

<activeByDefault>false</activeByDefault>

<property>

<name>java-11-precompile</name>

</property>

</activation>

<properties>

<sourcecode.test.version>11</sourcecode.test.version>

<bytecode.test.version>11</bytecode.test.version>

</properties>

<build>

<plugins>

<plugin>

<groupId>org.codehaus.mojo</groupId>

<artifactId>build-helper-maven-plugin</artifactId>

<version>${version.plugin.buildhelper}</version>

<executions>

<execution>

<id>java-11-precompile-test</id>

<phase>generate-sources</phase>

<goals>

<goal>add-test-source</goal>

</goals>

<configuration>

<sources>

<source>src/test/java-11</source>

</sources>

</configuration>

</execution>

</executions>

</plugin>

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-antrun-plugin</artifactId>

<version>${version.plugin.antrun}</version>

<executions>

<execution>

<id>java-11-precompile-test-copy</id>

<phase>process-test-classes</phase>

<goals>

<goal>run</goal>

</goals>

<configuration>

<target>

<copy todir="${project.basedir}/src/test/precompiled-11">

<fileset dir="${project.build.testOutputDirectory}" includes="net/bytebuddy/test/precompiled/v11/\*.class" />

</copy>

</target>

</configuration>

</execution>

</executions>

</plugin>

</plugins>

</build>

</profile>

<profile>

<id>java-11-no-precompile</id>

<activation>

<activeByDefault>true</activeByDefault>

<property>

<name>!java-11-precompile</name>

</property>

</activation>

<build>

<testResources>

<testResource>

<directory>src/test/precompiled-11</directory>

</testResource>

</testResources>

</build>

</profile>

<!-- Precompilation configuration for Java 16 class files. -->

<profile>

<id>java-16-precompile</id>

<activation>

<activeByDefault>false</activeByDefault>

<property>

<name>java-16-precompile</name>

</property>

</activation>

<properties>

<sourcecode.test.version>16</sourcecode.test.version>

<bytecode.test.version>16</bytecode.test.version>

</properties>

<build>

<plugins>

<plugin>

<groupId>org.codehaus.mojo</groupId>

<artifactId>build-helper-maven-plugin</artifactId>

<version>${version.plugin.buildhelper}</version>

<executions>

<execution>

<id>java-16-precompile-test</id>

<phase>generate-sources</phase>

<goals>

<goal>add-test-source</goal>

</goals>

<configuration>

<sources>

<source>src/test/java-16</source>

</sources>

</configuration>

</execution>

</executions>

</plugin>

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-antrun-plugin</artifactId>

<version>${version.plugin.antrun}</version>

<executions>

<execution>

<id>java-16-precompile-test-copy</id>

<phase>process-test-classes</phase>

<goals>

<goal>run</goal>

</goals>

<configuration>

<target>

<copy todir="${project.basedir}/src/test/precompiled-16">

<fileset dir="${project.build.testOutputDirectory}" includes="net/bytebuddy/test/precompiled/v16/\*.class" />

</copy>

</target>

</configuration>

</execution>

</executions>

</plugin>

</plugins>

</build>

</profile>

<profile>

<id>java-16-no-precompile</id>

<activation>

<activeByDefault>true</activeByDefault>

<property>

<name>!java-16-precompile</name>

</property>

</activation>

<build>

<testResources>

<testResource>

<directory>src/test/precompiled-16</directory>

</testResource>

</testResources>

</build>

</profile>

<!-- Precompilation configuration for Java 17 class files. -->

<profile>

<id>java-17-precompile</id>

<activation>

<activeByDefault>false</activeByDefault>

<property>

<name>java-17-precompile</name>

</property>

</activation>

<properties>

<sourcecode.test.version>17</sourcecode.test.version>

<bytecode.test.version>17</bytecode.test.version>

</properties>

<build>

<plugins>

<plugin>

<groupId>org.codehaus.mojo</groupId>

<artifactId>build-helper-maven-plugin</artifactId>

<version>${version.plugin.buildhelper}</version>

<executions>

<execution>

<id>java-17-precompile-test</id>

<phase>generate-sources</phase>

<goals>

<goal>add-test-source</goal>

</goals>

<configuration>

<sources>

<source>src/test/java-17</source>

</sources>

</configuration>

</execution>

</executions>

</plugin>

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-antrun-plugin</artifactId>

<version>${version.plugin.antrun}</version>

<executions>

<execution>

<id>java-17-precompile-test-copy</id>

<phase>process-test-classes</phase>

<goals>

<goal>run</goal>

</goals>

<configuration>

<target>

<copy todir="${project.basedir}/src/test/precompiled-17">

<fileset dir="${project.build.testOutputDirectory}" includes="net/bytebuddy/test/precompiled/v17/\*.class" />

</copy>

</target>

</configuration>

</execution>

</executions>

</plugin>

</plugins>

</build>

</profile>

<profile>

<id>java-17-no-precompile</id>

<activation>

<activeByDefault>true</activeByDefault>

<property>

<name>!java-17-precompile</name>

</property>

</activation>

<build>

<testResources>

<testResource>

<directory>src/test/precompiled-17</directory>

</testResource>

</testResources>

</build>

</profile>

</profiles>

</project>

Java-classmate –

<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<parent>

<groupId>com.fasterxml</groupId>

<artifactId>oss-parent</artifactId>

<version>35</version>

</parent>

<artifactId>classmate</artifactId>

<name>ClassMate</name>

<version>1.5.1</version>

<packaging>bundle</packaging>

<description>Library for introspecting types **with** full generic information

including resolving of field and method types.

</description>

<url>https://github.com/FasterXML/java-classmate</url>

<scm>

<connection>scm:git:git@github.com:FasterXML/java-classmate.git</connection>

<developerConnection>scm:git:git@github.com:FasterXML/java-classmate.git</developerConnection>

<url>https://github.com/FasterXML/java-classmate</url>

<tag>classmate-1.5.1</tag>

</scm>

<developers>

<developer>

<id>tatu</id>

<name>Tatu Saloranta</name>

<email>tatu@fasterxml.com</email>

</developer>

<developer>

<id>blangel</id>

<name>Brian Langel</name>

<email>blangel@ocheyedan.net</email>

</developer>

</developers>

<prerequisites>

<maven>2.2.1</maven>

</prerequisites>

<properties>

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

<version.jdk>1.6</version.jdk>

<osgi.export>com.fasterxml.classmate;version=${project.version},

com.fasterxml.classmate.\*;version=${project.version}

</osgi.export>

<osgi.private>com.fasterxml.classmate.util.\*</osgi.private>

<jdk.**module**.name>com.fasterxml.classmate</jdk.**module**.name>

</properties>

<!-- Licensing -->

<licenses>

<license>

<name>Apache License, Version 2.0</name>

<url>https://www.apache.org/licenses/LICENSE-2.0.txt</url>

<distribution>repo</distribution>

</license>

</licenses>

<organization>

<name>fasterxml.com</name>

<url>https://fasterxml.com</url>

</organization>

<dependencies>

<!-- and for testing, JUnit is needed -->

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>${version.junit}</version>

<scope>test</scope>

</dependency>

</dependencies>

<build>

<plugins>

<!-- 19-Oct-2019, tatu: Copied from

https://github.com/stephenc/git-timestamp-maven-plugin/blob/master/pom.xml#L327-L337

**to** simplify releases. I hope.

-->

<plugin>

<groupId>org.sonatype.plugins</groupId>

<artifactId>nexus-staging-maven-plugin</artifactId>

<version>1.6.6</version>

<extensions>true</extensions>

<configuration>

<serverId>sonatype-nexus-staging</serverId>

<nexusUrl>https://oss.sonatype.org/</nexusUrl>

<!--

<stagingProfileId>b34f19b9cc6224</stagingProfileId>

-->

</configuration>

</plugin>

<!-- As per [#38] add `Automatic-Module-Name` -->

<plugin>

<groupId>org.apache.felix</groupId>

<artifactId>maven-bundle-plugin</artifactId>

<configuration>

<instructions combine.children="merge">

<Automatic-Module-Name>${jdk.**module**.name}</Automatic-Module-Name>

</instructions>

</configuration>

</plugin>

<plugin>

<artifactId>maven-compiler-plugin</artifactId>

<configuration>

<source>${version.jdk}</source>

<target>${version.jdk}</target>

</configuration>

</plugin>

<plugin><!-- plug-in **to** attach source bundle in repo -->

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-source-plugin</artifactId>

<version>${version.plugin.source}</version>

<executions>

<execution>

<id>attach-sources</id>

<goals>

<goal>jar</goal>

</goals>

</execution>

</executions>

</plugin>

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-javadoc-plugin</artifactId>

<version>${version.plugin.javadoc}</version>

<configuration>

<source>${version.jdk}</source>

<target>${version.jdk}</target>

<encoding>UTF-8</encoding>

<maxmemory>512m</maxmemory>

<links>

<link>https://docs.oracle.com/javase/8/docs/api/</link>

</links>

</configuration>

<executions>

<execution>

<id>attach-javadocs</id>

<phase>verify</phase>

<goals>

<goal>jar</goal>

</goals>

</execution>

</executions>

</plugin>

<!-- 22-Mar-2019, tatu: Add rudimentary JDK9+ **module** info. To build **with** JDK 8

will have **to** use `moduleInfoFile` which is not optimal but anything else

**requires** JDK 9+.

-->

<plugin>

<groupId>org.moditect</groupId>

<artifactId>moditect-maven-plugin</artifactId>

<executions>

<execution>

<id>add-**module**-infos</id>

<phase>package</phase>

<goals>

<goal>add-**module**-info</goal>

</goals>

<configuration>

<overwriteExistingFiles>true</overwriteExistingFiles>

<**module**>

<moduleInfoFile>src/moditect/**module**-info.java</moduleInfoFile>

</**module**>

</configuration>

</execution>

</executions>

</plugin>

</plugins>

</build>

<profiles>

<profile>

<id>release-sign-artifacts</id>

<activation>

<property>

<name>performRelease</name>

<value>true</value>

</property>

</activation>

<build>

<plugins>

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-gpg-plugin</artifactId>

<executions>

<execution>

<id>sign-artifacts</id>

<phase>verify</phase>

<goals>

<goal>sign</goal>

</goals>

</execution>

</executions>

</plugin>

</plugins>

</build>

</profile>

</profiles>

<!-- NOTE: repositories from parent POM -->

</project>

Pom.xml –

<?xml version="1.0" encoding="UTF-8"?>

<!--

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

- JOnAS: Java(TM) Open Application Server

- Copyright (C) 2007-2008 Bull S.A.S.

- Contact: jonas-team@objectweb.org

-

- This library is free software; you can redistribute it and/or

- modify it under the terms of the GNU Lesser General Public

- License as published by the Free Software Foundation; either

- version 2.1 of the License, or any later version.

-

- This library is distributed in the hope that it will be useful,

- but WITHOUT ANY WARRANTY; without even the implied warranty of

- MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU

- Lesser General Public License for more details.

-

- You should have received a copy of the GNU Lesser General Public

- License along **with** this library; if not, write **to** the Free Software

- Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307

- USA

-

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

- $Id: pom.xml 12744 2008-01-29 18:15:13Z benoitf $

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

-->

<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4\_0\_0.xsd">

<parent>

<groupId>org.ow2.jonas.osgi</groupId>

<artifactId>jonas-externals</artifactId>

<version>5.0.1</version>

<relativePath>../pom.xml</relativePath>

</parent>

<modelVersion>4.0.0</modelVersion>

<groupId>org.ow2.jonas.osgi</groupId>

<artifactId>corba-api</artifactId>

<packaging>bundle</packaging>

<name>JOnAS :: Libraries :: Externals :: CORBA API</name>

<dependencies>

<!-- org.omg packages comes from jacorb.jar -->

<dependency>

<groupId>org.jacorb</groupId>

<artifactId>jacorb</artifactId>

</dependency>

</dependencies>

</project>