Migration to OPENJDK 12

Download openjdk 12 and unzip it in the java folder in your system. Set the environment variables in your system. Download the latest eclipse, June 2019 release and Tomcat 9.x

Install Server (as per requirement)

- apache-tomcat-9.0.16

Note: Set the environment variables to OpenJDK 12. Add the jar files to Tomcat folder as well.

Make sure that eclipse and Tomcat are running on java 12 version. Change the compling option in the eclipse to jdk12, and execution environment too using jdk12.

Removed from OpenJDK11

EE modules are deprecated in JDK 9 and removed in 11 and following are the replacements:

- java.xml.ws (JAX-WS) incl. jdk.xml.ws: wsgen, wsimport -> com.sun.xml.ws (by Sun/Oracle, *Eclipse Distribution License v. 1.0*)
- java.xml.bind (JAXB) incl. jdk.xml.bind : Schemagen -> com.sun.xml.bind:jaxb-impl + jaxb-api (by Sun/Oracle, *CDDL*)
- java.activation (JAF) -> com.sun.activation:javax.activation + avax.activation:javax.activation-api: ((by Sun/Oracle, CDDL)
- java.xml.ws.annotation (Common Annotations) -> javax.annotation + javax.annotation-api (by Glasfish, *CDDL*)
- java.transaction (JTA) -> 'org.glassfish:javax.transaction' (*CCDL*) + javax.transaction-api
- java.corba (CORBA) incl tools -> *no replacement*
- java.se.ee (all APIs from above)
- Geronimo-jpa_spec jar

Add external jar files **Project<Properties<Build Path** then under **Libraries** click **Add External Jars** add following jar files.

- javax.xml.soap
    - jakarta.activation-api-1.2.1.jar
    - jakarta.xml.bind-api-2.3.2.jar
    - jakarta.xml.soap-api-1.4.1.jar
    - mimepull-1.9.11.jar
    - saaj-api-1.3.5.jar
    - saaj-impl-1.5.1.jar
    - stax-ex-1.8.1.jar
- javax.xml.bind
    - javax.activation-api-1.2.0.jar
    - jaxb-api-2.3.0.jar
    - jaxb-core-2.3.0.jar
    - jaxb-impl-2.3.0-b170127.1453.jar
- javax.jws
    - javax.activation-api-1.2.0.jar

- javax.annotation-api-1.3.2.jar
- javax.jws-3.1.2.2.jar
- javax.xml.soap-api-1.4.0.jar
- jaxb-api-2.3.1.jar
- jaxws-api-2.3.1.jar

Note: If still exception persist, like noclassdeffounderror or javax.xml.soap.SOAPException etc. then Add required jar files to WEB-INF/lib

For Gradle Project

Step 1:

Upgrade Eclipse photon current version to 4.10 version.

Step 2:

Install gradle 5.2.0 Set the environment variables in your system.

To upgrade gradle wrapper of eclipse to version 5.2.0, go to command prompt and run command: gradle wrapper --gradle-version 5.2.0
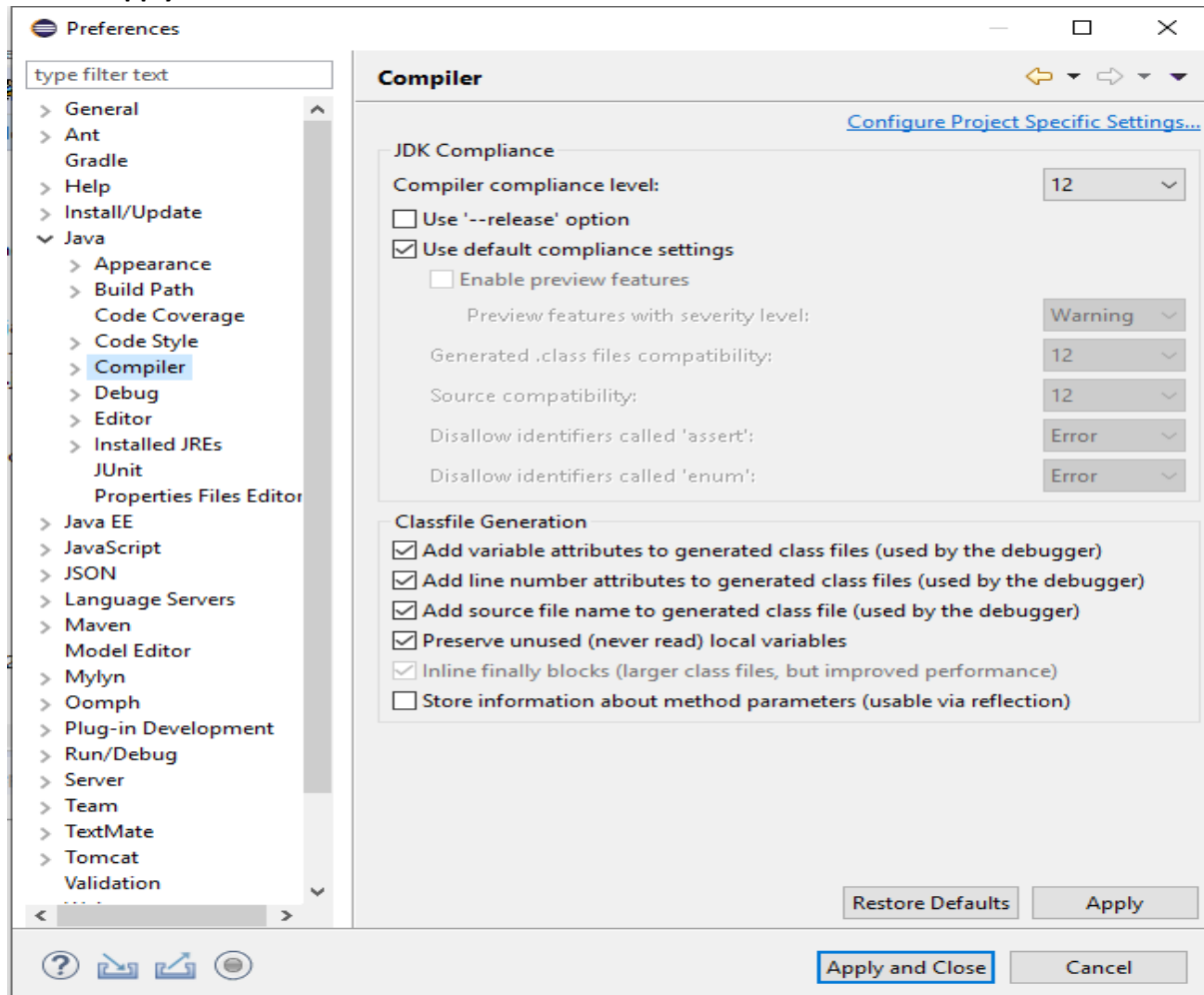
Step 3:

Click on **Window<Preferences<Java<Complier**. Set the values according to below given picture:

Click on **Window<Preferences<Java<Installed JREs**. Add jdk12 in the table.

Click on **Window<Preferences<Java<Installed JREs<Execution Environment**. Select JAVASE-12 under the heading Execution Environment and select the checkbox JDK 12 under heading Compatible JREs.

Click on **Window<Preferences<Servers<Runtime Environment.** Add servers as per requirement.
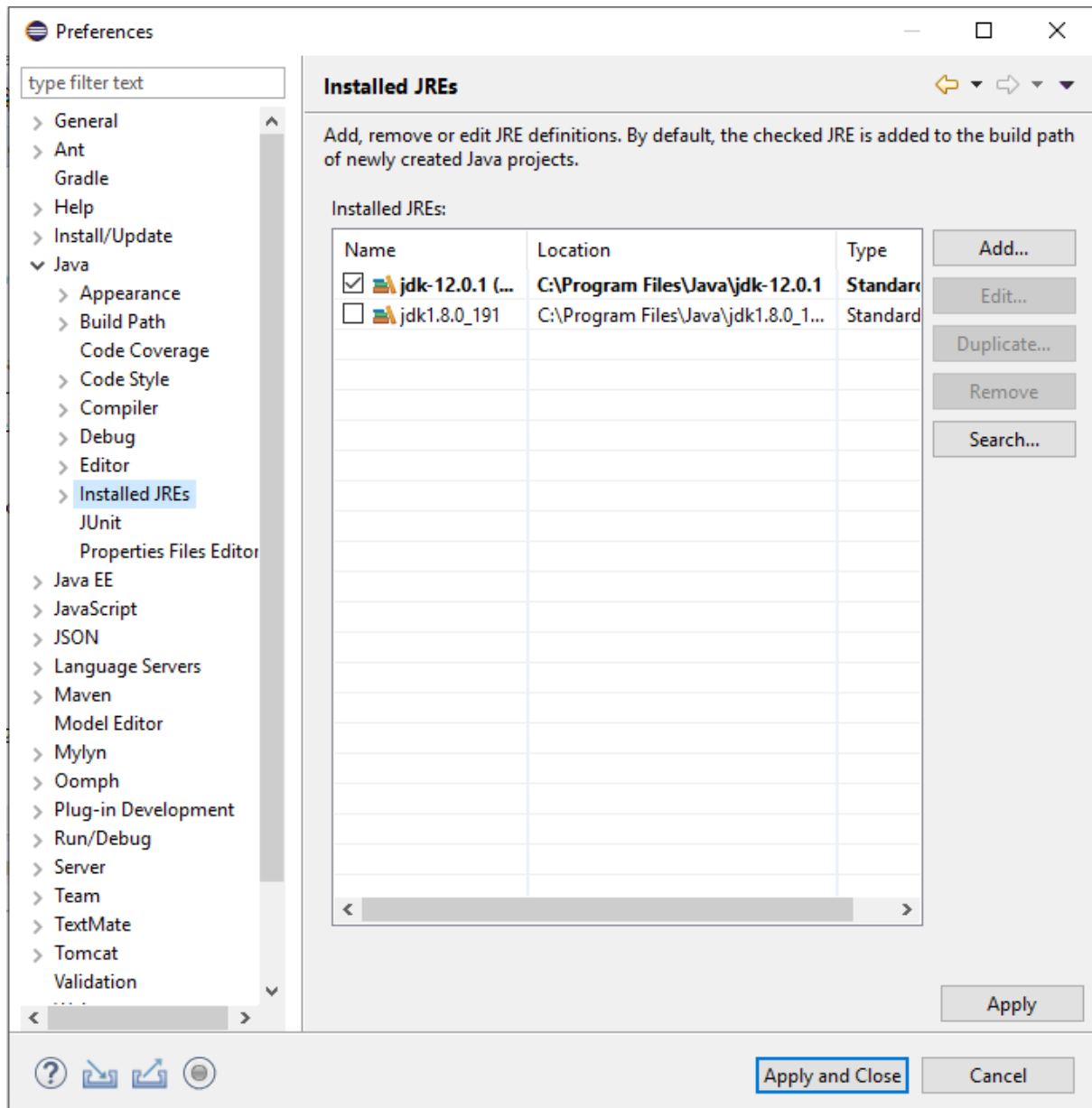
Click on **Apply and Close.**

Step 4:

Click on **Project<Properties<JavaBuildPath** add the required jar files in Classpath and add JRE System Library and link it with jdk12 in Modulepath.
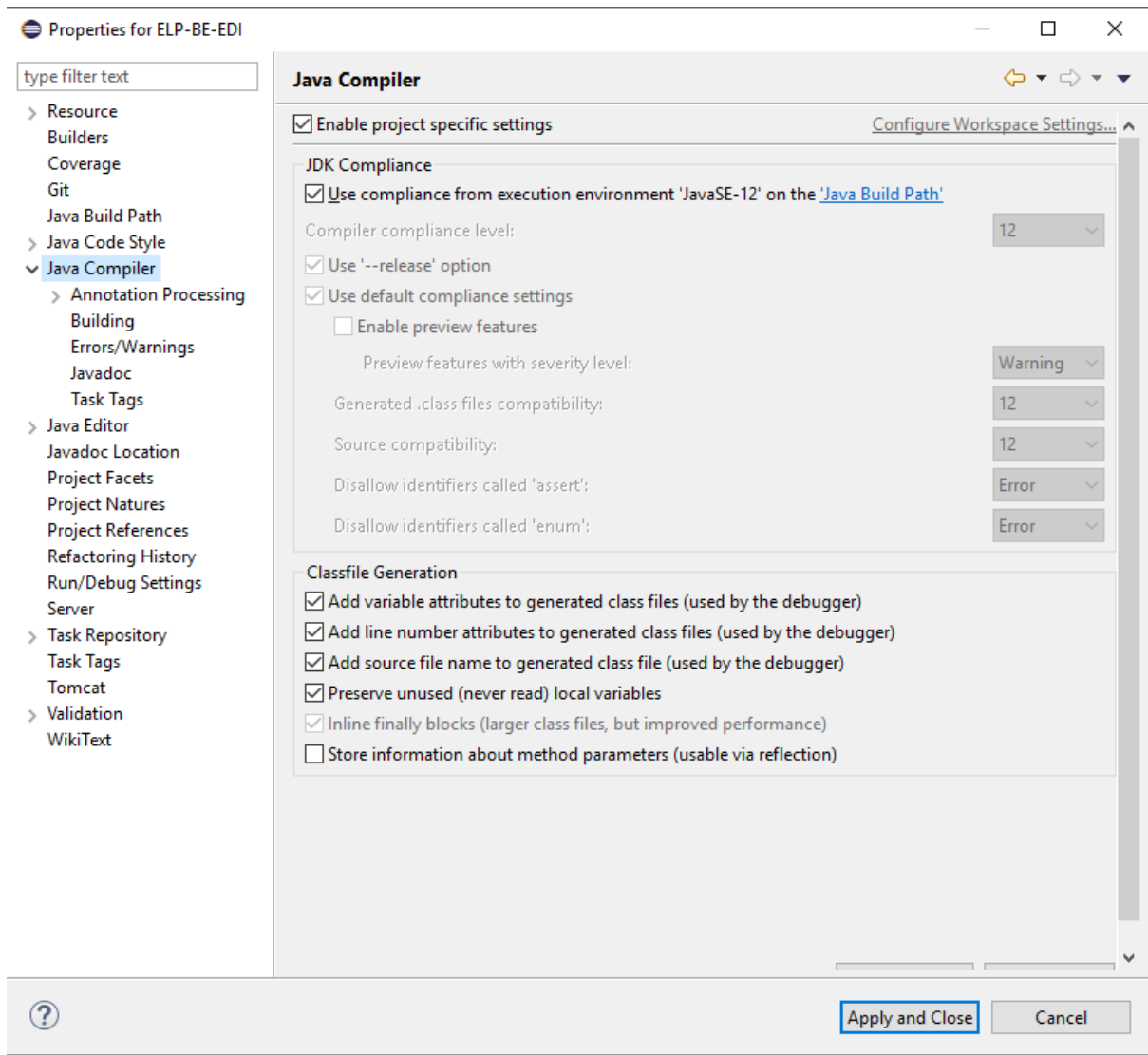
Set the execution environment to jdk12, as shown below.

Click on **Apply and Close.**

Note: Some code modification has to be carried out in build.gradle to make it compatible with newer version of gradle.

Click on **Project<Properties<Project Facets** set the java version to 12.

Errors and Exceptions:

1. If this " com.sun.net.ssl.HttpsURLConnection" is indicating as an error then-

   add one jar file named **com.sun.net.ssl.jar** in the build path of ELP-FE

   The above bug of eclipse has been solved in the June 2019 release.