DependEnCy Conflicts Assessment

How to use DECCA

DECCA can take a Maven based project (it should contain the complete Maven built project directory and file pom.xml) as input for analysis. The expected running environment is 64-bit Window operating system with JDK 1.8, or Linux operating system with Docker platform. **As Maven built projects need to download dependencies from Maven Central Repository, DECCA cannot work offline.**

You can run DECCA on the subjects based on the following steps:

Step 1: Unzip the **plugin-decca-windows.zip** to local directory. Recommended directory structure is:

```
D:\plugin-decca-windows
    ├──Decca
        ├──decca-3.0.jar :
        ├──decca-3.0.pom
        ├──soot-1.0.jar
        ├──apache-maven-3.2.5
        ├──soot-1.0.pom
    ├──testProject
        ├──result
        ├──        petstore-vertx-json-rx-server
```

*Note: To facilitate testing, please keep the unzip directory to be consistent with the above example. It should be noted that the location of data (e.g, D:\plugin-decca-windows) is not hardcoded, it can be replaced with user's actual unzip directory in the install commands.

Step 2: Install DECCA.

(a) Execute the following Windows CMD command to install soot:

```
D:\plugin-decca-windows\Decca\apache-maven-3.2.5\bin\mvn.bat install:install-file -Dfile=D:\plugin-decca\Decca\soot-1.0.jar -DgroupId=neu.lab -DartifactId=soot -Dversion=1.0 -Dpackaging=jar
```

(b) Execute the following Windows CMD command to install DECCA:

```
D:\plugin-decca-windows\Decca\apache-maven-3.2.5\bin\mvn.bat install:install-file -Dfile=D:\plugin-decca-windows\Decca\decca-3.0.jar -DgroupId=neu.lab -DartifactId=decca -Dversion=3.0 -Dpackaging=maven-plugin -DpomFile=D:\plugin-decca\Decca\decca-3.0.pom
```

Step 3: Detect and assess the dependency conflict issues.

Execute the following Windows CMD command to analyze the project:

```
D:\plugin-decca-windows\Decca\apache-maven-3.2.5\bin\mvn.bat -f=D:\plugin-decca-windows\testProject\petstore-
```

vertx-json-rx-server\pom.xml -DresultFilePath=D:\plugin-decca-windows\testProject\result -DsubdivisionLevel=false -DfromHostSearch=true -Dmaven.test.skip=true neu.lab:decca:3.0:printRiskLevel -e

Command explanation:

(1) -f=pom.file : Specify the project under analysis;

(2) -DresultFilePath=output issue report directory : Output the issue report to the specified file;

(3) -Dappend=Boolean : Specify the result output mode (whether in append mode or not);

(4) -DsubdivisionLevel= Boolean : Specify the assessment mode. (True: MODE 1; False MODE 2);

(5) -DDfromHostSearch= Boolean : Specify the assessment mode. (False: MODE 3).

Then you can get the dependency issue report in your specified directory (e.g., **D:\plugin-decca-windows\testProject\result\io_swagger_petstore-vertx-json-rx-server_1_0_0-SNAPSHOT.xml**).