

O-Parent (oparent)

Goals

The goals of oparent are to centrally define shared parent POM definitions and configurations such as nexus (distributionManagement) location, coding styles, license checks, coding style checks, sonar setup, etc.

- Isolate all the common external dependencies, default version, dependency management, plugin management, etc.
- Avoid duplicate/conflicting settings for each project
- Each project sets its parent to inherit the defaults from ONAP Parent
- Project level external dependencies and versions can be overridden if necessary

Ultimately this will ensure consistency across ONAP projects, and free individual projects from redundant work whenever the standard configurations need to be changed.

Release Notes

Frankfurt

<https://docs.onap.org/projects/onap-integration/en/frankfurt/release-notes.html#release-notes>

How to Implement for Your Project

Inherit from oparent

To use oparent, make sure that the oparent POM is set as the ultimate parent POM for all your POM files. If your project already has its own parent POM, then you just need to make this change at that one POM file.

Set the parent POM in your pom.xml as follows using the appropriate version:

oparent master (Honolulu) JDK 11

```
<parent>
<groupId>org.onap.oparent</groupId>
<artifactId>oparent</artifactId>
<version>3.2.0</version>
<relativePath/>
</parent>
```

Formatting Plugin Now Available Starting in guilin- 25 Jun 2020

Oparent now has a formatting plugin available for projects that need to get their source code formatted to meet checkstyle guidelines. Instructions are in the pom.xml on how to use it.

```
# Clone oparent somewhere local
> git clone https://github.com/onap/oparent.git

# 1st - your project should be inheriting from this oparent java dependency
> cd <my-repo>
> vi pom.xml

# ensure pom.xml is pointing to 3.1.0-SNAPSHOT or later

# 2nd - go into your project's source directory you wish to reformat
> cd <my-repo-to-reformat>

# 3rd - type in the following and make sure you set the path to where you have oparent cloned and its
      onap-java-formatter.xml file

> mvn formatter:format spotless:apply process-sources -Dproject.parent.basedir=<oparent-clone-location>

# formatter will re-format your source files

# check that the source compiles
> mvn clean install

# the source changes can now be uploaded via git review process
```

CVE Profile Now Available starting in guilin- 26 Jun 2020

This profile can be used offline to check a repository for CVE issues in the codebase. Useful for contributors to check a new dependency without waiting for code to be merged and a CLM report job to be run.

NOTE: Downloading the CVE database can take awhile and require some bandwidth.

```
# # Be sure your project is inheriting from oparent java dependency
#
> mvn verify -P cve

# should start seeing the following output:
[INFO] Processing Complete for NVD CVE - 2019 (50630 ms)
[INFO] Download Started for NVD CVE - Modified
[INFO] Download Complete for NVD CVE - Modified (594 ms)
[INFO] Processing Started for NVD CVE - Modified
[INFO] Processing Complete for NVD CVE - Modified (592 ms)
[INFO] Begin database maintenance
[INFO] Updated the CPE ecosystem on 661 NVD records

...

#
# Look for these types of messages
#
apache-log4j-extras-1.2.17.jar (pkg:maven/log4j/apache-log4j-extras@1.2.17, cpe:2.3:a:apache:log4j:1.2.17:*:*:*:*:*:* : CVE-2019-17571, CVE-2020-9488
dme2-3.1.200-oss.jar/META-INF/maven/com.google.guava/guava/pom.xml (pkg:maven/com.google.guava/guava@19.0, cpe:2.3:a:google:guava:19.0:*:*:*:*:*:* : CVE-2018-10237
dme2-3.1.200-oss.jar/META-INF/maven/com.hazelcast/hazelcast-client-protocol/pom.xml (pkg:maven/com.hazelcast/hazelcast-client-protocol@1.2.0, cpe:2.3:a:hazelcast:hazelcast:1.2.0:*:*:*:*:*:* : CVE-2016-10750
dme2-3.1.200-oss.jar/META-INF/maven/com.hazelcast/hazelcast/pom.xml (pkg:maven/com.hazelcast/hazelcast@3.7.2, cpe:2.3:a:hazelcast:hazelcast:3.7.2:*:*:*:*:*:* : CVE-2016-10750
dme2-3.1.200-oss.jar/META-INF/maven/commons-beanutils/commons-beanutils/pom.xml (pkg:maven/commons-beanutils/commons-beanutils@1.9.2, cpe:2.3:a:apache:commons_beanutils:1.9.2:*:*:*:*:*:* : CVE-2019-10086
dme2-3.1.200-oss.jar/META-INF/maven/commons-collections/commons-collections/pom.xml (pkg:maven/commons-collections/commons-collections@3.2.1, cpe:2.3:a:apache:commons_collections:3.2.1:*:*:*:*:*:* : CVE-2015-6420, CVE-2017-15708, Remote code execution
dme2-3.1.200-oss.jar/META-INF/maven/org.eclipse.jetty.websocket/websocket-core/pom.xml (pkg:maven/org.eclipse.jetty.websocket/websocket-core@9.0.0.M2, cpe:2.3:a:eclipse:jetty:9.0.0.m2:*:*:*:*:**, cpe:2.3:a:jetty:jetty:9.0.0.m2:*:*:*:*:**) : CVE-2017-7656, CVE-2017-7657, CVE-2017-7658, CVE-2018-12536
dme2-3.1.200-oss.jar/META-INF/maven/org.eclipse.jetty/jetty-server/pom.xml (pkg:maven/org.eclipse.jetty/jetty-server@9.3.12.v20160915, cpe:2.3:a:eclipse:jetty:9.3.12:20160915:*:*:*:*:**, cpe:2.3:a:jetty:jetty:9.3.12:20160915:*:*:*:*:**) : CVE-2017-7656, CVE-2017-7657, CVE-2017-7658, CVE-2017-9735, CVE-2018-12536, CVE-2018-12545, CVE-2019-10241, CVE-2019-10247
dme2-3.1.200-oss.jar/META-INF/maven/org.eclipse.jetty/jetty-xml/pom.xml (pkg:maven/org.eclipse.jetty/jetty-xml@9.3.12.v20160915, cpe:2.3:a:eclipse:jetty:9.3.12:20160915:*:*:*:*:**, cpe:2.3:a:jetty:jetty:9.3.12:20160915:*:*:*:*:**) : CVE-2017-7656, CVE-2017-7657, CVE-2017-7658, CVE-2018-12536, CVE-2018-12545, CVE-2019-10241, CVE-2019-10247
kotlin-stdlib-1.3.20.jar (pkg:maven/org.jetbrains.kotlin/kotlin-stdlib@1.3.20, cpe:2.3:a:jetbrains:kotlin:1.3.20:*:*:*:*:*:* : CVE-2019-10101, CVE-2019-10102, CVE-2019-10103
```

Previous Versions of Oparent

oparent Jakarta branch - JDK 11 (IN DEVELOPMENT)

```
<!-- RELEASED VERSION WILL BE AVAILABLE 1st HALF 2022 -->
<!-- AVAILABLE SNAPSHOT -->

<parent>
  <groupId>org.onap.oparent</groupId>
  <artifactId>oparent</artifactId>
  <version>3.3.0-SNAPSHOT</version>
  <relativePath/>
</parent>
```

oparent Istanbul branch - JDK 11

```
<!-- RELEASED VERSION -->

<parent>
<groupId>org.onap.oparent</groupId>
<artifactId>oparent</artifactId>
<version>3.2.1</version>
<relativePath/>
</parent>

<!-- AVAILABLE SNAPSHOT -->

<parent>
<groupId>org.onap.oparent</groupId>
<artifactId>oparent</artifactId>
<version>3.2.2-SNAPSHOT</version>
<relativePath/>
</parent>
```

oparent Honolulu branch - JDK 11

```
<!-- RELEASED VERSION -->

<parent>
<groupId>org.onap.oparent</groupId>
<artifactId>oparent</artifactId>
<version>3.1.0</version>
<relativePath/>
</parent>
```

oparent Guilin branch - JDK 11

```
<!-- RELEASED VERSION -->

<parent>
<groupId>org.onap.oparent</groupId>
<artifactId>oparent</artifactId>
<version>3.1.0</version>
<relativePath/>
</parent>
```

oparent Frankfurt Branch - JDK 11 - NO LONGER SUPPORTED

```
<!-- LAST AVAILABLE RELEASE -->
<parent>
  <groupId>org.onap.oparent</groupId>
  <artifactId>oparent</artifactId>
  <version>3.0.2</version>
  <relativePath/>
</parent>
```

oparent EI Alto branch JDK 8 - NO LONGER SUPPORTED

```
<!-- LAST AVAILABLE RELEASE -->
<parent>
  <groupId>org.onap.oparent</groupId>
  <artifactId>oparent</artifactId>
  <version>2.1.0</version>
  <relativePath/>
</parent>
```

Remove redundant configuration items

Once the POMs have been modified to inherit from oparent, you can now remove the redundant configuration items such as nexus (distributionManagement) location, coding styles, license checks, coding style checks, sonar setup, etc. from your own project POM files. Any necessary changes to the above can now be managed centrally without your project having to incur additional overhead.

Providing Feedback

If the oparent POM does not provide something that you need or causes conflicts, please provide feedback to the Integration team so that we can update oparent accordingly.

Notes on Releasing OPARENT

1. Update the dependencies/pom.xml and dependencies-clm/pom.xml with any updated versions of upstream projects.
 - a. this should be done with the security team to identify the acceptable or better version
 - b. project team repos should also be validated to make sure they understand any implications on client code before they up-rev
 - c. The merge of this change should create a maven merge job log that can be referenced
 - d. This should be X.Y.Z-SNAPSHOT as the revision.
2. Discuss at the PTL call that an updated version of OPARENT will be released.
3. Use the LF self-release process to create the releases/X.Y.Z.yaml file that points to the maven merge job and the release number
 - a. <https://gerrit.onap.org/r/#/c/oparent/+94862/>
4. The self release jobs should run and publish the maven artifact for X.Y.Z release.
5. After that job runs you need to up-rev the Patch version of oparent
 - a. <https://gerrit.onap.org/r/#/c/oparent/+94863/>
 - b. This changes the revision to X.Y.Z+1-SNAPSHOT (or whatever X,Y,Z combination is appropriate) in all the affected files.
6. oparent repository would now be ready for step 1.