Aggregating Javadocs For Multi-Projects

For example, consider the following directory structure:

```
Project
  |-- pom.xml
  |-- Module1
  |    `-- pom.xml
  |-- Module2
  |    `-- pom.xml
  `-- Module3
       `-- pom.xml
```

Using The <aggregate/> Parameter (deprecated)

Note: <aggregate/> parameter is **deprecated** since 2.5. Please use the aggregate goal instead of.

The <aggregate/> parameter can be used to generate javadocs for multi-module projects. It gives the option to generate one javadoc report for the entire project (all modules) or generate one javadoc report for each module.

When you execute javadoc:javadoc from Project directory with aggregate set to **true**, a javadoc report will be created in the target directory of Project with all the javadocs of Project's modules included. If aggregate is set to **false** (default), a javadoc report for Module1 will be generated in the target directory of Module1, a javadoc report for Module2 will be generated in the target directory of Module2, and a javadoc report for Module3 will be generated in the target directory of Module3.

```
<project>
  ...
  <reporting> (or <build>)
    <plugins>
      <plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-javadoc-plugin</artifactId>
        <version>2.10.3</version>
        <configuration>
          ...
          <aggregate>true</aggregate>
          ...
        </configuration>
      </plugin>
    </plugins>
    ...
  </reporting> (or </build>)
  ...
</project>
```

Using The aggregate Goals

The <aggregate/> parameter doesn't include generate source directories defined using the build-helper:add-source. In this case, you need to use the aggregate goal and test-aggregate goals. You could define these goals in the <build/> element (using the <execution/> tag) or <reporting/> element (using the <reportSet/> tag) as shown below. For more information, refer to the Selective Javadocs Reports page.

```
<project>
  ...
  <build>
    <plugins>
      <plugin>
        <groupId>org.apache.maven.plugins</groupId>
```

```
        <artifactId>maven-javadoc-plugin</artifactId>
        <version>2.10.3</version>
        <configuration>
          <!-- Default configuration for all reports -->
          ...
        </configuration>
        <executions>
          <execution>
            <id>aggregate</id>
            <goals>
              <goal>aggregate</goal>
            </goals>
            <phase>site</phase>
            <configuration>
              <!-- Specific configuration for the aggregate report -->
              ...
            </configuration>
          </execution>
          ...
        </executions>
      </plugin>
      ...
    </plugins>
  </build>
  ...
</project>
```

```
<project>
  ...
  <reporting>
    <plugins>
      <plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-javadoc-plugin</artifactId>
        <version>2.10.3</version>
        <configuration>
          <!-- Default configuration for all reports -->
          ...
        </configuration>
        <reportSets>
          <reportSet>
            <id>non-aggregate</id>
            <configuration>
              <!-- Specific configuration for the non aggregate report -->
              ...
            </configuration>
            <reports>
              <report>javadoc</report>
            </reports>
          </reportSet>
          <reportSet>
            <id>aggregate</id>
            <configuration>
              <!-- Specific configuration for the aggregate report -->
              ...
            </configuration>
            <reports>
              <report>aggregate</report>
            </reports>
          </reportSet>
          ...
        </reportSets>
      </plugin>
      ...
    </plugins>
  </reporting>
```

```
  ...
</project>
```

The Javadoc plugin contains several `aggregate` goals to be use with an aggregator project. Here is the full list of all `aggregate` goals:

- javadoc:aggregate to generate the Javadoc files.
- javadoc:test-aggregate to generate the test Javadoc files.
- javadoc:aggregate-jar to create an archive file of the Javadoc files.
- javadoc:test-aggregate-jar to create an archive file of the test Javadoc files.