Pre-defined Descriptor Files

There are four predefined descriptor formats available for reuse, packaged within the Assembly Plugin. Their descriptorIds are:

- bin
- jar-with-dependencies
- src
- project

bin

Use `bin` as the `descriptorRef` of your assembly-plugin configuration in order to create a binary distribution archive of your project. This built-in descriptor produces an assembly with the classifier `bin` in three archive formats: tar.gz, tar.bz2, and zip.

The assembled archive contains the binary JAR produced by running `mvn package` plus any README, LICENSE, and NOTICE files available in the project root directory.

Below is the `bin` descriptor format:

```
<assembly xmlns="http://maven.apache.org/ASSEMBLY/2.1.1"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/ASSEMBLY/2.1.1 https://maven.apache.org/xsd/assembly-2.1.1.xsd">
  <id>bin</id>
  <formats>
    <format>tar.gz</format>
    <format>tar.bz2</format>
    <format>zip</format>
  </formats>
  <fileSets>
    <fileSet>
      <directory>${project.basedir}</directory>
      <outputDirectory></outputDirectory>
      <includes>
        <include>README*</include>
        <include>LICENSE*</include>
        <include>NOTICE*</include>
      </includes>
    </fileSet>
    <fileSet>
      <directory>${project.build.directory}</directory>
      <outputDirectory></outputDirectory>
      <includes>
        <include>*.jar</include>
      </includes>
    </fileSet>
    <fileSet>
      <directory>${project.build.directory}/site</directory>
      <outputDirectory>docs</outputDirectory>
    </fileSet>
  </fileSets>
</assembly>
```

jar-with-dependencies

Use `jar-with-dependencies` as the `descriptorRef` of your assembly-plugin configuration in order to create a JAR which contains the binary output of your project, along its the unpacked dependencies. This built-in descriptor produces an assembly with the classifier `jar-with-dependencies` using the JAR archive format.

Note that `jar-with-dependencies` provides only basic support for uber-jars. For more control, use the Maven Shade Plugin (https://maven.apache.org/plugins/maven-shade-plugin/) .

Below is the `jar-with-dependencies` descriptor format:

```
<assembly xmlns="http://maven.apache.org/ASSEMBLY/2.1.1"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/ASSEMBLY/2.1.1 https://maven.apache.org/xsd/assembly-2.1.1.xsd">
  <!-- TODO: a jarjar format would be better -->
  <id>jar-with-dependencies</id>
  <formats>
    <format>jar</format>
  </formats>
  <includeBaseDirectory>false</includeBaseDirectory>
  <dependencySets>
    <dependencySet>
      <outputDirectory>/</outputDirectory>
      <useProjectArtifact>true</useProjectArtifact>
      <unpack>true</unpack>
      <scope>runtime</scope>
    </dependencySet>
  </dependencySets>
</assembly>
```

src

Use `src` as the `descriptorRef` in your assembly-plugin configuration to create source archives for your project. The archive will contain the contents of your project's `/src` directory structure, for reference by your users. The `src` descriptorId produces an assembly archive with the classifier `src` in three formats: tar.gz, tar.bz2, and zip.

Below is the `src` descriptor format:

```
<assembly xmlns="http://maven.apache.org/ASSEMBLY/2.1.1"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/ASSEMBLY/2.1.1 https://maven.apache.org/xsd/assembly-2.1.1.xsd">
  <id>src</id>
  <formats>
    <format>tar.gz</format>
    <format>tar.bz2</format>
    <format>zip</format>
  </formats>
  <fileSets>
    <fileSet>
      <directory>${project.basedir}</directory>
      <includes>
        <include>README*</include>
        <include>LICENSE*</include>
        <include>NOTICE*</include>
        <include>pom.xml</include>
      </includes>
      <useDefaultExcludes>true</useDefaultExcludes>
    </fileSet>
    <fileSet>
      <directory>${project.basedir}/src</directory>
      <useDefaultExcludes>true</useDefaultExcludes>
    </fileSet>
  </fileSets>
</assembly>
```

project

Using the `project` `<descriptorRef>` in your Assembly Plugin configuration will produce an assembly containing your entire project, minus any build output that lands in the `/target` directory. The resulting assembly should allow your users to build your project using Maven, Ant, or whatever build system you have configured in your project's normal SCM working directory. It produces assemblies with the classifier `project` in three archive formats: tar.gz, tar.bz2, and zip.

The following is the assembly descriptor for the `project` descriptorRef:

```
<assembly xmlns="http://maven.apache.org/ASSEMBLY/2.1.1"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/ASSEMBLY/2.1.1 https://maven.apache.org/xsd/assembly-2.1.1.xsd">
  <id>project</id>
  <formats>
    <format>tar.gz</format>
    <format>tar.bz2</format>
    <format>zip</format>
  </formats>
  <fileSets>
    <fileSet>
      <directory>${project.basedir}</directory>
      <outputDirectory></outputDirectory>
      <useDefaultExcludes>true</useDefaultExcludes>
      <excludes>
        <exclude>**/*.log</exclude>
        <exclude>**/${project.build.directory}/**</exclude>
      </excludes>
    </fileSet>
  </fileSets>
</assembly>
```