Release Handle JAR to Maven Central

No longer necessary as of 2020

CNRI is now releasing their own versions of the Handle JAR in Maven. They are NOT releasing to Maven Central, but instead have their own repository at https://handle.net/maven (which is included in our parent POM. So, we are now able to pull in the proper version of Handle JAR via a dependency like this:

```
<dependency>
   <groupId>net.handle</groupId>
   <artifactId>handle</artifactId>
   <version>9.3.0</version>
</dependency>
```

This procedure was used to release the latest version of Handle.net JAR to Maven Central (via Sonatype) in April 2019.

Sonatype Release requirements

As of 2019, Sonatype has strict validation requirements for all JAR projects. They include:

- Must include the JAR itself
- Must include a "*-sources.jar"
- Must include a "*-javadoc.jar"
- Must include a valid "pom.xml"
- All JARs must be signed (an ".asc" signature file must

Download Handle Software and Create Release Files

1. Download Handle software distribution package: http://handle.net/download_hnr.html
2. Unzip the distribution package.
    a. The main "handle-[version].jar" can be found in the "/lib" folder
    b. The "handle-[version]-sources.jar" can be created from the embedded "handle-[version]-src.zip". Just extract that, and rename it to "handle-[version]-sources.jar"
    c. The "handle-[version]-javadoc.jar" can be created from the "/doc/apidoc" folder. Just extract that, zip it up and rename it to "handle-[version]-javadoc.jar"
3. Create a valid POM for this release. Here's the one created for v9.2.0 (with help from CNRI staff). **NOTE: We've renamed this release to "9.2.0.v20190814" based on the date it was released to Maven Central.**

```
<?xml version="1.0" encoding="UTF-8"?>
<project xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.
xsd" xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <modelVersion>4.0.0</modelVersion>
  <groupId>org.dspace</groupId>
  <artifactId>handle</artifactId>
  <version>9.2.0.v20190814</version>
  <name>CNRI Handle.net software</name>
  <description>
           CNRI Handle.net Software (Permission granted for redistribution by Giridhar Manepalli at CNRI)
  </description>
  <url>http://handle.net/</url>
  <dependencies>
    <dependency>
      <groupId>com.google.code.gson</groupId>
      <artifactId>gson</artifactId>
      <version>2.8.5</version>
      <scope>compile</scope>
    </dependency>
    <dependency>
      <groupId>org.dspace</groupId>
      <artifactId>cnriutil</artifactId>
      <version>2.0.v20190416</version>
      <scope>compile</scope>
    </dependency>
    <dependency>
      <groupId>commons-codec</groupId>
      <artifactId>commons-codec</artifactId>
      <version>1.11</version>
      <scope>compile</scope>
    </dependency>
    <dependency>
      <groupId>org.python</groupId>
      <artifactId>jython</artifactId>
      <version>2.2.1</version>
```

```
    <scope>compile</scope>
    <optional>true</optional>
  </dependency>
  <dependency>
    <groupId>org.bouncycastle</groupId>
    <artifactId>bcprov-jdk15on</artifactId>
    <version>1.59</version>
    <scope>compile</scope>
  </dependency>
  <dependency>
    <groupId>org.bouncycastle</groupId>
    <artifactId>bcpkix-jdk15on</artifactId>
    <version>1.59</version>
    <scope>compile</scope>
  </dependency>
  <dependency>
    <groupId>javax.servlet</groupId>
    <artifactId>javax.servlet-api</artifactId>
    <version>3.0.1</version>
    <scope>compile</scope>
    <optional>true</optional>
  </dependency>
  <dependency>
    <groupId>org.eclipse.jetty.aggregate</groupId>
    <artifactId>jetty-all</artifactId>
    <version>8.1.22.v20160922</version>
    <scope>compile</scope>
    <exclusions>
      <exclusion>
        <artifactId>javax.servlet</artifactId>
        <groupId>org.eclipse.jetty.orbit</groupId>
      </exclusion>
    </exclusions>
  </dependency>
  <dependency>
    <groupId>com.sleepycat</groupId>
    <artifactId>je</artifactId>
    <version>7.5.11</version>
    <scope>compile</scope>
    <optional>true</optional>
  </dependency>
  <dependency>
    <groupId>com.github.oshi</groupId>
    <artifactId>oshi-core</artifactId>
    <version>3.13.3</version>
    <scope>compile</scope>
    <optional>true</optional>
  </dependency>
  <dependency>
    <groupId>junit</groupId>
    <artifactId>junit</artifactId>
    <version>4.12</version>
    <scope>test</scope>
  </dependency>
  <dependency>
    <groupId>org.hamcrest</groupId>
    <artifactId>hamcrest-core</artifactId>
    <version>1.3</version>
    <scope>test</scope>
  </dependency>
  <dependency>
    <groupId>net.minidev</groupId>
    <artifactId>json-smart</artifactId>
    <version>2.3</version>
    <scope>test</scope>
  </dependency>
  <dependency>
    <groupId>com.googlecode.json-simple</groupId>
    <artifactId>json-simple</artifactId>
    <version>1.1.1</version>
    <scope>test</scope>
```

```
    </dependency>
    <dependency>
      <groupId>com.nimbusds</groupId>
      <artifactId>nimbus-jose-jwt</artifactId>
      <version>2.26.1</version>
      <scope>test</scope>
    </dependency>
  </dependencies>
  <licenses>
    <license>
      <name>Handle.Net Public License Agreement (ver.2)</name>
      <url>http://hdl.handle.net/20.1000/112</url>
      <distribution>manual</distribution>
    </license>
  </licenses>
  <developers>
    <developer>
      <name>CNRI</name>
      <email>hdladmin@cnri.reston.va.us</email>
      <url>http://handle.net/hnr_support.html</url>
    </developer>
  </developers>
  <scm>
    <url>http://handle.net/download_hnr.html</url>
  </scm>
</project>
```

Download CNRIUtil and Create Release Files

Because Handle.jar requires cnriutil.jar, we also need to release a copy of it to Maven Central. This process is the same as with Handle.jar (above). Again, we've appended the date of release on to the version number below.

1. Obtain cnriutil.jar from the Handle downloadable release (or from CNRI staff)
    a. The cnriutil-sources.jar and cnriutil-javadoc.jar were obtained from CNRI Staff
2. Create a valid POM for this release. Here's the one created for v9.1.0 (with help from CNRI staff). **NOTE: We've renamed this release to "2.0. v20190416" based on the date it was released to Maven Central.**

```
<?xml version="1.0" encoding="UTF-8"?>
<project xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.
xsd" xmlns="http://maven.apache.org/POM/4.0.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <modelVersion>4.0.0</modelVersion>
  <groupId>org.dspace</groupId>
  <artifactId>cnriutil</artifactId>
  <version>2.0.v20190416</version>
  <name>CNRI Utility Library (Required by Handle.net software)</name>
  <description>
          CNRI Utility Library (Permission granted for redistribution by Giridhar Manepalli at CNRI)
  </description>
  <url>http://handle.net/</url>
  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>4.12</version>
      <scope>test</scope>
    </dependency>
  </dependencies>
  <licenses>
    <license>
          <name>Handle.Net Public License Agreement (ver.2)</name>
          <url>http://hdl.handle.net/20.1000/112</url>
          <distribution>manual</distribution>
        </license>
  </licenses>
  <developers>
    <developer>
      <name>CNRI</name>
      <email>hdladmin@cnri.reston.va.us</email>
      <url>http://handle.net/hnr_support.html</url>
    </developer>
  </developers>
  <scm>
    <url>http://handle.net/download_hnr.html</url>
  </scm>
</project>
```

Sign each CNRIUtil artifact & Upload to Sonatype

There are two ways of doing this, as noted in the Sonatype docs here: https://central.sonatype.org/pages/manual-staging-bundle-creation-and-deployment.html

You can either use maven-gpg-plugin to sign & deploy each file.

Or, you can manually sign each, and upload them to Sonatype (via the web UI).

I chose the latter. Here's what I did:

1. Signed each artifact
    a. gpg2 -ab cnriutil-2.0.v20190416.pom
    b. gpg2 -ab cnriutil-2.0.v20190416.jar
    c. gpg2 -ab cnriutil-2.0.v20190416-sources.jar
    d. gpg2 -ab cnriutil-2.0.v20190416-javadoc.jar
2. Now, a total of 8 files exist. Each of the above, and a corresponding "*.asc" signature.
3. Bundle them into a single JAR (bundle.jar):

```
jar -cvf cnriutil-bundle.jar cnriutil-2.0.v20190416.pom cnriutil-2.0.v20190416.pom.asc cnriutil-2.0.
v20190416.jar cnriutil-2.0.v20190416.jar.asc cnriutil-2.0.v20190416-sources.jar cnriutil-2.0.v20190416-
sources.jar.asc cnriutil-2.0.v20190416-javadoc.jar cnriutil-2.0.v20190416-javadoc.jar.asc
```

4. Login to https://oss.sonatype.org/ and upload the bundle.jar on the "Staging Upload" page
    a. Select "Artifact Bundle" upload mode
    b. Upload the "cnriutil-bundle.jar" you created
5. Check the "Staging Repositories" to see if all validation succeeded. If not, correct any errors and try again.
6. Otherwise, check if everything looks correct. Click Release! Now, wait for it to appear in Maven Central

Sign each Handle artifact & Upload to Sonatype

There are two ways of doing this, as noted in the Sonatype docs here: https://central.sonatype.org/pages/manual-staging-bundle-creation-and-deployment.html

You can either use maven-gpg-plugin to sign & deploy each file.

Or, you can manually sign each, and upload them to Sonatype (via the web UI).

I chose the latter. Here's what I did:

1. Signed each artifact
    a. gpg2 -ab handle-9.2.0.v20190814.pom
    b. gpg2 -ab handle-9.2.0.v20190814.jar
    c. gpg2 -ab handle-9.2.0.v20190814-sources.jar
    d. gpg2 -ab handle-9.2.0.v20190814-javadoc.jar
2. Now, a total of 8 files exist. Each of the above, and a corresponding "*.asc" signature.
3. Bundle them into a single JAR (bundle.jar):

```
jar -cvf handle-bundle.jar handle-9.2.0.v20190814.pom handle-9.2.0.v20190814.pom.asc handle-9.2.0.v20190814.jar handle-9.2.0.v20190814.jar.asc handle-9.2.0.v20190814-sources.jar handle-9.2.0.v20190814-sources.jar.asc handle-9.2.0.v20190814-javadoc.jar handle-9.2.0.v20190814-javadoc.jar.asc
```

4. Login to https://oss.sonatype.org/ and upload the bundle.jar on the "Staging Upload" page
    a. Select "Artifact Bundle" upload mode
    b. Upload the "handle-bundle.jar" you created
5. Check the "Staging Repositories" to see if all validation succeeded. If not, correct any errors and try again.
6. Otherwise, check if everything looks correct. Click Release! Now, wait for it to appear in Maven Central