The Java™ Tutorials

Trail: Deployment
Lesson: Packaging Programs in JAR Files
Section: Signing and Verifying JAR Files

The Java Tutorials have been written for JDK 8. Examples and practices described in this page don't take advantage of improvements introduced in later releases.

Verifying Signed JAR Files

Typically, verification of signed JAR files will be the responsibility of your Java™ Runtime Environment. Your browser will verify signed applets that it downloads. Signed applications invoked with the `-jar` option of the interpreter will be verified by the runtime environment.

However, you can verify signed JAR files yourself by using the `jarsigner` tool. You might want to do this, for example, to test a signed JAR file that you've prepared.

The basic command to use for verifying a signed JAR file is:

```
jarsigner -verify jar-file
```

This command will verify the JAR file's signature and ensure that the files in the archive haven't changed since it was signed. You'll see the following message if the verification is successful:

```
jar verified.
```

If you try to verify an unsigned JAR file, the following message results:

```
jar is unsigned. (signatures missing or not parsable)
```

If the verification fails, an appropriate message is displayed. For example, if the contents of a JAR file have changed since the JAR file was signed, a message similar to the following will result if you try to verify the file:

```
jarsigner: java.lang.SecurityException: invalid SHA1
signature file digest for test/classes/Manifest.class
```

Note: The JDK treates a signed JAR file as unsigned if the signed JAR file uses any algorithm that's specified in the `jdk.jar.disabledAlgorithms` Security Property in the *`java.home`*`/lib/security/java.security` file (where *`java.home`* is the directory where you installed your JRE).

Previous page: Signing JAR Files
Next page: Using JAR-related APIs