

The Java™ Tutorials

Trail: Deployment

Lesson: Packaging Programs in JAR Files

Section: Working with Manifest Files: The Basics

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.

Sealing Packages within a JAR File

Packages within JAR files can be optionally sealed, which means that all classes defined in that package must be archived in the same JAR file. You might want to seal a package, for example, to ensure version consistency among the classes in your software.

You seal a package in a JAR file by adding the `Sealed` header in the manifest, which has the general form:

```
Name: myCompany/myPackage/  
Sealed: true
```

The value `myCompany/myPackage/` is the name of the package to seal.

Note that the package name must end with a `/`.

An Example

We want to seal two packages `firstPackage` and `secondPackage` in the JAR file `MyJar.jar`.

We first create a text file named `Manifest.txt` with the following contents:

```
Name: myCompany/firstPackage/  
Sealed: true  
  
Name: myCompany/secondPackage/  
Sealed: true
```

Warning: The text file must end with a new line or carriage return. The last line will not be parsed properly if it does not end with a new line or carriage return.

We then create a JAR file named `MyJar.jar` by entering the following command:

```
jar cfm MyJar.jar Manifest.txt MyPackage/*.class
```

This creates the JAR file with a manifest with the following contents:

```
Manifest-Version: 1.0  
Created-By: 1.7.0_06 (Oracle Corporation)  
Name: myCompany/firstPackage/  
Sealed: true  
Name: myCompany/secondPackage/  
Sealed: true
```

Sealing JAR Files

If you want to guarantee that all classes in a package come from the same code source, use JAR sealing. A sealed JAR specifies that all packages defined by that JAR are sealed unless overridden on a per-package basis.

To seal a JAR file, use the `Sealed` manifest header with the value `true`. For example,

```
Sealed: true
```

specifies that all packages in this archive are sealed unless explicitly overridden for particular packages with the `Sealed` attribute in a manifest entry.

