

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.

Setting an Application's Entry Point

If you have an application bundled in a JAR file, you need some way to indicate which class within the JAR file is your application's entry point. You provide this information with the `Main-Class` header in the manifest, which has the general form:

```
Main-Class: classname
```

The value *classname* is the name of the class that is your application's entry point.

Recall that the entry point is a class having a method with signature `public static void main(String[] args)`.

After you have set the `Main-Class` header in the manifest, you then run the JAR file using the following form of the `java` command:

```
java -jar JAR-name
```

The `main` method of the class specified in the `Main-Class` header is executed.

An Example

We want to execute the `main` method in the class `MyClass` in the package `MyPackage` when we run the JAR file.

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

```
Main-Class: MyPackage.MyClass
```

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)
Main-Class: MyPackage.MyClass
```

When you run the JAR file with the following command, the `main` method of `MyClass` executes:

```
java -jar MyJar.jar
```

Setting an Entry Point with the JAR Tool

The `'e'` flag (for 'entrypoint') creates or overrides the manifest's `Main-Class` attribute. It can be used while creating or updating a JAR file. Use it to specify the application entry point without editing or creating the manifest file.

For example, this command creates `app.jar` where the `Main-Class` attribute value in the manifest is set to `MyApp`:

```
jar cfe app.jar MyApp MyApp.class
```

You can directly invoke this application by running the following command:

```
java -jar app.jar
```

If the entrypoint class name is in a package it may use a `'.'` (dot) character as the delimiter. For example, if `Main.class` is in a package called `foo` the entry point can be specified in the following ways:

```
jar cfe Main.jar foo.Main foo/Main.class
```

Previous page: [Modifying a Manifest File](#)

Next page: [Adding Classes to the JAR File's Classpath](#)