

## The Java™ Tutorials

**Trail:** Deployment

**Lesson:** Packaging Programs in JAR Files

**Section:** Using JAR 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.*

## Running JAR-Packaged Software

Now that you have learned how to create JAR files, how do you actually run the code you packaged? Consider these scenarios:

- Your JAR file contains an applet that is to be run inside a browser.
- Your JAR file contains an application that is to be started from the command line.
- Your JAR file contains code that you want to use as an extension.

This section will cover the first two situations. A separate trail in the tutorial on the [extension mechanism](#) covers the use of JAR files as extensions.

### Applets Packaged in JAR Files

To start any applet from an HTML file for running inside a browser, you use the `applet` tag. For more information, see the [Java Applets](#) lesson. If the applet is bundled as a JAR file, the only thing you need to do differently is to use the `archive` parameter to specify the relative path to the JAR file.

As an example, use the TicTacToe demo applet. The `applet` tag in the HTML file that displays the applet can be marked up like this:

```
<applet code=TicTacToe.class
        width="120" height="120">
</applet>
```

If the TicTacToe demo was packaged in a JAR file named `TicTacToe.jar`, you can modify the `applet` tag with the addition of an `archive` parameter:

```
<applet code=TicTacToe.class
        archive="TicTacToe.jar"
        width="120" height="120">
</applet>
```

The `archive` parameter specifies the relative path to the JAR file that contains `TicTacToe.class`. For this example it is assumed that the JAR file and the HTML file are in the same directory. If they are not, you must include the JAR file's relative path in the `archive` parameter's value. For example, if the JAR file was one directory below the HTML file in a directory called `applets`, the `applet` tag would look like this:

```
<applet code=TicTacToe.class
        archive="applets/TicTacToe.jar"
        width="120" height="120">
</applet>
```

### JAR Files as Applications

You can run JAR packaged applications with the Java launcher (`java` command). The basic command is:

```
java -jar jar-file
```

The `-jar` flag tells the launcher that the application is packaged in the JAR file format. You can only specify one JAR file, which must contain all of the application-specific code.

Before you execute this command, make sure that the runtime environment has information about which class within the JAR file is the application's entry point.

To indicate which class is the application's entry point, you must add a `Main-Class` header to the JAR file's manifest. The header takes the form:

```
Main-Class: classname
```

The header's value, `classname`, is the name of the class that is the application's entry point.

For more information, see the [Setting an Application's Entry Point](#) section.

When the `Main-Class` is set in the manifest file, you can run the application from the command line:

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

To run the application from the JAR file that is in another directory, you must specify the path of that directory: `java -jar path/app.jar`

---

[About Oracle](#) | [Contact Us](#) | [Legal Notices](#) | [Terms of Use](#) | [Your Privacy Rights](#)

Copyright © 1995, 2017 Oracle and/or its affiliates. All rights reserved.

**Previous page:** Updating a JAR File

**Next page:** Working with Manifest Files: The Basics