

**Supplement: Packaging and Deploying Java Projects  
For Introduction to Java Programming  
By Y. Daniel Liang**

1 Introduction

Your project may consist of many classes and supporting files, such as image files and audio files. To make your programs run on the end-user side, you need to provide end-users with all these files. For convenience, Java supports an archive file that can be used to group all the project files in a compressed file.

Archiving makes it possible for Java applications, applets, and their requisite components (.class files, images, and sounds) to be transported in a single file. This single file can be deployed on an end-user's machine as an application. It also can be downloaded to a browser in a single HTTP transaction, rather than opening a new connection for each piece. This greatly simplifies application deployment and improves the speed with which an applet can be loaded onto a Web page and begin functioning.

You can use the JDK **jar** command to create an archive file from the class directory. The following command creates an archive file named TicTacToe.jar for the TicTacToe program.

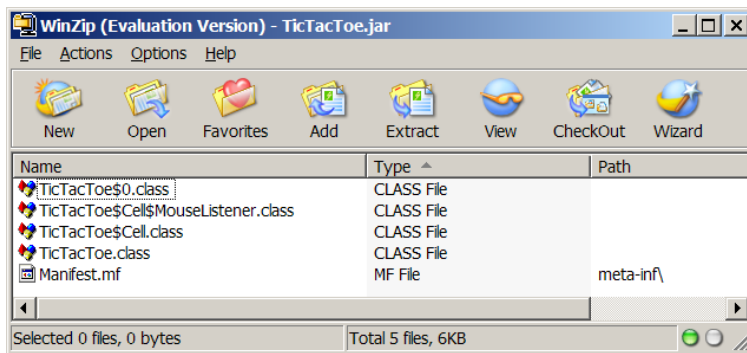
```
c:\book>jar -cf TicTacToe.jar TicTacToe.class  
TicTacToe$Cell.class TicTacToe$0.class  
TicTacToe$Cell$MouseListener.class
```

The **-c** option is for creating a new archive file, and the **-f** option specifies the archive file's name. Note that the TicTacToe program consists of the classes TicTacToe.class and inner classes TicTacToe\$Cell.class, TicTacToe\$0.class, and TicTacToe\$Cell\$MouseListener.class.

**NOTE**

**<subtitle: viewing .jar contents>**

The Java archive file format (JAR) is based on the popular ZIP file format. You can view the contents of a .jar file using WinZip, a popular compression utility for Windows, as shown in Figure 1.



**Figure 1**

*You can view the files contained in the archive file using the WinZip utility.*

**\*\*\*end of NOTE**

2 The Manifest File

As shown in Figure 17.17, a manifest file was created with the path name `meta-inf\`. The manifest is a special file that contains information about the files packaged in a JAR file. For example, the manifest file in Figure 17.17 contains the following information:

```
Manifest-Version: 1.0
Created-By: 1.6.0 (Sun Microsystems Inc.)
```

You can modify the information contained in the manifest file to enable the JAR file to be used for a variety of purposes. For instance, you can add information to specify a main class to run an application using the .jar file.

2 Running Archived Projects

You can package all the class files and dependent resource files in an archive file for distribution to the end-user. If the project is a Java application, the user should have a Java-running environment already installed. If it is not installed, the user can download the Java Runtime Environment (JRE) from [java.sun.com](http://java.sun.com) and install it.

**NOTE**

**<subtitle: JRE>**

The Java Runtime Environment is the minimum standard Java platform for running Java programs. It contains the Java interpreter, Java core classes, and supporting files. The JRE does not contain any development tools (such as Applet Viewer or javac) or classes that pertain only to a development environment. The JRE is a subset of JDK.

To run TicTacToe as an application, take the following steps:

1. Update the manifest file to insert an entry for the main class. You need to create a text file containing the following two lines:

```
Main-Class: TicTacToe  
Sealed: true
```

The first line specifies the main class. The second line is necessary to ensure that the first line can be inserted into an existing manifest file in a jar. Assume that these two lines are contained in the file temp.mf.

2. Execute the jar command to insert the main class line into the manifest file in TicTacToe.jar, as follows:

```
jar -uvmf temp.mf TicTacToe.jar
```

The **-u** option is for updating an existing jar file, the **-v** option is for displaying command output, the **-m** option is for appending the contents in temp.mf to the manifest file in the archive, and the **-f** option specifies the archive file's name.

3. Run the .jar file using the java command from the directory that contains TicTacToe.jar, as follows:

```
java -jar TicTacToe.jar
```

#### **NOTE**

You can write an installation procedure that creates the necessary directories and subdirectories on the end-user's computer. The installation can also create an icon that the end-user can double-click to start the program. For information on creating Windows desktop icons, please see Supplement I.E, "Creating Shortcuts for Java Applications on Windows."

To run TicTacToe as an applet, modify the <applet> tag in the HTML file to include an archive attribute. The archive attribute specifies the archive file in which the applet is contained. For example, the HTML file for running TicTacToe can be modified as shown below:

```
<applet  
code = "TicTacToe.class"  
  
archive = "TicTacToe.jar"  
  
width = 400  
  
height = 300  
  
hspace = 0  
  
vspace = 0  
  
align = mddle  
  
>  
</applet>
```