

---

# How to Run Java Programs Using Metrowerks CodeWarrior

---

**C. Thomas Wu**

**Supplement material to accompany *An Introduction to Object-Oriented Programming with Java***

*WCB/McGraw-Hill*  
© *The McGraw-Hill Companies, Inc.*, 1999

---

In this document, we will explain how to run Java programs using Metrowerk's CodeWarrior for Java. Please remember that the procedure described in this document is just one way to create and run Java programs with CodeWarrior. Among the many possible variations, we present here the most straightforward.

---

## **1.0 Installing the Software and javabook Package**

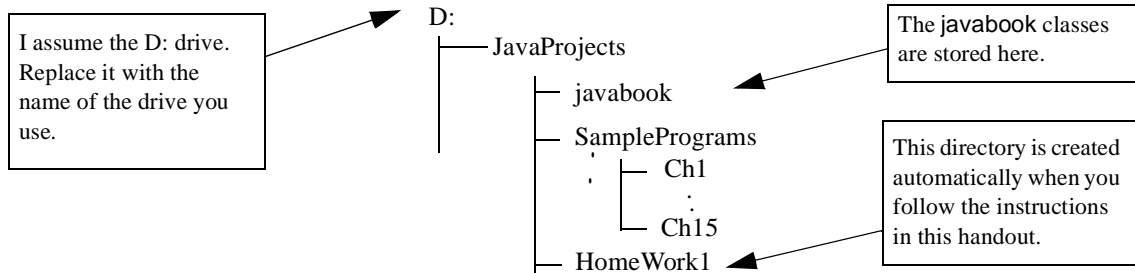
---

Install CodeWarrior for Java on your machine following the instructions of the installation program. Also, download the javabook package and sample programs from the DrCaffeine web site ([www.drcaffeine.com](http://www.drcaffeine.com)). Notice that the Java package is organized as a folder (directory), so you will see the javabook package as a folder that contains the javabook classes (bytecode files) and the corresponding source files.

## 2.0 Directory Structure

---

You can place the javabook package and sample programs anywhere you want. However, in order to present concrete examples of how to run programs that use javabook classes, we will assume the following directory structure in this handout:



## 3.0 Starting CodeWarrior for Java

---

Start CodeWarrior by selecting its program icon in the Start|Program|CodeWarrior Pro 4 Java group (or in whichever group the CodeWarrior icon is located).

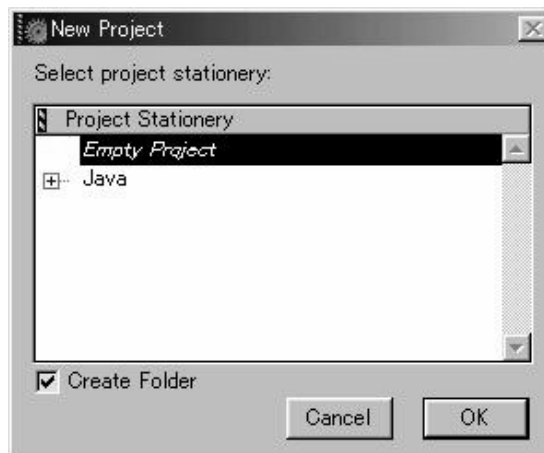
## 4.0 Creating a Project

---

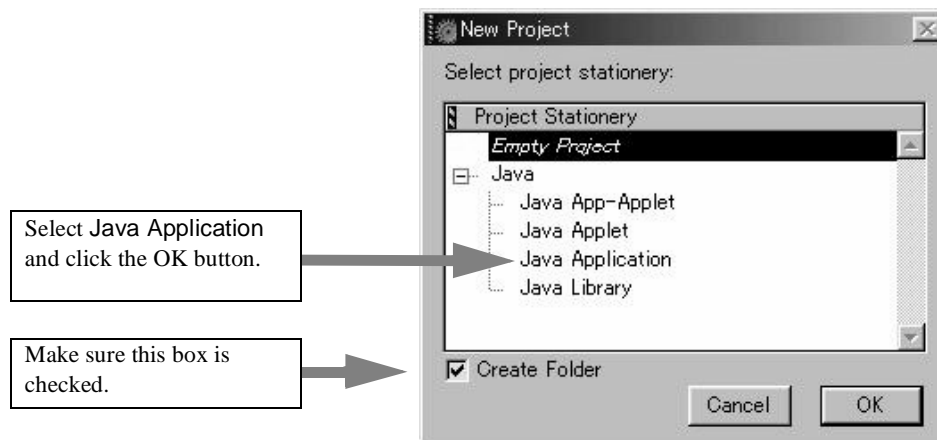
CodeWarrior and other Java IDEs (Integrated Development Environment) use project-based concepts. A project consists of various types of files that are necessary for compiling and running programs. For the sample programs from the earlier chapters, the projects will include only one or two source files (note: we are counting here only the files we create ourselves). In the later chapters, projects will include over 10 files.

To create a project, select the menu choice File|New Project. The New Project dialog will appear on the screen.

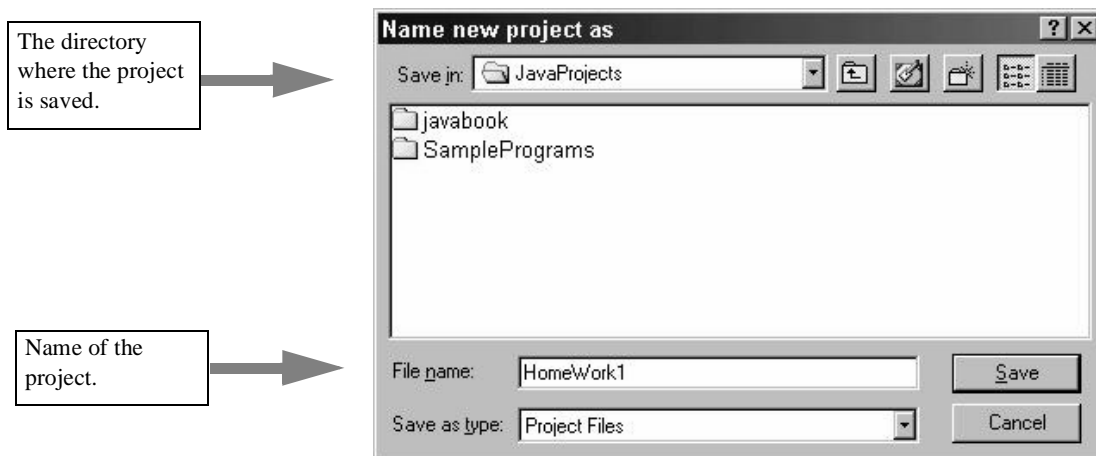
Click the plus sign (+) to list the available types of Java projects.



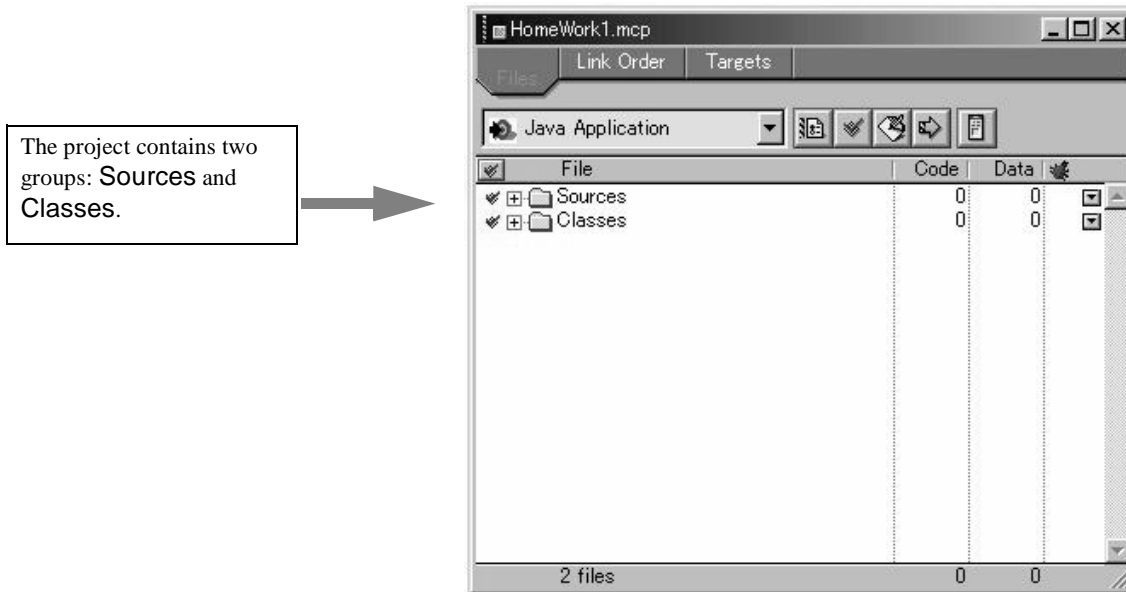
Click the plus sign to expand the Java category:



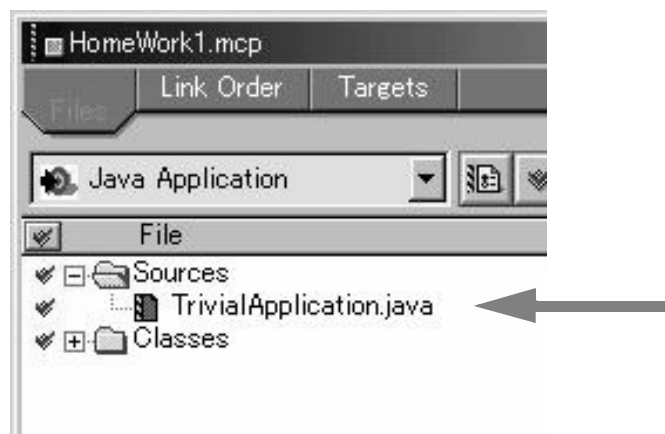
Select the project type Java Application, and click the OK button. The Name new project as dialog will show up. Move to the JavaProjects directory, and enter HomeWork1 as the project name. The dialog now looks like this:



When everything is entered correctly, click the **Save** button. The Project Window will appear on the screen:



Click the plus sign next to the **Sources** group. This item with a folder icon is called a *group* by CodeWarrior. We will use this terminology in this document. You will see the source file named `TrivialApplication.java` included in the group. *NOTE: The group in the Project Window does not correspond to any folder of the operating system.*



This source file includes a very small program that prints the message **Hello World** in a DOS window. We will replace it with our program **FunTime** from Chapter 1. We have two ways to make this replacement. The first is to open this file, replace the contents with the **FunTime** program, and save the file with the new name `FunTime.java`. The second is to remove this file from the group and create a new file that includes the **Fun-**

Time program. We will use the second way since you need to learn how to create a new file when you create a project with multiple classes anyway. To remove `TrivialApplication.java`, click the item with the right mouse button. Select the choice **Remove Selected Items** from the popup menu. *Note: Removing an item from a group in the Project Window does not remove the file from the hard disk.*

---

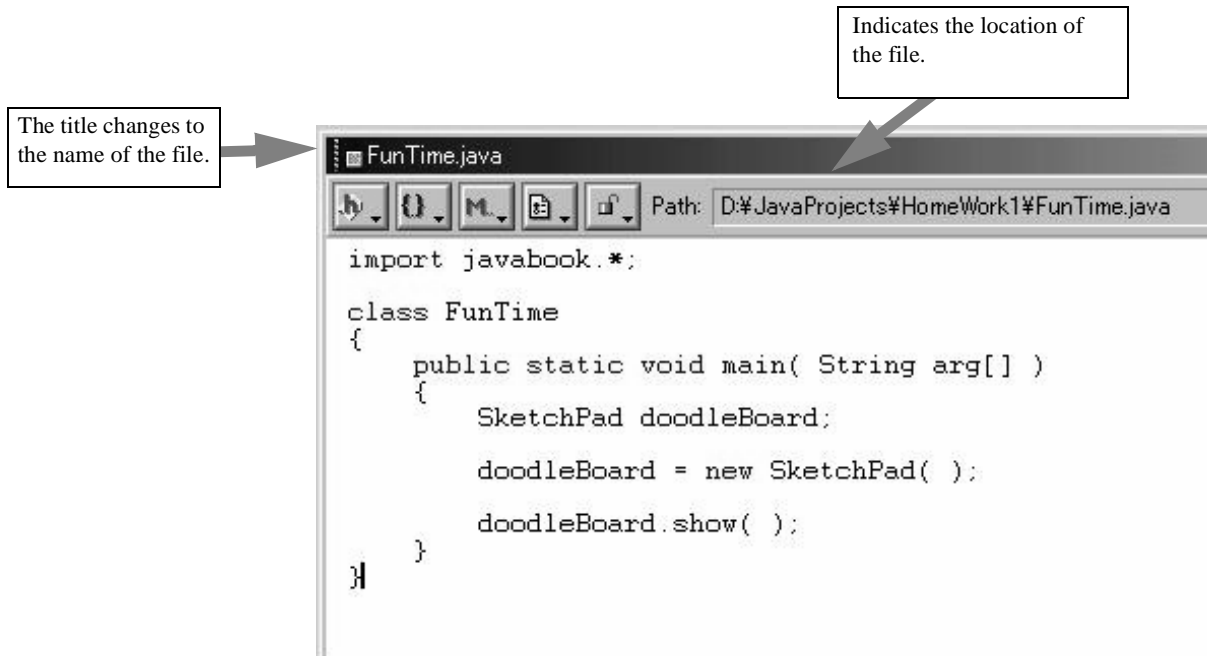
### 5.0 Creating a Source File and Adding it to the Project

---

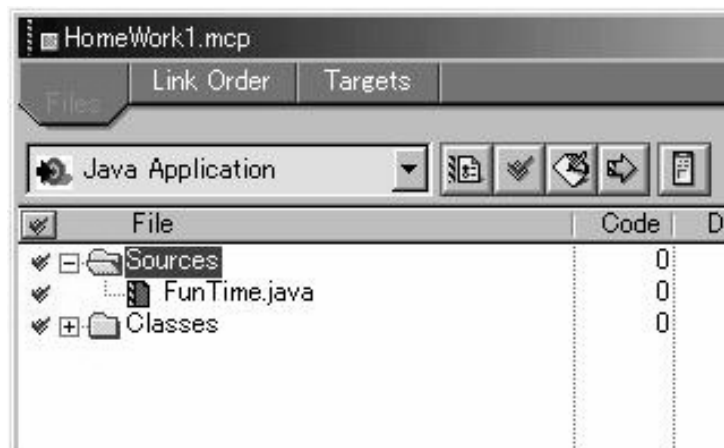
To create a new source file, select the menu choice **File|New**. An Editor window will appear:



Enter the FunTime program, and save the program as FunTime.java. Make sure to place the file in the HomeWork1 directory. The Path field will tell you where the file is saved. After the program is saved, the editor window will look like this:



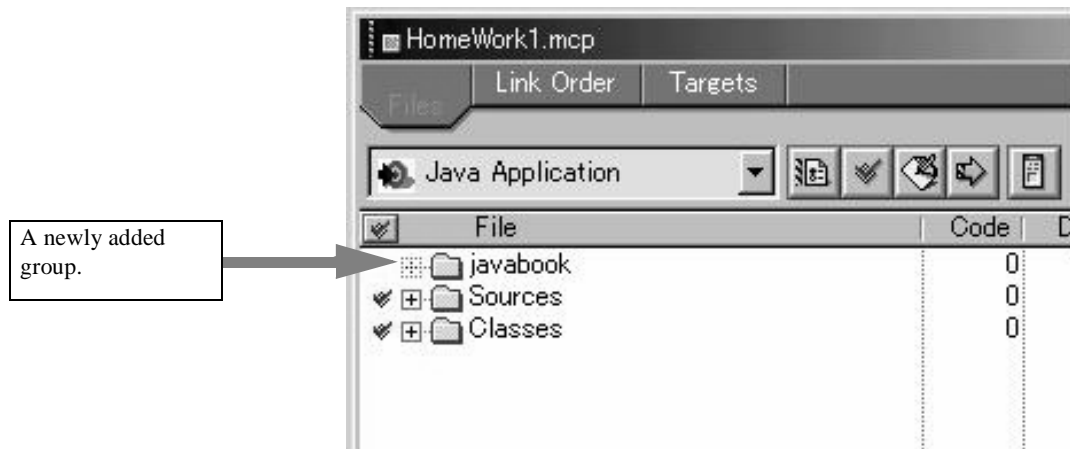
Now we need to add this source file to the Sources group in the Project Window. First, click the Sources group. It becomes highlighted. Then, select the menu choice Project|Add Files. A Select files to add dialog will appear. Using this dialog, select the source file FunTime.java in the HomeWork1 directory and click the Add button. If the file is added correctly, you can expand the Sources group (by clicking the plus sign) and see the Project Window that looks like this:



## 6.0 Add javabook Classes to the Project

The FunTime program refers to a javabook class called SketchPad. For the CodeWarrior to be able to execute the program, we need to add javabook classes used by the program to the project. Although it is only necessary to add the javabook classes used by the program, it is not always simple to detect the classes that need to be added to the project. This is because a program may use a javabook class directly or indirectly. For example, if you use `MessageBox` in your program, in addition to adding `MessageBox` to the project, you must also add the `JavaBookDialog` class because `MessageBox` is a subclass of `JavaBookDialog`. Instead of trying to figure out exactly which classes need to be added to the project, we will simply add all javabook classes to the project.

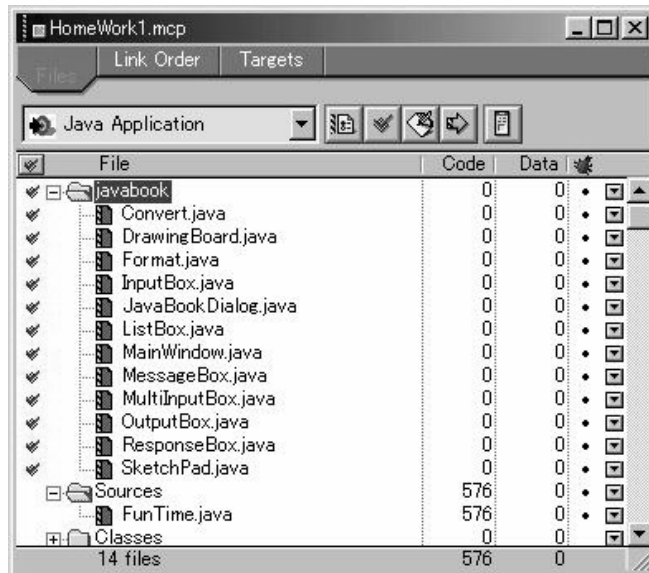
First we create a new project group. Select the menu choice `Project|Create New Group`. Name the new group `javabook`. **NOTE: Make sure no folder is highlighted when you select `Create New Group`.** The Project Window should now look like this:



Next we add the javabook classes to the newly created javabook group. Select the javabook group by clicking on it. It gets highlighted. Select the menu choice `Project|Add Files`. Using the `Select files to add` dialog, select the classes in the `D:\JavaProjects\javabook` directory. *Note: You can select all files at once using the standard Windows technique.*

As a default, the dialog lists the source files only, and you can add just the javabook source files. This will work. Optionally, you can add only the bytecode files. The advantage of adding source files is that you can open the files from the project and read their source code. Adding the source files, however, will make the CodeWarrior compile the javabook source files when you try to run the FunTime program for the first time. If you prefer this compilation of the javabook classes not happen, then you can instead add only the bytecode javabook classes. To select the bytecode files, set the File Type to All Files so the bytecode files get listed in the dialog. Since the compilation of javabook classes happens only once for the project and the compilation won't take that

long, we'll add only the source files. After the source files are added to the javabook group, you can click the plus sign and see the listing of the javabook source files:



When the files are added, a Message dialog will appear stating that the access path has been added to your project. This is what you expect to happen. Just close the dialog.

---

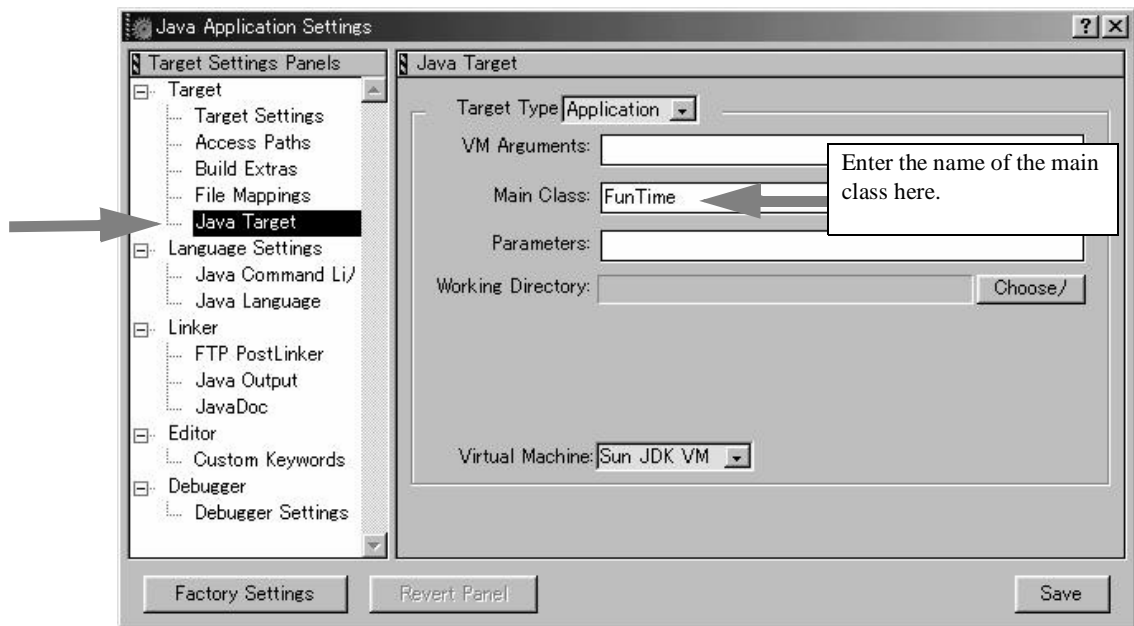
## 7.0 Specifying the Application Settings

---

We have one more step left before running the program. We must specify the main class for this project. Select the menu choice **Edit|Java Application Settings**. The Java



Application Settings dialog appears. Click on the *Java Target* item, and enter the name of the main class in the Main Class field. The dialog now looks like this:



Click the Save button to save the new setting and close the dialog.

---

## 8.0 Running a Java Application Program

---

We are finally ready to run the program. Choose the menu choice **Project|Run**, or press the F5 key. If the program contains no error, the **SketchPad** window will appear:



---

## **9.0 Quick Summary**

---

Here's a quick summary of the steps you take to enter, compile, and run a Java application using CodeWarrior for Java:

1. Create a project of type **Java Application**.
2. Remove the automatically added source file **TrivialApplication.java** from the project.
3. Create a new java source file. Name the source file with the class name. Example: The class **Funtime** is saved in the file **Funtime.java**.
4. Add the newly created source file to the project.
5. Create the **javabook** group, and add the **javabook** classes to the group.
6. Set the main class of the program inside the **Java Application Settings** dialog.
7. Run the application.

---

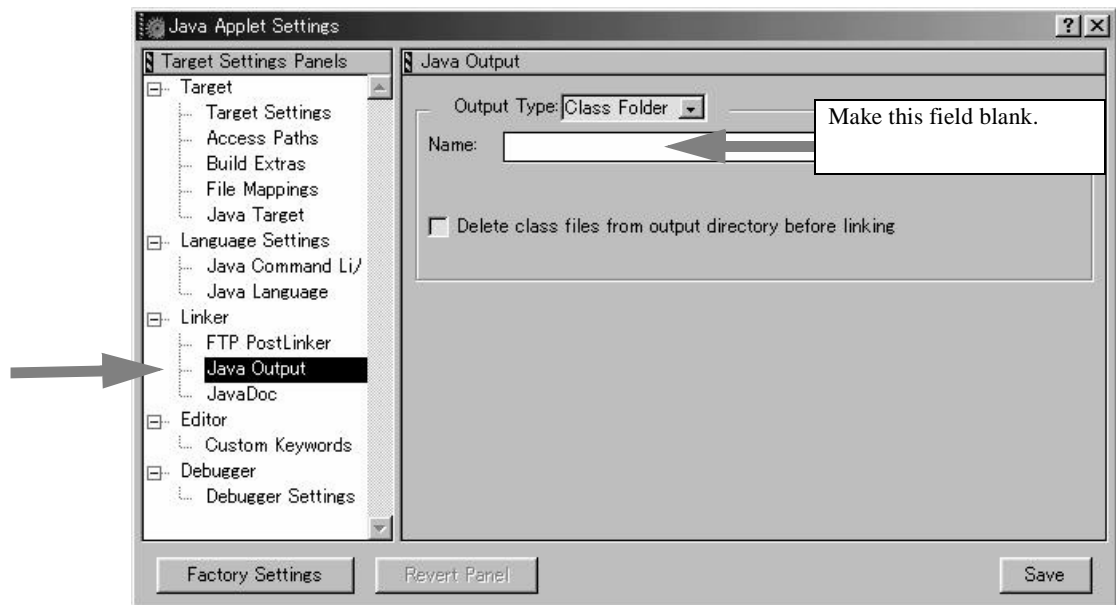
## **10.0 Running Applets**

---

Steps for compiling and executing applets are basically identical to those for applications. The main difference is that an applet requires an HTML file. The role of HTML files is explained in Chapter 2. We will explain how to run applets in a summary format. We assume the folder for the project is **D:\JavaProjects\MyApplet**.

1. Create a project of type **Java Applet**. Name the project **MyApplet**. This should create the folder **D:\JavaProjects\MyApplet**. Please review Section 4.0.
2. Remove the automatically added source files **TrivialApplet.htm** and **TrivialApplet.java** from the project.
3. Create a new java source file and an HTML file. Name the source file and the HTML file with the class name. Example: The source file for **MyFirstApplet** is saved as **MyFirstApplet.java**, and the corresponding HTML file is saved as **MyFirstApplet.html** (or **MyFirstApplet.htm**). Please refer to Chapter 2.  
Save the files in the folder **D:\JavaProjects\MyApplet**.
4. Add the newly created source and HTML files to the project.
5. Set the output folder for the bytecode files inside the **Java Applet Settings** dialog. Select the menu choice **Edit|Java Applet Settings**. Click the **Java Output** item

under the Linker category. The value for the Name field is Java Classes. Erase this text. Click the **Save** button, and close the dialog.



This setting specifies where the bytecode files are saved for the applet. By default, they are saved in the **Java Classes** folder under **MyApplet**. We want the bytecode files to be saved at the project folder level, that is, **D:\JavaProjects\MyApplet**. We specify this by leaving the **Name** field blank. *Note: this setting is required only for applets.*

6. Run the applet. *Note: When you run an applet for the very first time, you may get a dialog asking you to accept the Sun license agreement or not. Click the **Accept** button.*

*Note: You don't add the **javabook** classes to an applet project unless, of course, your applet uses non-user interface classes such as **Convert** and **Format**.*