
How to Run Java Programs Using Microsoft J++

C. Thomas Wu

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

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

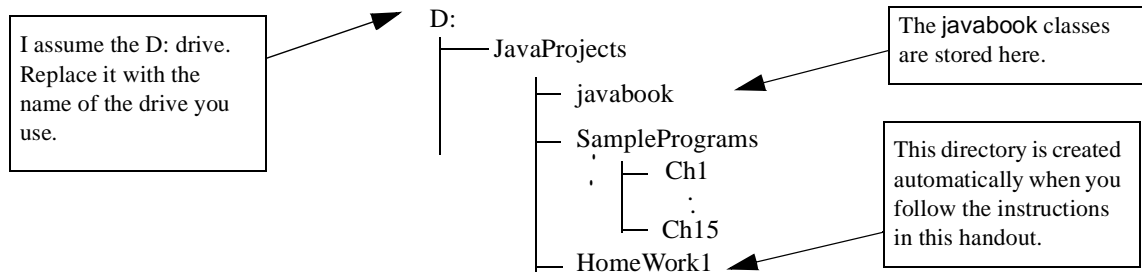
In this document, we will explain how to run Java programs using Microsoft J++ version 6.0.

1.0 Installing the Software and javabook Package

Install Microsoft J++ 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). 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. The javabook folder also includes the source files so you can study them and have an option of modifying them.

2.0 Directory Structure

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



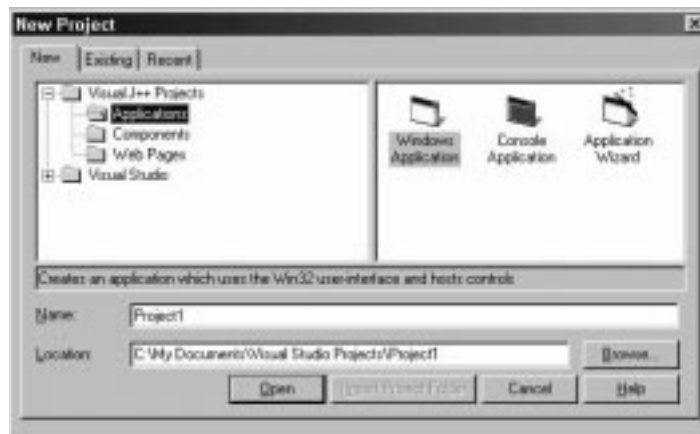
3.0 Starting J++

Start J++ by selecting its program icon in the Start|Program|Microsoft Visual J++ 6.0 group (or whichever group the J++ icon is located).

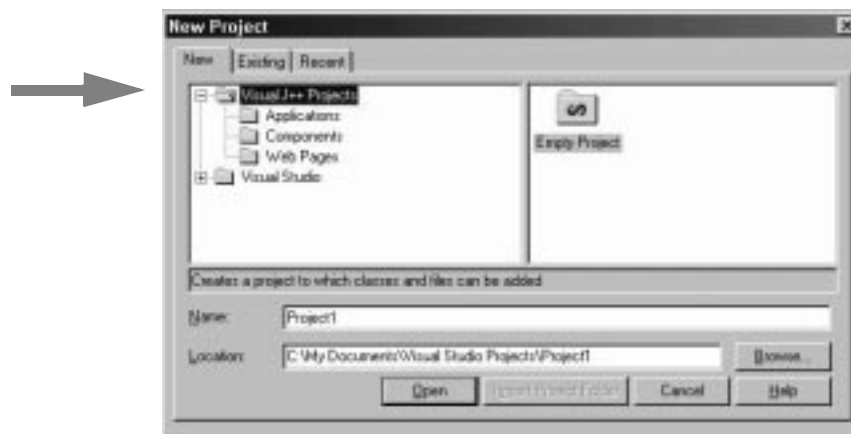
4.0 Creating a Project

J++ 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. In the later chapters, projects will include over 10 files.

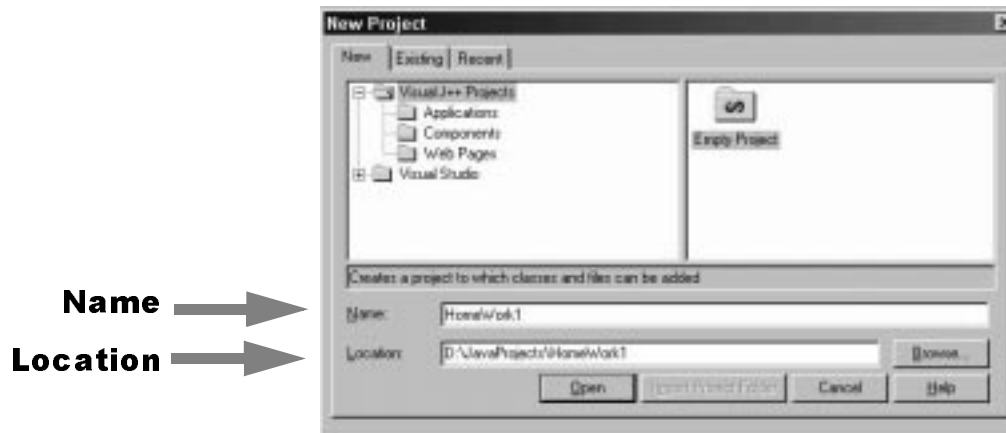
To create a project, select the menu choice File|New Project. The Project Wizard dialog appears on the screen.



In this dialog, a Windows application is selected as the basis for a new project. We do not use any of these predefined templates for our programs. Instead, we will create an empty project. Click the Visual J++ Project folder icon. The dialog will change to



We enter the name of a new project in the **Name** field and place this project under the directory D:\JavaProjects. We will call this project **Homework1**. To set the location of the directory, you can click the **Browse** button and select the directory you want. You need to experiment here to see the various ways of setting the **Name** and **Location** fields. Before you continue, make sure the dialog looks as follows:



When everything is set correctly, click the **Open** button. J++ IDE will create a new folder called **HomeWork1** and place all files related to this project in this folder. You will notice a new project called **HomeWork1** appearing in the Project Explorer window at the top righthand corner. Note: Depending on the option, the Project Explorer window can appear elsewhere. If you don't see the Project Explorer window, select the menu choice **View|Project Explorer**.

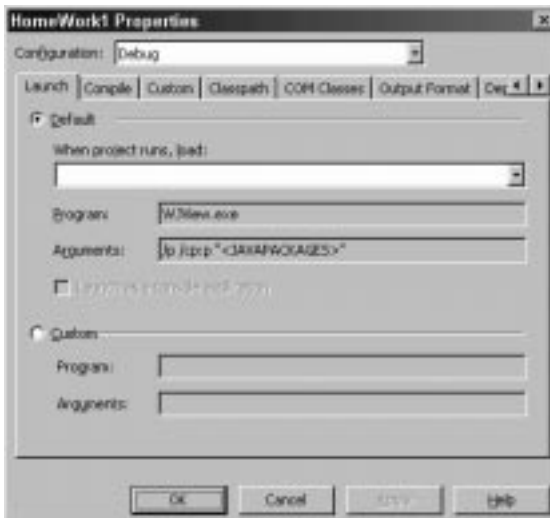


5.0 Specifying the CLASSPATH Property of the Project (Very IMPORTANT!)

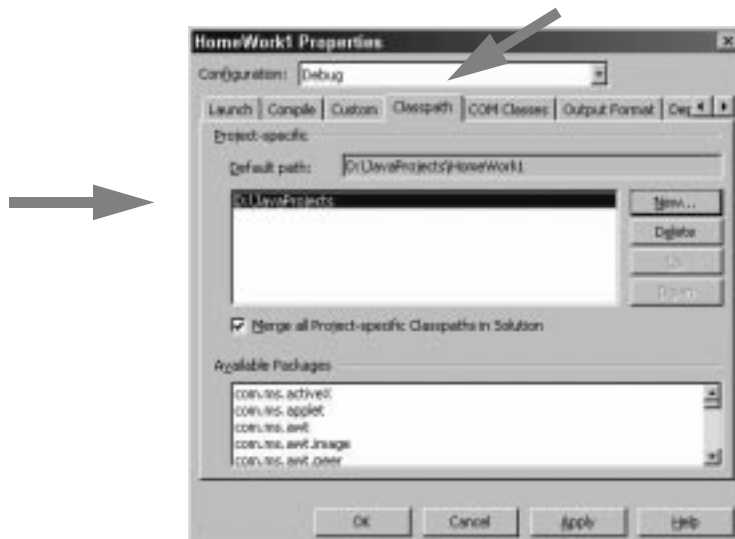
Now that a new project is created, we need to set its CLASSPATH property so the java-book classes can be accessed from the project. Choose the menu choice

Project|HomeWork 1 Properties

Note: If you named the project XYZ, then the menu text would be XYZ Properties. When you select this menu choice, the Properties dialog appears on the screen.



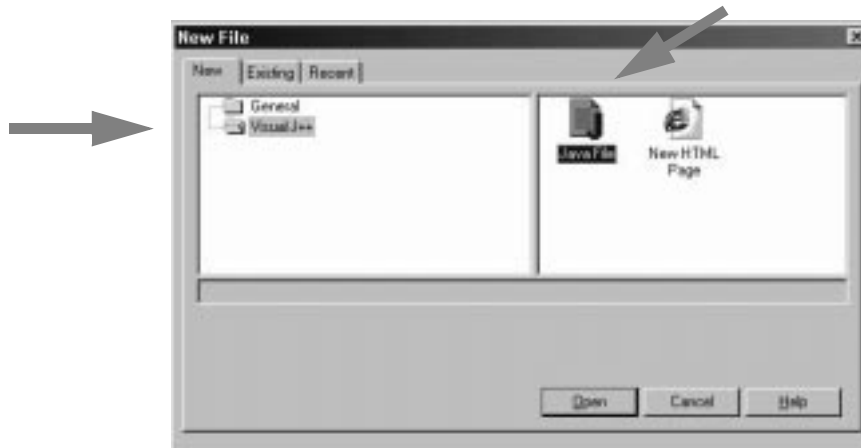
Click the **Classpath** tab. Inside this tab, click the **New** button to add a new class path. Enter the classpath **D:\JavaProjects**. We specify this directory as a new class path because this directory contains the javabook package. Notice that a Java package is stored in a hard disk as a directory. The Properties dialog now looks as follows:



Click the OK button to set this property and close the dialog.

6.0 Creating a Class File (Source Code)

We are now ready to add a class to the project. Let's create the FunTime program from Chapter 1. Choose the File|New File menu choice. The New File dialog opens. Click the Visual J++ folder icon and then select the Java File icon in the right pane. The dialog looks as follows:

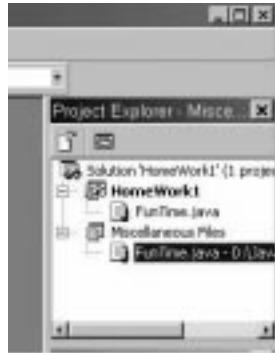


Click the Open button. An editor window to enter the program appears. Type in the FunTime program from Chapter 1. After you entered the program, the window should look something like this:



After the program is entered, choose the File|Save As menu choice to save the program. (The actual text will read Save Java File1 As. In general, the text is Save XXX As with XXX replaced by the temporary name given by the IDE.) Save the file as **Fun-Time.java**. *Make sure you enter the name using lower and uppercases correctly and adding the suffix **java**.*

The Project window now looks like this:



The save source file FunTime.java is added to the project HomeWork1. Notice the same file is added to a new entry called Miscellaneous Files. This portion will be gone when you compile the program correctly and close the editor window.

NOTE: Instead of typing the program, you can add existing source files to the project. To run sample programs, for example, all you have to do is to copy the source files to the project. Let's say you want to test run the DrawShape program from Chapter 6. You can do so by copying the sample files from the Chapter 6 folder to the HomeWork1 folder. Files copied to the HomeWork1 folder will appear automatically in the Project Explorer window. If you do this, make sure you remove other files from the project. I suggest that you create a project called Sample and use this project to run sample programs. Everytime you want to run a new sample program, remove any files currently in the project and copy the source files for the sample program. **Warning: If you remove a source file from a project, the file is also deleted from the project folder also.**

7.0 Running a Java Application Program

We are finally ready to run the program. Choose the menu choice Debug|Start or click the VCR-like start toolbar icon:



If there's no error in the program, the SketchPad window will appear on the screen.

8.0 Quick Summary

Here's a quick summary of the steps you take to enter, compile, and run a Java program using J++.

1. Create a project.
2. Set the project's class path property to point to the javabook package.
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. Run the program.

9.0 Running Applets

Steps for compiling and executing applets are similar 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. In Step 3 of Quick Summary, you create a corresponding HTML file for the applet. Example: `MyFirstApplet.java` for the applet source file and `MyFirstApplet.html` for the corresponding HTML file.