
How to Run Java Programs Using JBuilder2

C. Thomas Wu

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

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

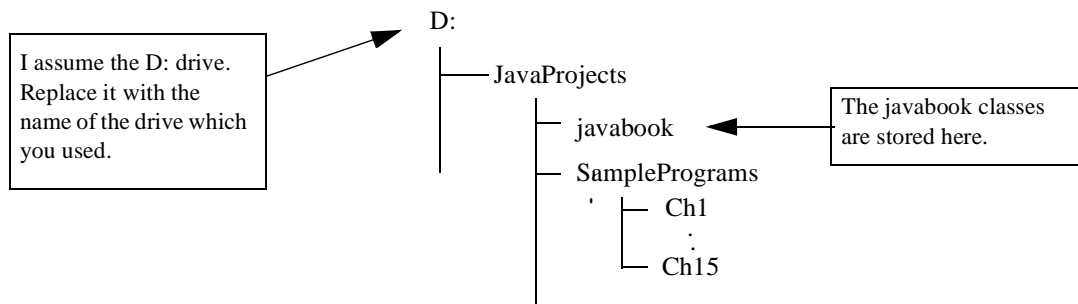
In this document, I will explain how to run Java programs using Borland JBuilder2. JBuilder2 comes in three versions: Standard, Professional, and Client/Server. Necessary steps for running programs are essentially the same for all three versions. I will use the Standard edition for this document. Borland also provides a University edition, which is free to a qualified education institution. For more information on the University, please visit www.inprise.com/programs/education/.

1.0 Installing the Software and javabook Package

Install JBuilder2 on your machine following the directions given by its installation program. Install the javabook package and the sample programs.

2.0 Directory Structure

In this document, I assume the following directory structure:



3.0 Starting JBuilder2

Start JBuilder2 by selecting its program icon in the Start/Program/JBuilder2 group (or whichever group the JBuilder2 icon is located).

4.0 Creating a Project

JBuilder2 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 the File text field, you need to provide the full pathname for the project. A project file will have the suffix `jpr`. You can click on the **Browse** button and use the regular **Open File** dialog and provide the name. You should use the name of the main class as the project name. For instance, we will name the project file for the `MyFirstAppliation` class in the `Ch2` folder `MyFirstApplication.jpr` and place this project file in the directory `D:\JavaProjects\SamplePrograms\Ch2`.

To illustrate the steps in creating a new project from scratch, we will create a project `MyFuntime.jpr` in the `JavaProjects\Homework1` folder. Before continuing, create a folder `Homework1` under the `JavaProjects` folder. Using the **Browse** feature or entering directly into the File text field, specify the full pathname `D:\JavaProjects\Homework1\MyFuntime.jpr`.

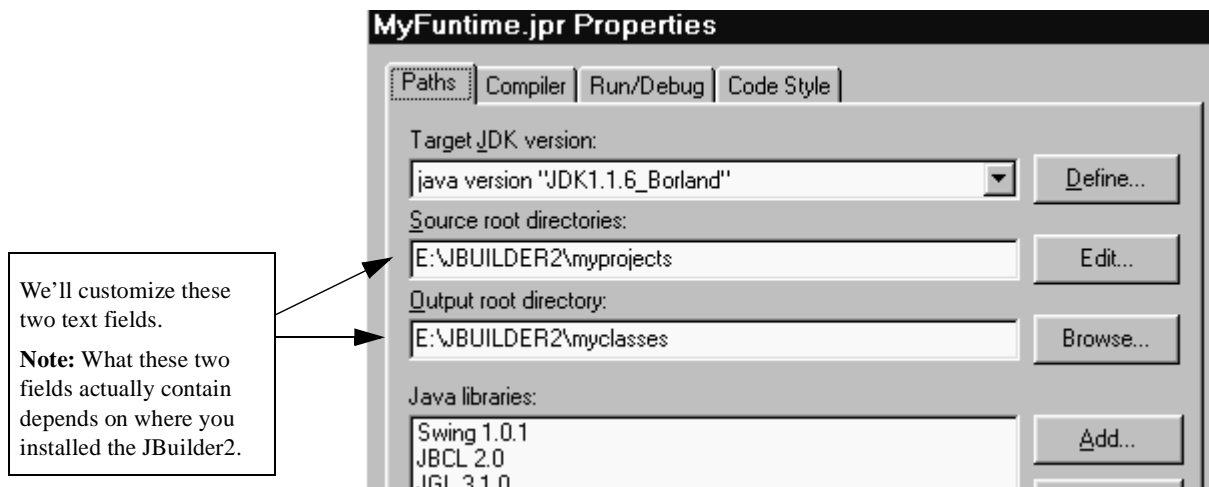
After you specify the project name, click the Finish button. When the project is created properly, you will see the following project window:



Notice that the project includes one HTML file named MyFuntime.html. This file is automatically added by JBuilder2, and you can see its contents on the right side of the project window. This is not a useful feature; you can delete it (by clicking on the file and then clicking the icon with a red Minus sign, right next to the green Plus sign).

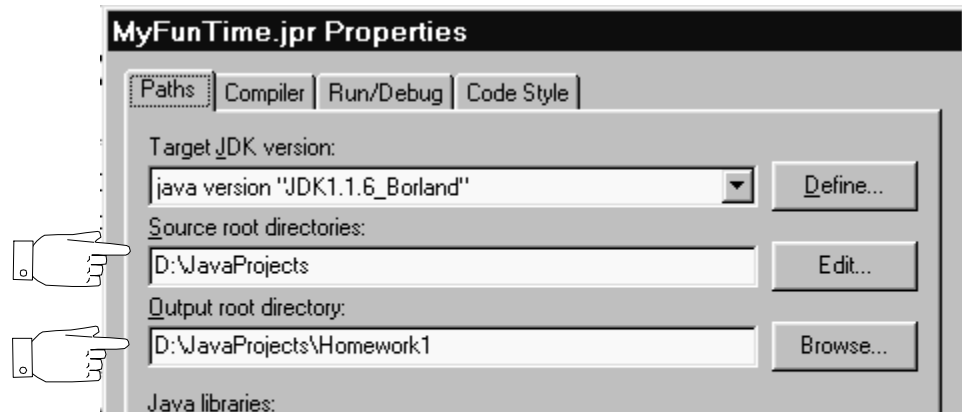
5.0 Setting the Project Properties (Very IMPORTANT!)

JBuilder2 is very picky on where we place files and how we name them. To make your project work correctly, it is very important to set the project's properties first. We use the Project Property dialog to tell JBuilder2 where to find files and where to place the generated class files. To open this dialog, select the menu choice File/Project Properties... The following dialog appears on the screen:



We need to modify the source root directories and the output root directory. The source root directory specifies where to place the source files and where to find the existing source files. The output root directory specifies where the corresponding class (byte-

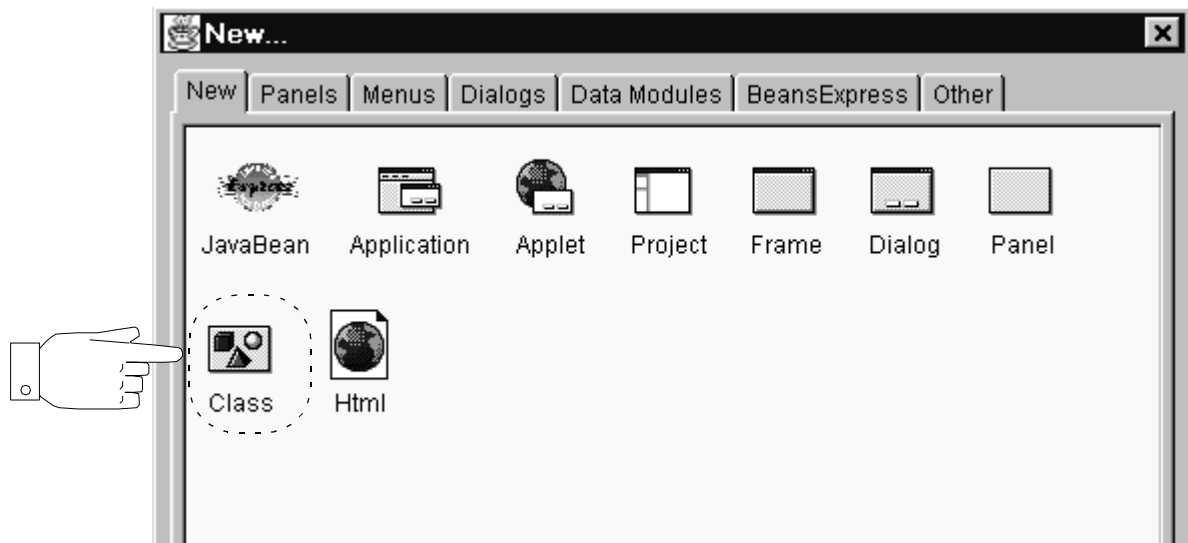
code) files will be saved. You can only specify one output root directory, but multiple source root directories.



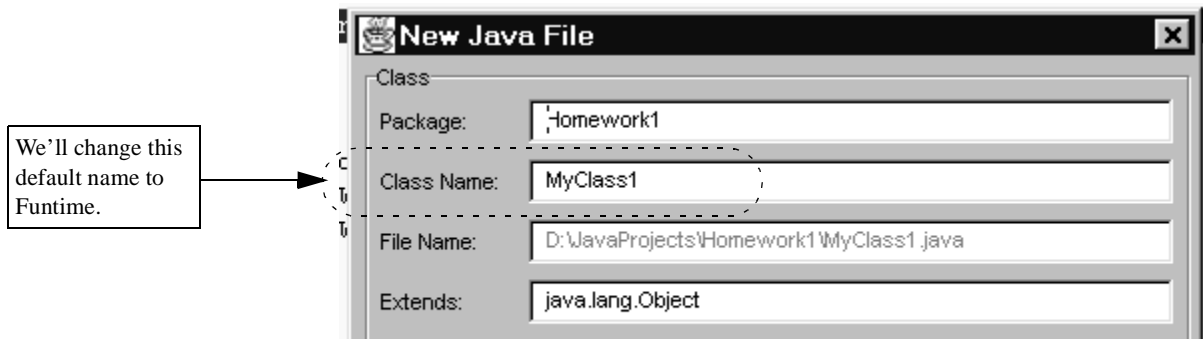
For the source root directories, we put D:\JavaProjects because this directory contains the javabook package. You can include D:\JavaProjects\Homework1 also, but it is not necessary. JBuilder2 will find the source files in the current directory. NOTE: JBuilder2 will create a subdirectory javabook under Homework1. The Homework1/javabook contains the bytecode files. Other IDE won't do this.

6.0 Creating a Class File (Source Code)

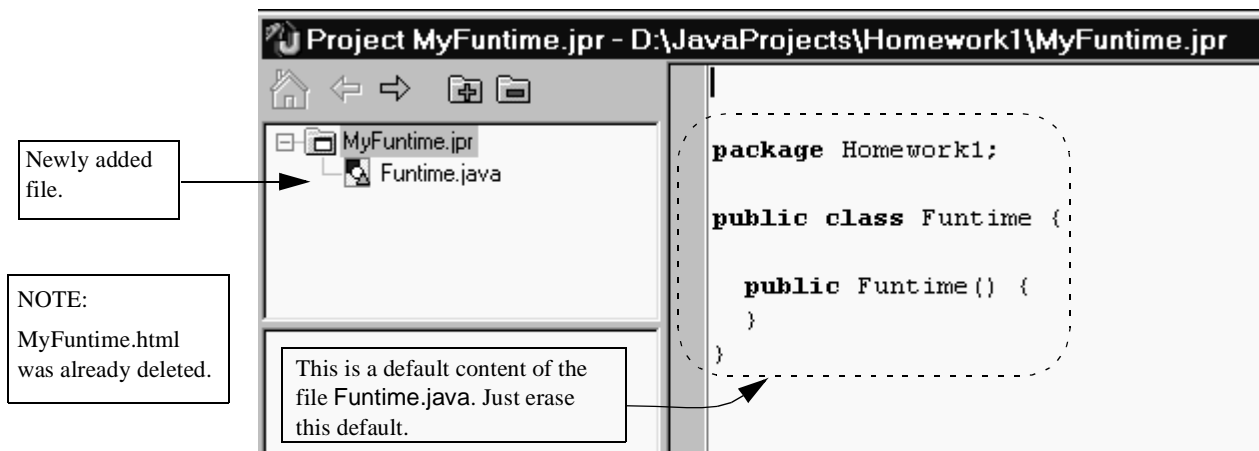
We are now ready to add a (main) class to this project. To add a new class, select the menu choice File/New... The New dialog shows up.



Select the Class icon and click the OK button. You will see the New Java File dialog:



We will change the Class Name to Funtime. Notice that the File Name entry is automatically updated as we type the name in the Class Name text field. Click the OK button to close this dialog. The project window now includes this file:



Erase the whole content of the file that is automatically entered by JBuilder2 and type in the following:

```
/*
    Program Funtime
*/
import javabook.*;

class Funtime
{
    public static void main (String args[])
    {
        SketchPad  doodleBoard;
        doodleBoard = new SketchPad( );
        doodleBoard.show();
    }
}
```

Save the file after you finished entering the text.

WARNING:

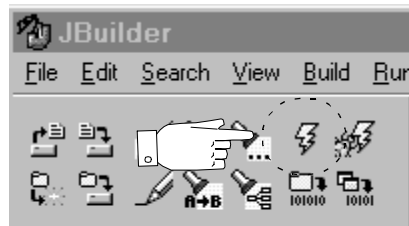
If you use JBuilder 2 University Edition Version 3 (the CD says Third Edition, but when you view its About box, it says JBuilder 2 University Version 2.01), you must put the modifier `public` for the class declaration as in

```
public class FunTime
```

I won't get into the reasons here why the compiler requires this, but in my opinion, the University edition should not impose any additional requirements not required by the other editions. It certainly should not require something that the Java language does not require.

7.0 Running a Java Application Program

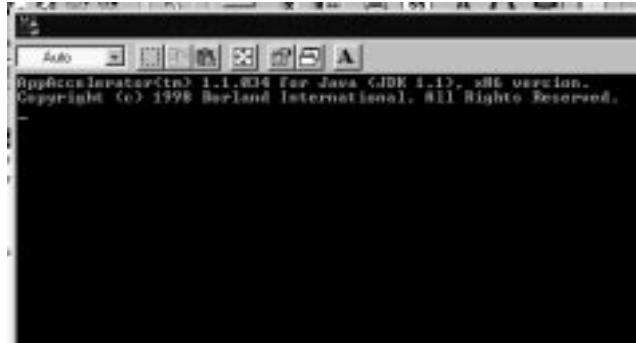
Now we are ready to compile and run the program. Click the thunderbolt icon to compile and run the program.



If everything is done right, the program will run and you will a Sketchpad window:



You will also see the DOS window behind the Sketchpad window:



8.0 Quick Summary

Here's a quick summary of the steps you take to enter, compile, and run a Java program using JBuilder2.

1. Create a project. Name this project differently from the directory that will contain this project. Example: The directory **Homework1** contains the project **MyFun-time.jpr**.
2. Set the project property. Change the source root directories and the output root directory.
3. Create a new class file. Name the source file with the class name. Example: The class **Funtime** is saved in the file **Funtime.java**.
4. Click the thunderbolt icon to compile and 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.