Using NetBeans™ to Compile and Run Java Programs

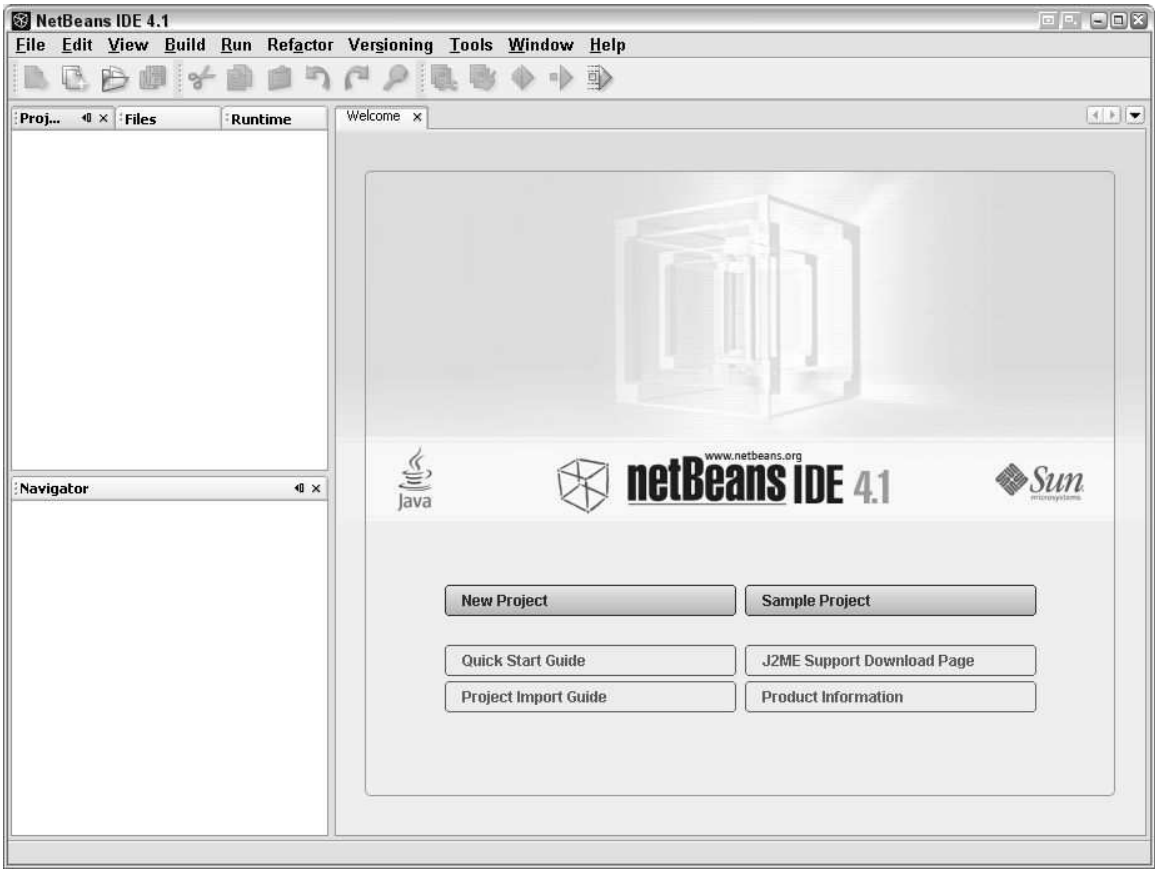
(This document is based on NetBeans 4.1. For information on other versions of NetBeans, contact the author via email: *BeginProg2 at BurdBrain dot com*.)

1. Download NetBeans by visiting http://www.netbeans.org.

2. Install the downloaded NetBeans program on your computer.

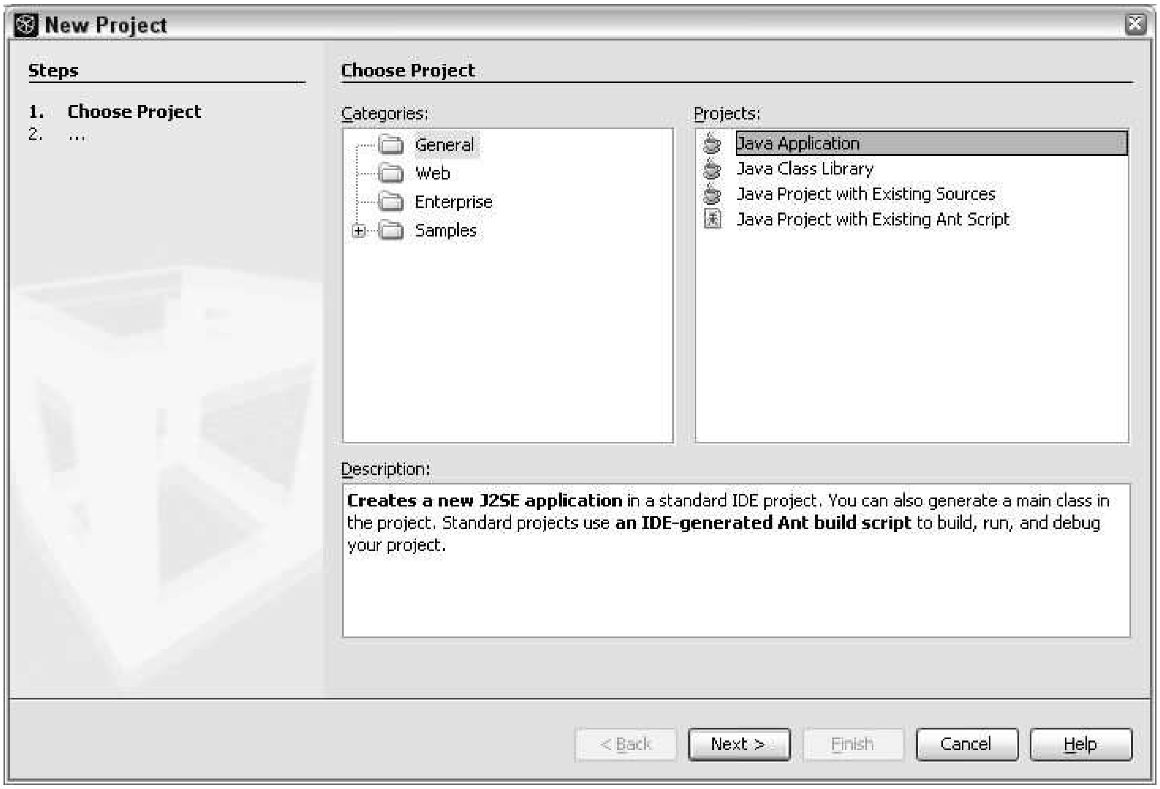
3. Run NetBeans by clicking the NetBeans icon.

You see the NetBeans opening screen.



4. Click New Project.

The New Project dialog appears.

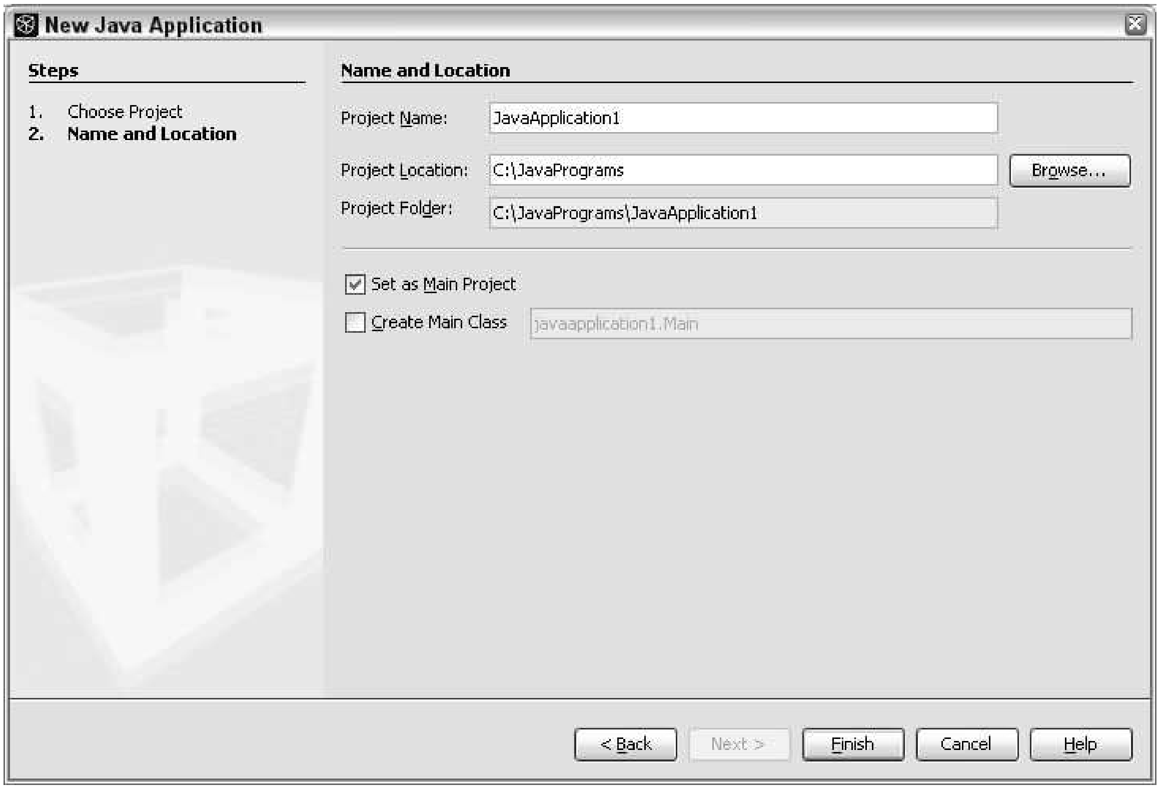


Creates a new J2SE application in a standard IDE project. You can also generate a main class in the project. Standard projects use **an IDE-generated Ant build script** to build, run, and debug your project.

5. In the Categories pane, select General, and in the Projects pane, select Java Application.

6. Click Next.

The New Java Application dialog appears.



New Java Application

Steps

1. Choose Project
2. **Name and Location**

Name and Location

Project Name: JavaApplication1

Project Location: C:\JavaPrograms

Project Folder: C:\JavaPrograms\JavaApplication1

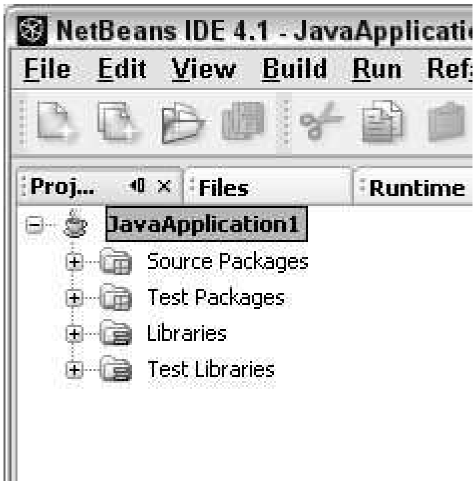
☑ Set as Main Project

□ Create Main Class javaapplication1.Main

7. Unless you're fussy, you can accept whatever defaults you see in the Project Name, Project Location, and Project Folder fields. Leave Set as Main Project checked. But in this example, you should uncheck Create Main Class.

8. Click Finish.

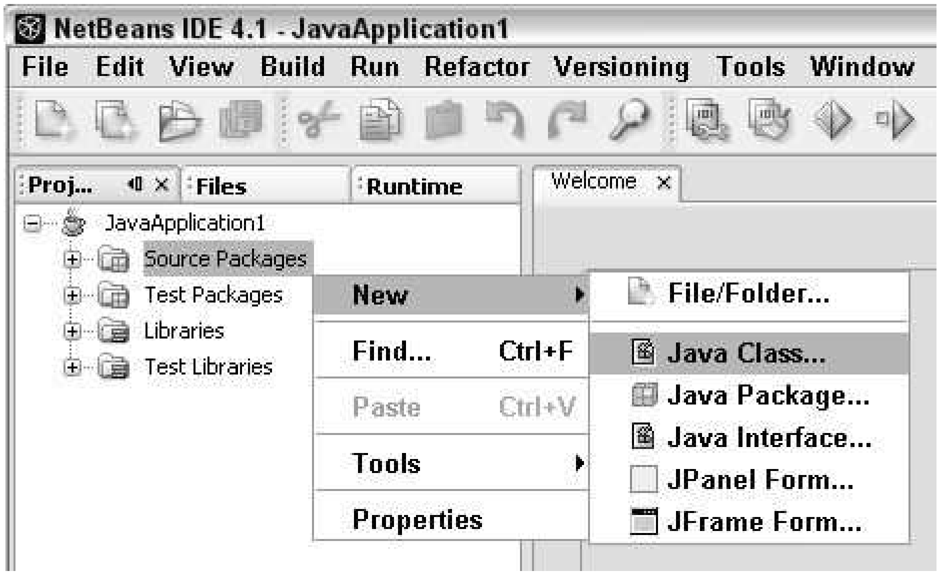
A new project tree appears in the NetBeans Projects window.



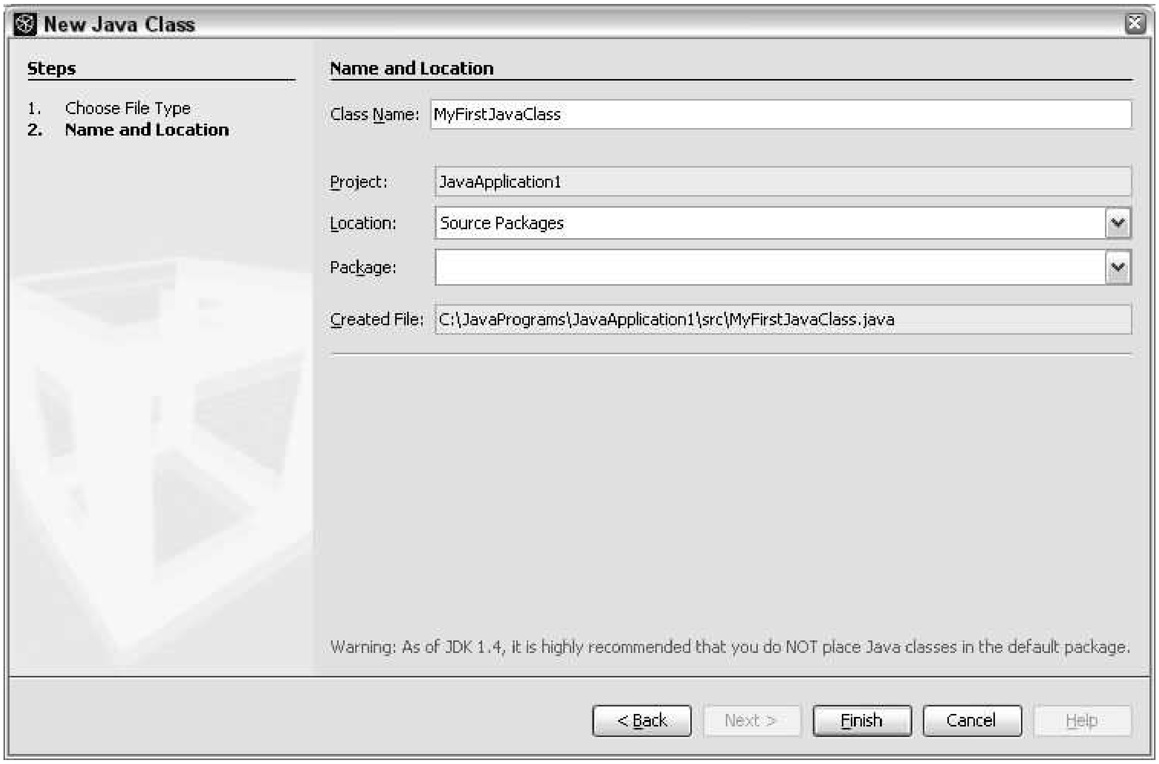
NetBeans IDE 4.1 - JavaApplicati

9. On the project tree, right-click the Source Package branch. In the resulting context menus, select New → Java Class.

On a Macintosh, substitute Ctrl-Click for right-click.



The New Java Class dialog appears.



Class Name: MyFirstJavaClass

Project: JavaApplication1

Location: Source Packages

Package:

Created File: C:\JavaPrograms\JavaApplication1\src\MyFirstJavaClass.java

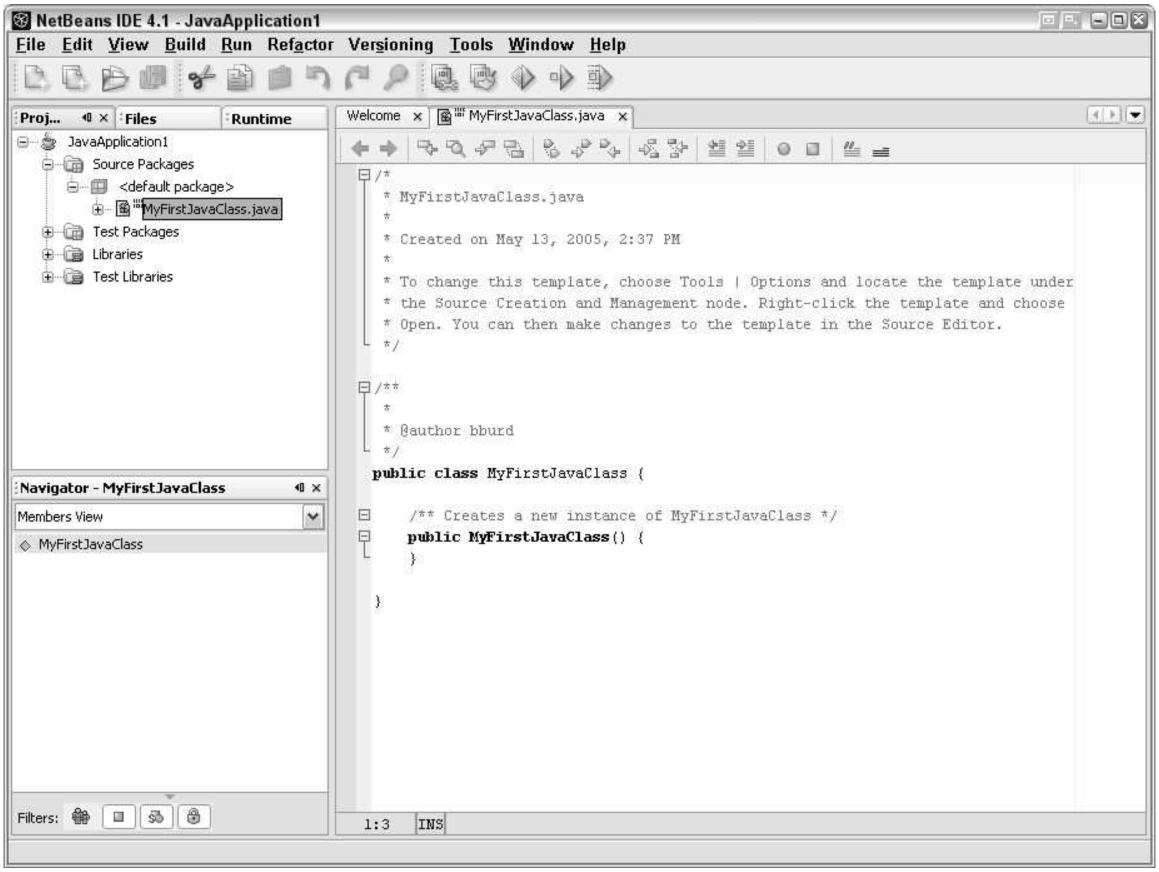
Warning: As of JDK 1.4, it is highly recommended that you do NOT place Java classes in the default package.

10. In the Class Name field, type the name of the class that you want to create.

In this example, I type **MyFirstJavaClass**, because I intend to run the code in Listing 3-1. The name of the Java source file is `MyFirstJavaClass.java`, but in the Class Name field, I type **MyFirstJavaClass**, without the `.java` extension.

11. Click Finish.

The new class appears in the Projects window. Some skeletal code for the new class appears in the editor.



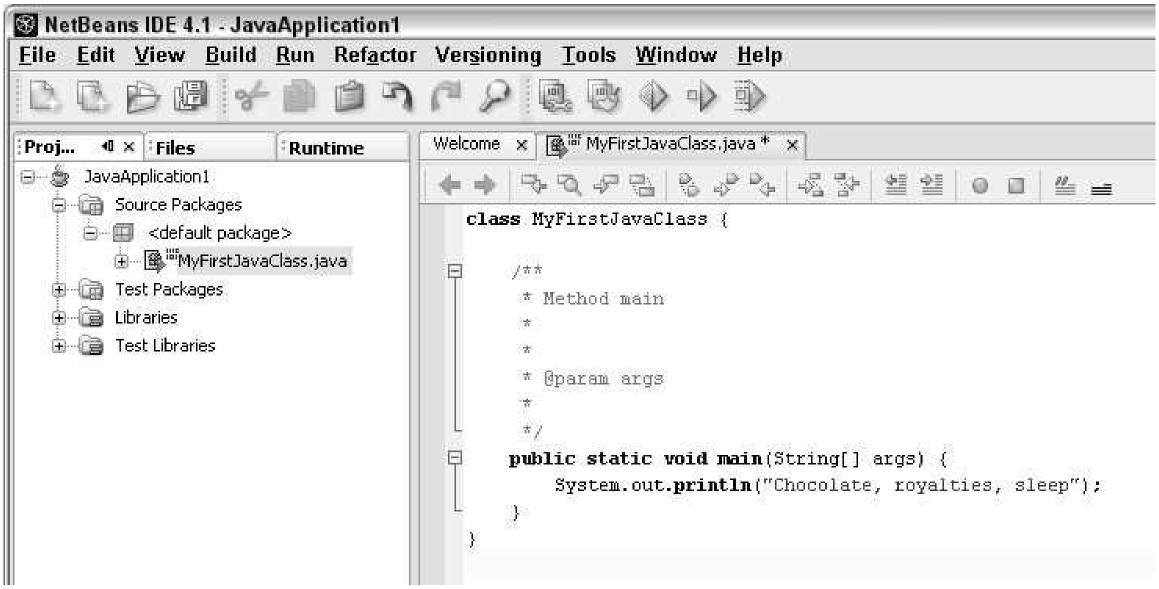
```
/*
 * MyFirstJavaClass.java
 *
 * Created on May 13, 2005, 2:37 PM
 *
 * To change this template, choose Tools | Options and locate the template under
 * the Source Creation and Management node. Right-click the template and choose
 * Open. You can then make changes to the template in the Source Editor.
 */

/**
 *
 * @author bburd
 */
public class MyFirstJavaClass {

    /** Creates a new instance of MyFirstJavaClass */
    public MyFirstJavaClass() {
    }

}
```

12. If you're creating code from scratch, you can type additional code in the editor. But if you're running an example from the book, copy that example's code, and paste the code into the NetBeans editor (replacing the skeletal code that's already in the editor).

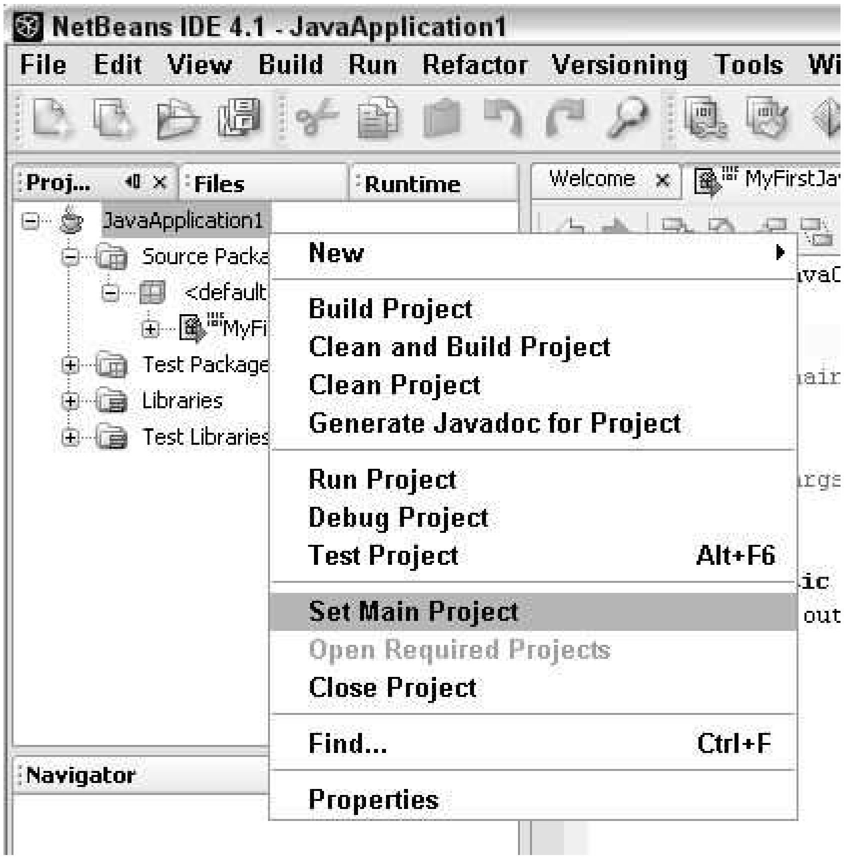


```
class MyFirstJavaClass {

    /**
     * Method main
     *
     *
     * @param args
     *
     */
    public static void main(String[] args) {
        System.out.println("Chocolate, royalties, sleep");
    }
}
```

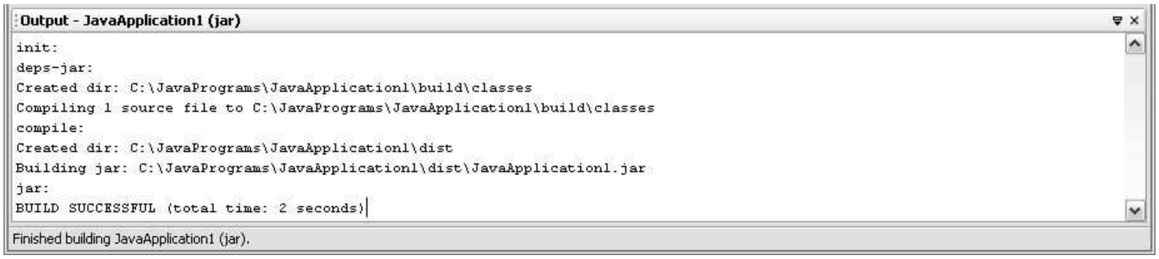
13. On the main NetBeans menu, choose File → Save.

14. In the NetBeans Projects window, right-click the current project branch. In the resulting context menu, select Set Main Project.



15. On the main NetBeans menu, choose Build → Build Main Project.

The output window shows a successful compilation.



Output - JavaApplication1 (jar)

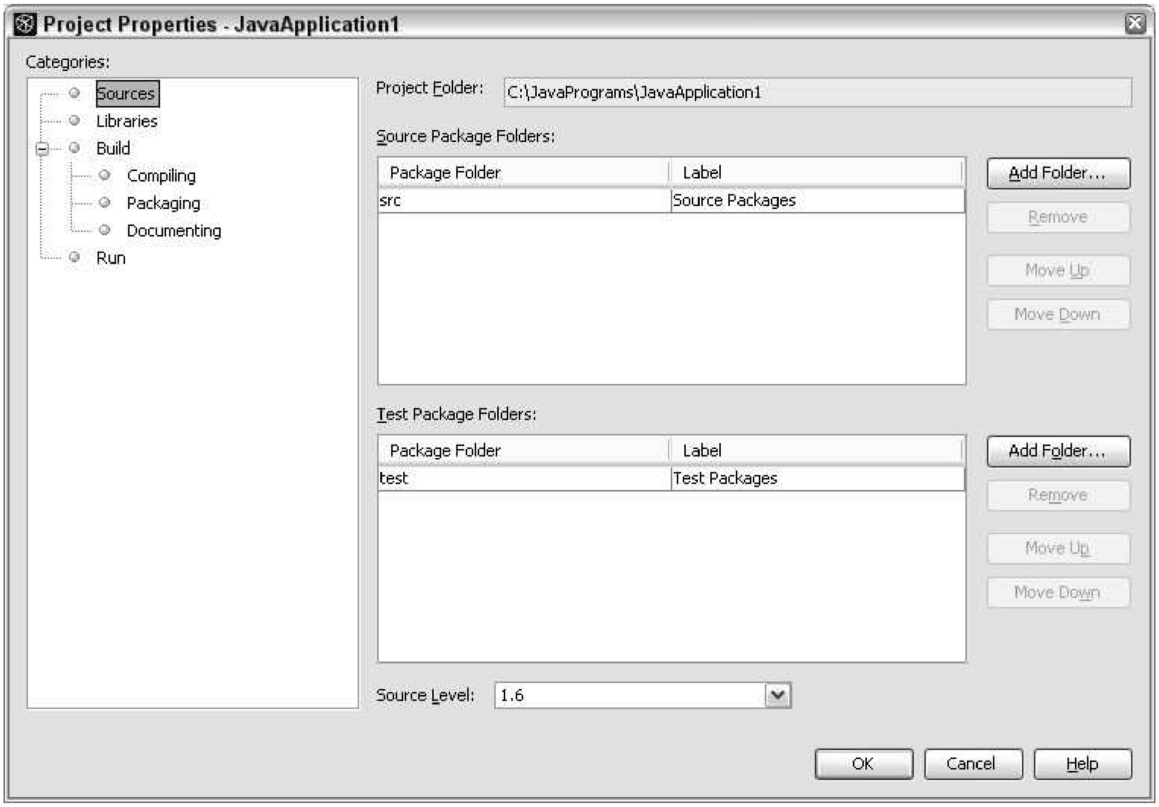
```
init:
deps-jar:
Created dir: C:\JavaPrograms\JavaApplication1\build\classes
Compiling 1 source file to C:\JavaPrograms\JavaApplication1\build\classes
compile:
Created dir: C:\JavaPrograms\JavaApplication1\dist
Building jar: C:\JavaPrograms\JavaApplication1\dist\JavaApplication1.jar
jar:
BUILD SUCCESSFUL (total time: 2 seconds)
```

Finished building JavaApplication1 (jar).

If the compilation isn't successful, make changes to the code in the editor window, and try compiling again.

16. In the Projects window, right-click the current project's branch. In the resulting context menu, select Properties.

The Project Properties dialog appears.



Project Properties - JavaApplication1

Project Folder: C:\JavaPrograms\JavaApplication1

Source Package Folders:

Package Folder	Label
src	Source Packages

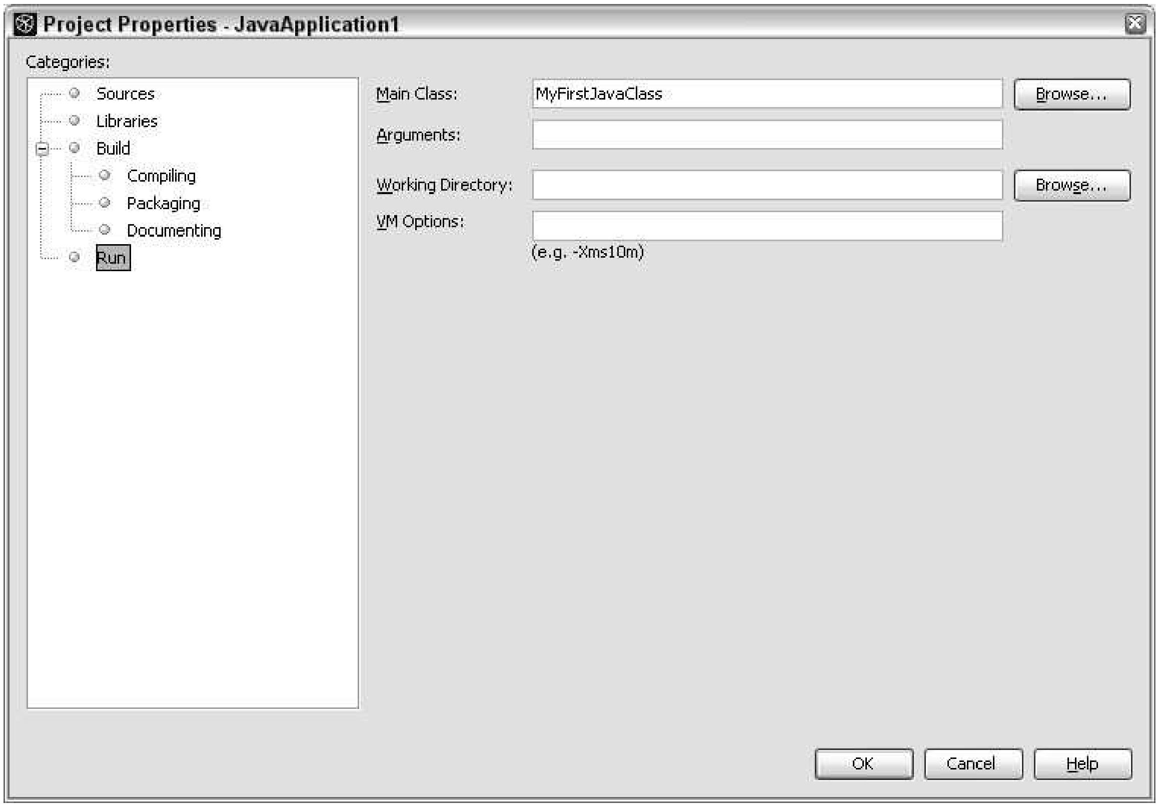
Test Package Folders:

Package Folder	Label
test	Test Packages

Source Level: 1.6

17. On the left side of the Project Properties dialog, select Run.

The fields on the right side change to include Main Class, Arguments, and so on.



Main Class: MyFirstJavaClass

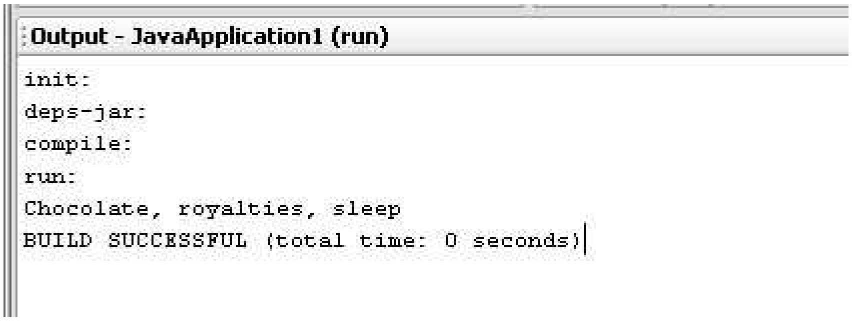
18. In the Main Class field, type the name of the class that contains the main method.

In this example, I type **MyFirstJavaClass**, because I intend to run the code in Listing 3-1. The name of the Java source file is `MyFirstJavaClass.java`, but in the Main Class field, I type **MyFirstJavaClass**, without the `.java` extension.

19. Click OK to dismiss the Project Properties dialog.

20. On the main NetBeans menu, choose Run → Run Main Project.

The program's output appears in the NetBeans Output window, and everybody is happy.



```
init:
deps-jar:
compile:
run:
Chocolate, royalties, sleep
BUILD SUCCESSFUL (total time: 0 seconds)
```