

Importing this book's sample programs

This import business can be tricky. As you move from one dialog box to the next, many of the options have similar names. That's because Eclipse offers many different ways to import many different kinds of things. Anyway, if you follow these instructions, you'll be okay.

1. In Eclipse's main menu, select **File**→**Import**. (See Figure 1.)

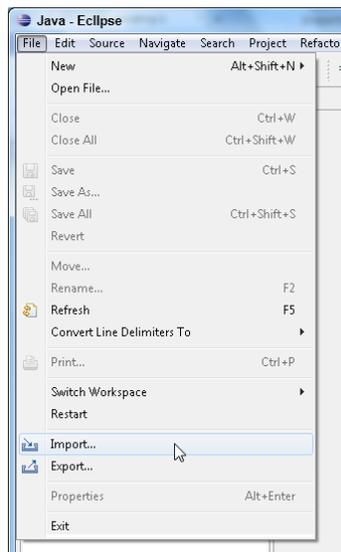


Figure 1: Starting to import this book's code.

As a result, Eclipse displays an Import dialog box.

2. In the Import dialog box's tree, expand the **General** branch.
3. In the **General** branch, double-click the *Existing Projects into Workspace* sub-branch. (See Figure 2.)

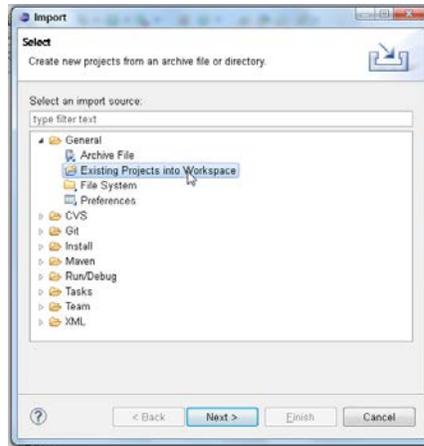


Figure 2: Among all the options, select Existing Projects into Workspace.

As a result, an Import Projects dialog box appears.

4. In the Import Projects dialog box, choose the *Select archive file* radio button. (See Figure 3.)

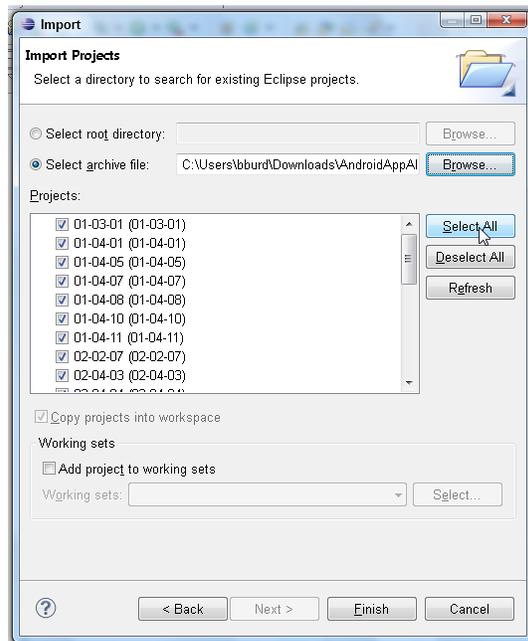


Figure 3: The Import Projects dialog box.

This book's code lives in an archive file named `AndroidAppAllinOneDummies.zip`.

5. Click the Browse button to find the `AndroidAppAllinOneDummies.zip` file on your computer's hard drive.

If you're not sure where to find the file, look first in a folder named *Downloads*.

After you find the `AndroidAppAllinOneDummies.zip` file, Eclipse's Import Projects dialog box displays the names of the projects inside the file. (Again, see Figure 3.)

6. Click the Select All button.

This book's examples are so exciting that you want to import all of them!

7. Click the Finish button.

As a result, the main Eclipse workbench reappears. The left side of the workbench displays the names of this book's Java projects. (See Figure 4.)

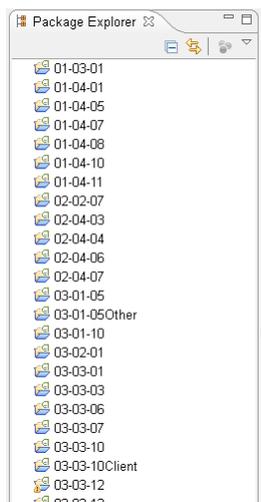


Figure 4: Eclipse displays a bunch of Java projects.

The project numbers are generally of the form

(minibook number)-(chapter number)-(listing number)

For example, project 01-03-01 contains the code from Book I, Listing 3-1.

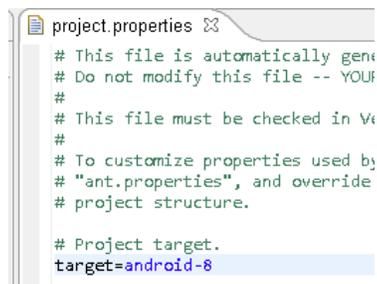


I'm randomly seeing error messages saying the following:

Project has no project.properties file!

If you see errors of this kind, try the following steps:

1. In the Package Explorer, double-click the offending project's project.properties file.
2. In the project.properties file, the last line is something like target=android-8. (See Figure 5.) Make a note of the stuff after the equal sign.



```
project.properties
# This file is automatically generated
# Do not modify this file -- YOUR CHANGES WILL BE LOST
#
# This file must be checked in Version Control Systems
#
# To customize properties used by the build system, use the
# "ant.properties" file and override the values there.
# Project target.
target=android-8
```

Figure 5: The project.properties file.

In Figure 5, the text to remember is android-8.

3. (On Windows:) Right-click the project's branch. (On a Mac:) Control-click the project's branch.
4. In the resulting context menu, click Properties. (See Figure 6.)

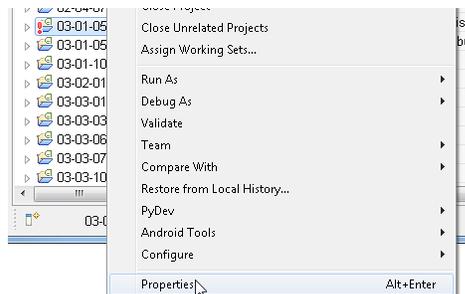


Figure 6: Starting to change a project's properties.

As a result, a Properties dialog box appears. (See Figure 7.)

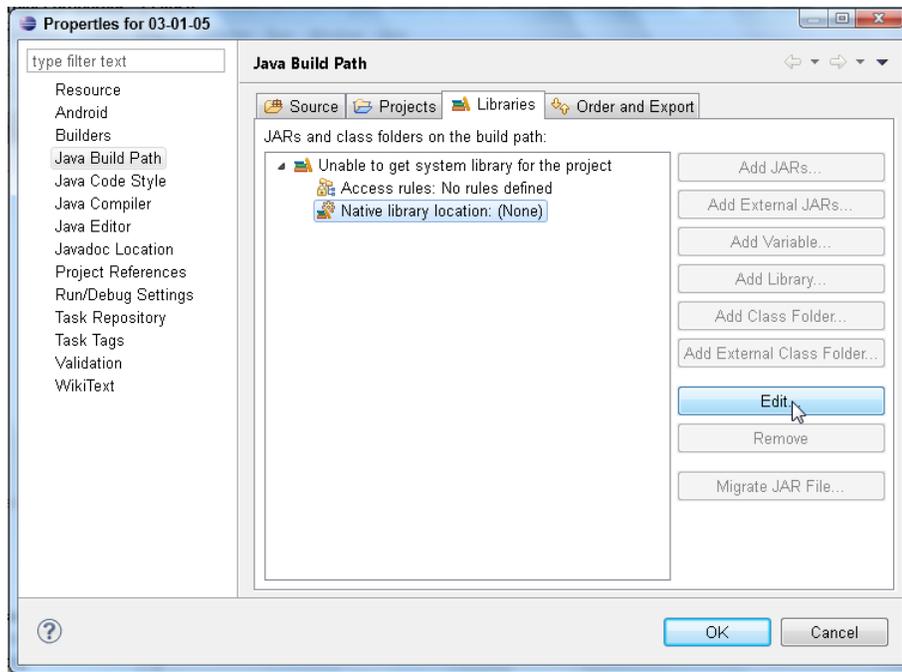


Figure 7: The Properties dialog box.

5. On the left side of the Properties dialog box, look for the Java Build Path.
6. In the body of the Properties dialog box, select the Libraries tab.
7. Expand the entry labeled *Unable to get system library for the project*.
8. Select the sub-entry labeled *Native library location: (None)*.
9. Click the Edit button.

As a result, a Native Library Folder Configuration dialog box appears. (See Figure 8.)

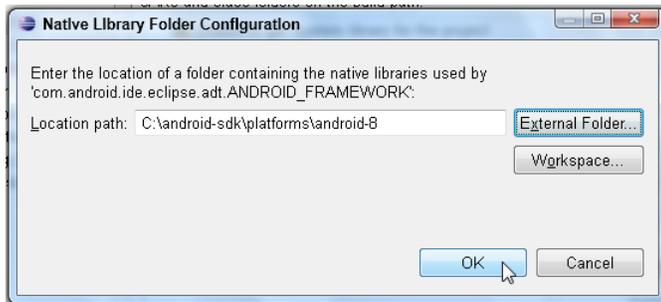


Figure 8: The Native Library Folder Configuration dialog box.

- 10. Browse to the External Folder containing the platform that you found in Step 2.**
- 11. With that External Folder's name in the Location path field, click OK.**
As a result, you're back to the Properties dialog box.
- 12. Click OK again to dismiss the Properties dialog box.**
As a result, you're back to the Eclipse workbench.
- 13. (On Windows:) Right-click the project's branch. (On a Mac:) Control-click the project's branch.**
- 14. In the resulting context menu, select Close Project.**
When the project is closed, the little icon on the project's branch changes into a tiny closed folder.
- 15. Double-click the project's branch.**
Double-clicking reopens the project. The problem is fixed! (... and if the problem isn't fixed, send email to android@allmycode.com.)