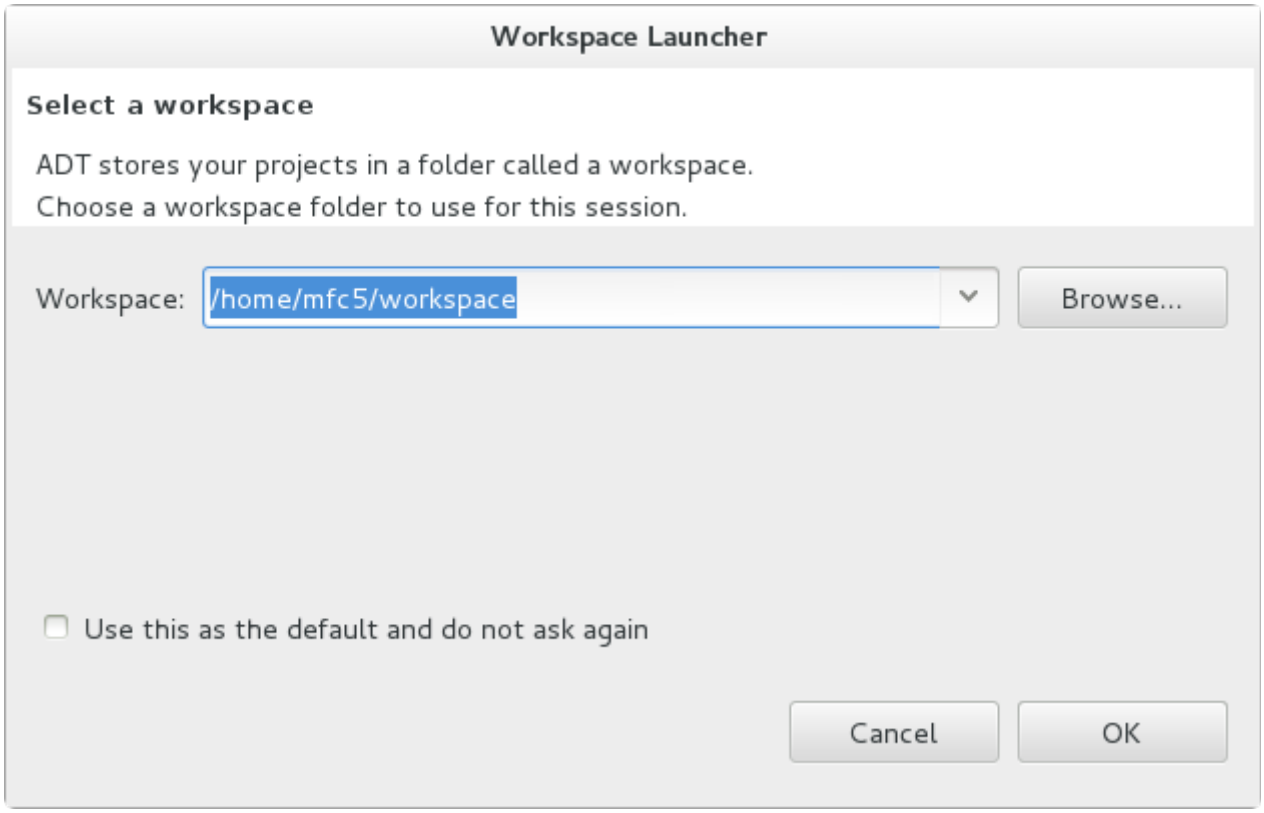
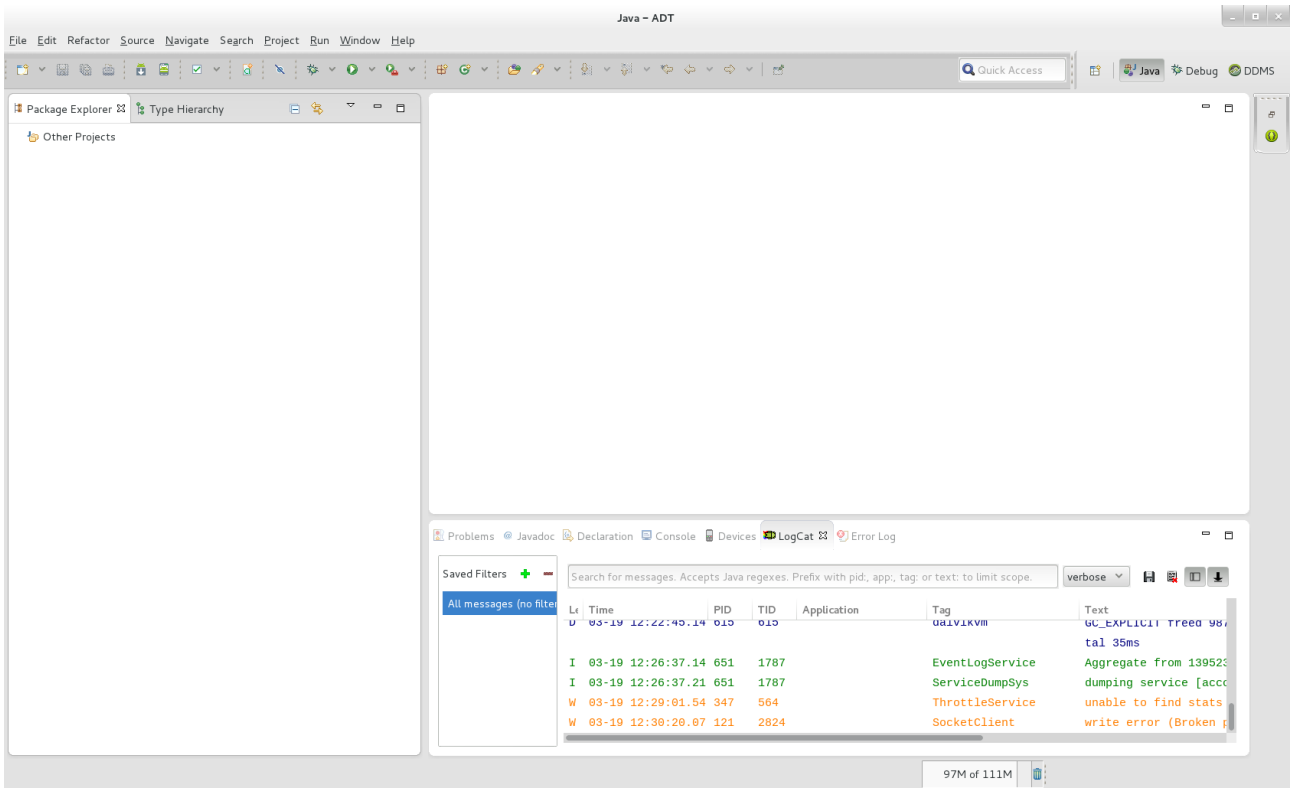
IMPORTING THE ANDROID PROJECT

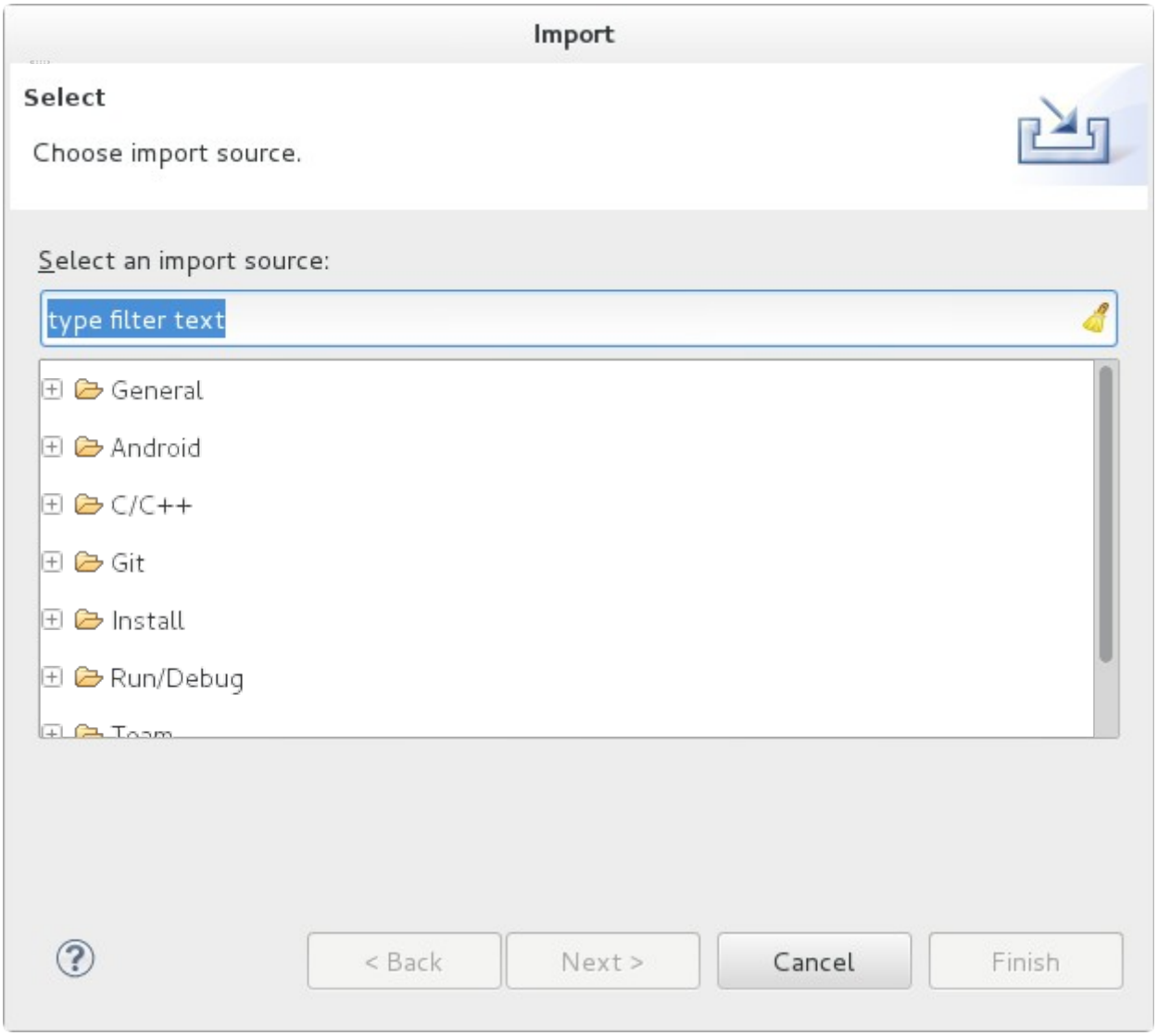
1. Load Eclipse:



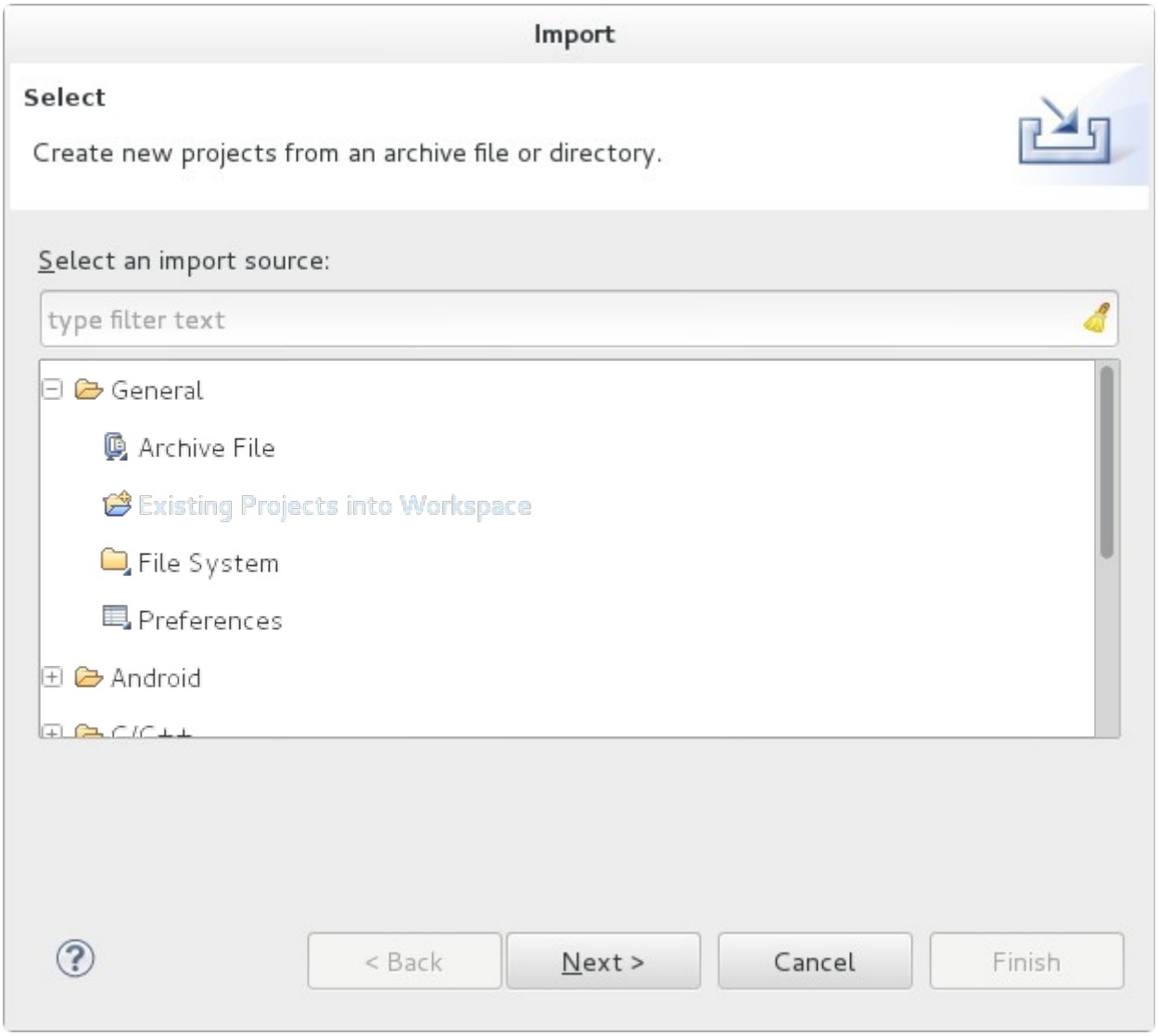
2. Using the default workspace is perfectly fine, simply click OK. The Eclipse IDE will load.



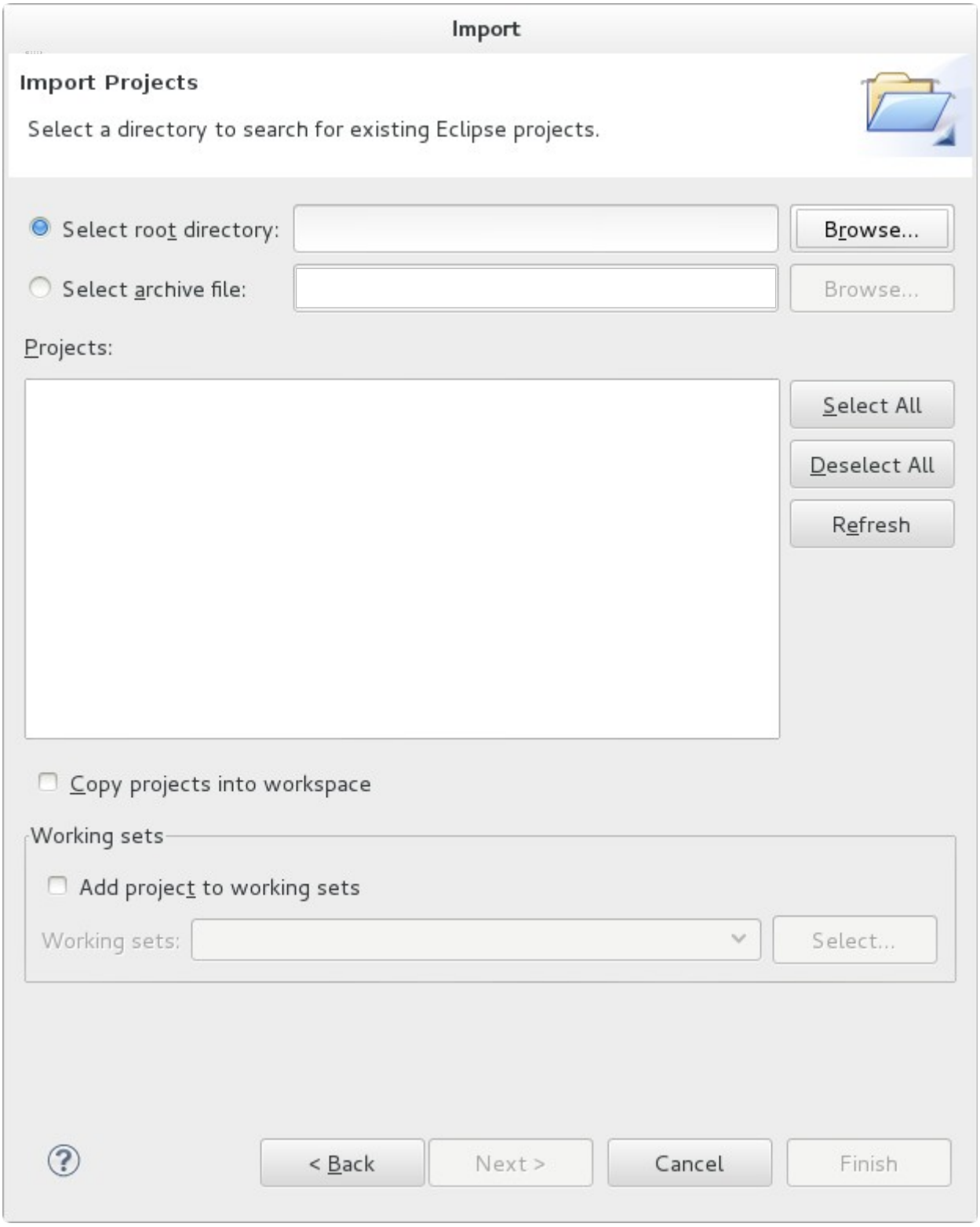
3. Under “Other Projects” on the left hand side, right click and select the menu option “Import...”. The Import dialog box will open.



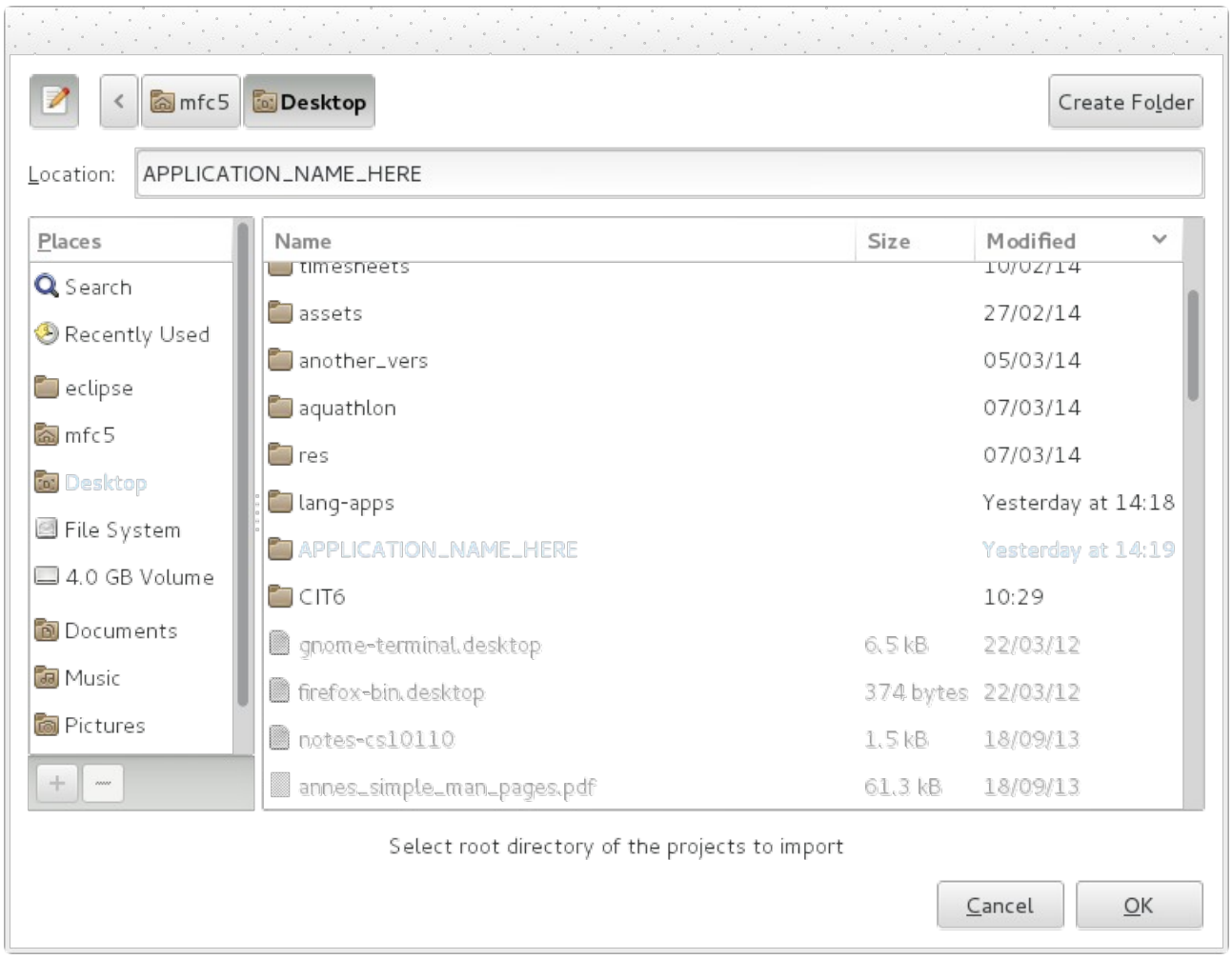
4. Do not click on “Android”. Instead, click on “General” and then click on “Existing Projects into Workspace”.



5. Click “Next”, and then click the first “Browse” button on the right hand side of the next dialog box labeled “Select root directory:”.



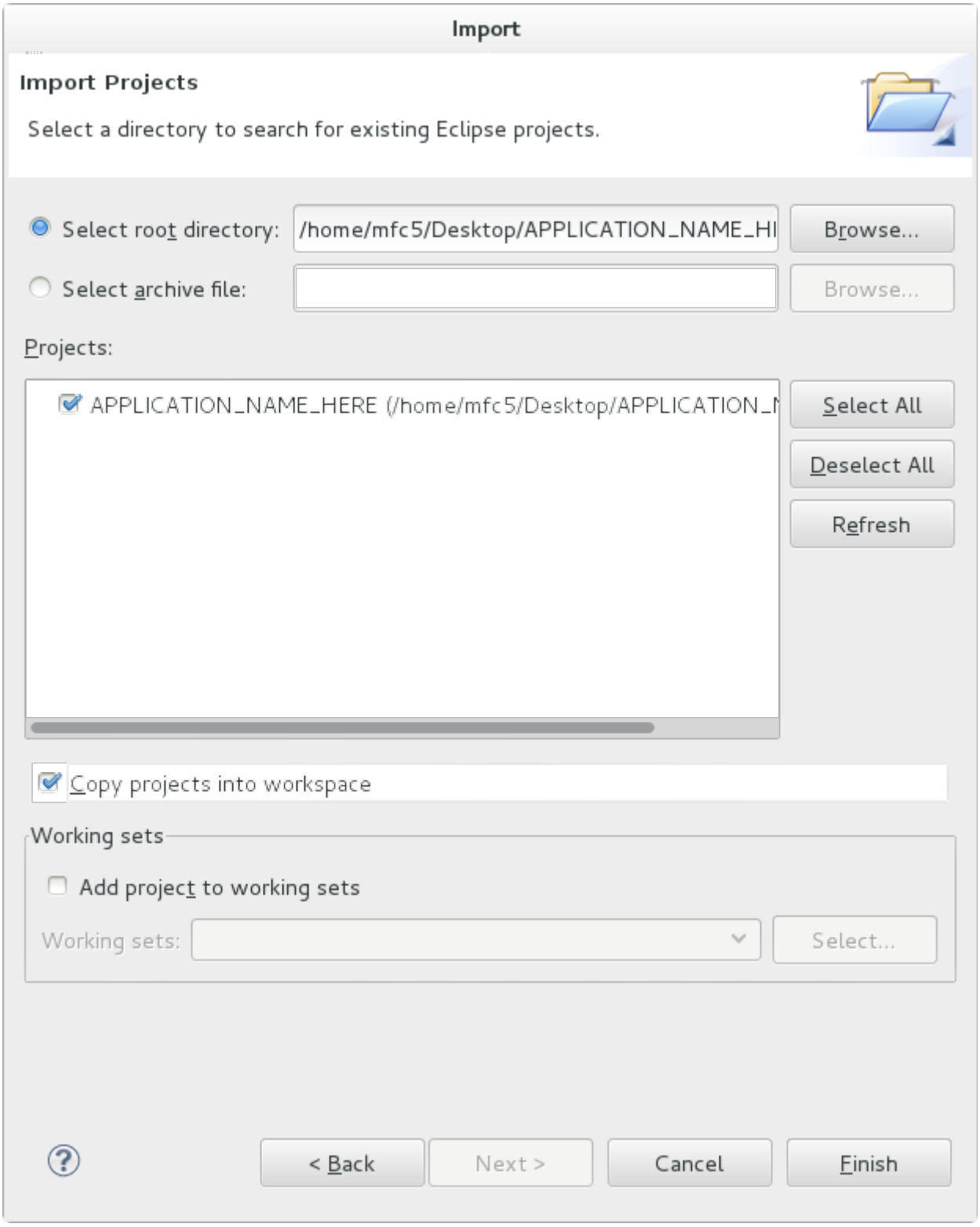
6. Browse to, and select the downloaded Android project folder. It should be called “APPLICATION_NAME_HERE”.



7. Click OK.

8. Back at the “Import” window, make sure you select the option:

“Copy Project into Workspace”.

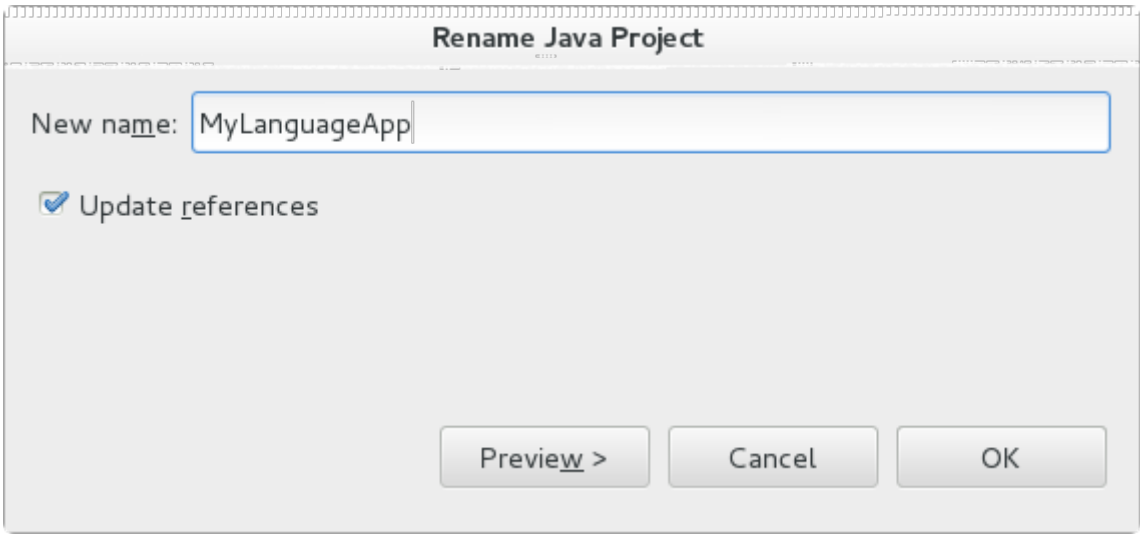


9. Click “Finish”.

You now have the project in your workspace, and we're ready to customize it for deployment.

CUSTOMIZING THE ANDROID PROJECT

1. Right click on the project name, on the left, which is probably called "APPLICATION_NAME_HERE". Select Refactor → Rename.



2. Enter a name for your application, for example "MyLanguageApp" and click OK.

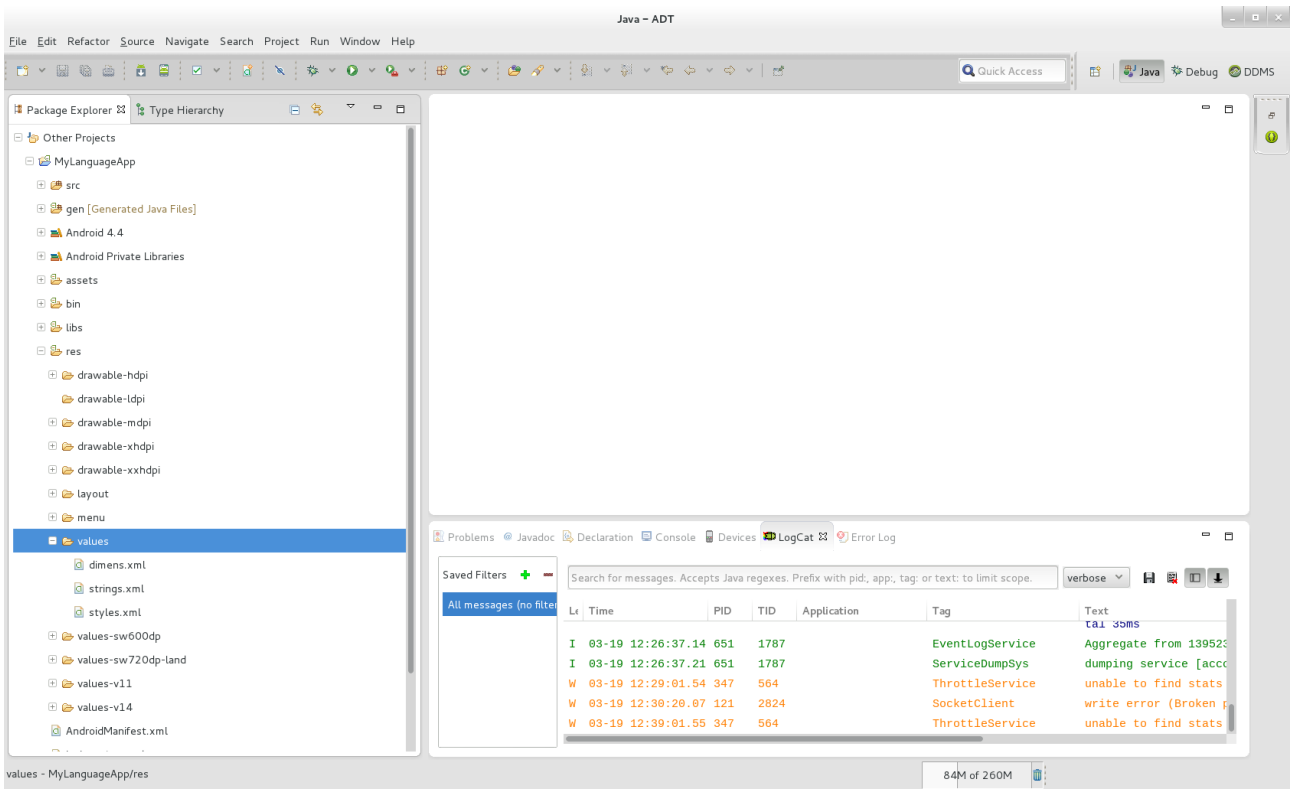
All we have done so far is rename the folder the app is held in, in our workspace. This allows us to work on multiple apps at the same time. We now need to set the application name, so that it is displayed correctly on the users device on the application menu list.

To do this:

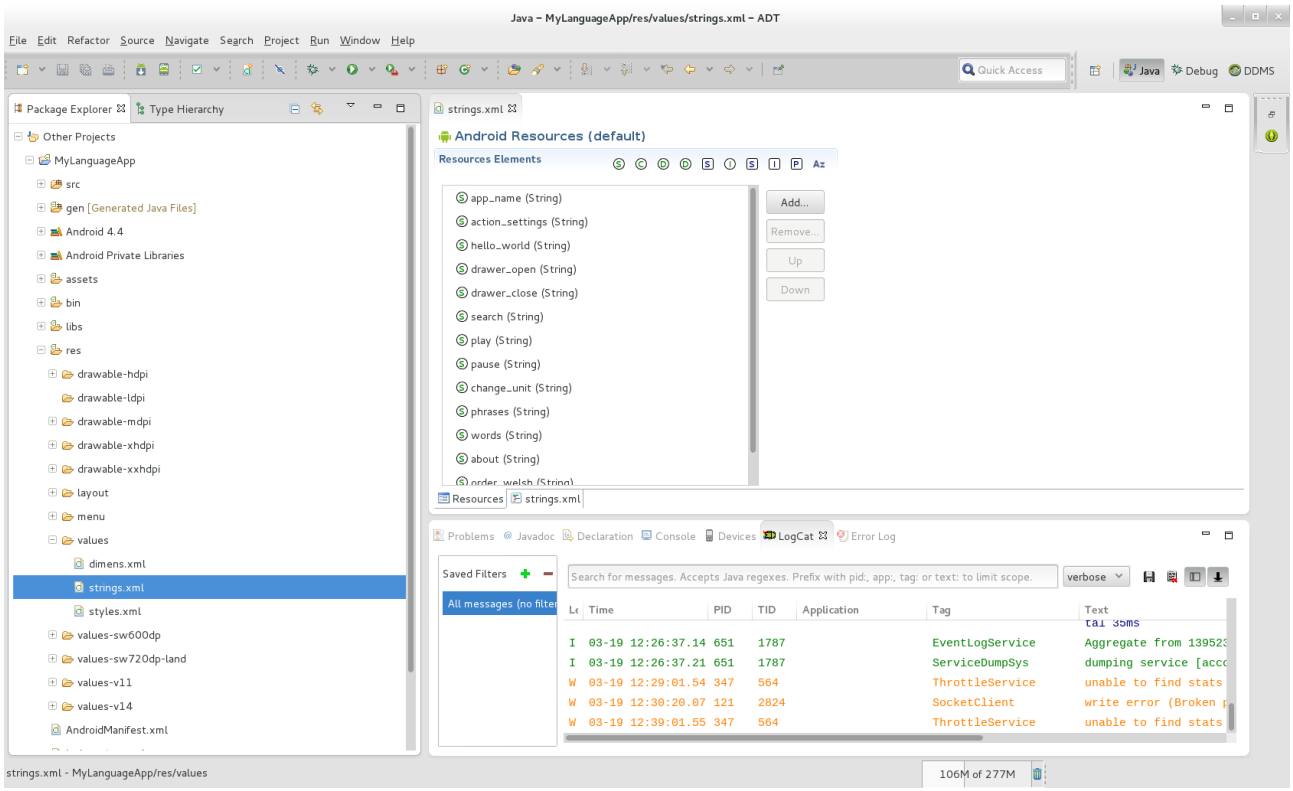
3. Expand the application tree, until you get to the folder:

MyLanguageApp → res → values

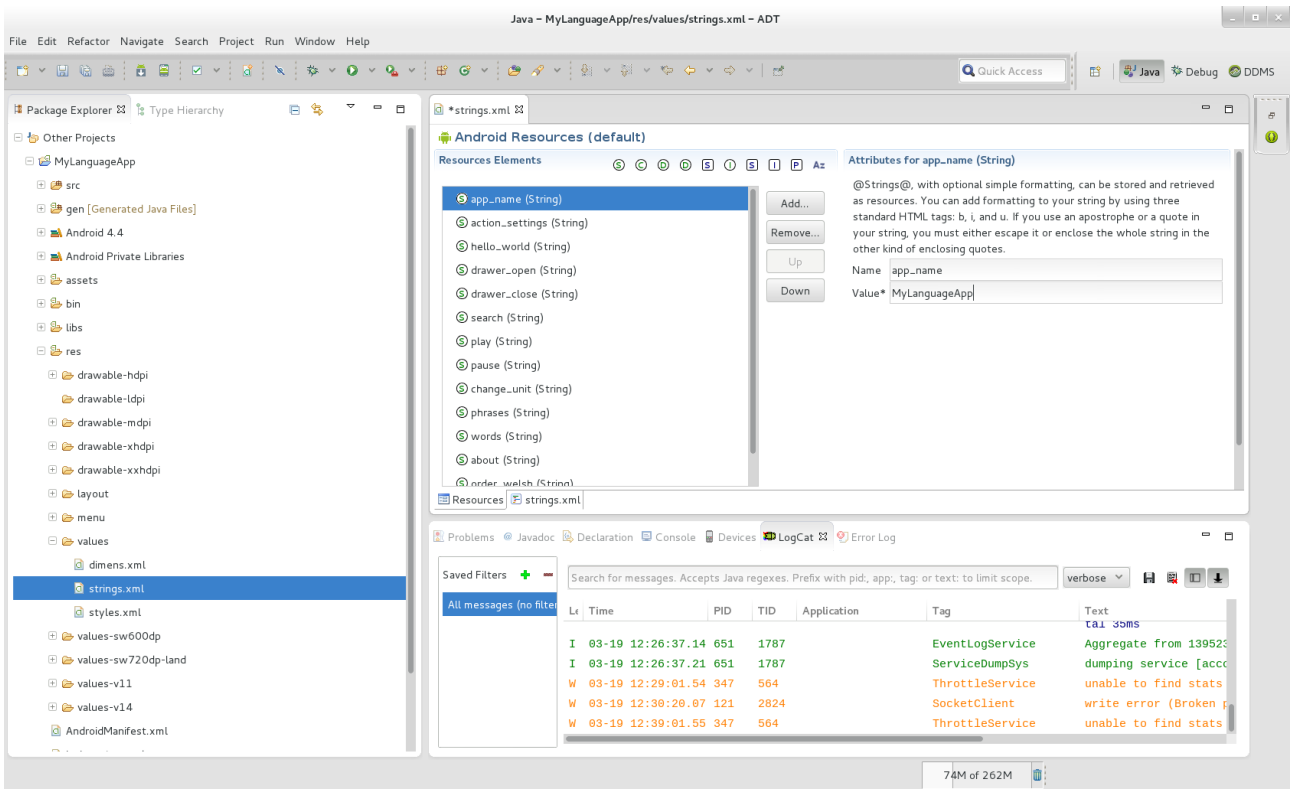
You should see a something like the following:



4. Open the file called “strings.xml” by double clicking on it.



5. Click on the string called “app_name”. On the right hand side you can see it's name, and it's value. It's value should be “APPLICATION_NAME_HERE”. Enter a new name for your application, for example, “MyLanguageApp”.

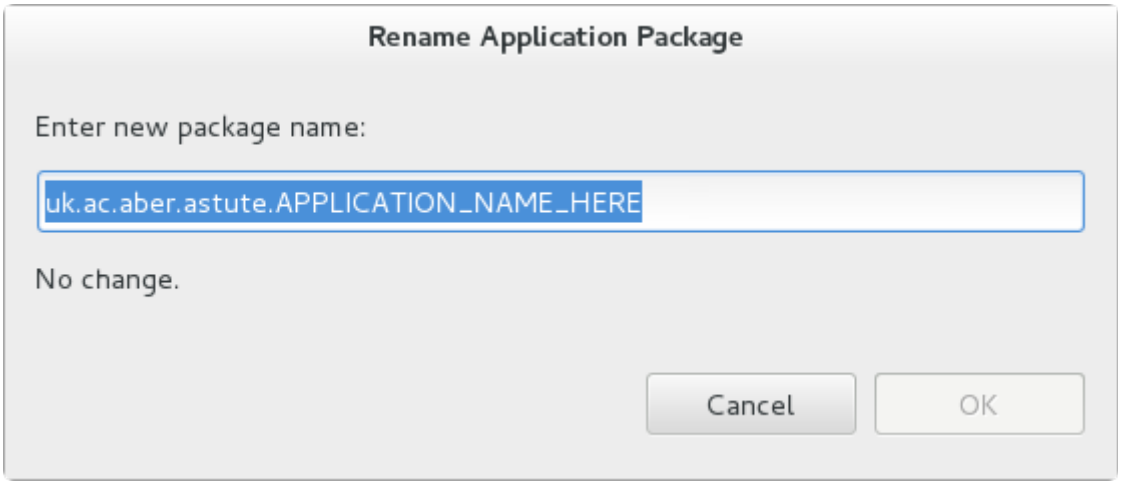


6. Click the “save” icon in the top left of the screen on the menu bar.

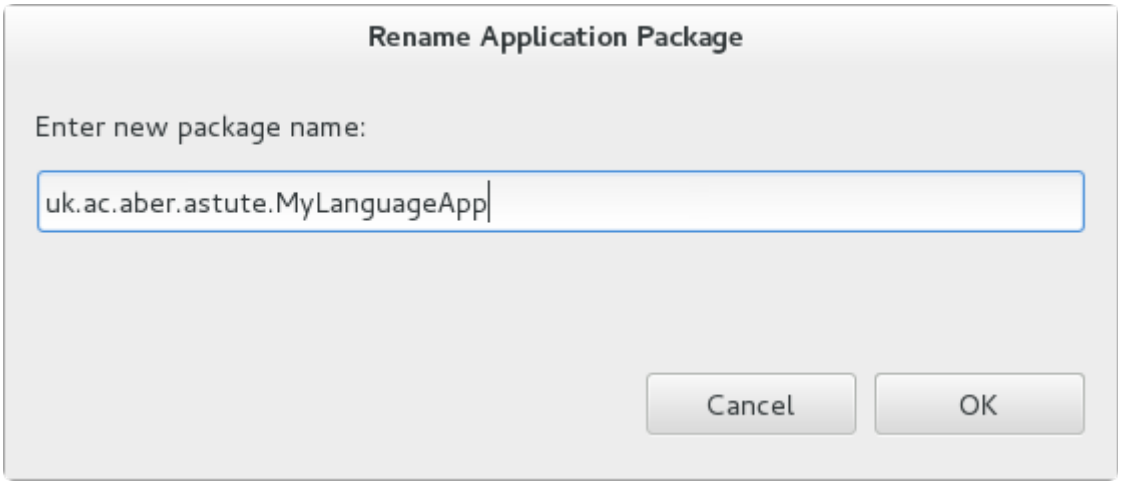
We have no renamed the “application” launcher icon. However, we have more work to do. Android requires that all applications have a completely seperate “workspace” or “package” to function properly. To give our application a new package name we need to use the android tools.

7. Right click on the project name in the tree, as you did to rename the project folder (see step 1). This time, navigate the menus until you get to:

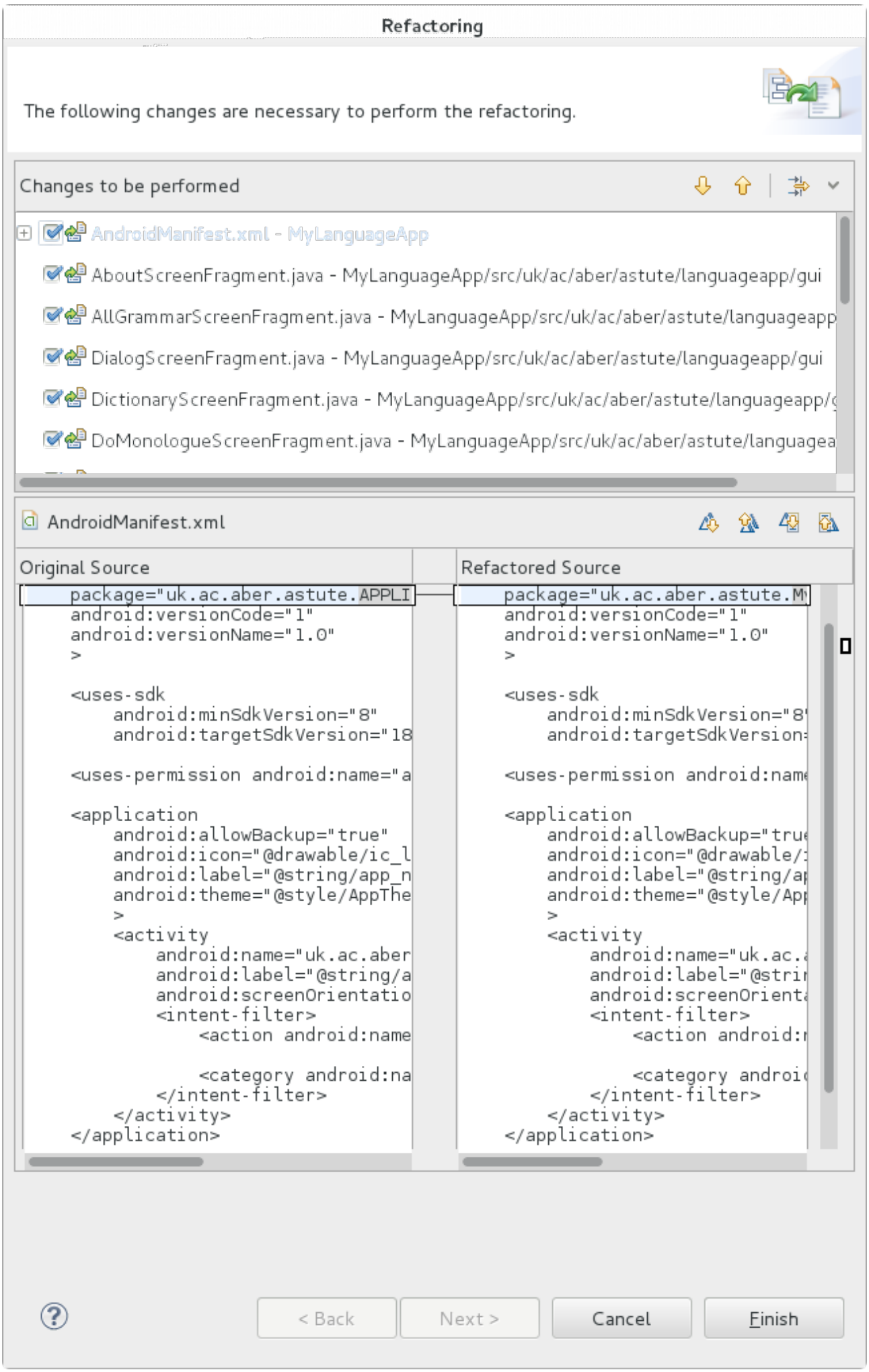
Right click → Android Tools → Rename Application Package



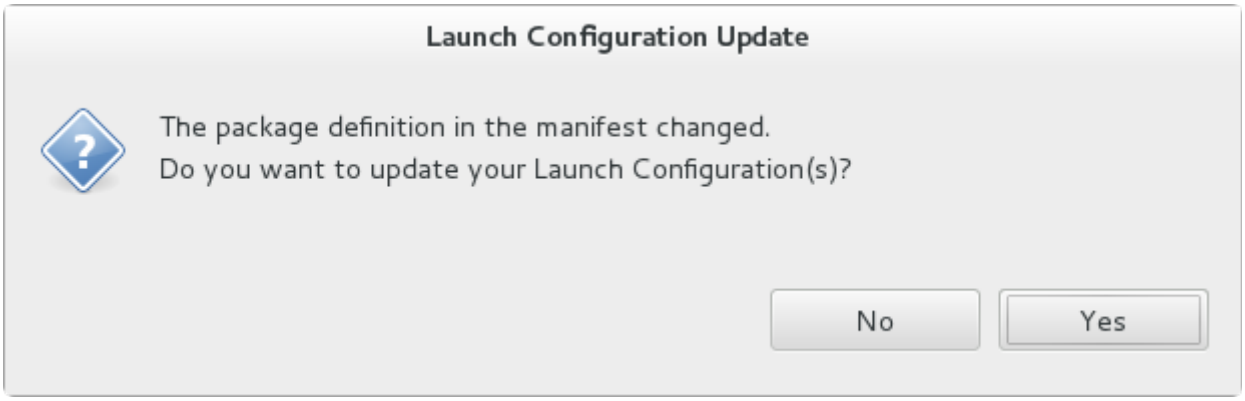
8. You can use any unique package name her. However, it must be in the format of “a.b.c.something.blah”. It cannot have spaces. A simple solution, would be to just use the default but change “APPLICATION_NAME_HERE” to, in our case, “MyLanguageApp”.



9. Click “OK” and you will get a screen that looks like the following:

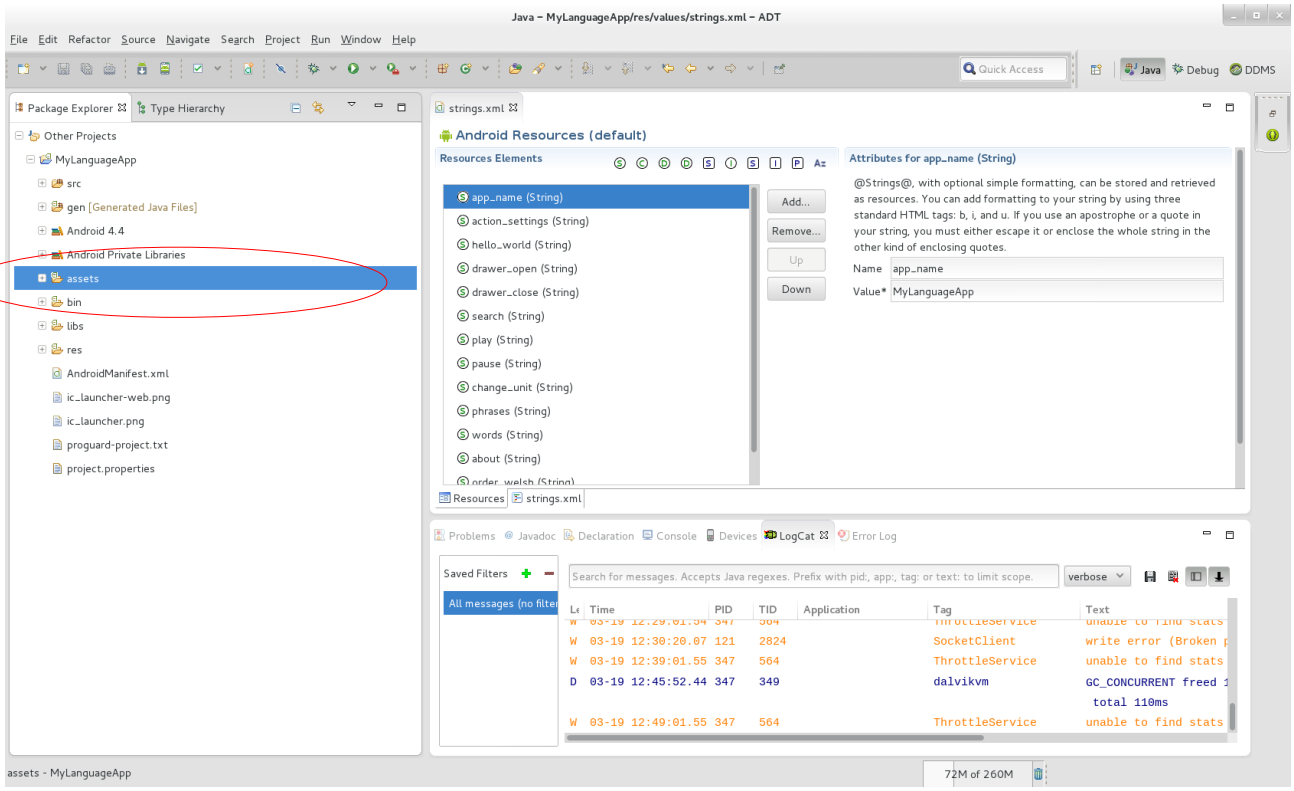

Refactoring
The following changes are necessary to perform the refactoring.
Changes to be performed
AndroidManifest.xml - MyLanguageApp
AboutScreenFragment.java - MyLanguageApp/src/uk/ac/aber/astute/languageapp/gui
AllGrammarScreenFragment.java - MyLanguageApp/src/uk/ac/aber/astute/languageapp
DialogScreenFragment.java - MyLanguageApp/src/uk/ac/aber/astute/languageapp/gui
DictionaryScreenFragment.java - MyLanguageApp/src/uk/ac/aber/astute/languageapp/
DoMonologueScreenFragment.java - MyLanguageApp/src/uk/ac/aber/astute/languagea
AndroidManifest.xml
Original Source
Refactored Source
< Back
Next >
Cancel
Finish

10. Click “Finish” at this screen.

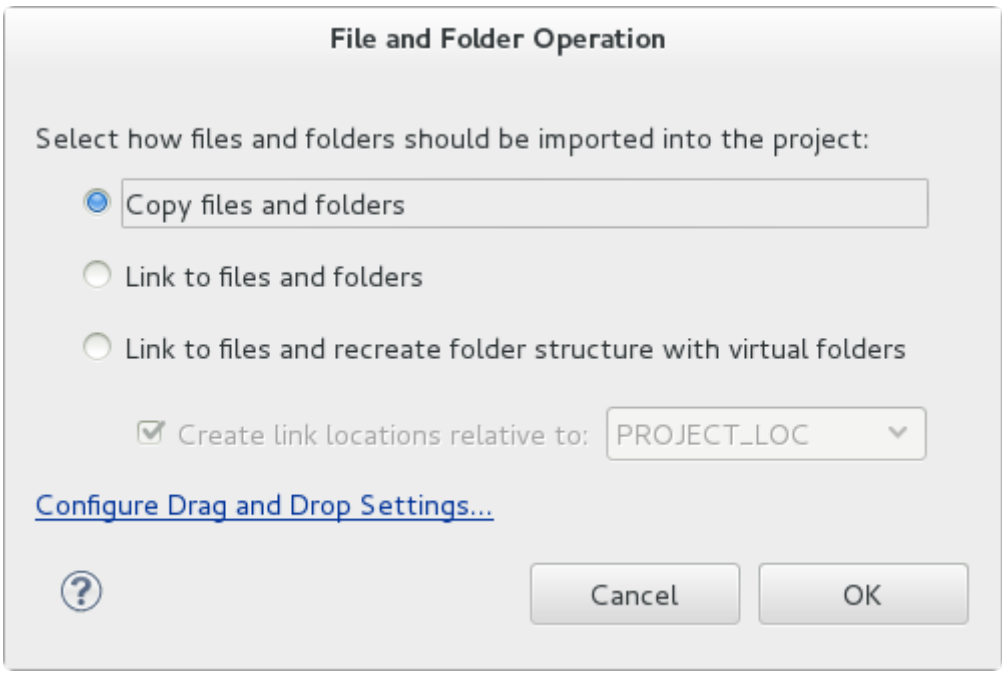


11. And click “YES” to this screen.

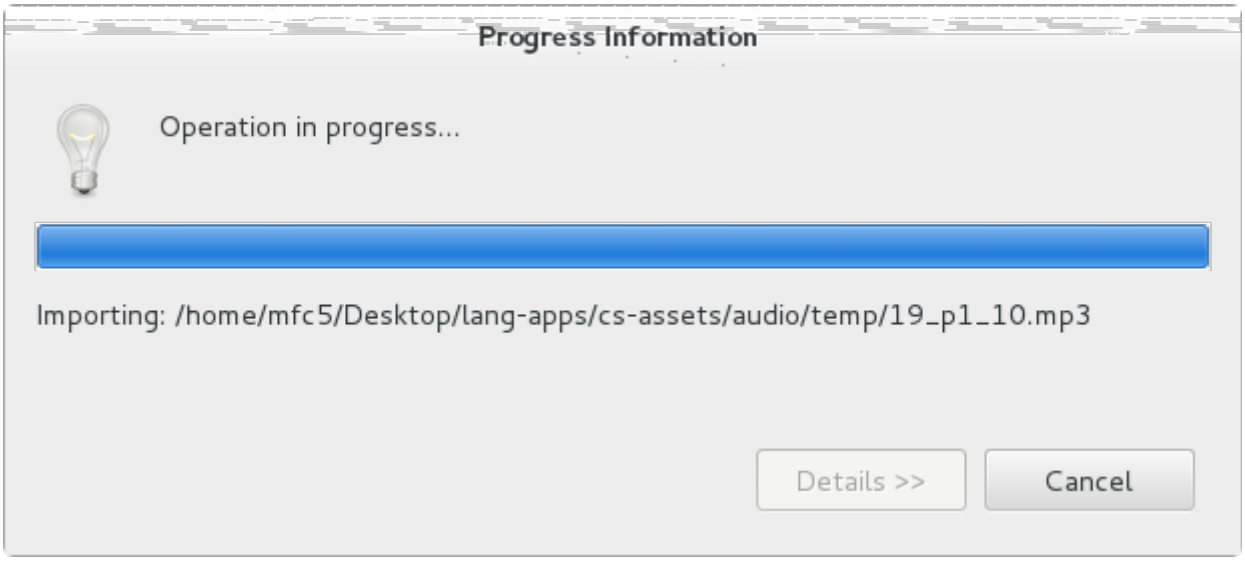
Your app is now ready for the data! It will now be called “MyLanguageApp”. To load the data, click and drag the “langDB.sqlite3”, “html”, “images” and “sounds” folders generated by the content creator into the “assets” folder under your project tree:

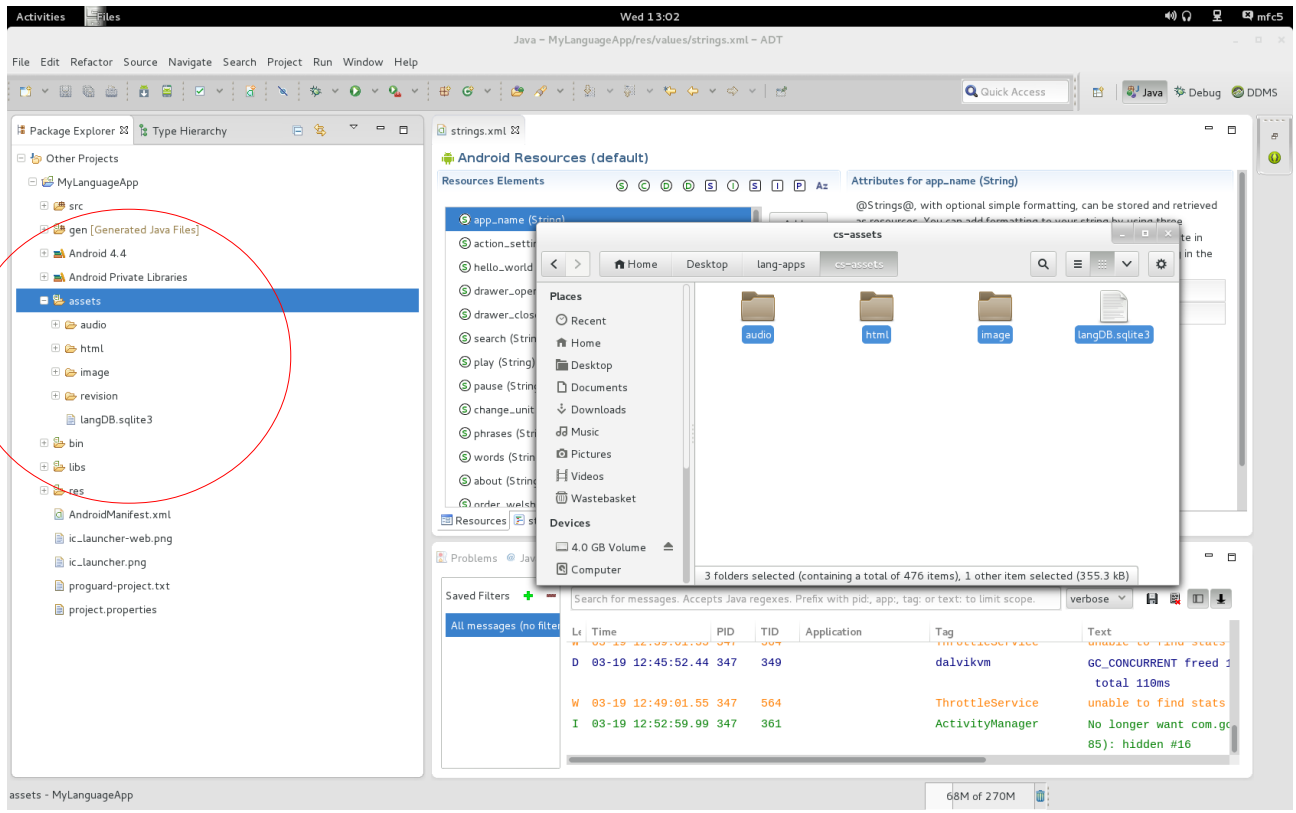


12. When asked what kind of “file and folder” operation you wish to do. Select Copy files and folders and click OK.



Whilst the files copy, you will see the progress.





When done, the files will show in the project tree.

At this point, you're ready to deploy.

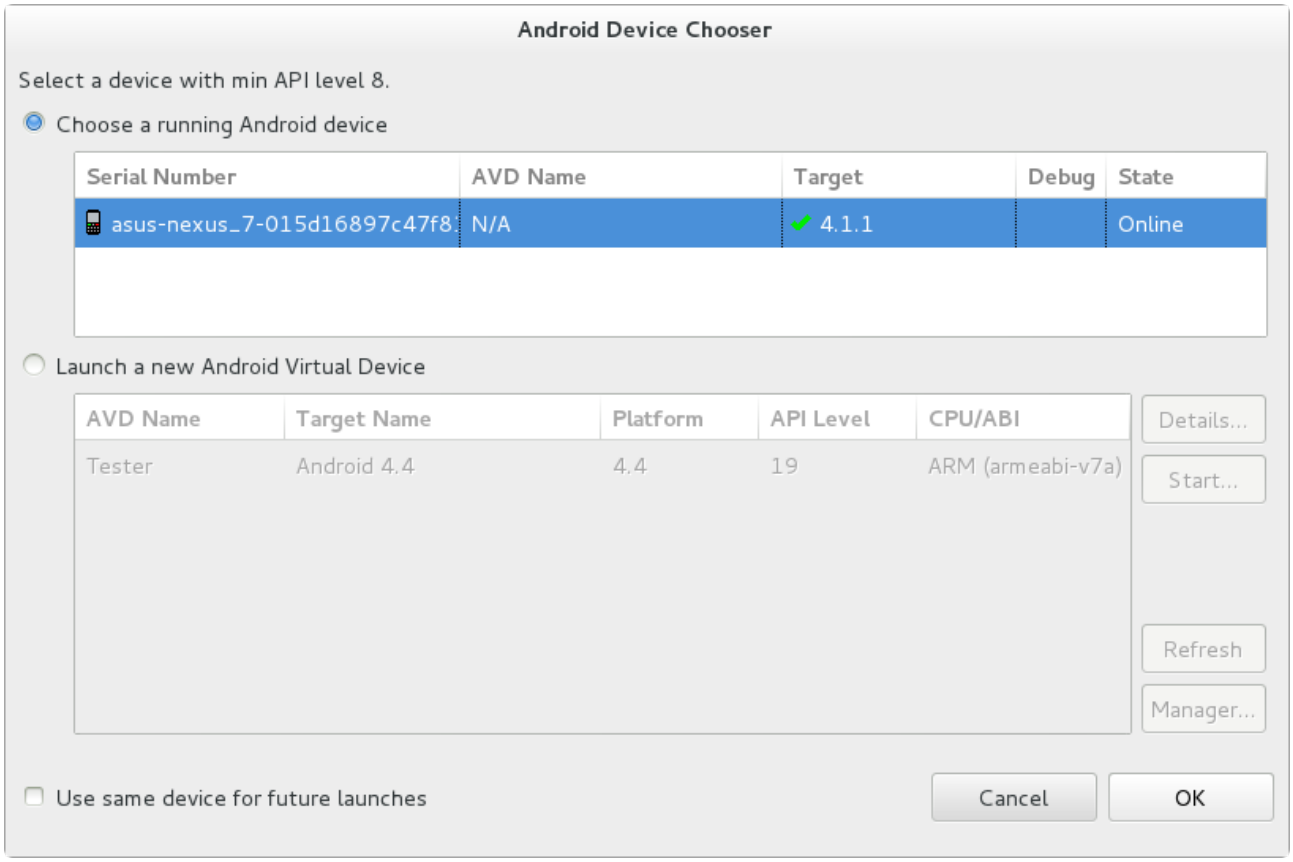
BUILDING AND DEPLOYING YOUR PROJECT

Your project is now ready to go.

To run your project:

Right click the project, select Run As, then click “1 Android Application”.

If you have a phone or other android device plugged in, select it from the list, and click OK. If you don't, use one of the emulators available.



NOTE

Depending on how many images and sounds you have, it may take a while for the app to be downloaded to the phone or device. Be patient, it will work!

When it's done, the app will automatically start. But you'll also be able to find it under the “installed apps” menu or list on your device.