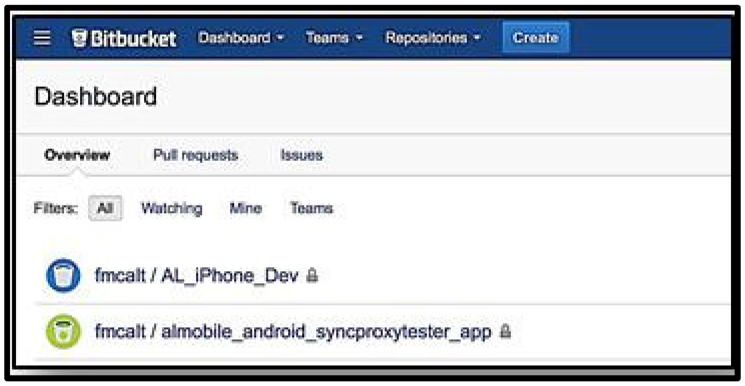
Import Procedure of Proxy into Android Studio

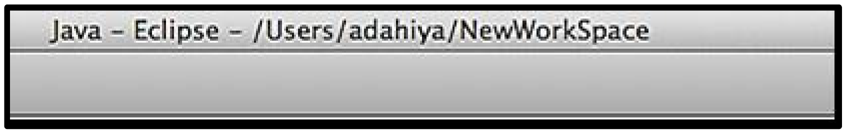


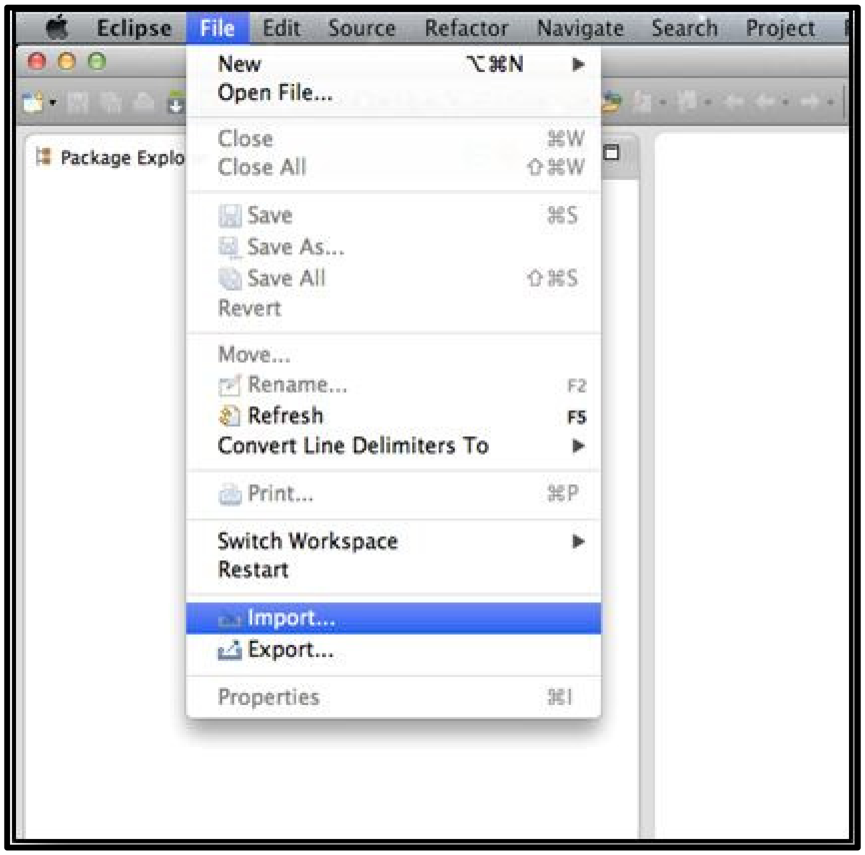
1. Clone the most recent package from BitBucket.

2. Once you have cloned your package make sure you check out the branch on which you are working.

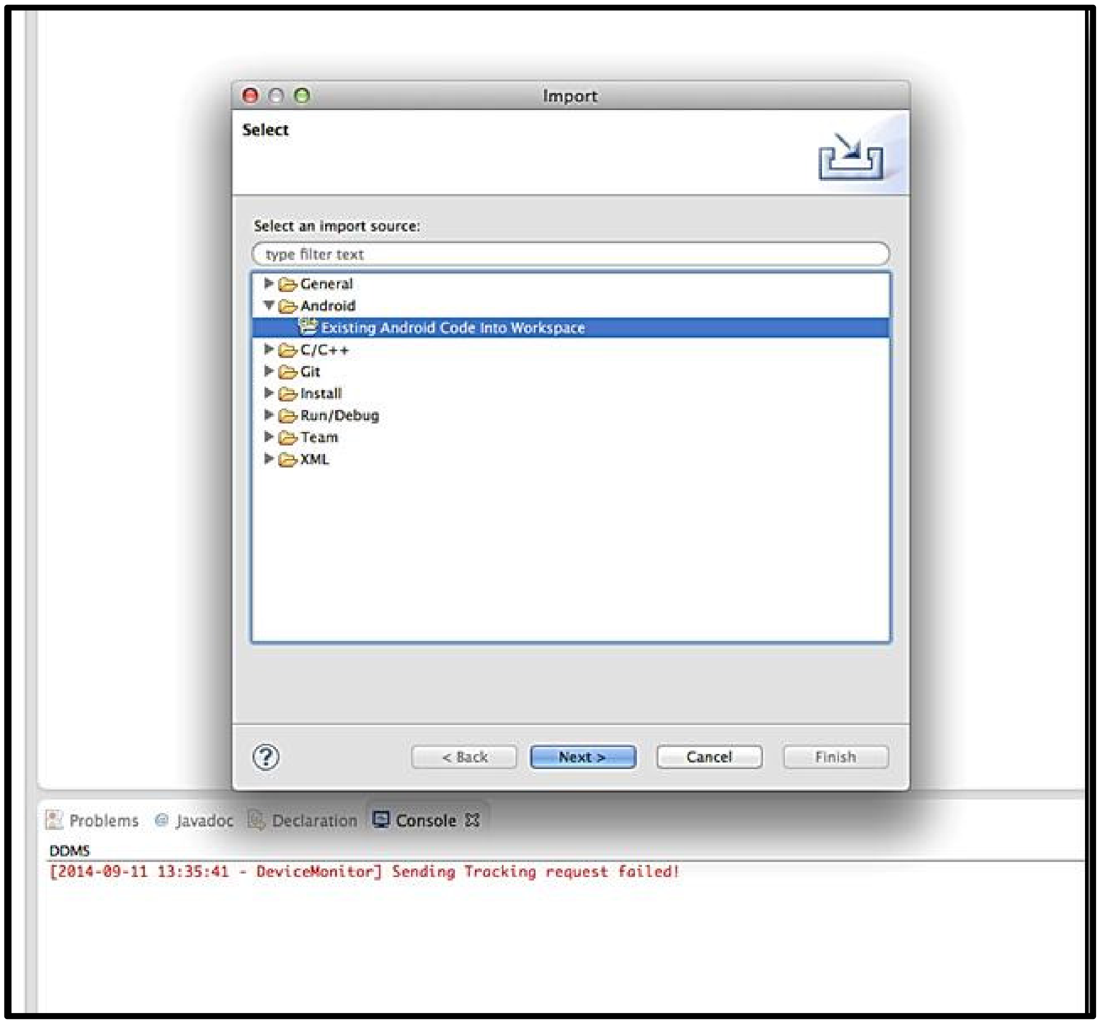
3. Next open Eclipse.

4. You will be prompted to create a new workspace. Your new work space will be displayed at the top of your screen.





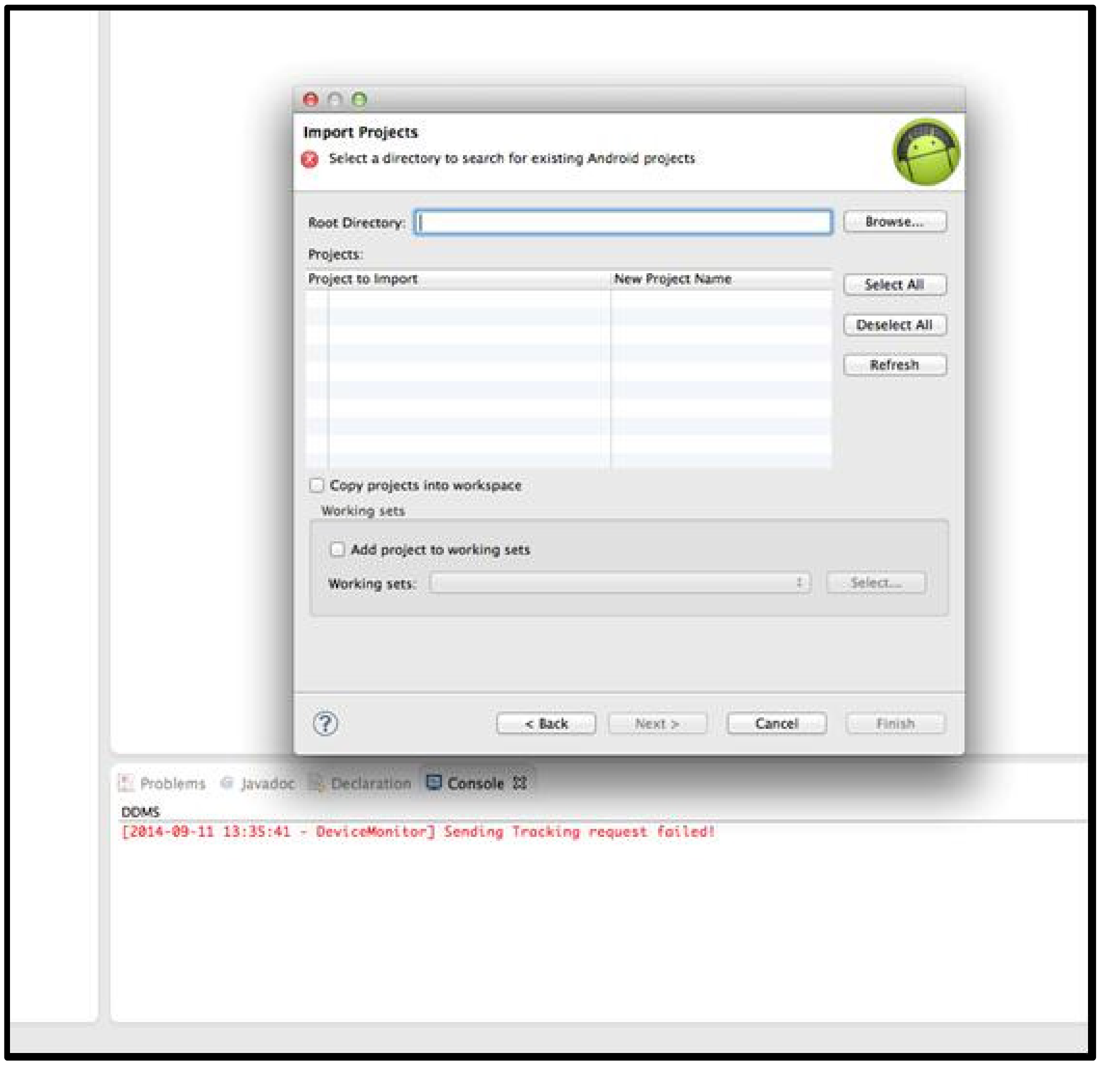
5. Under File select Import. Now you can import the new package into Eclipse.



DDMS

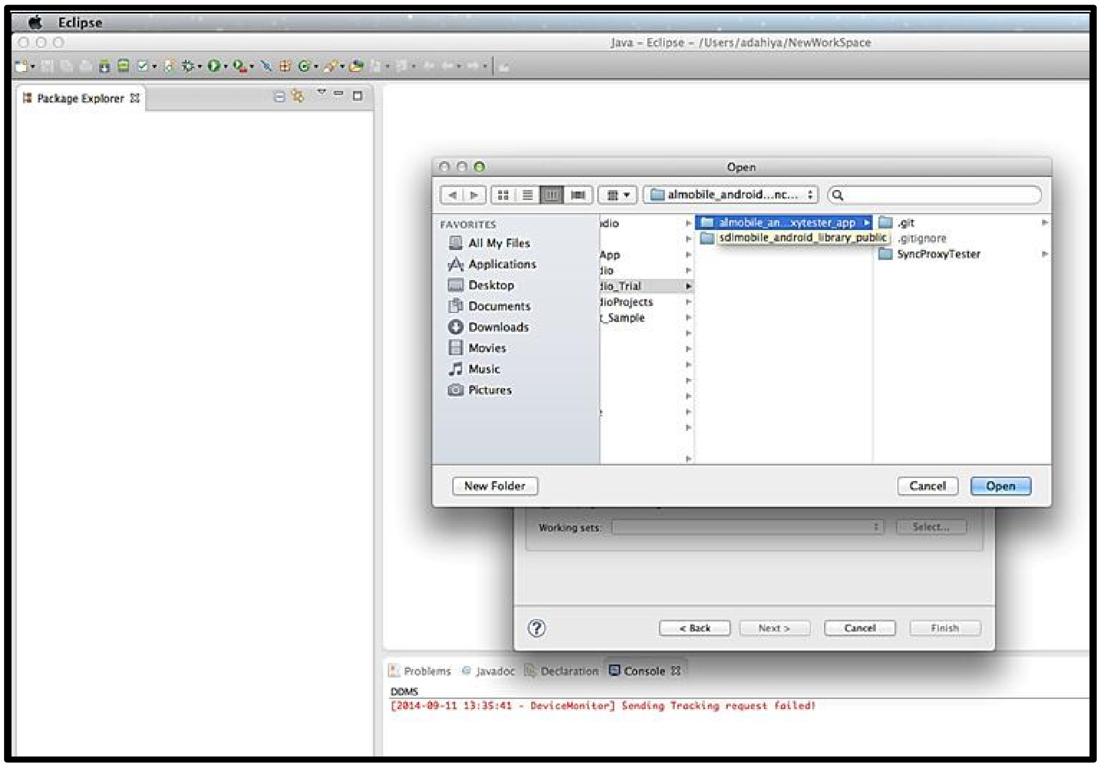
```
[2014-09-11 13:35:41 - DeviceMonitor] Sending Tracking request failed!
```

6. Once the Select menu appears choose import android code into existing workspace.
7. Click Next.

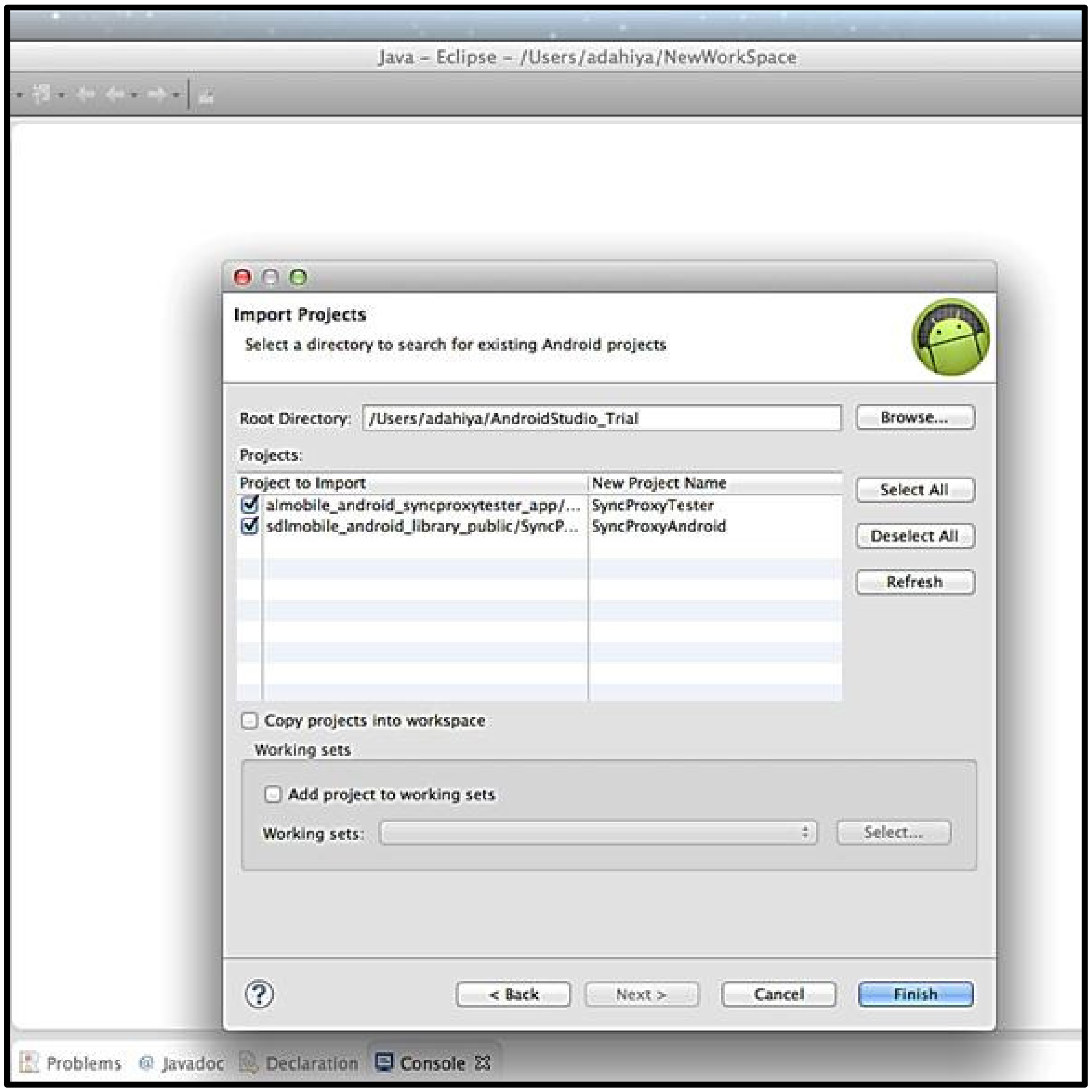


```
[2014-09-11 13:35:41 - DeviceMonitor] Sending Tracking request failed!
```

8. You will be prompted to browse for your code. Make sure to select both the library and the code.



```
[2014-09-11 13:35:41 - DeviceMonitor] Sending Tracking request failed!
```



Java – Eclipse – /Users/adahiya/NewWorkSpace

Import Projects

Select a directory to search for existing Android projects

Root Directory: /Users/adahiya/AndroidStudio_Trial

Projects:

Project to Import	New Project Name
✓ almobile_android_syncproxytester_app/...	SyncProxyTester
✓ sdlmobile_android_library_public/SyncP...	SyncProxyAndroid

☐ Copy projects into workspace

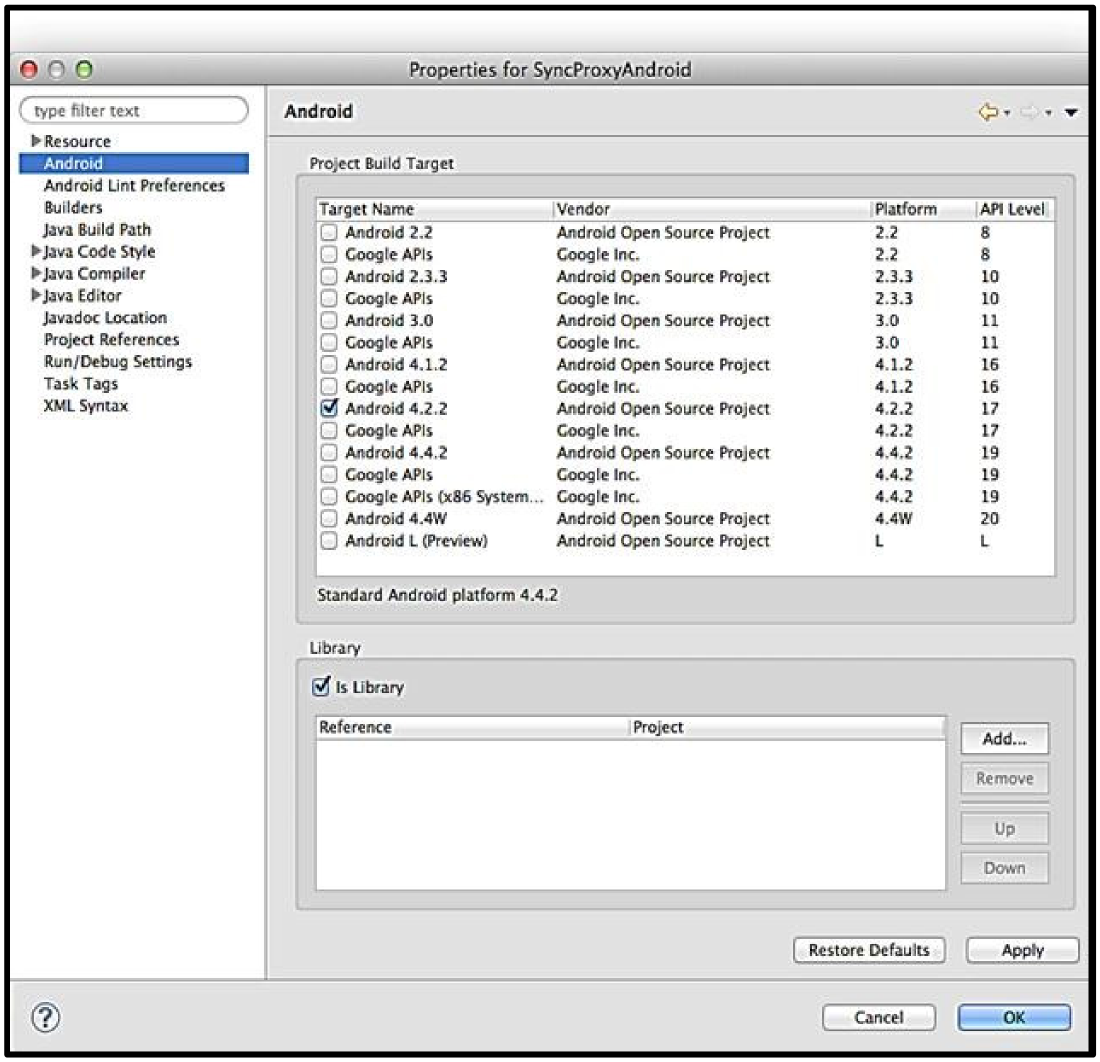
Working sets

☐ Add project to working sets

Working sets:

9. Import both the library and the application by selecting the blue check boxes under Project to Import.

10. Next click the Finish button.



Android

Project Build Target

Target Name	Vendor	Platform	API Level
☐ Android 2.2	Android Open Source Project	2.2	8
☐ Google APIs	Google Inc.	2.2	8
☐ Android 2.3.3	Android Open Source Project	2.3.3	10
☐ Google APIs	Google Inc.	2.3.3	10
☐ Android 3.0	Android Open Source Project	3.0	11
☐ Google APIs	Google Inc.	3.0	11
☐ Android 4.1.2	Android Open Source Project	4.1.2	16
☐ Google APIs	Google Inc.	4.1.2	16
☑ Android 4.2.2	Android Open Source Project	4.2.2	17
☐ Google APIs	Google Inc.	4.2.2	17
☐ Android 4.4.2	Android Open Source Project	4.4.2	19
☐ Google APIs	Google Inc.	4.4.2	19
☐ Google APIs (x86 System...	Google Inc.	4.4.2	19
☐ Android 4.4W	Android Open Source Project	4.4W	20
☐ Android L (Preview)	Android Open Source Project	L	L

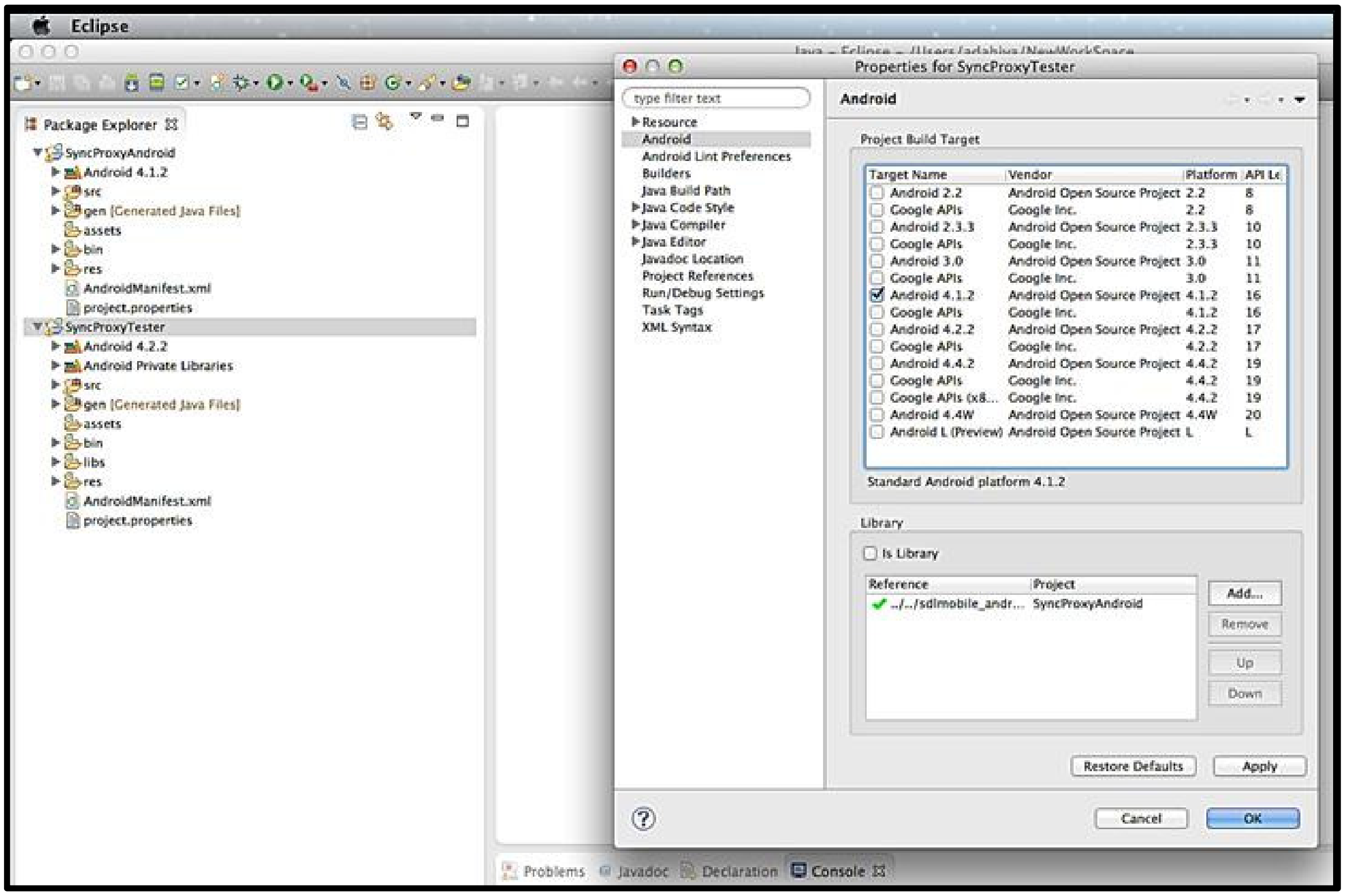
Standard Android platform 4.4.2

Library

☑ Is Library

Reference	Project

11. Make sure to change your target under properties you can also select the type of phone you are using under properties


Properties for SyncProxyTester

Android

Project Build Target

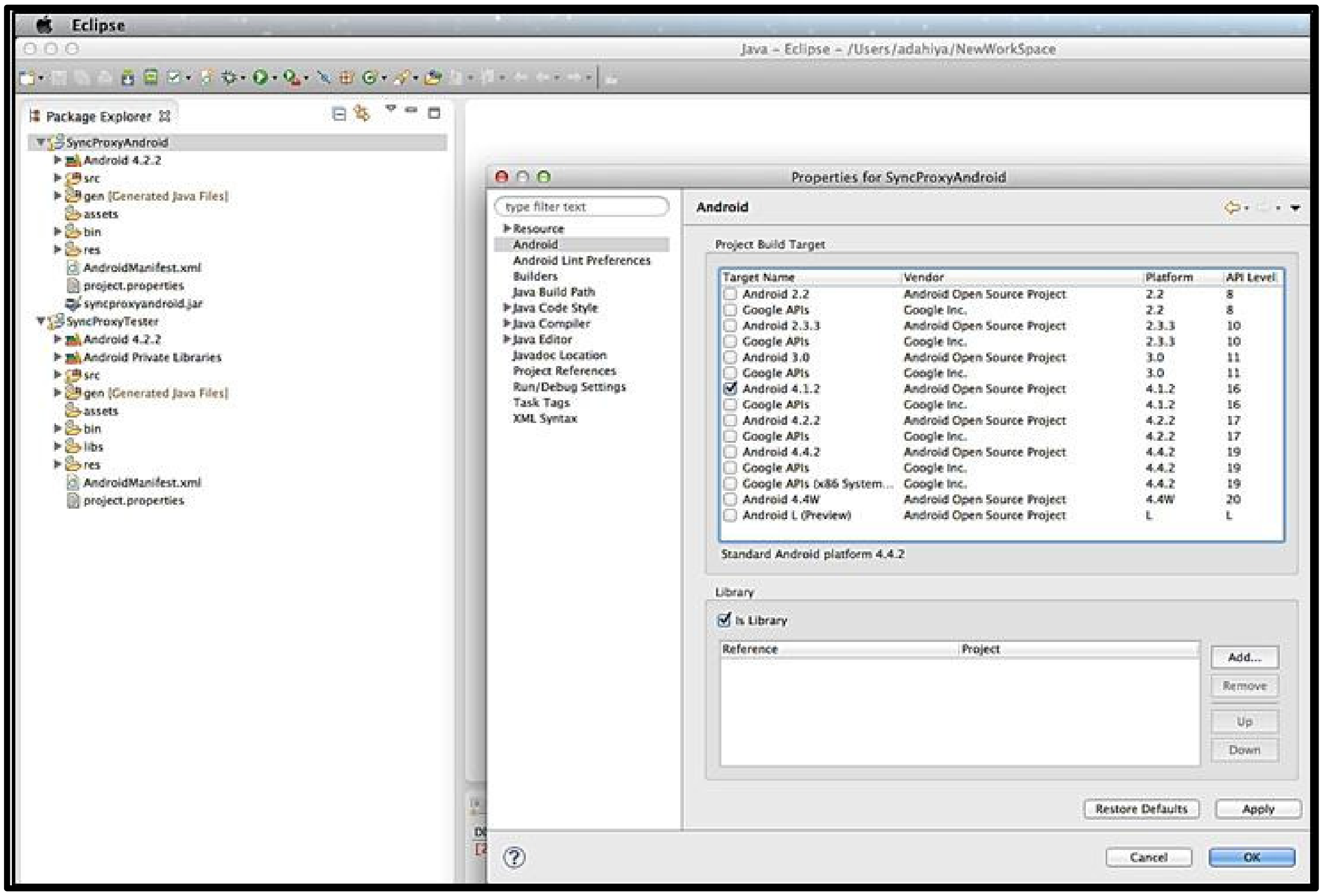
Target Name	Vendor	Platform	API Le
☐ Android 2.2	Android Open Source Project	2.2	8
☐ Google APIs	Google Inc.	2.2	8
☐ Android 2.3.3	Android Open Source Project	2.3.3	10
☐ Google APIs	Google Inc.	2.3.3	10
☐ Android 3.0	Android Open Source Project	3.0	11
☐ Google APIs	Google Inc.	3.0	11
☑ Android 4.1.2	Android Open Source Project	4.1.2	16
☐ Google APIs	Google Inc.	4.1.2	16
☐ Android 4.2.2	Android Open Source Project	4.2.2	17
☐ Google APIs	Google Inc.	4.2.2	17
☐ Android 4.4.2	Android Open Source Project	4.4.2	19
☐ Google APIs	Google Inc.	4.4.2	19
☐ Google APIs (x8...	Google Inc.	4.4.2	19
☐ Android 4.4W	Android Open Source Project	4.4W	20
☐ Android L (Preview)	Android Open Source Project	L	L

Standard Android platform 4.1.2

Library

☐ Is Library

Reference	Project
✓ ../../sdlmobile_andr...	SyncProxyAndroid



Android

Project Build Target

	Target Name	Vendor	Platform	API Level
☐	Android 2.2	Android Open Source Project	2.2	8
☐	Google APIs	Google Inc.	2.2	8
☐	Android 2.3.3	Android Open Source Project	2.3.3	10
☐	Google APIs	Google Inc.	2.3.3	10
☐	Android 3.0	Android Open Source Project	3.0	11
☐	Google APIs	Google Inc.	3.0	11
✓	Android 4.1.2	Android Open Source Project	4.1.2	16
☐	Google APIs	Google Inc.	4.1.2	16
☐	Android 4.2.2	Android Open Source Project	4.2.2	17
☐	Google APIs	Google Inc.	4.2.2	17
☐	Android 4.4.2	Android Open Source Project	4.4.2	19
☐	Google APIs	Google Inc.	4.4.2	19
☐	Google APIs (x86 System...	Google Inc.	4.4.2	19
☐	Android 4.4W	Android Open Source Project	4.4W	20
☐	Android L (Preview)	Android Open Source Project	L	L

Standard Android platform 4.4.2

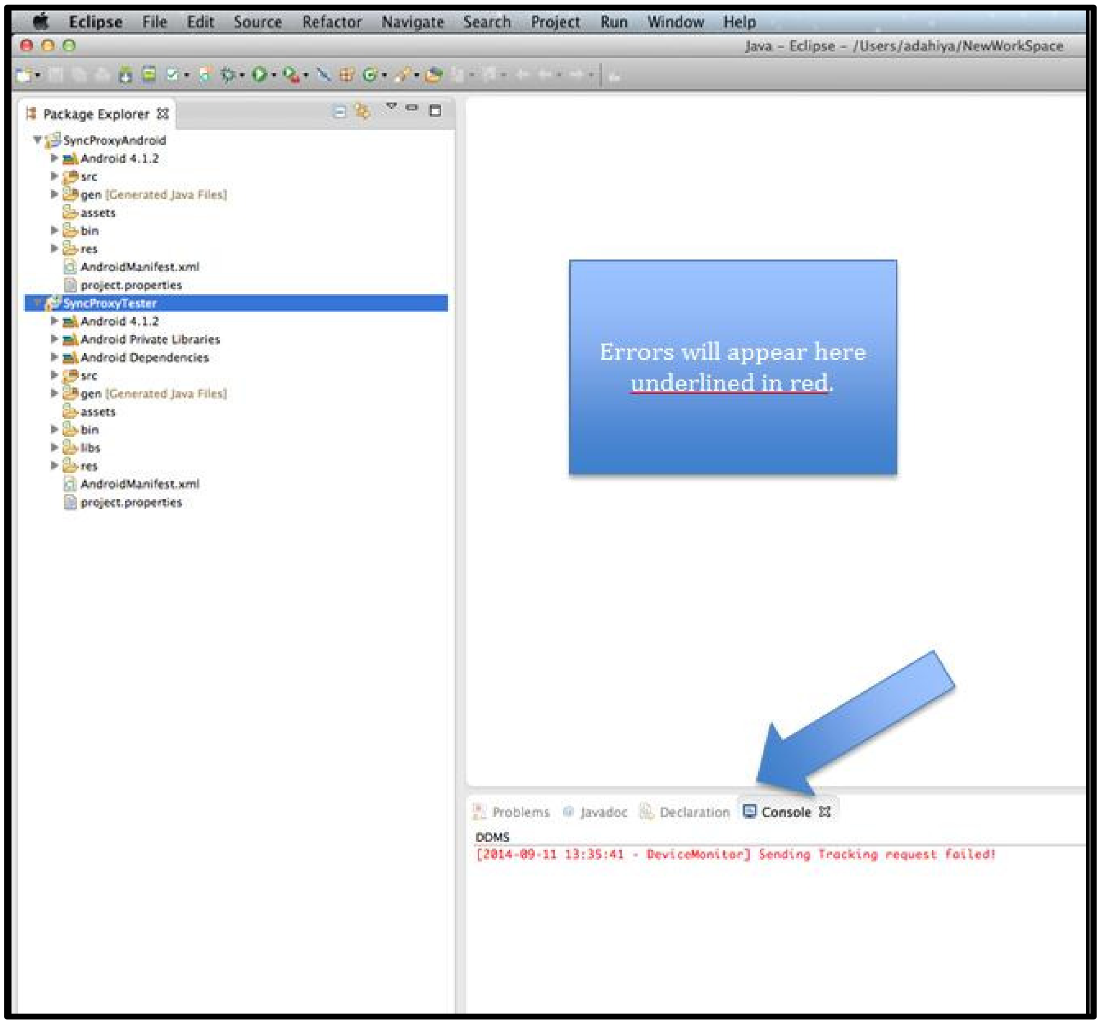
Library

✓ Is Library

Reference	Project

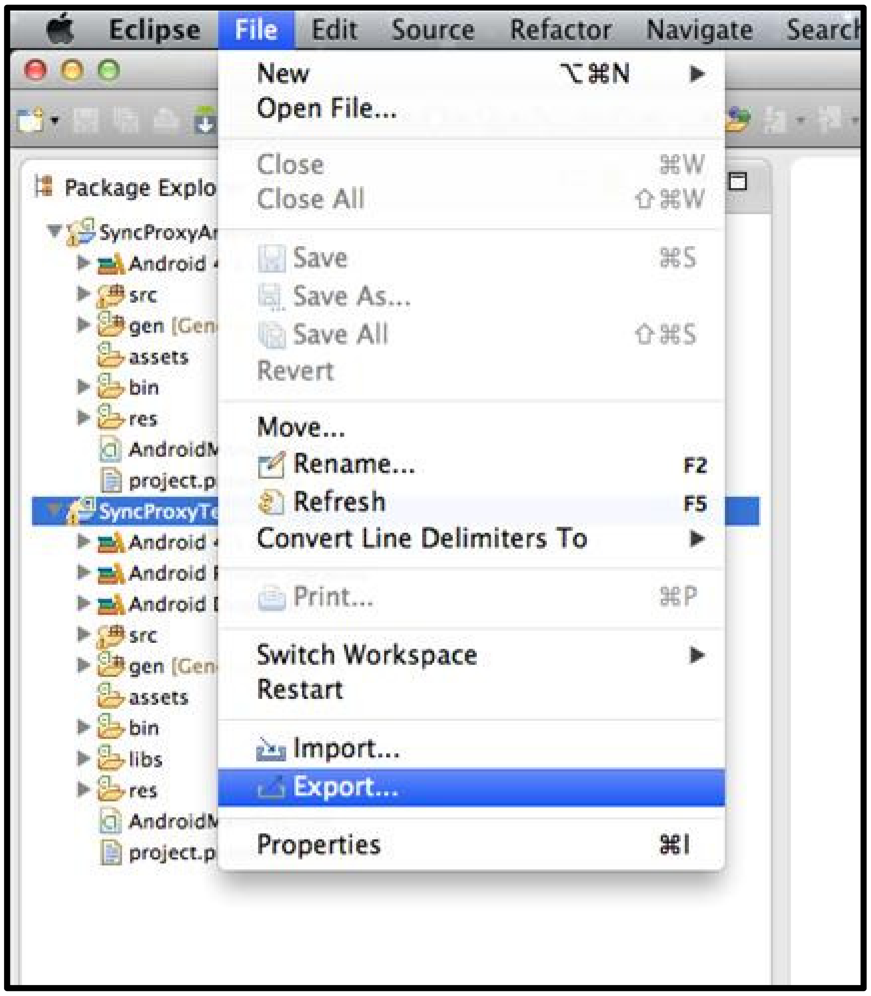
12. You need to use the same target for your library as well

13. Make sure your library project is included in your application.
14. (Under package explore both will be the same)


Eclipse File Edit Source Refactor Navigate Search Project Run Window Help
Java – Eclipse – /Users/adahiya/NewWorkSpace
Package Explorer
SyncProxyAndroid
Android 4.1.2
src
gen [Generated Java Files]
assets
bin
res
AndroidManifest.xml
project.properties
SyncProxyTester
Android 4.1.2
Android Private Libraries
Android Dependencies
src
gen [Generated Java Files]
assets
bin
libs
res
AndroidManifest.xml
project.properties
Errors will appear here underlined in red.
Problems Javadoc Declaration Console
DDMS

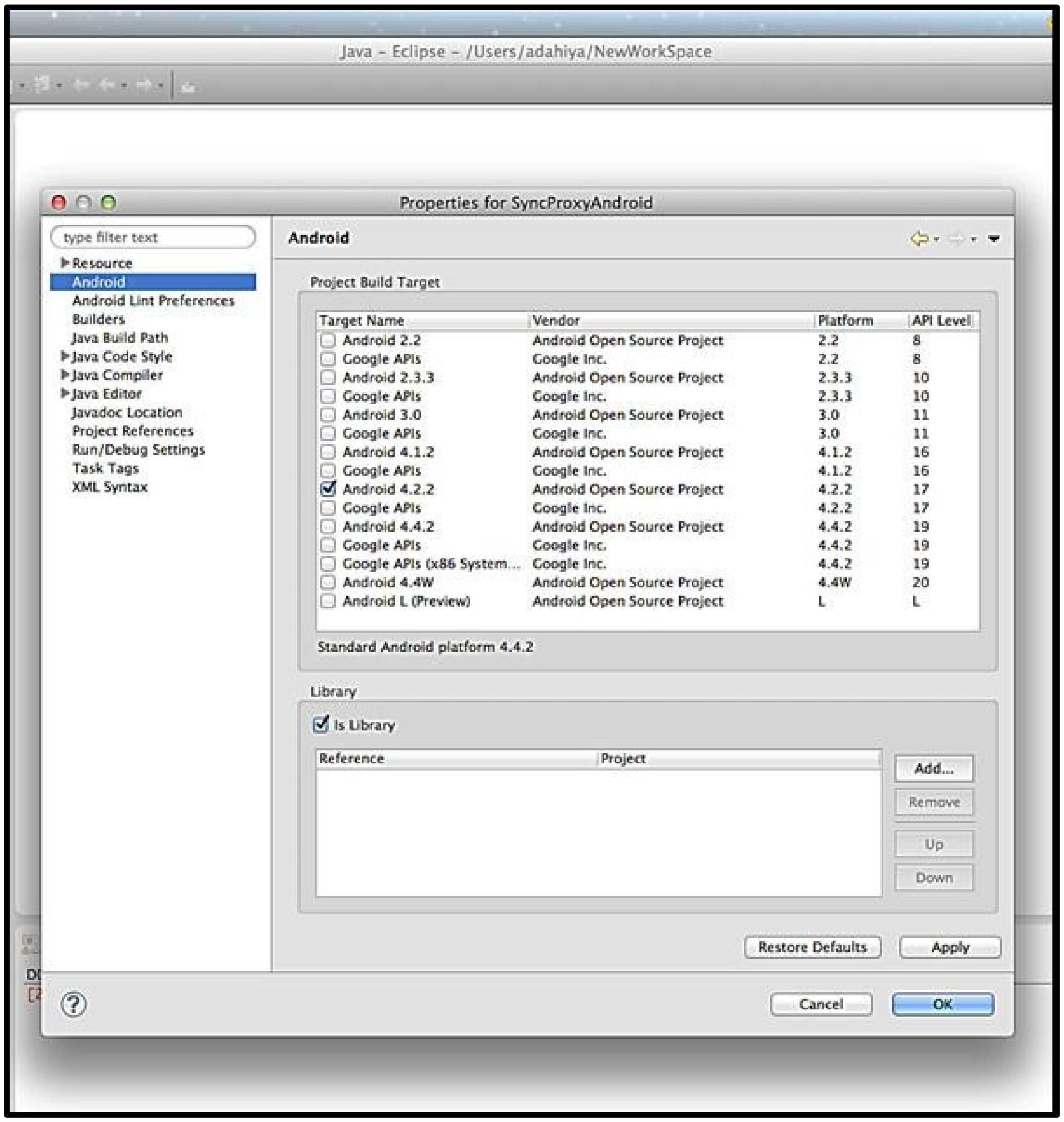
```
[2014-09-11 13:35:41 - DeviceMonitor] Sending Tracking request failed!
```

15. Make sure there are not any errors. Problems also appear in the console in red text.



16. Next go to file

17. Then go to export


Java – Eclipse – /Users/adahiya/NewWorkSpace

Properties for SyncProxyAndroid

Android

Project Build Target

Target Name	Vendor	Platform	API Level
☐ Android 2.2	Android Open Source Project	2.2	8
☐ Google APIs	Google Inc.	2.2	8
☐ Android 2.3.3	Android Open Source Project	2.3.3	10
☐ Google APIs	Google Inc.	2.3.3	10
☐ Android 3.0	Android Open Source Project	3.0	11
☐ Google APIs	Google Inc.	3.0	11
☐ Android 4.1.2	Android Open Source Project	4.1.2	16
☐ Google APIs	Google Inc.	4.1.2	16
✓ Android 4.2.2	Android Open Source Project	4.2.2	17
☐ Google APIs	Google Inc.	4.2.2	17
☐ Android 4.4.2	Android Open Source Project	4.4.2	19
☐ Google APIs	Google Inc.	4.4.2	19
☐ Google APIs (x86 System...	Google Inc.	4.4.2	19
☐ Android 4.4W	Android Open Source Project	4.4W	20
☐ Android L (Preview)	Android Open Source Project	L	L

Standard Android platform 4.4.2

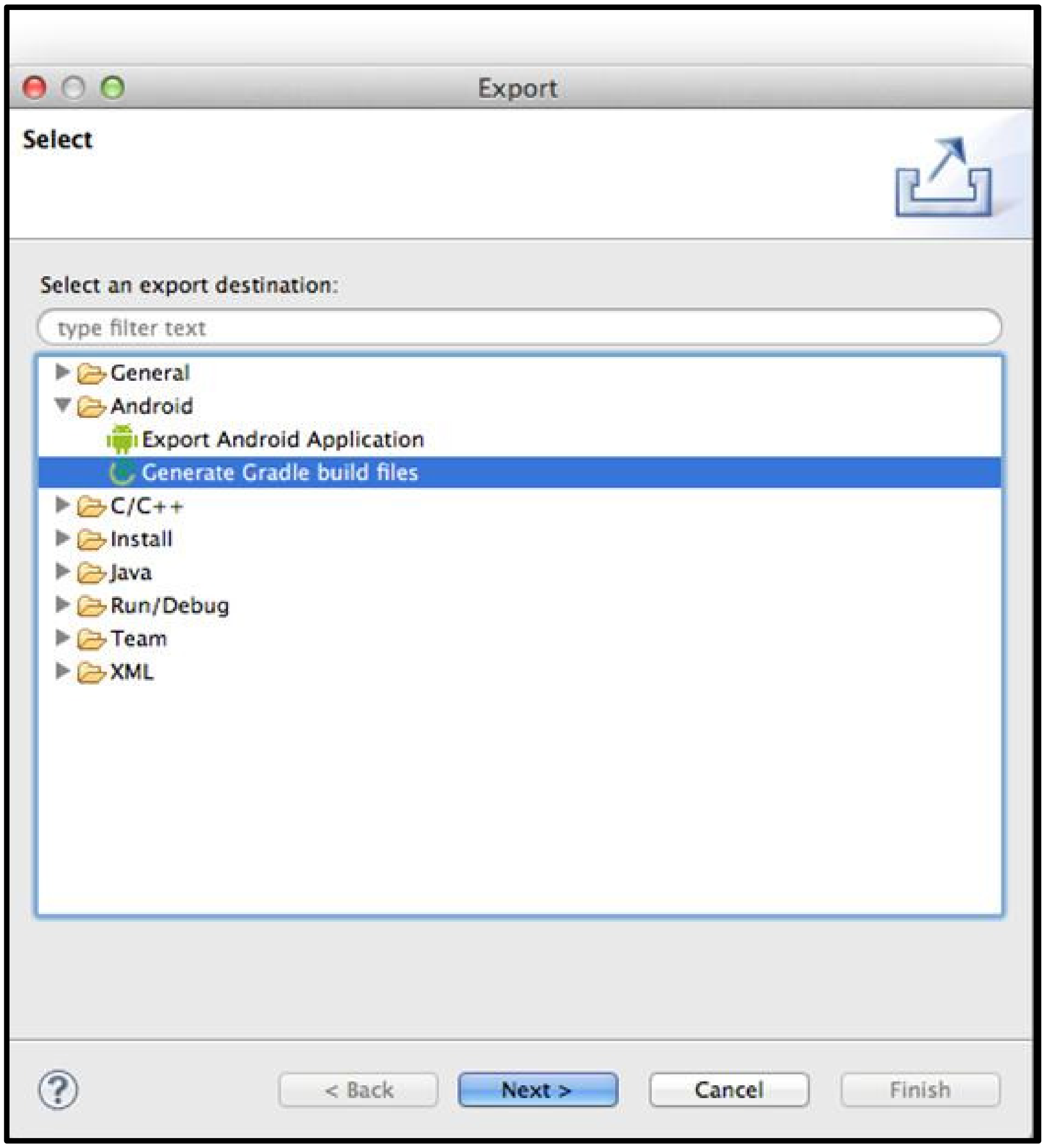
Library

✓ Is Library

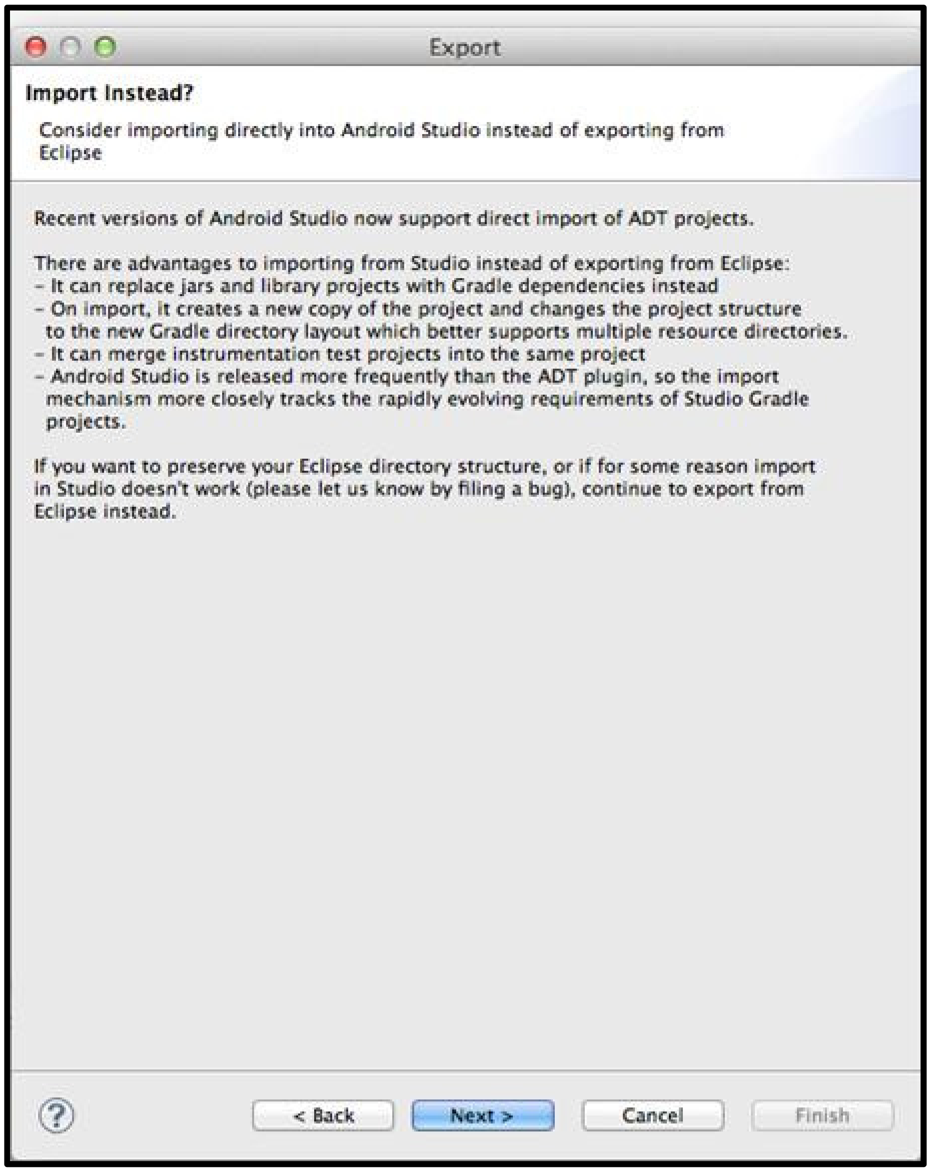
Reference	Project

18. Then select Android from Resource.

19. Make sure the check box is selected in the Project Build Target window.



20. From the Select menu choose Generate Gradle build files then click next.



Import Instead?

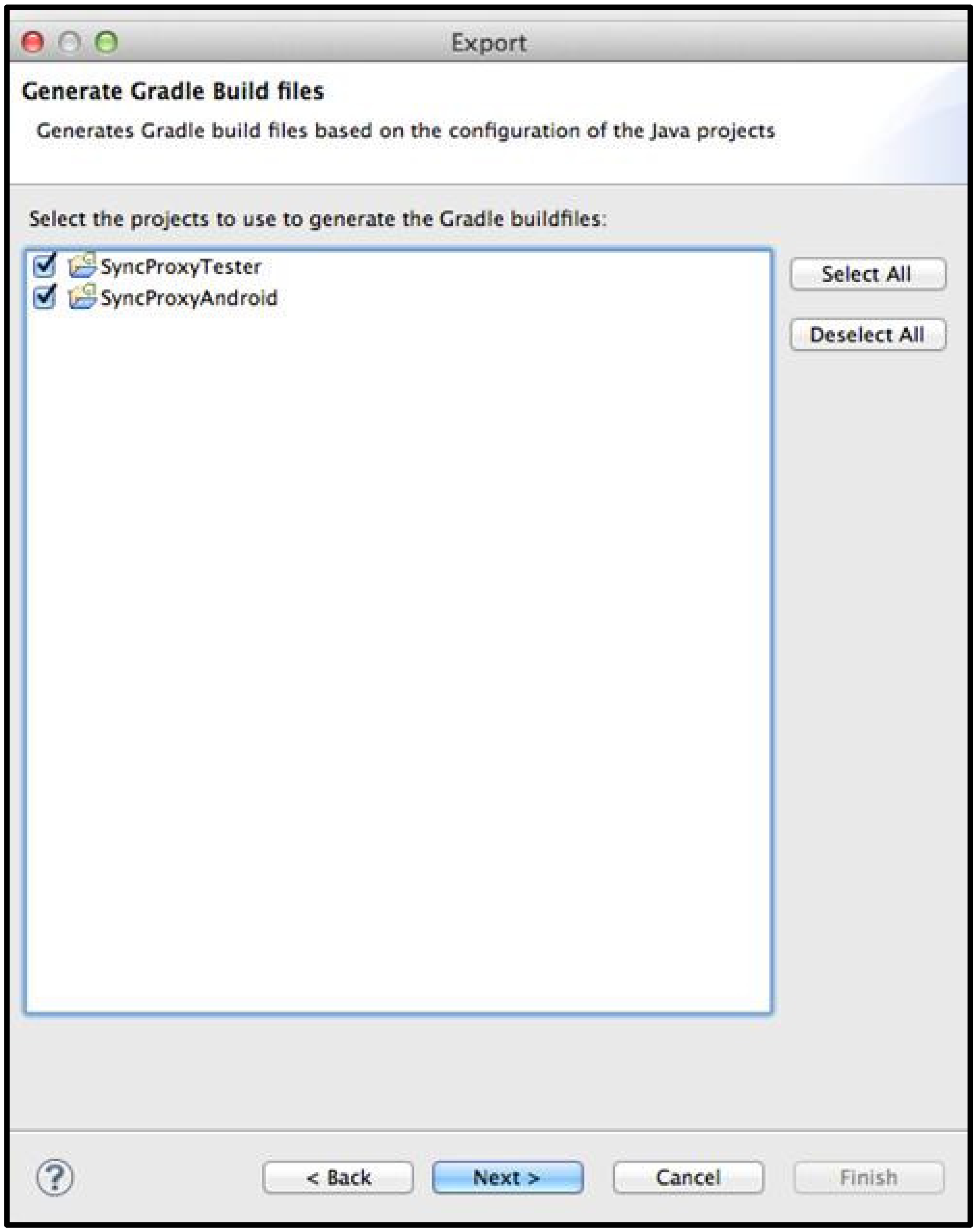
Consider importing directly into Android Studio instead of exporting from Eclipse

Recent versions of Android Studio now support direct import of ADT projects.

There are advantages to importing from Studio instead of exporting from Eclipse:
- It can replace jars and library projects with Gradle dependencies instead
- On import, it creates a new copy of the project and changes the project structure to the new Gradle directory layout which better supports multiple resource directories.
- It can merge instrumentation test projects into the same project
- Android Studio is released more frequently than the ADT plugin, so the import mechanism more closely tracks the rapidly evolving requirements of Studio Gradle projects.

If you want to preserve your Eclipse directory structure, or if for some reason import in Studio doesn't work (please let us know by filing a bug), continue to export from Eclipse instead.

21. You will be given the option to import instead. Disregard this message and Click Next.



Export

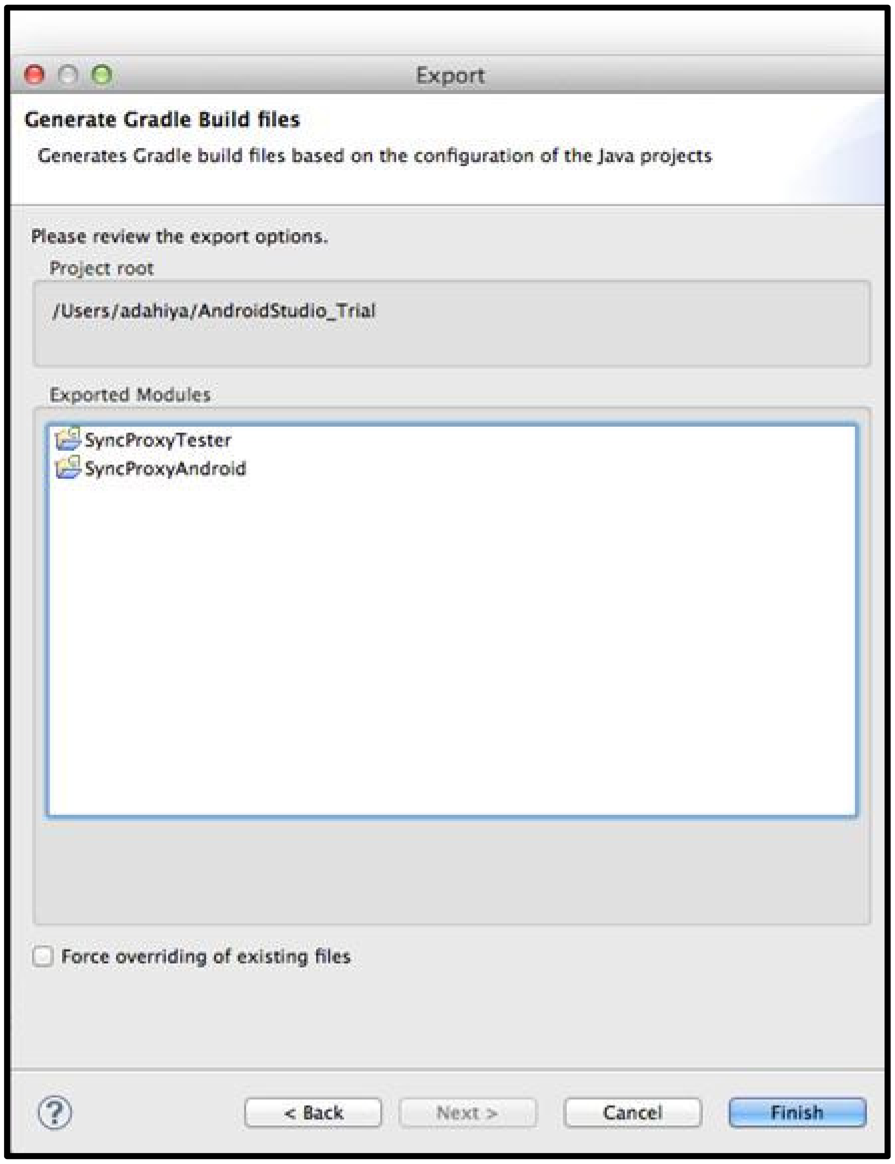
Generate Gradle Build files

Generates Gradle build files based on the configuration of the Java projects

Select the projects to use to generate the Gradle buildfiles:

- [✓] SyncProxyTester
- [✓] SyncProxyAndroid

22. You will have the following two options: Click Next


Generate Gradle Build files

Generates Gradle build files based on the configuration of the Java projects

Please review the export options.

Project root

/Users/adahiya/AndroidStudio_Trial

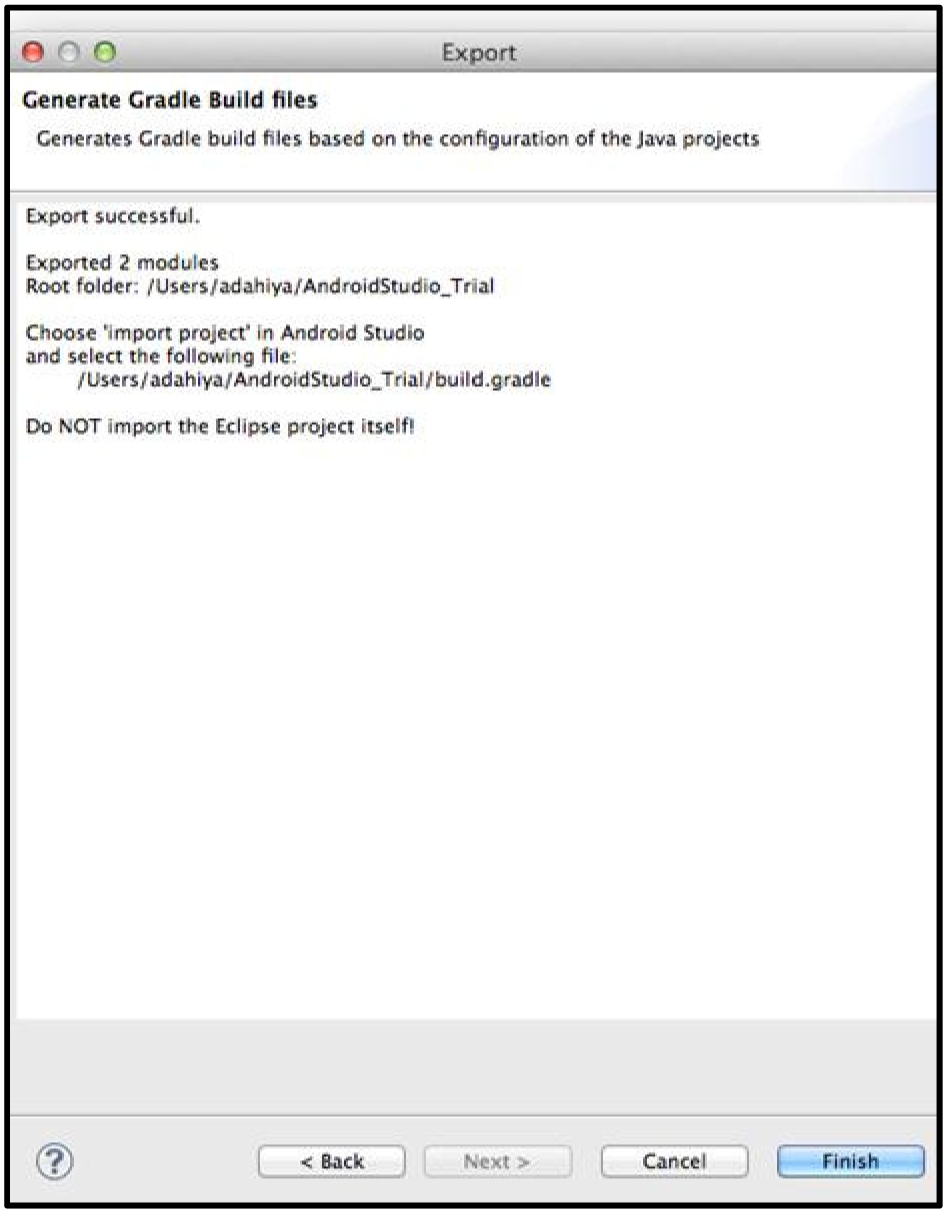
Exported Modules

SyncProxyTester
SyncProxyAndroid

Force overriding of existing files

23. Make sure you select both the application and the library

24. Click Finish

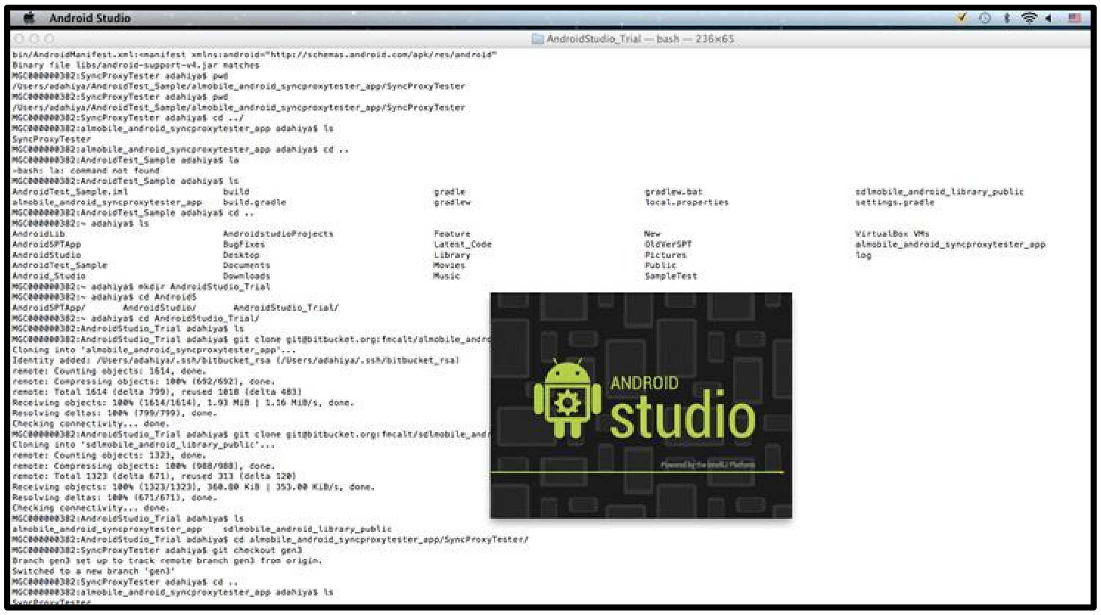


Generate Gradle Build files

Generates Gradle build files based on the configuration of the Java projects

```
Export successful.

Exported 2 modules
Root folder: /Users/adahiya/AndroidStudio_Trial

Choose 'import project' in Android Studio
and select the following file:
      /Users/adahiya/AndroidStudio_Trial/build.gradle

Do NOT import the Eclipse project itself!
```

25. Once you are finished you will receive a message that reads export successful. You can now exit eclipse

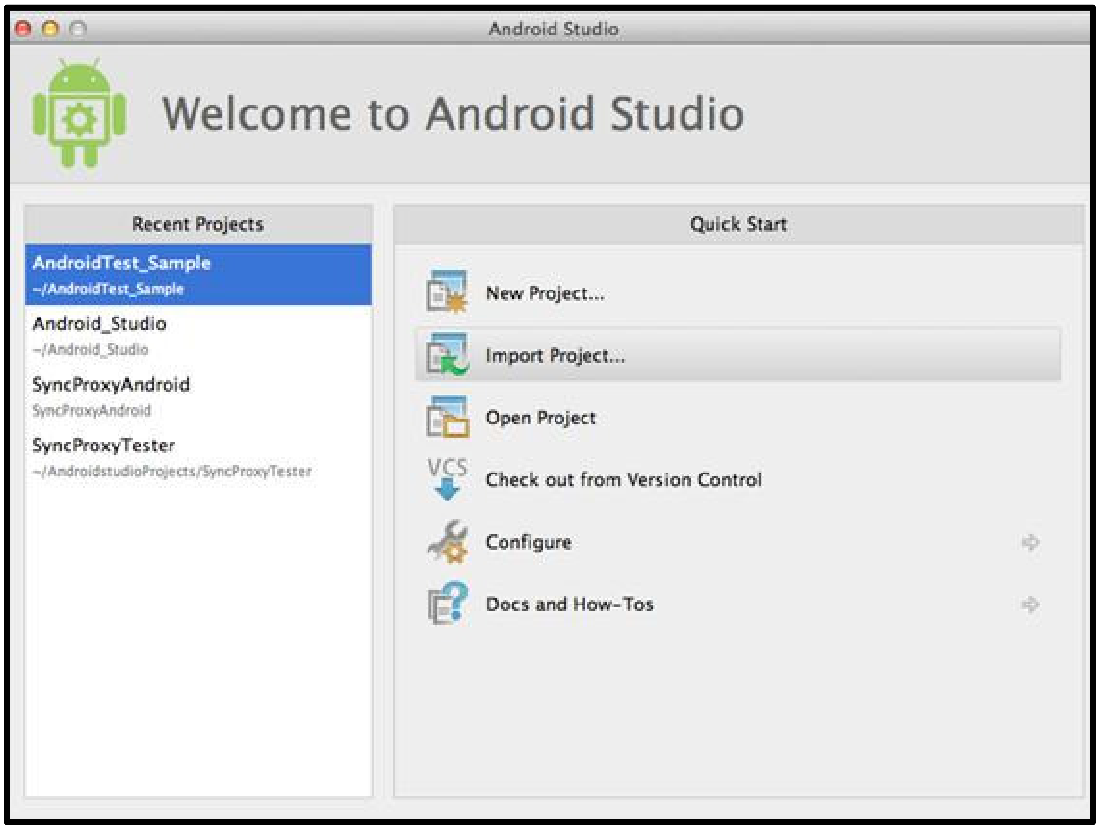


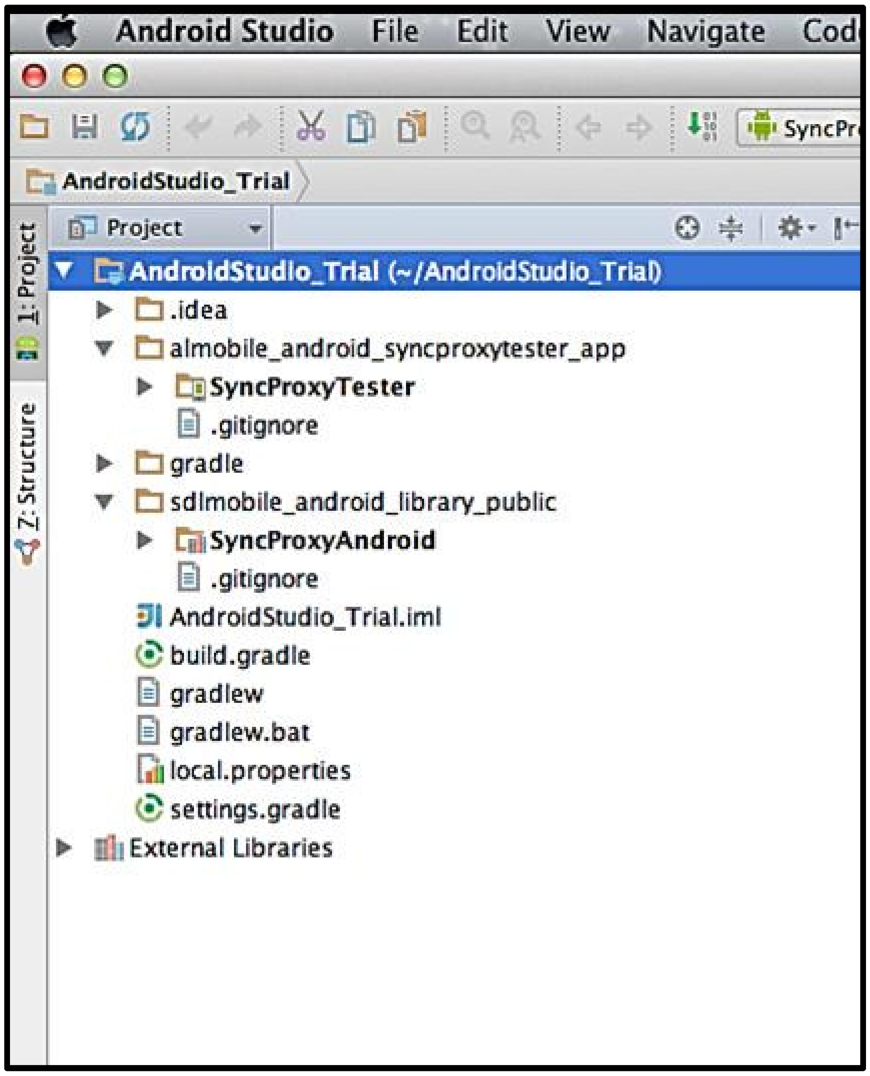
```
bin/AndroidManifest.xml:<manifest xmlns:android="http://schemas.android.com/apk/res/android"
Binary file libs/android-support-v4.jar matches
MGC000000382:SyncProxyTester adahiya$ pwd
/Users/adahiya/AndroidTest_Sample/almobile_android_syncproxytester_app/SyncProxyTester
MGC000000382:SyncProxyTester adahiya$ pwd
/Users/adahiya/AndroidTest_Sample/almobile_android_syncproxytester_app/SyncProxyTester
MGC000000382:SyncProxyTester adahiya$ cd ../
MGC000000382:almobile_android_syncproxytester_app adahiya$ ls
SyncProxyTester
MGC000000382:almobile_android_syncproxytester_app adahiya$ cd ..
MGC000000382:AndroidTest_Sample adahiya$ la
-bash: la: command not found
MGC000000382:AndroidTest_Sample adahiya$ ls
AndroidTest_Sample.iml                build                                 gradle                                gradlew.bat                           sdlmobile_android_library_public
almobile_android_syncproxytester_app  build.gradle                          gradlew                               local.properties                      settings.gradle
MGC000000382:AndroidTest_Sample adahiya$ cd ..
MGC000000382:~ adahiya$ ls
AndroidLib                            AndroidstudioProjects                 Feature                               New                                   VirtualBox VMs
AndroidSPTApp                         BugFixes                              Latest_Code                           OldVerSPT                             almobile_android_syncproxytester_app
AndroidStudio                         Desktop                               Library                               Pictures                              log
AndroidTest_Sample                    Documents                             Movies                                Public
Android_Studio                        Downloads                             Music                                 SampleTest
MGC000000382:~ adahiya$ mkdir AndroidStudio_Trial
MGC000000382:~ adahiya$ cd AndroidS
AndroidSPTApp/       AndroidStudio/       AndroidStudio_Trial/
MGC000000382:~ adahiya$ cd AndroidStudio_Trial/
MGC000000382:AndroidStudio_Trial adahiya$ ls
MGC000000382:AndroidStudio_Trial adahiya$ git clone git@bitbucket.org:fmcalt/almobile_andro
Cloning into 'almobile_android_syncproxytester_app'...
Identity added: /Users/adahiya/.ssh/bitbucket_rsa (/Users/adahiya/.ssh/bitbucket_rsa)
remote: Counting objects: 1614, done.
remote: Compressing objects: 100% (692/692), done.
remote: Total 1614 (delta 799), reused 1018 (delta 483)
Receiving objects: 100% (1614/1614), 1.93 MiB | 1.16 MiB/s, done.
Resolving deltas: 100% (799/799), done.
Checking connectivity... done.
MGC000000382:AndroidStudio_Trial adahiya$ git clone git@bitbucket.org:fmcalt/sdlmobile_andr
Cloning into 'sdlmobile_android_library_public'...
remote: Counting objects: 1323, done.
remote: Compressing objects: 100% (988/988), done.
remote: Total 1323 (delta 671), reused 313 (delta 120)
Receiving objects: 100% (1323/1323), 360.00 KiB | 353.00 KiB/s, done.
Resolving deltas: 100% (671/671), done.
Checking connectivity... done.
MGC000000382:AndroidStudio_Trial adahiya$ ls
almobile_android_syncproxytester_app    sdlmobile_android_library_public
MGC000000382:AndroidStudio_Trial adahiya$ cd almobile_android_syncproxytester_app/SyncProxyTester/
MGC000000382:SyncProxyTester adahiya$ git checkout gen3
Branch gen3 set up to track remote branch gen3 from origin.
Switched to a new branch 'gen3'
MGC000000382:SyncProxyTester adahiya$ cd ..
MGC000000382:almobile_android_syncproxytester_app adahiya$ ls
SyncProxyTester
```

26. You will need to install android studio on your computer.

27. Next open Android studio

28. Select Import Project from the Quick Start menu.





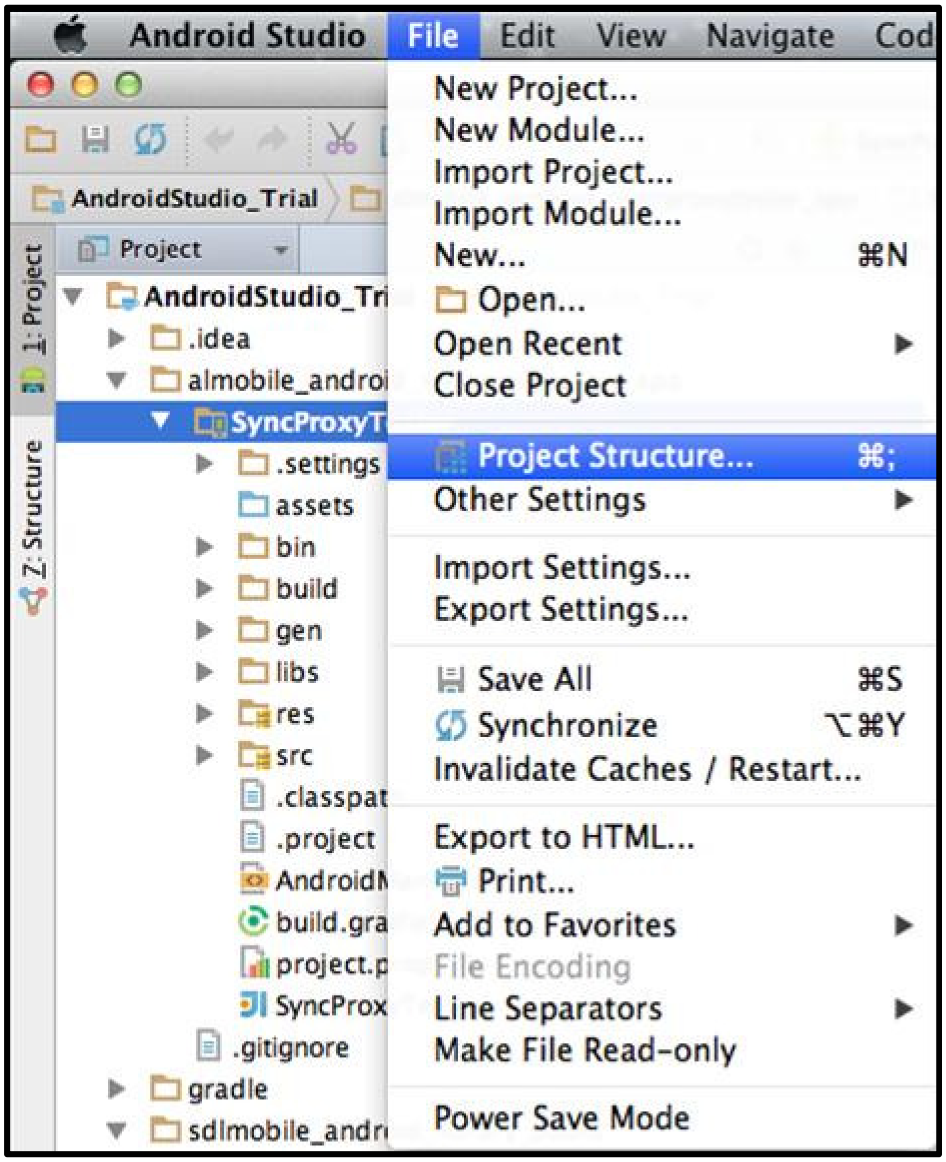
29. Go to the directory where you saved your project

30. Select your project and press OK

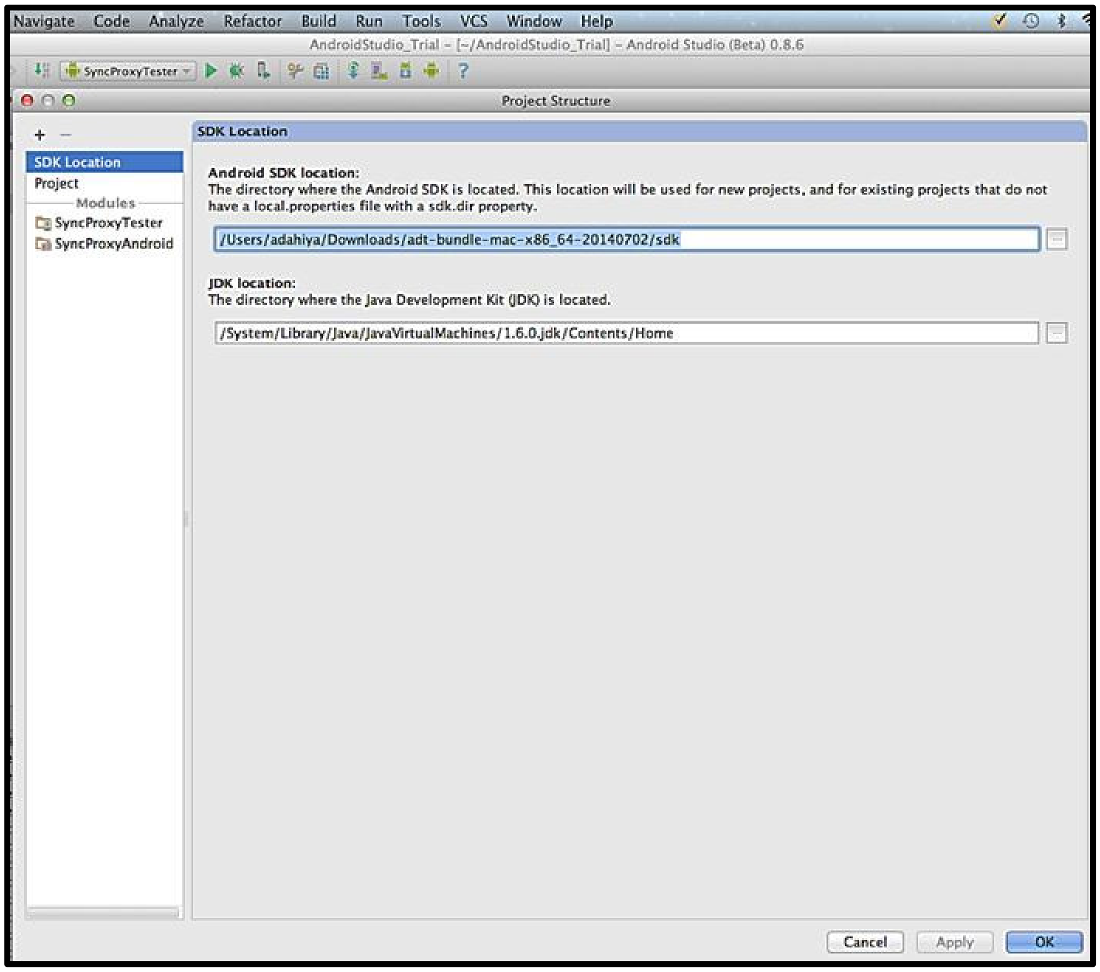
31. The project is now loading into Android Studio

32. You can expand the project window to see your application and library

33. Both folders will be visible and each folder will have its own build.gradle folder



34. Under File select Project Structure


Navigate Code Analyze Refactor Build Run Tools VCS Window Help
AndroidStudio_Trial - [~/AndroidStudio_Trial] - Android Studio (Beta) 0.8.6
SyncProxyTester
Project Structure
SDK Location
Project
Modules
SyncProxyTester
SyncProxyAndroid
Cancel
Apply
OK

SDK Location

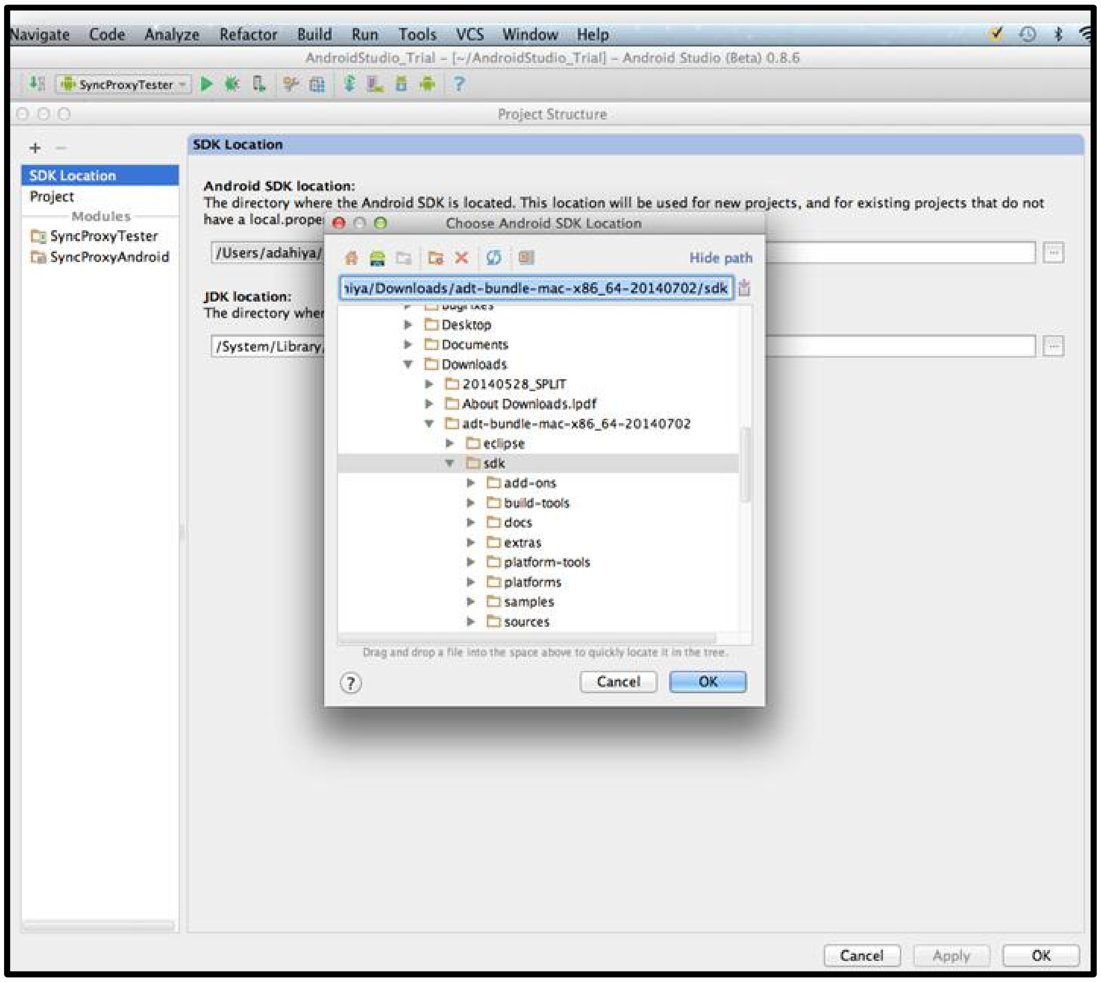
Android SDK location:
The directory where the Android SDK is located. This location will be used for new projects, and for existing projects that do not have a local.properties file with a sdk.dir property.

```
/Users/adahiya/Downloads/adt-bundle-mac-x86_64-20140702/sdk
```

JDK location:
The directory where the Java Development Kit (JDK) is located.

```
/System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home
```

35. By default it will be set up to SDK for Android Studio


Navigate Code Analyze Refactor Build Run Tools VCS Window Help
AndroidStudio_Trial - [~/AndroidStudio_Trial] - Android Studio (Beta) 0.8.6
SyncProxyTester
Project Structure
SDK Location
Project
Modules
SyncProxyTester
SyncProxyAndroid
Choose Android SDK Location
Hide path
Desktop
Documents
Downloads
20140528_SPLIT
About Downloads.lpdf
adt-bundle-mac-x86_64-20140702
eclipse
sdk
add-ons
build-tools
docs
extras
platform-tools
platforms
samples
sources
Drag and drop a file into the space above to quickly locate it in the tree.
Cancel
OK
Cancel
Apply
OK

Android SDK location:
The directory where the Android SDK is located. This location will be used for new projects, and for existing projects that do not have a local.prope

/Users/adahiya/

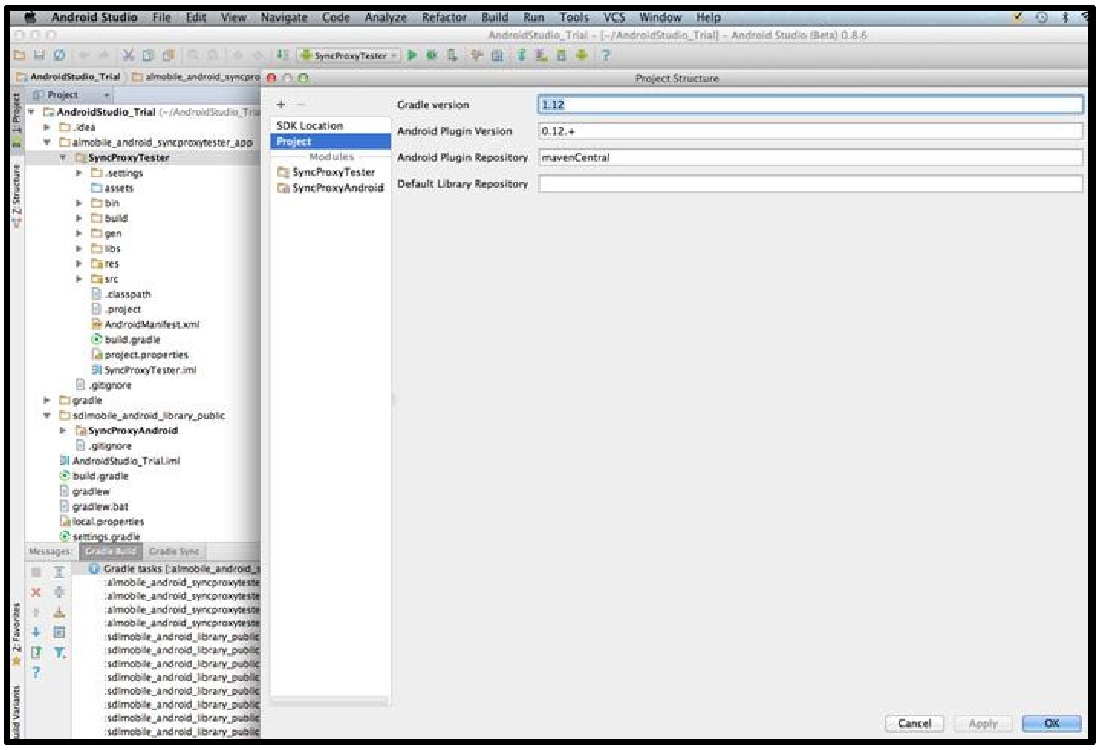
iiya/Downloads/adt-bundle-mac-x86_64-20140702/sdk

JDK location:
The directory wher

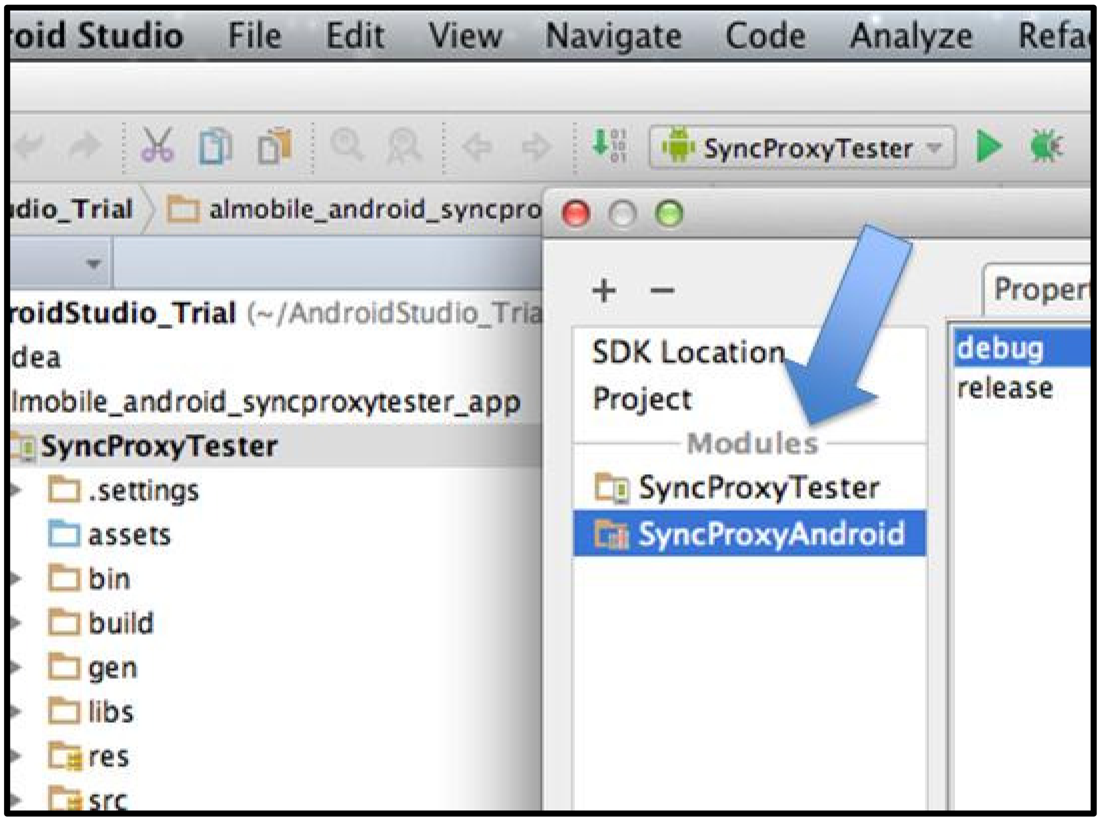
/System/Library/

36. (if you want you can again download an SDK or you can use your existing SDK from Eclipse) From this screen you can point to your existing SDK.

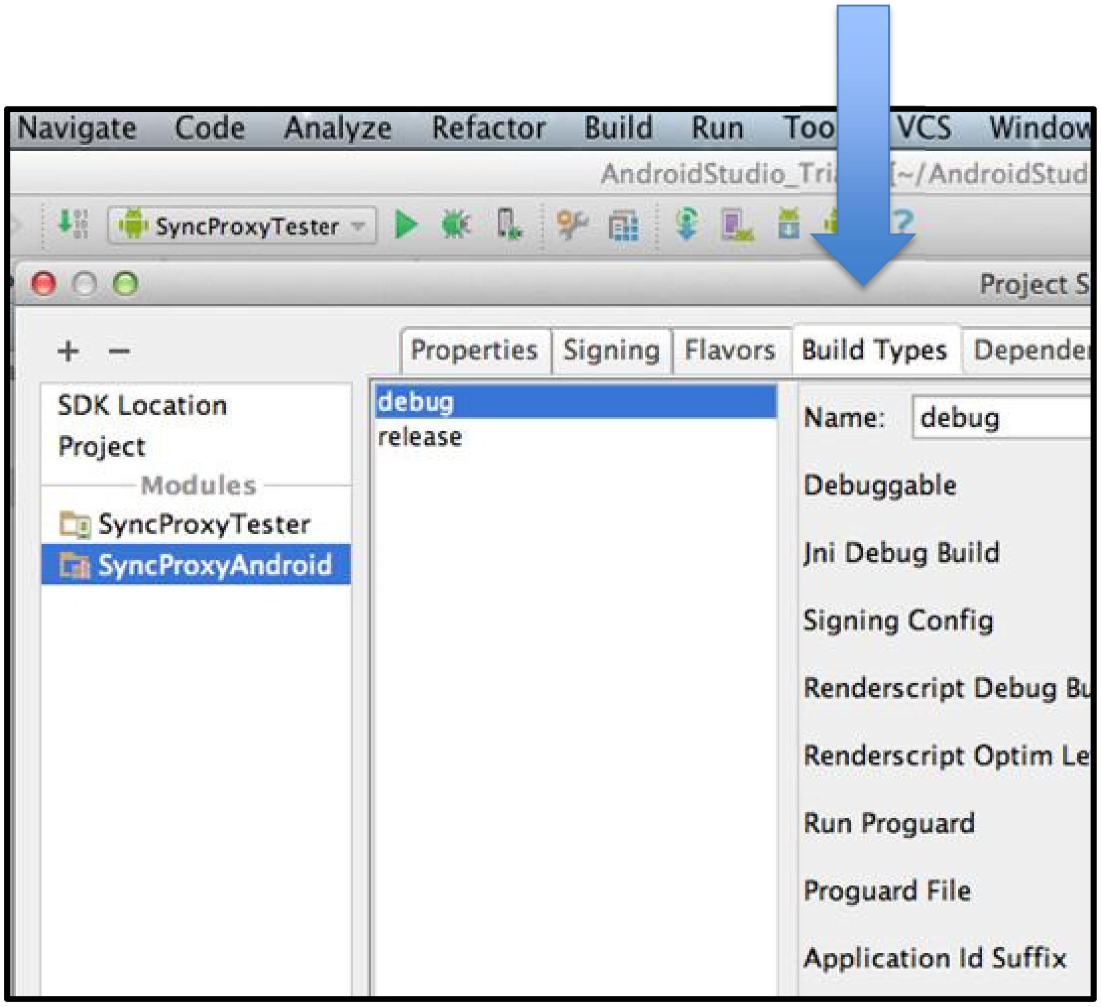
37. This has the build environment. Sometimes there is compatibility issues it is important to note which build version you are working from to avoid errors.


Android Studio File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
AndroidStudio_Trial - [~/AndroidStudio_Trial] - Android Studio (Beta) 0.8.6
Project Structure
SDK Location
Project
Modules
SyncProxyTester
SyncProxyAndroid
Cancel
Apply
OK

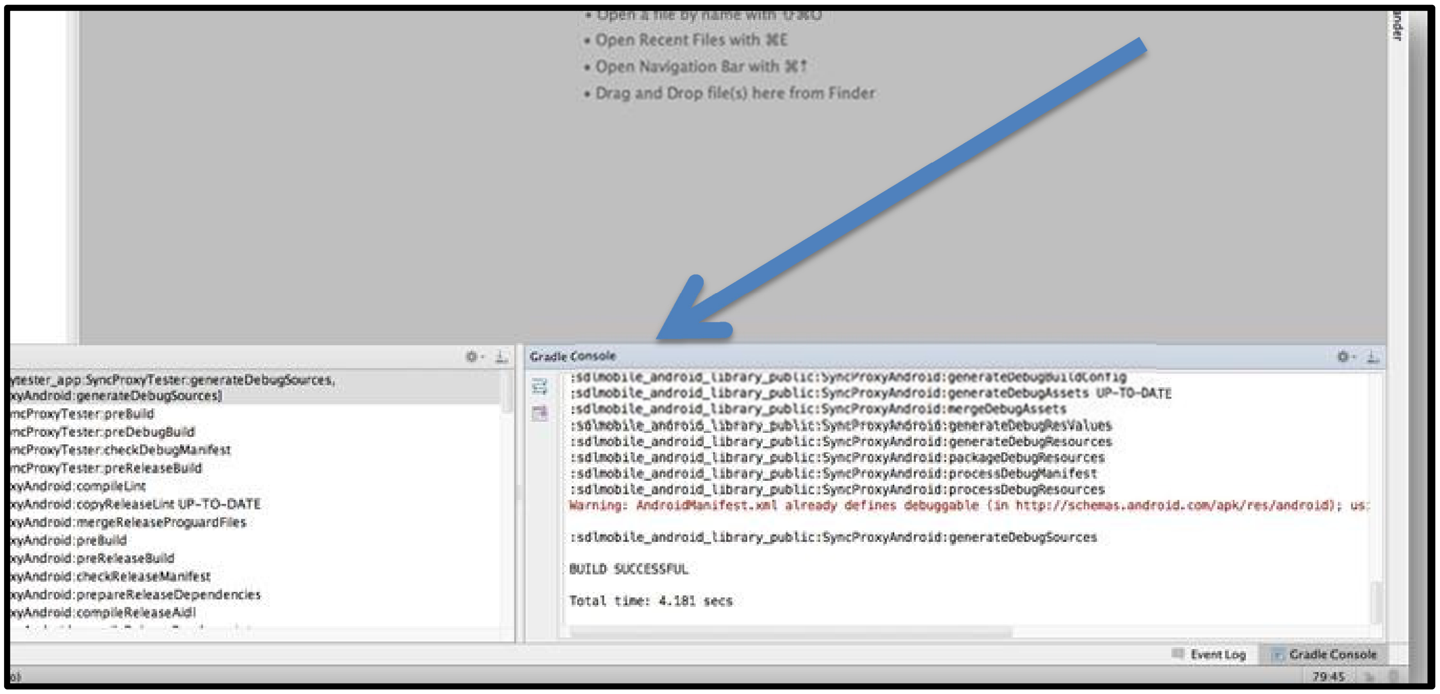
Gradle version: 1.12
Android Plugin Version: 0.12.+
Android Plugin Repository: mavenCentral
Default Library Repository:



38. Under modules you will see your library and application. Make sure to configure to the same setting you had in eclipse. This includes not only the import but also the first build in android studio.
39. You can also select any parameters you want from this screen.



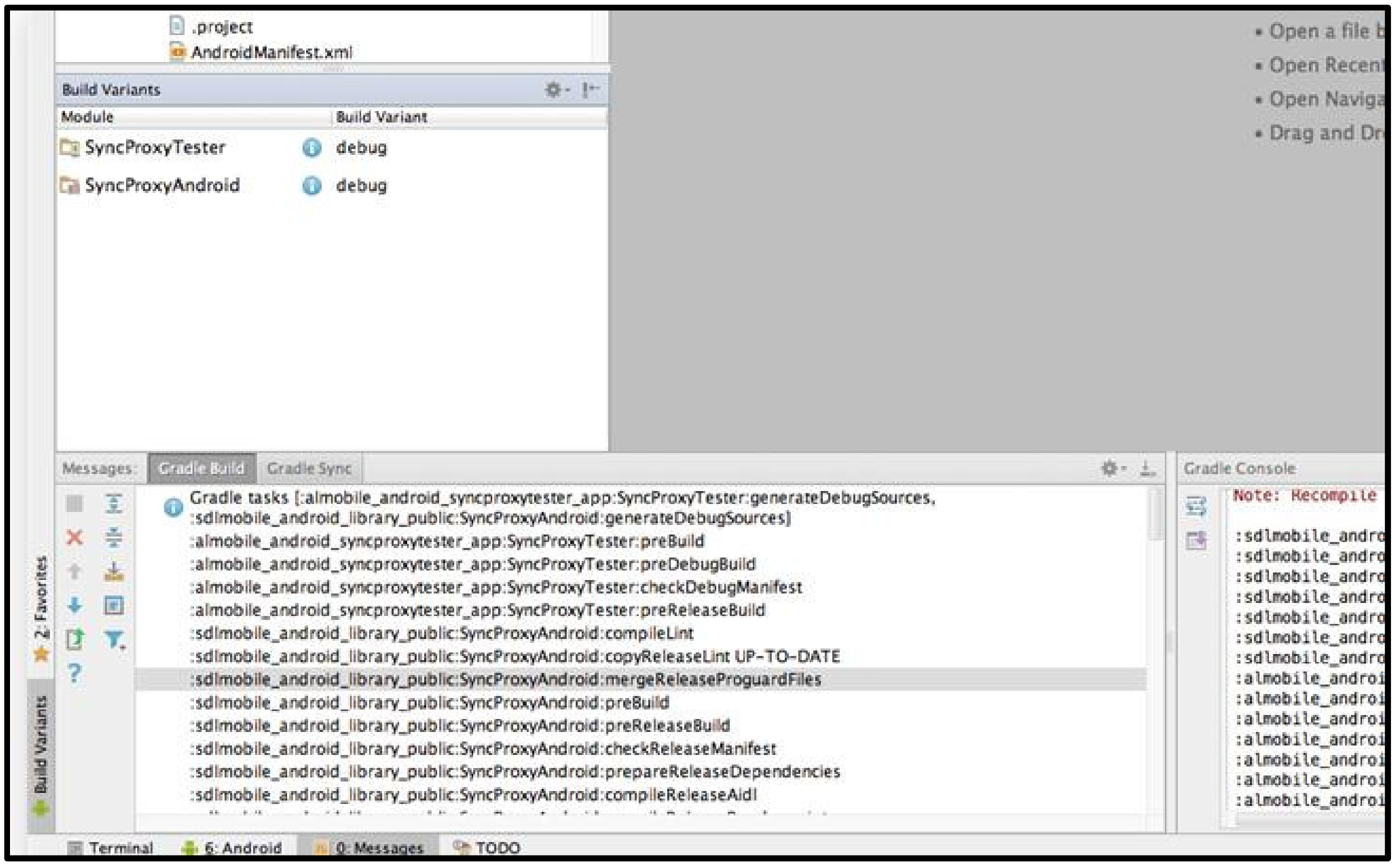
40. Under build types there are two options debug and release. You can use debug for testing and release for uploading your final build. Your build will be configured per your selection of debug or release.


• Open a file by name with ⇧⌘O
• Open Recent Files with ⌘E
• Open Navigation Bar with ⌘↑
• Drag and Drop file(s) here from Finder
Gradle Console
Event Log
Gradle Console
79:45

```
ytester_app:SyncProxyTester:generateDebugSources,
xyAndroid:generateDebugSources]
ncProxyTester:preBuild
ncProxyTester:preDebugBuild
ncProxyTester:checkDebugManifest
ncProxyTester:preReleaseBuild
xyAndroid:compileLint
xyAndroid:copyReleaseLint UP-TO-DATE
xyAndroid:mergeReleaseProguardFiles
xyAndroid:preBuild
xyAndroid:preReleaseBuild
xyAndroid:checkReleaseManifest
xyAndroid:prepareReleaseDependencies
xyAndroid:compileReleaseAidl
```

```
:sdlmobile_android_library_public:SyncProxyAndroid:generateDebugBuildConfig
:sdlmobile_android_library_public:SyncProxyAndroid:generateDebugAssets UP-TO-DATE
:sdlmobile_android_library_public:SyncProxyAndroid:mergeDebugAssets
:sdlmobile_android_library_public:SyncProxyAndroid:generateDebugResValues
:sdlmobile_android_library_public:SyncProxyAndroid:generateDebugResources
:sdlmobile_android_library_public:SyncProxyAndroid:packageDebugResources
:sdlmobile_android_library_public:SyncProxyAndroid:processDebugManifest
:sdlmobile_android_library_public:SyncProxyAndroid:processDebugResources
Warning: AndroidManifest.xml already defines debuggable (in http://schemas.android.com/apk/res/android); us

:sdlmobile_android_library_public:SyncProxyAndroid:generateDebugSources

BUILD SUCCESSFUL

Total time: 4.181 secs
```

41. On the bottom right hand side of the screen make sure the Gradle console is open. This will show griddle environment log. It will display any errors in code.

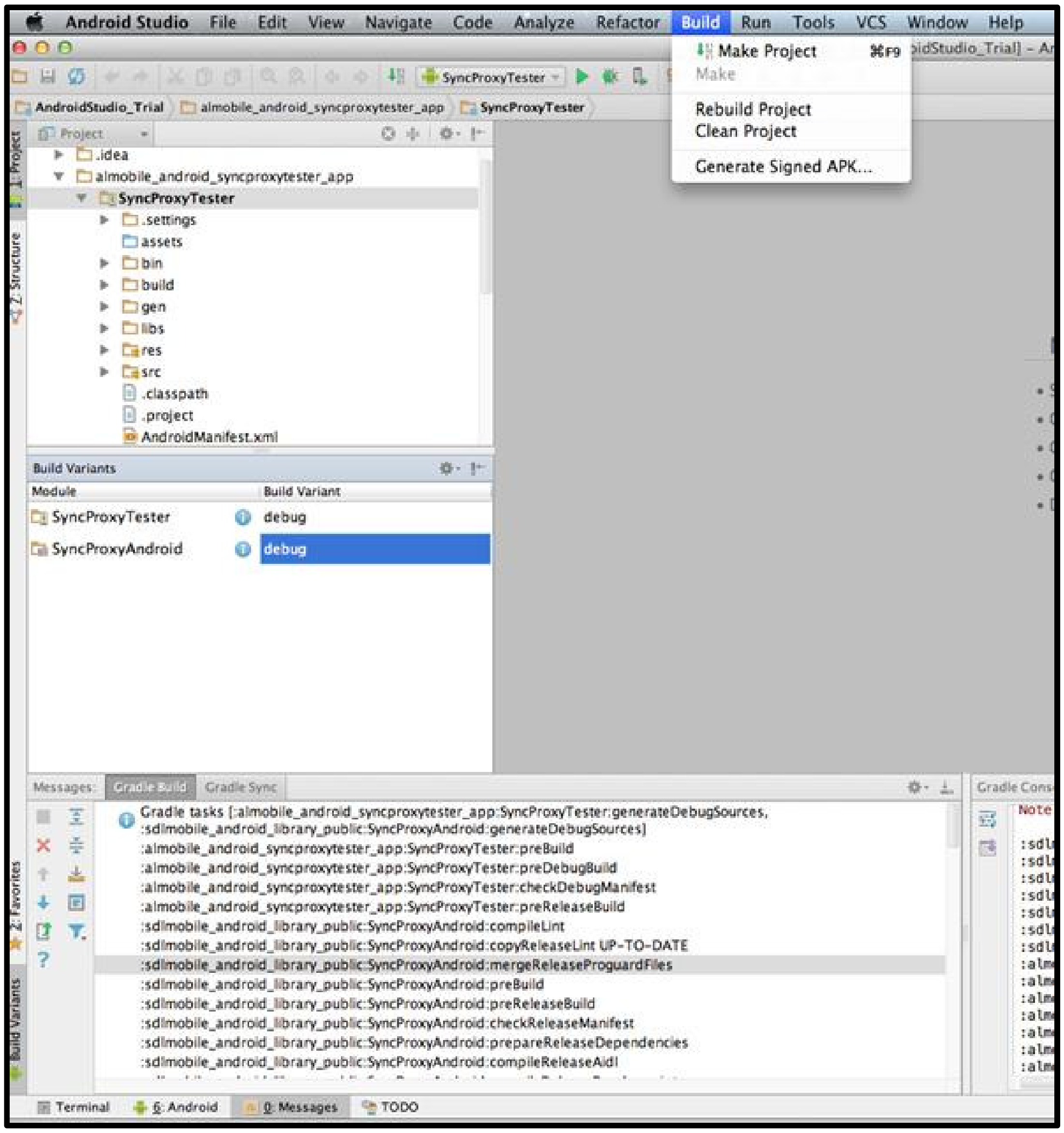


Module	Build Variant
SyncProxyTester	debug
SyncProxyAndroid	debug

```
Gradle tasks [:almobile_android_syncproxytester_app:SyncProxyTester:generateDebugSources,
:sdlmobile_android_library_public:SyncProxyAndroid:generateDebugSources]
:almobile_android_syncproxytester_app:SyncProxyTester:preBuild
:almobile_android_syncproxytester_app:SyncProxyTester:preDebugBuild
:almobile_android_syncproxytester_app:SyncProxyTester:checkDebugManifest
:almobile_android_syncproxytester_app:SyncProxyTester:preReleaseBuild
:sdlmobile_android_library_public:SyncProxyAndroid:compileLint
:sdlmobile_android_library_public:SyncProxyAndroid:copyReleaseLint UP-TO-DATE
:sdlmobile_android_library_public:SyncProxyAndroid:mergeReleaseProguardFiles
:sdlmobile_android_library_public:SyncProxyAndroid:preBuild
:sdlmobile_android_library_public:SyncProxyAndroid:preReleaseBuild
:sdlmobile_android_library_public:SyncProxyAndroid:checkReleaseManifest
:sdlmobile_android_library_public:SyncProxyAndroid:prepareReleaseDependencies
:sdlmobile_android_library_public:SyncProxyAndroid:compileReleaseAidl
```

```
Note: Recompile

:sdlmobile_andro
:sdlmobile_andro
:sdlmobile_andro
:sdlmobile_andro
:sdlmobile_andro
:sdlmobile_andro
:sdlmobile_andro
:almobile_androi
:almobile_androi
:almobile_androi
:almobile_androi
:almobile_androi
:almobile_androi
:almobile_androi
```

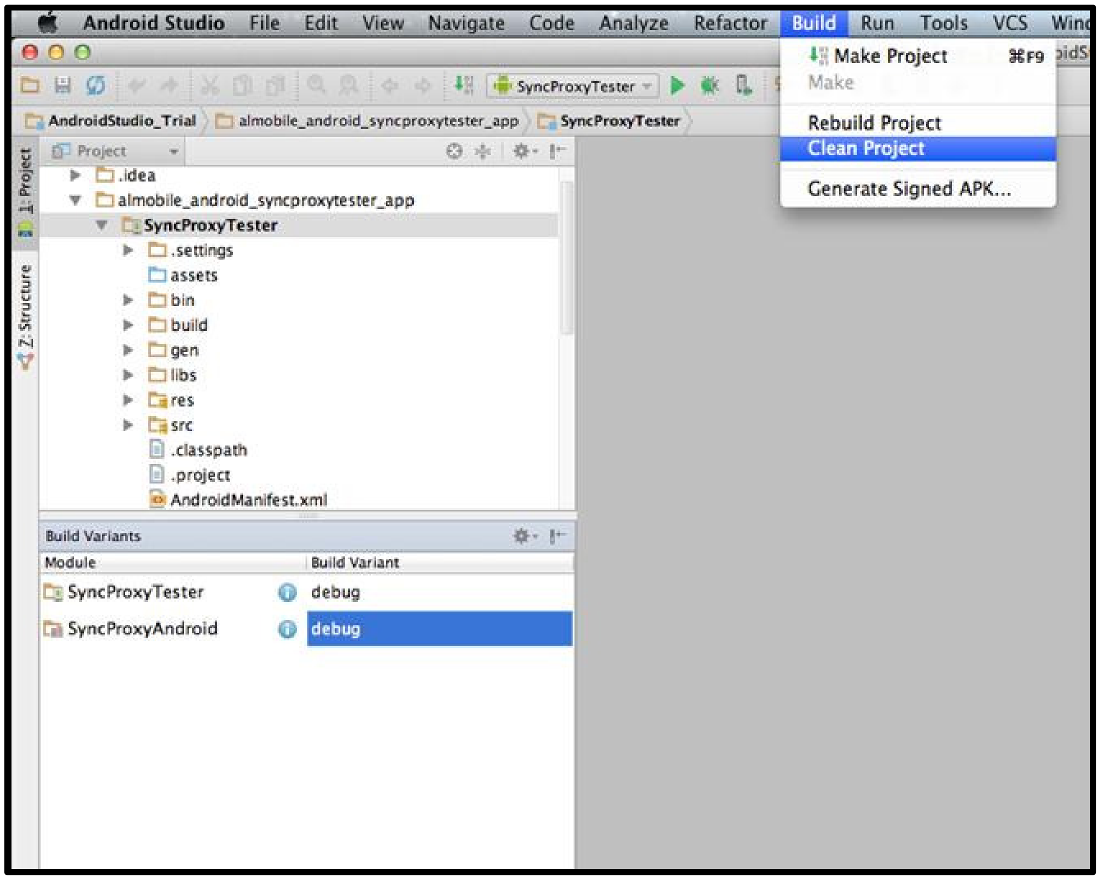
42. Another Important window is build variants. It will display the options to debug or release. Select the option you need.



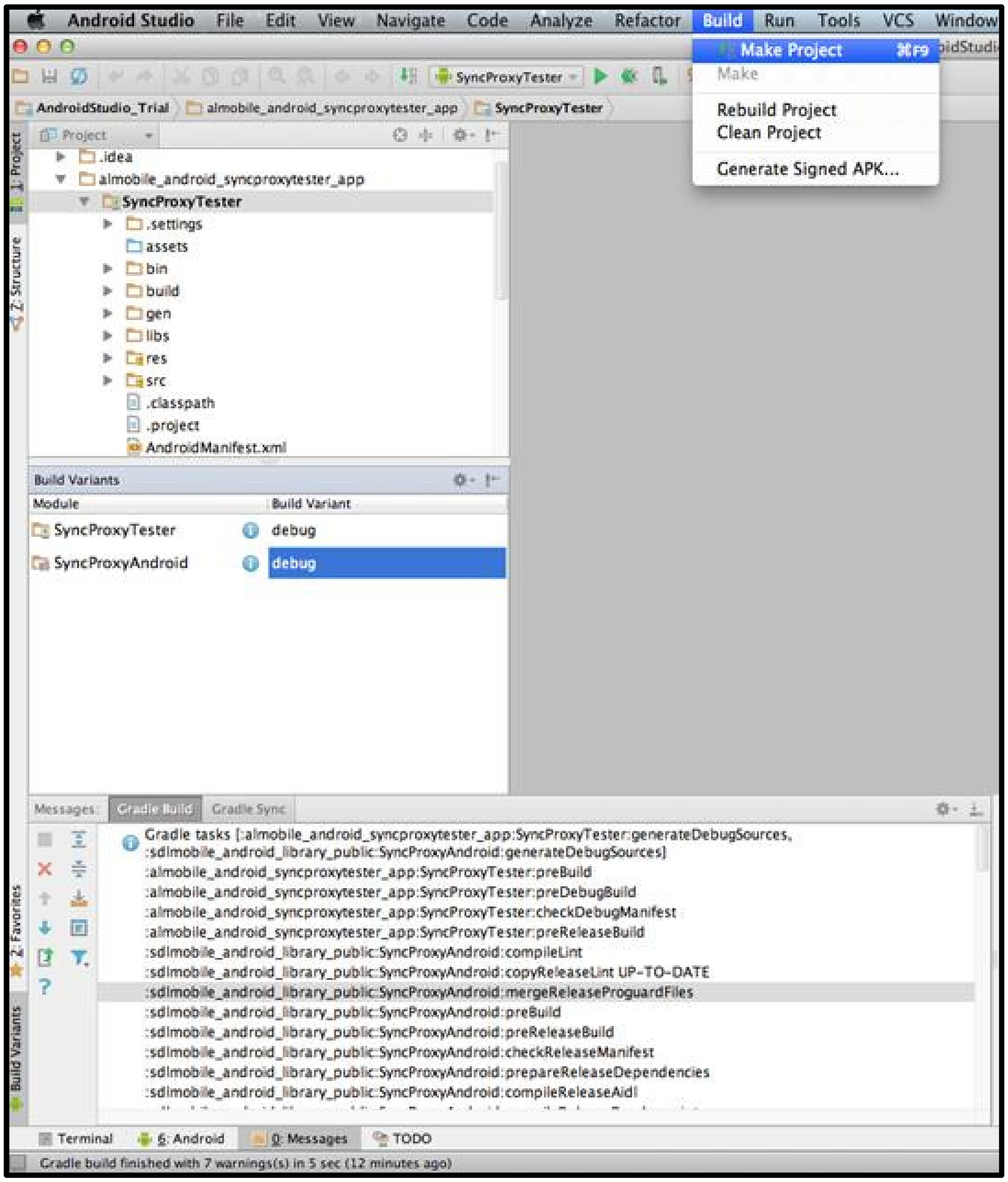
Module	Build Variant
SyncProxyTester	debug
SyncProxyAndroid	debug

```
Gradle tasks [:almobile_android_syncproxytester_app:SyncProxyTester:generateDebugSources,
:sdlmobile_android_library_public:SyncProxyAndroid:generateDebugSources]
:almobile_android_syncproxytester_app:SyncProxyTester:preBuild
:almobile_android_syncproxytester_app:SyncProxyTester:preDebugBuild
:almobile_android_syncproxytester_app:SyncProxyTester:checkDebugManifest
:almobile_android_syncproxytester_app:SyncProxyTester:preReleaseBuild
:sdlmobile_android_library_public:SyncProxyAndroid:compileLint
:sdlmobile_android_library_public:SyncProxyAndroid:copyReleaseLint UP-TO-DATE
:sdlmobile_android_library_public:SyncProxyAndroid:mergeReleaseProguardFiles
:sdlmobile_android_library_public:SyncProxyAndroid:preBuild
:sdlmobile_android_library_public:SyncProxyAndroid:preReleaseBuild
:sdlmobile_android_library_public:SyncProxyAndroid:checkReleaseManifest
:sdlmobile_android_library_public:SyncProxyAndroid:prepareReleaseDependencies
:sdlmobile_android_library_public:SyncProxyAndroid:compileReleaseAidl
```

43. Next select the Build option from the top of the screen.



44. Select Clean Project.



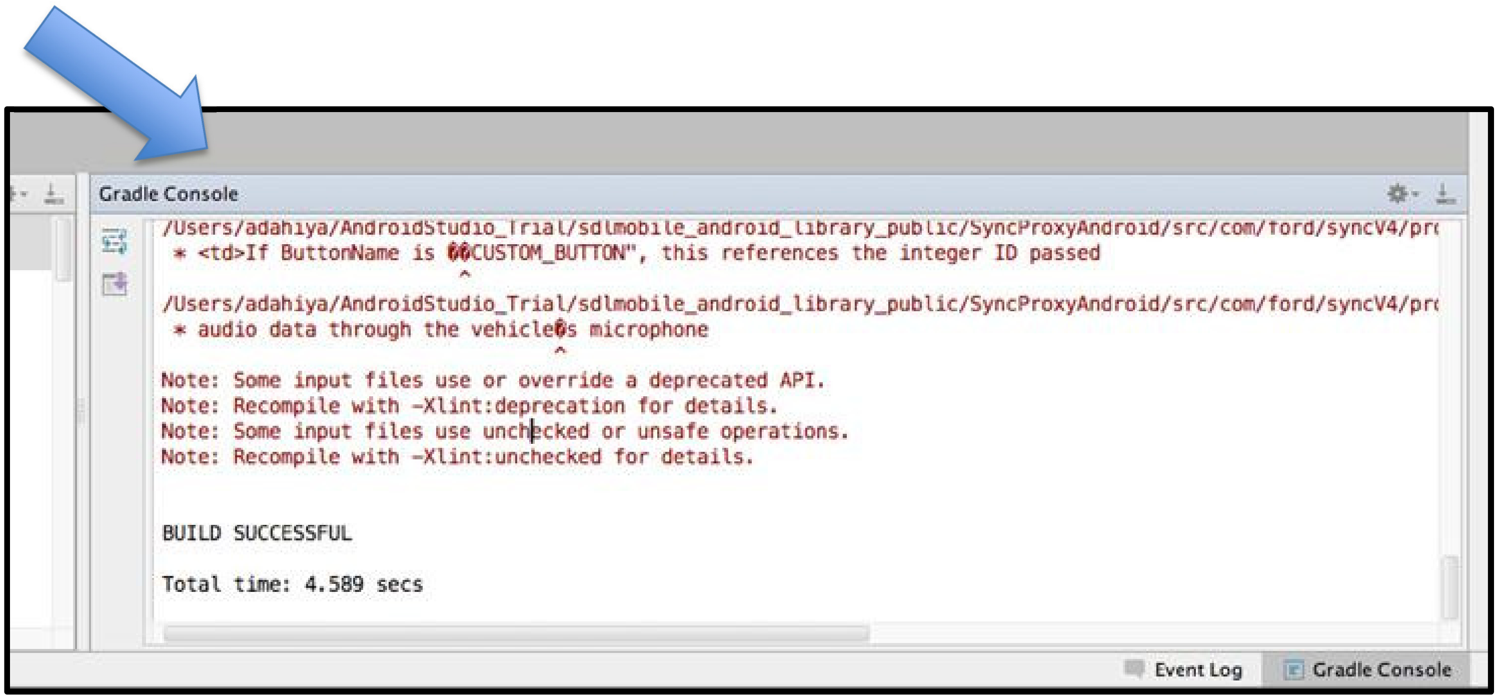
Build Variants

Module	Build Variant
SyncProxyTester	debug
SyncProxyAndroid	debug

```
Gradle tasks [:almobile_android_syncproxytester_app:SyncProxyTester:generateDebugSources,
:sdlmobile_android_library_public:SyncProxyAndroid:generateDebugSources]
:almobile_android_syncproxytester_app:SyncProxyTester:preBuild
:almobile_android_syncproxytester_app:SyncProxyTester:preDebugBuild
:almobile_android_syncproxytester_app:SyncProxyTester:checkDebugManifest
:almobile_android_syncproxytester_app:SyncProxyTester:preReleaseBuild
:sdlmobile_android_library_public:SyncProxyAndroid:compileLint
:sdlmobile_android_library_public:SyncProxyAndroid:copyReleaseLint UP-TO-DATE
:sdlmobile_android_library_public:SyncProxyAndroid:mergeReleaseProguardFiles
:sdlmobile_android_library_public:SyncProxyAndroid:preBuild
:sdlmobile_android_library_public:SyncProxyAndroid:preReleaseBuild
:sdlmobile_android_library_public:SyncProxyAndroid:checkReleaseManifest
:sdlmobile_android_library_public:SyncProxyAndroid:prepareReleaseDependencies
:sdlmobile_android_library_public:SyncProxyAndroid:compileReleaseAidl
```

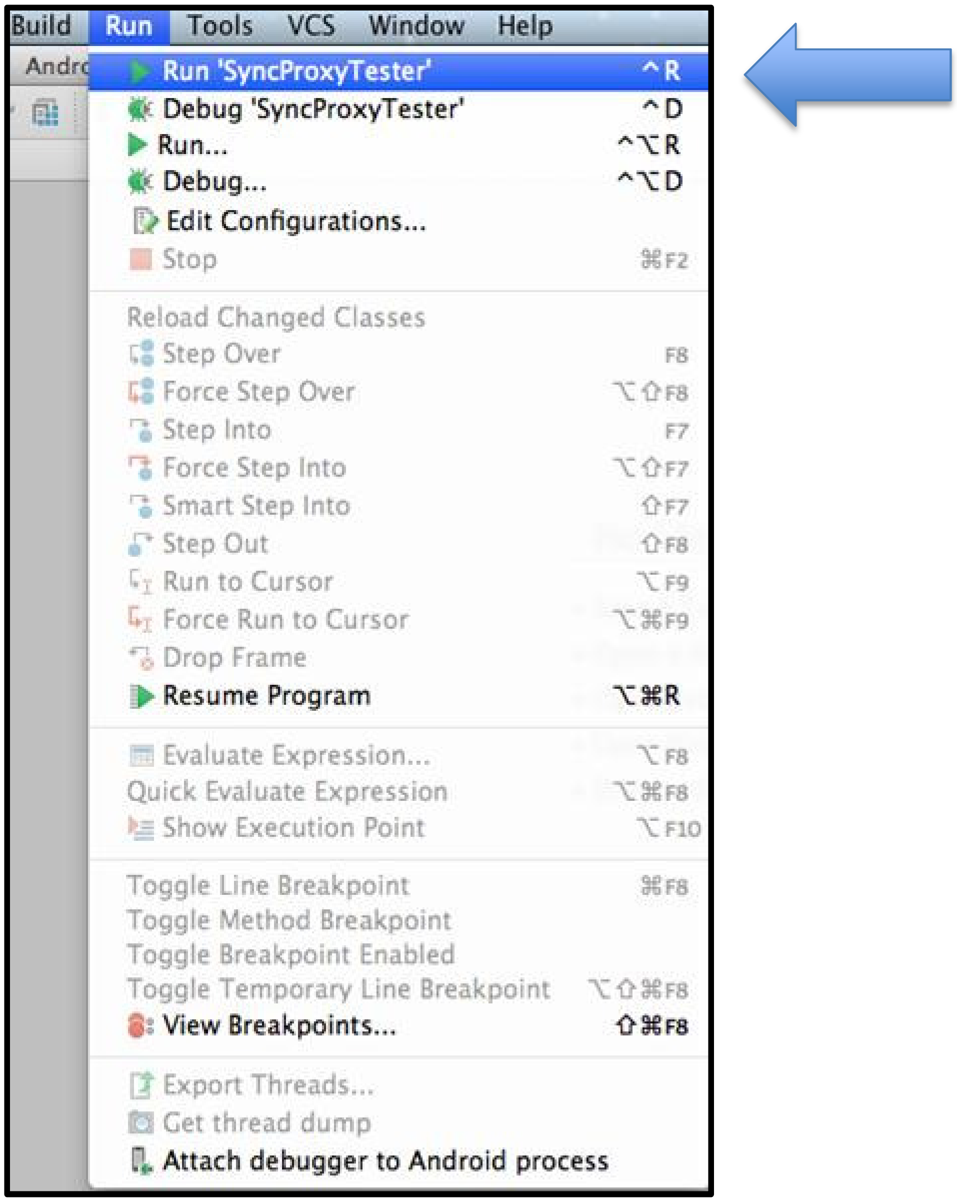
Gradle build finished with 7 warnings(s) in 5 sec (12 minutes ago)

45. Next select Build from the top of the screen again this time select Make Project.

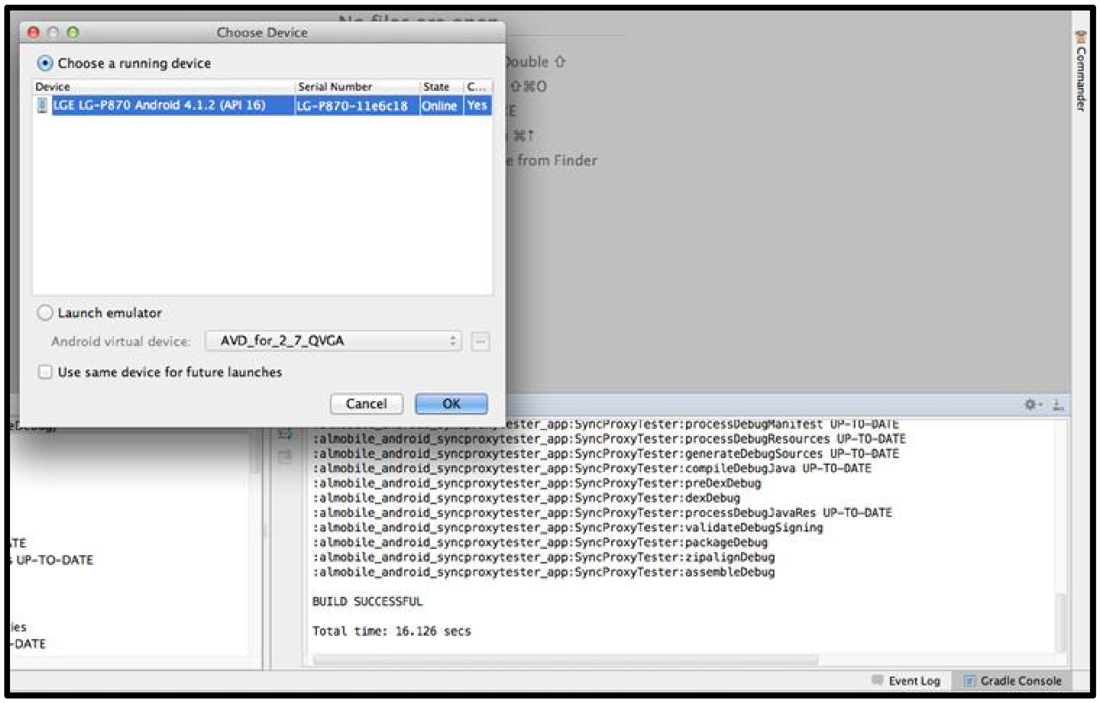


```
/Users/adahiya/AndroidStudio_Trial/sdlmobile_android_library_public/SyncProxyAndroid/src/com/ford/syncV4/pr
 * <td>If ButtonName is ��CUSTOM_BUTTON", this references the integer ID passed
                          ^
/Users/adahiya/AndroidStudio_Trial/sdlmobile_android_library_public/SyncProxyAndroid/src/com/ford/syncV4/pr
 * audio data through the vehicle�s microphone
                                  ^
Note: Some input files use or override a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
Note: Some input files use unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.


BUILD SUCCESSFUL

Total time: 4.589 secs
```

46. If there are any warnings they will be displayed in the Gradle Console. You should get a build successful message in the console.



47. Once your build is complete. You can now run it on your phone.
48. Connect your phone go to run press run Sync Proxy Tester

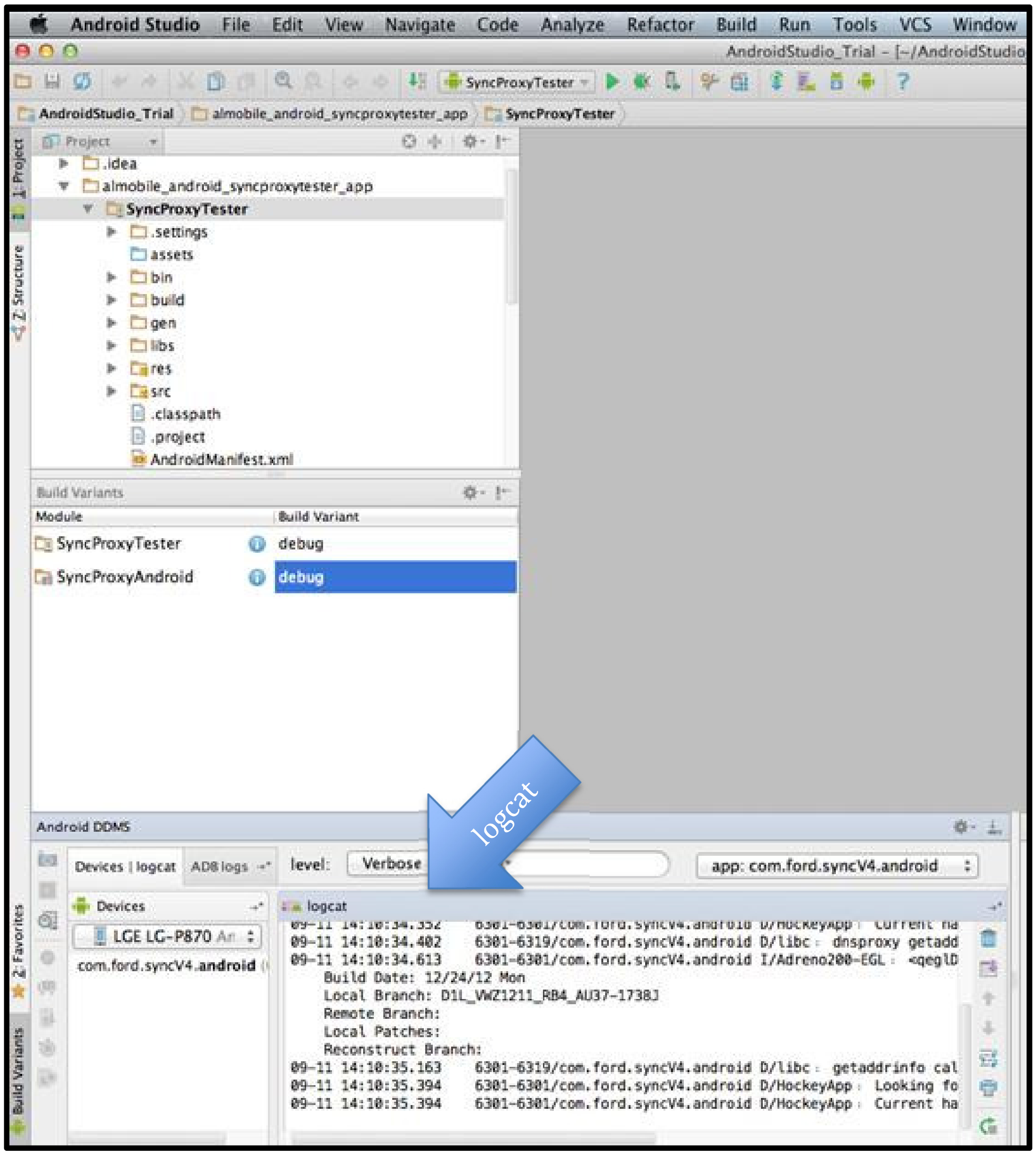


Device	Serial Number	State	C...
LGE LG-P870 Android 4.1.2 (API 16)	LG-P870-11e6c18	Online	Yes

Android virtual device: AVD_for_2_7_QVGA

```
...ester_app:SyncProxyTester:processDebugManifest UP-TO-DATE
:almobile_android_syncproxytester_app:SyncProxyTester:processDebugResources UP-TO-DATE
:almobile_android_syncproxytester_app:SyncProxyTester:generateDebugSources UP-TO-DATE
:almobile_android_syncproxytester_app:SyncProxyTester:compileDebugJava UP-TO-DATE
:almobile_android_syncproxytester_app:SyncProxyTester:preDexDebug
:almobile_android_syncproxytester_app:SyncProxyTester:dexDebug
:almobile_android_syncproxytester_app:SyncProxyTester:processDebugJavaRes UP-TO-DATE
:almobile_android_syncproxytester_app:SyncProxyTester:validateDebugSigning
:almobile_android_syncproxytester_app:SyncProxyTester:packageDebug
:almobile_android_syncproxytester_app:SyncProxyTester:zipalignDebug
:almobile_android_syncproxytester_app:SyncProxyTester:assembleDebug

BUILD SUCCESSFUL

Total time: 16.126 secs
```

49. Your logs will be displayed in the Griddle Console. It will prompt you to choose a device. Select your device and click OK.



Module	Build Variant
SyncProxyTester	debug
SyncProxyAndroid	debug

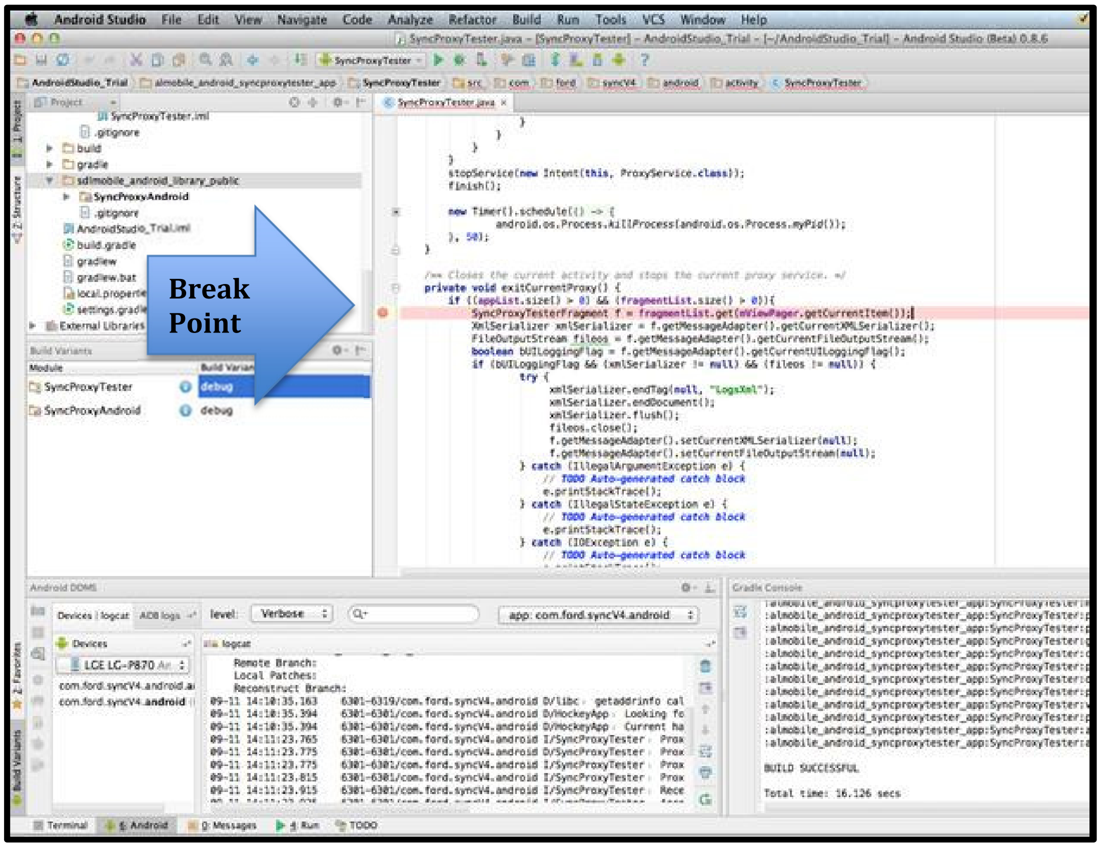
level: Verbose

app: com.ford.syncV4.android

```
09-11 14:10:34.352    6301-6301/com.ford.syncV4.android D/HockeyApp:  Current ha
09-11 14:10:34.402    6301-6319/com.ford.syncV4.android D/libc:  dnsproxy getadd
09-11 14:10:34.613    6301-6301/com.ford.syncV4.android I/Adreno200-EGL:  <qeglD
    Build Date: 12/24/12 Mon
    Local Branch: D1L_VWZ1211_RB4_AU37-1738J
    Remote Branch:
    Local Patches:
    Reconstruct Branch:
09-11 14:10:35.163    6301-6319/com.ford.syncV4.android D/libc:  getaddrinfo cal
09-11 14:10:35.394    6301-6301/com.ford.syncV4.android D/HockeyApp:  Looking fo
09-11 14:10:35.394    6301-6301/com.ford.syncV4.android D/HockeyApp:  Current ha
```

50. Once the application starts running you will see the logcat.

51. Your application logs (debug logs will be visible under logcat)



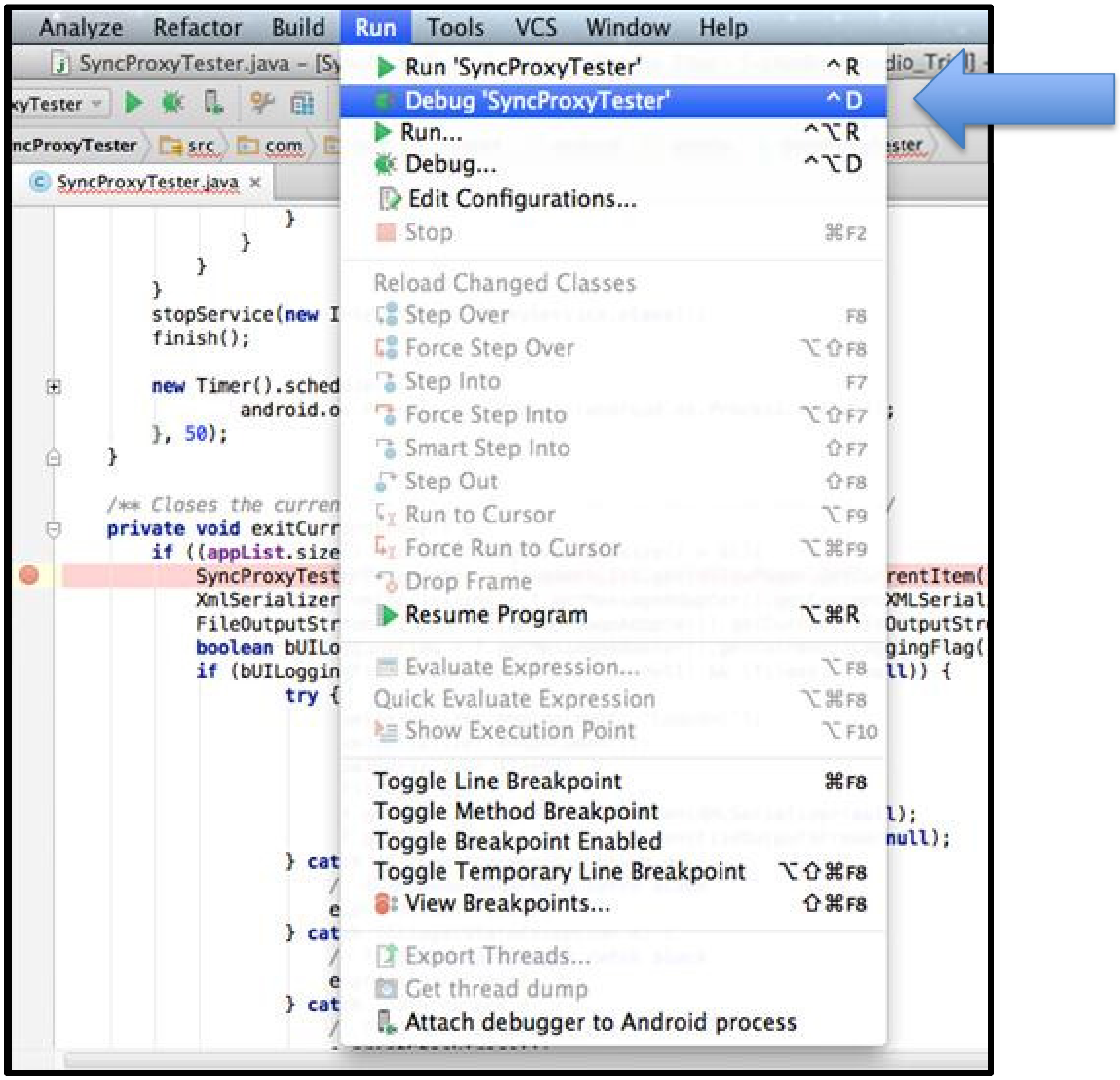
```
                    }
                }
            }
        }
        stopService(new Intent(this, ProxyService.class));
        finish();

        new Timer().schedule(() -> {
                android.os.Process.killProcess(android.os.Process.myPid());
        }, 50);
    }

    /** Closes the current activity and stops the current proxy service. */
    private void exitCurrentProxy() {
        if ((appList.size() > 0) && (fragmentList.size() > 0)){
            SyncProxyTesterFragment f = fragmentList.get(mViewPager.getCurrentItem());
            XmlSerializer xmlSerializer = f.getMessageAdapter().getCurrentXMLSerializer();
            FileOutputStream fileos = f.getMessageAdapter().getCurrentFileOutputStream();
            boolean bUILoggingFlag = f.getMessageAdapter().getCurrentUILoggingFlag();
            if (bUILoggingFlag && (xmlSerializer != null) && (fileos != null)) {
                    try {
                        xmlSerializer.endTag(null, "LogsXml");
                        xmlSerializer.endDocument();
                        xmlSerializer.flush();
                        fileos.close();
                        f.getMessageAdapter().setCurrentXMLSerializer(null);
                        f.getMessageAdapter().setCurrentFileOutputStream(null);
                    } catch (IllegalArgumentException e) {
                      // TODO Auto-generated catch block
                      e.printStackTrace();
                    } catch (IllegalStateException e) {
                      // TODO Auto-generated catch block
                      e.printStackTrace();
                    } catch (IOException e) {
                      // TODO Auto-generated catch block
```

level: Verbose   app: com.ford.syncV4.android

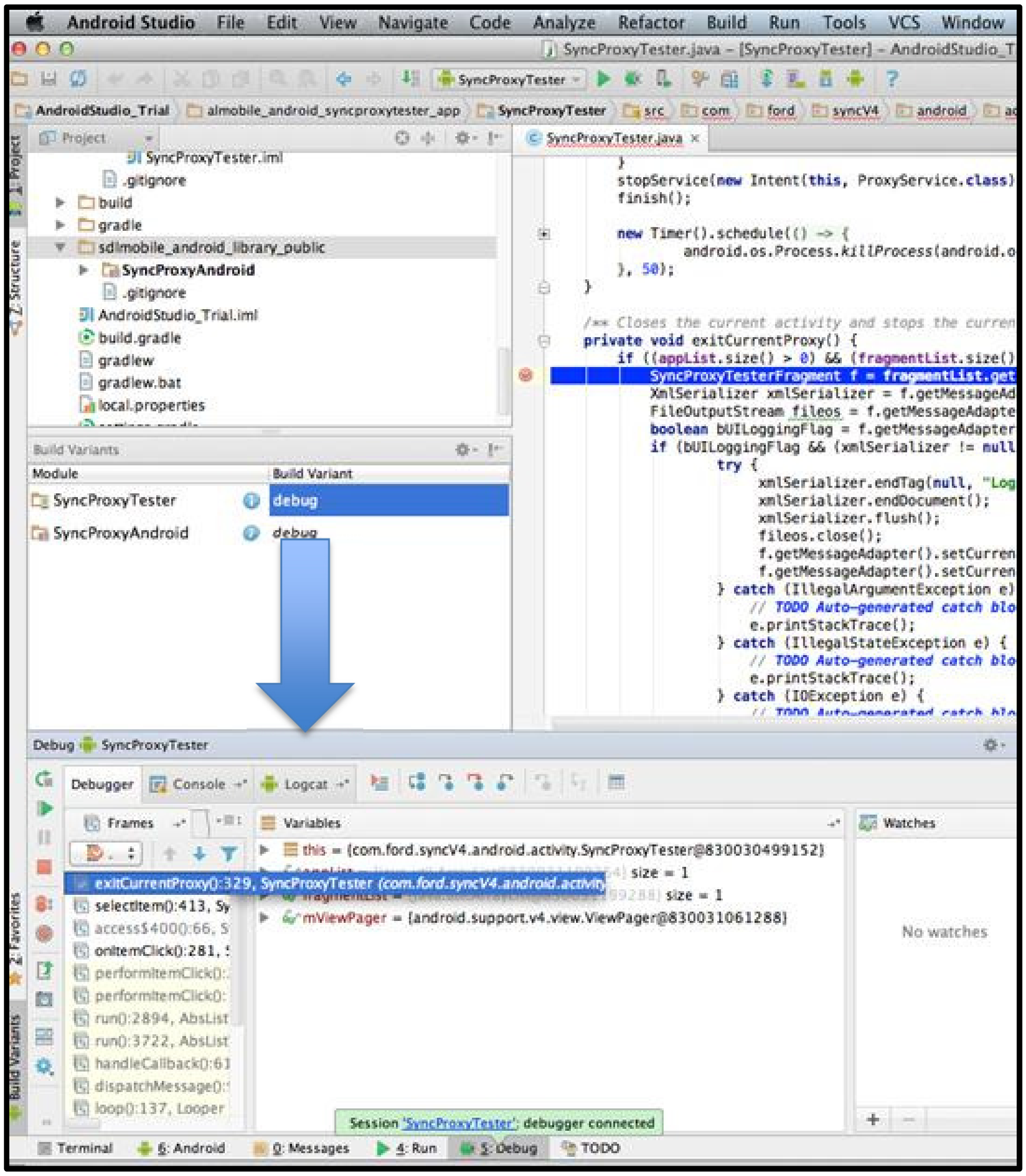
```
    Remote Branch:
    Local Patches:
    Reconstruct Branch:
09-11 14:10:35.163    6301-6319/com.ford.syncV4.android D/libc﹕ getaddrinfo cal
09-11 14:10:35.394    6301-6301/com.ford.syncV4.android D/HockeyApp﹕ Looking fo
09-11 14:10:35.394    6301-6301/com.ford.syncV4.android D/HockeyApp﹕ Current ha
09-11 14:11:23.765    6301-6301/com.ford.syncV4.android I/SyncProxyTester﹕ Prox
09-11 14:11:23.775    6301-6301/com.ford.syncV4.android D/SyncProxyTester﹕ Prox
09-11 14:11:23.775    6301-6301/com.ford.syncV4.android I/SyncProxyTester﹕ Prox
09-11 14:11:23.815    6301-6301/com.ford.syncV4.android I/SyncProxyTester﹕ Prox
09-11 14:11:23.915    6301-6301/com.ford.syncV4.android I/SyncProxyTester﹕ Rece
```

```
:almobile_android_syncproxytester_app:SyncProxyTester:p
:almobile_android_syncproxytester_app:SyncProxyTester:p
:almobile_android_syncproxytester_app:SyncProxyTester:g
:almobile_android_syncproxytester_app:SyncProxyTester:c
:almobile_android_syncproxytester_app:SyncProxyTester:p
:almobile_android_syncproxytester_app:SyncProxyTester:c
:almobile_android_syncproxytester_app:SyncProxyTester:p
:almobile_android_syncproxytester_app:SyncProxyTester:v
:almobile_android_syncproxytester_app:SyncProxyTester:p
:almobile_android_syncproxytester_app:SyncProxyTester:a
:almobile_android_syncproxytester_app:SyncProxyTester:a

BUILD SUCCESSFUL

Total time: 16.126 secs
```

52. For debugging you must put in a break point in the code.



53. Once the breakpoint is inserted go to Run and select debug SyncProxyTester. It will again prompt you to select a device. Select the desired device and click OK.



```
        }
        stopService(new Intent(this, ProxyService.class)
        finish();

        new Timer().schedule(() -> {
                android.os.Process.killProcess(android.o
        }, 50);
    }

    /** Closes the current activity and stops the curren
    private void exitCurrentProxy() {
        if ((appList.size() > 0) && (fragmentList.size()
            SyncProxyTesterFragment f = fragmentList.get
            XmlSerializer xmlSerializer = f.getMessageAd
            FileOutputStream fileos = f.getMessageAdapte
            boolean bUILoggingFlag = f.getMessageAdapter
            if (bUILoggingFlag && (xmlSerializer != null
                    try {
                        xmlSerializer.endTag(null, "Log
                        xmlSerializer.endDocument();
                        xmlSerializer.flush();
                        fileos.close();
                        f.getMessageAdapter().setCurren
                        f.getMessageAdapter().setCurren
                    } catch (IllegalArgumentException e)
                        // TODO Auto-generated catch blo
                        e.printStackTrace();
                    } catch (IllegalStateException e) {
                        // TODO Auto-generated catch blo
                        e.printStackTrace();
                    } catch (IOException e) {
```

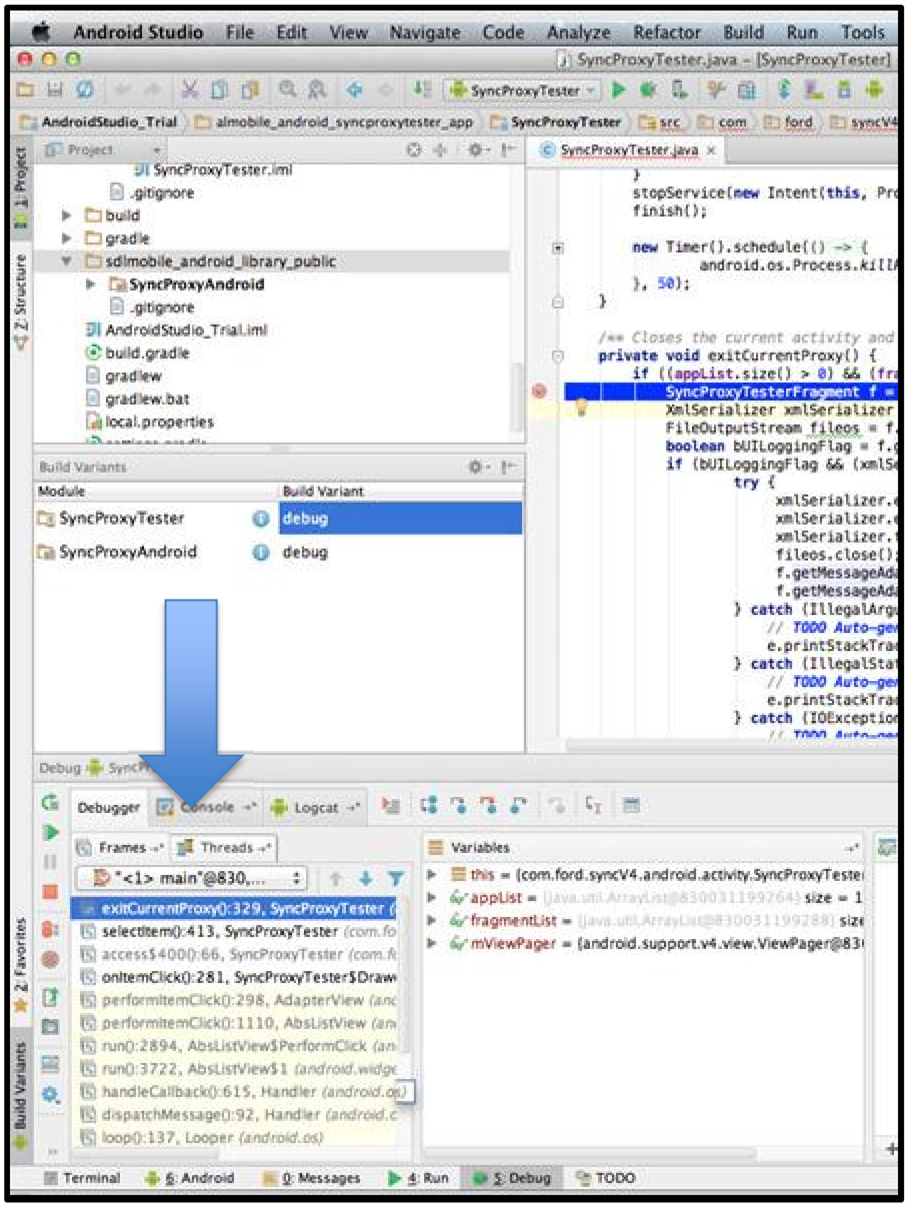
```
exitCurrentProxy():329, SyncProxyTester (com.ford.syncV4.android.activity
selectItem():413, Sy
access$4000():66, S
onItemClick():281, S
performItemClick():
performItemClick():
run():2894, AbsList
run():3722, AbsList
handleCallback():61
dispatchMessage():
loop():137, Looper
```

```
this = {com.ford.syncV4.android.activity.SyncProxyTester@830030499152}
size = 1
size = 1
mViewPager = {android.support.v4.view.ViewPager@830031061288}
```

Session 'SyncProxyTester': debugger connected

Module	Build Variant
SyncProxyTester	debug
SyncProxyAndroid	debug

54. The app will be open on your phone and logs will be displayed in logcat.



```
SyncProxyTester.iml
.gitignore
build
gradle
sdlmobile_android_library_public
    SyncProxyAndroid
    .gitignore
AndroidStudio_Trial.iml
build.gradle
gradlew
gradlew.bat
local.properties
```

Build Variants

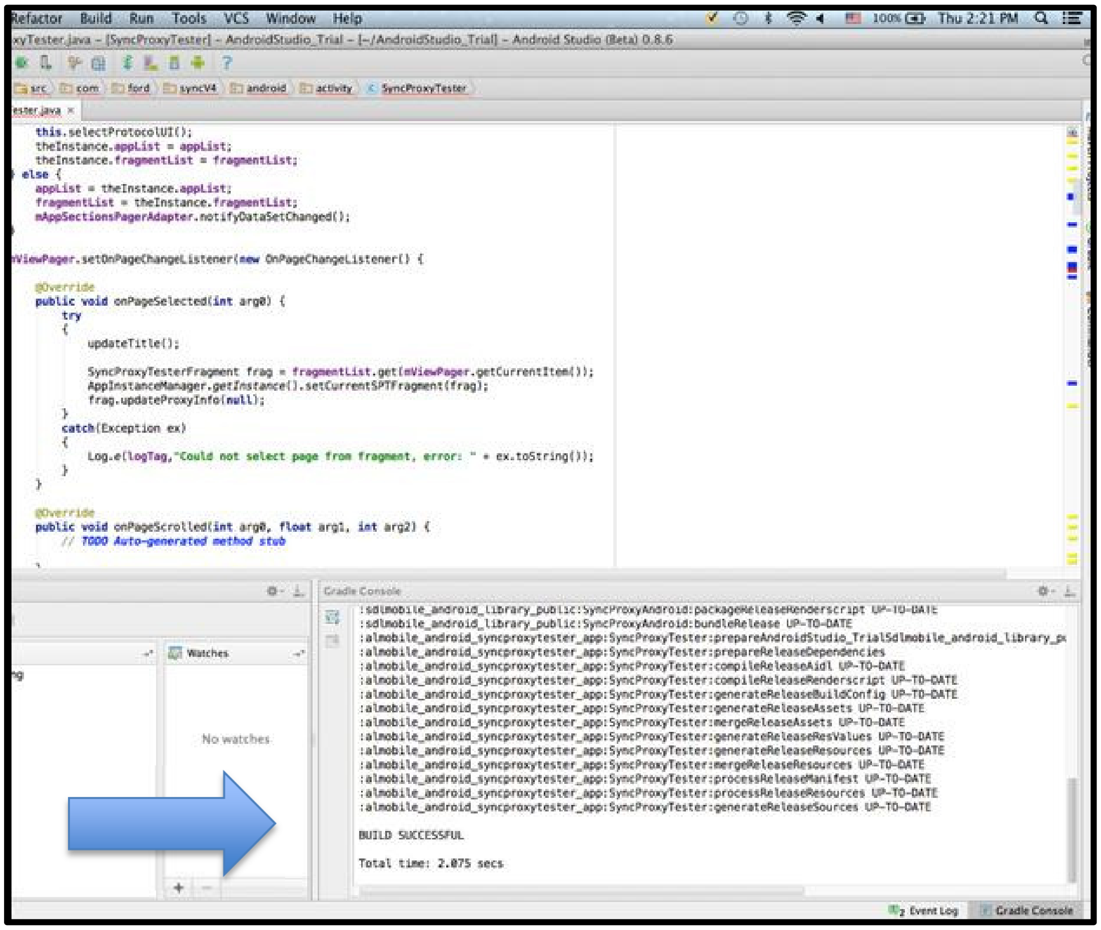
Module	Build Variant
SyncProxyTester	debug
SyncProxyAndroid	debug

```
        }
        stopService(new Intent(this, Pro
        finish();

        new Timer().schedule(() -> {
                android.os.Process.killA
        }, 50);
    }

    /** Closes the current activity and
    private void exitCurrentProxy() {
        if ((appList.size() > 0) && (fra
            SyncProxyTesterFragment f =
            XmlSerializer xmlSerializer
            FileOutputStream fileos = f.
            boolean bUILoggingFlag = f.g
            if (bUILoggingFlag && (xmlSe
                try {
                    xmlSerializer.e
                    xmlSerializer.e
                    xmlSerializer.t
                    fileos.close();
                    f.getMessageAd
                    f.getMessageAd
                } catch (IllegalArgu
                   // TODO Auto-gen
                   e.printStackTrac
                } catch (IllegalStat
                   // TODO Auto-gen
                   e.printStackTrac
                } catch (IOException
```

```
"<1> main"@830,...
exitCurrentProxy():329, SyncProxyTester (
selectItem():413, SyncProxyTester (com.fo
access$400():66, SyncProxyTester (com.fo
onItemClick():281, SyncProxyTester$Draw
performItemClick():298, AdapterView (anc
performItemClick():1110, AbsListView (an
run():2894, AbsListView$PerformClick (an
run():3722, AbsListView$1 (android.widge
handleCallback():615, Handler (android.os)
dispatchMessage():92, Handler (android.c
loop():137, Looper (android.os)
```

```
this = {com.ford.syncV4.android.activity.SyncProxyTeste
appList = {java.util.ArrayList@830031199264} size = 1
fragmentList = {java.util.ArrayList@830031199288} size
mViewPager = {android.support.v4.view.ViewPager@83
```

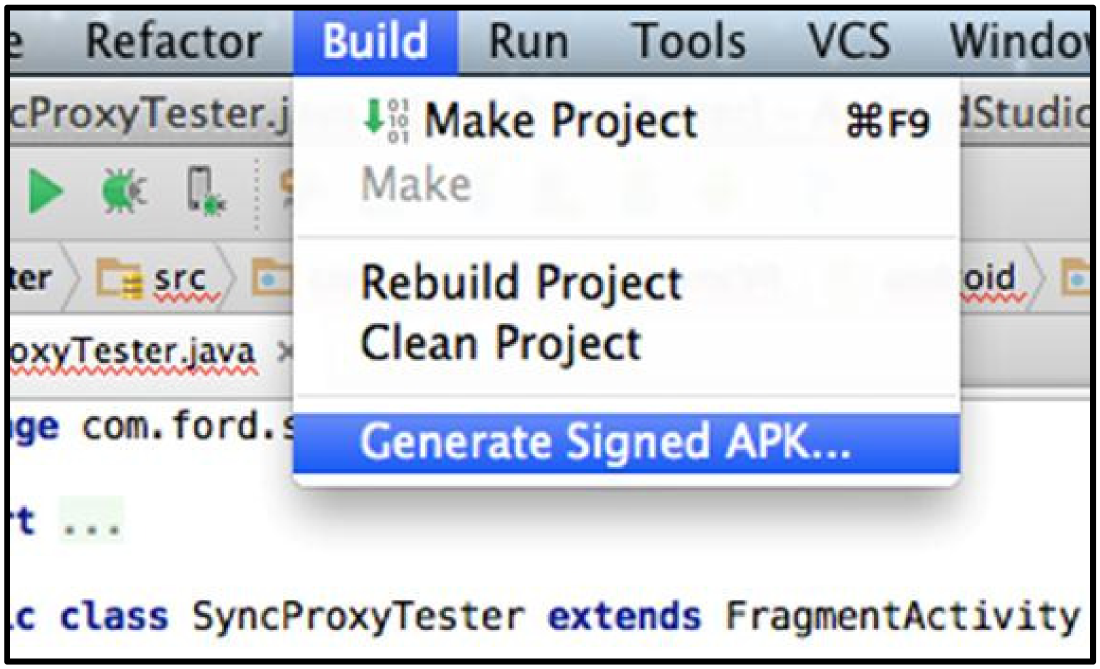
55. Execute the function where you inserted your breakpoint. Select exit SPT and it will be highlighted in the code. Your stack traces will be displayed here.


Refactor Build Run Tools VCS Window Help
xyTester.java – [SyncProxyTester] – AndroidStudio_Trial – [~/AndroidStudio_Trial] – Android Studio (Beta) 0.8.6
src
com
ford
syncV4
android
activity
SyncProxyTester
ester.java
Gradle Console
Watches
No watches
Event Log
Gradle Console

```
    this.selectProtocolUI();
    theInstance.appList = appList;
    theInstance.fragmentList = fragmentList;
} else {
    appList = theInstance.appList;
    fragmentList = theInstance.fragmentList;
    mAppSectionsPagerAdapter.notifyDataSetChanged();
}

mViewPager.setOnPageChangeListener(new OnPageChangeListener() {

    @Override
    public void onPageSelected(int arg0) {
        try
        {
            updateTitle();

            SyncProxyTesterFragment frag = fragmentList.get(mViewPager.getCurrentItem());
            AppInstanceManager.getInstance().setCurrentSPTFragment(frag);
            frag.updateProxyInfo(null);
        }
        catch(Exception ex)
        {
            Log.e(logTag,"Could not select page from fragment, error: " + ex.toString());
        }
    }

    @Override
    public void onPageScrolled(int arg0, float arg1, int arg2) {
        // TODO Auto-generated method stub
```

```
:sdlmobile_android_library_public:SyncProxyAndroid:packageReleaseRenderscript UP-TO-DATE
:sdlmobile_android_library_public:SyncProxyAndroid:bundleRelease UP-TO-DATE
:almobile_android_syncproxytester_app:SyncProxyTester:prepareAndroidStudio_TrialSdlmobile_android_library_pu
:almobile_android_syncproxytester_app:SyncProxyTester:prepareReleaseDependencies
:almobile_android_syncproxytester_app:SyncProxyTester:compileReleaseAidl UP-TO-DATE
:almobile_android_syncproxytester_app:SyncProxyTester:compileReleaseRenderscript UP-TO-DATE
:almobile_android_syncproxytester_app:SyncProxyTester:generateReleaseBuildConfig UP-TO-DATE
:almobile_android_syncproxytester_app:SyncProxyTester:generateReleaseAssets UP-TO-DATE
:almobile_android_syncproxytester_app:SyncProxyTester:mergeReleaseAssets UP-TO-DATE
:almobile_android_syncproxytester_app:SyncProxyTester:generateReleaseResValues UP-TO-DATE
:almobile_android_syncproxytester_app:SyncProxyTester:generateReleaseResources UP-TO-DATE
:almobile_android_syncproxytester_app:SyncProxyTester:mergeReleaseResources UP-TO-DATE
:almobile_android_syncproxytester_app:SyncProxyTester:processReleaseManifest UP-TO-DATE
:almobile_android_syncproxytester_app:SyncProxyTester:processReleaseResources UP-TO-DATE
:almobile_android_syncproxytester_app:SyncProxyTester:generateReleaseSources UP-TO-DATE

BUILD SUCCESSFUL

Total time: 2.075 secs
```

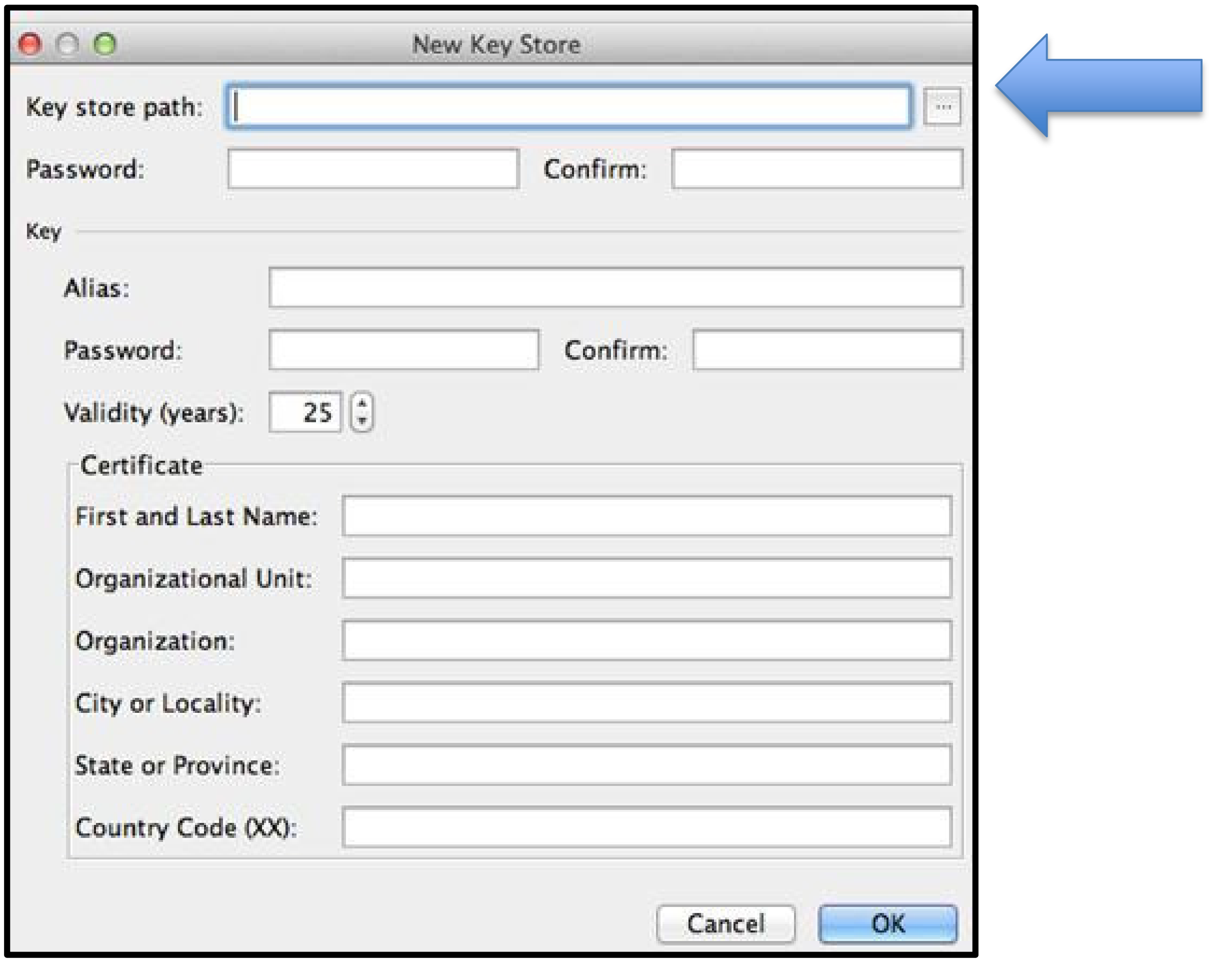
56. Now that your debugging is complete you can resume the application.
57. Simply use the same steps for release. It will automatically build. You should get a BUILD SUCCESSFUL message in the Gradle Console.



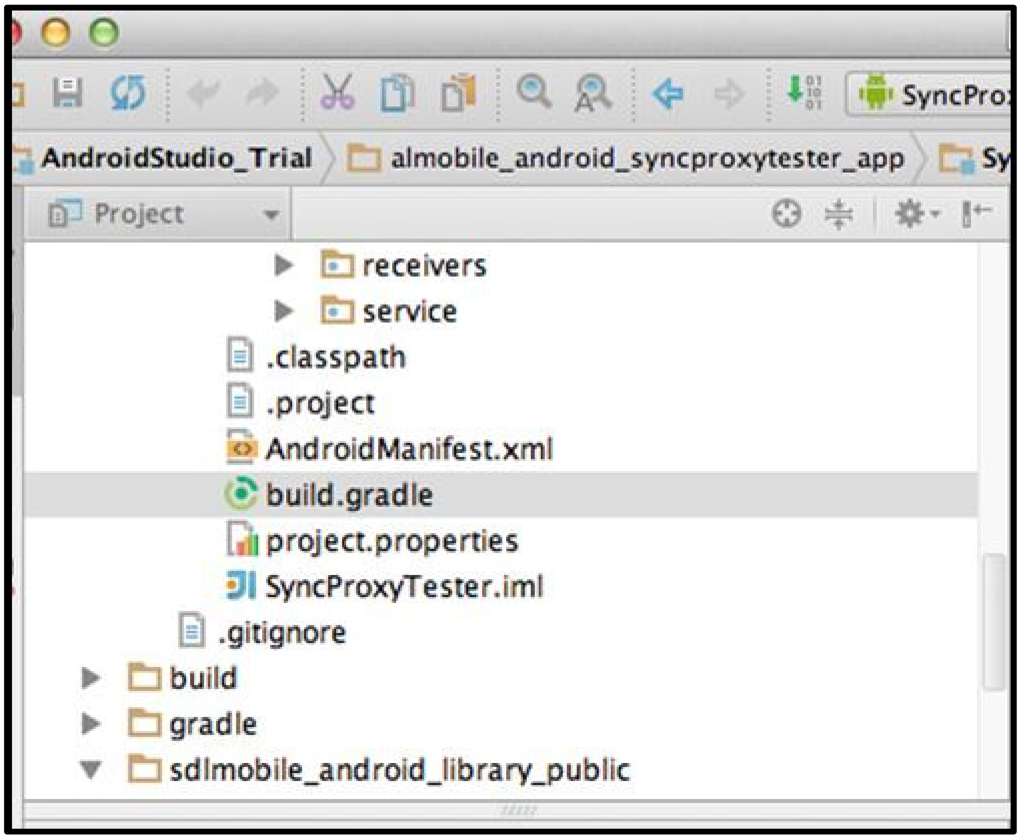
```
ge com.ford.s
t ...
c class SyncProxyTester extends FragmentActivity
```

58. You can now upload your build. Under Build Generate Signed APK

59. You will be asked which model to use. Select your desired model. Then click NEXT.



60. You will now need to create a key. Fill out all the fields. And then click OK



61. Your APK will be generated under your app folder

62. Build.gradle in this file you can configure all the dependencies for the application. You will not need to use Eclipse once you have imported the application into Android Studio.