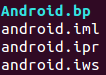
使用Android Studio進行runtime debug framework source code

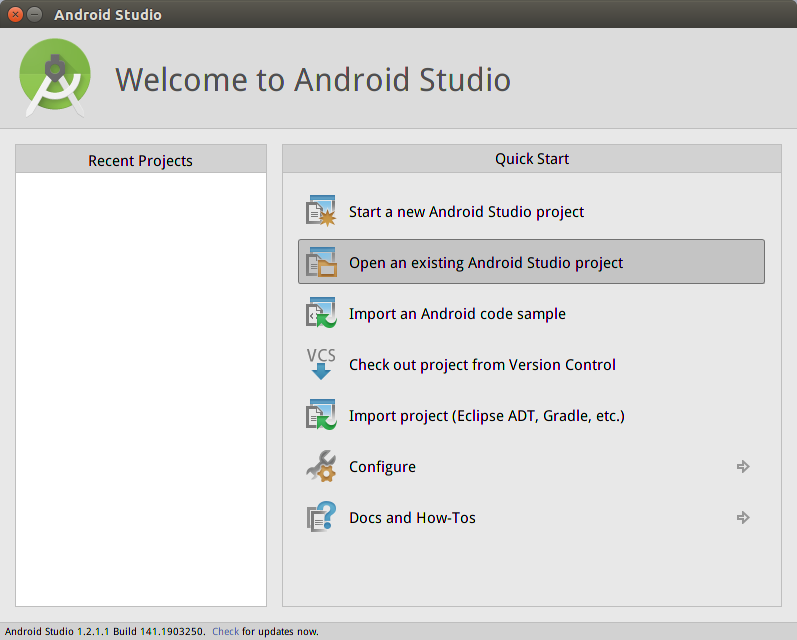
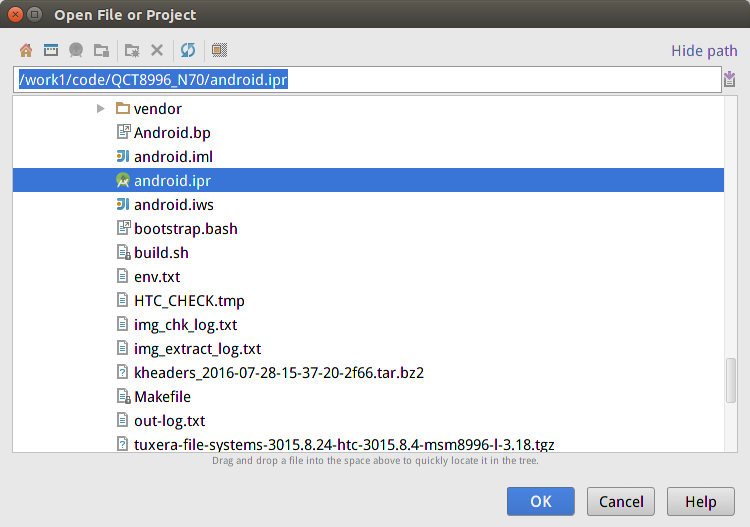
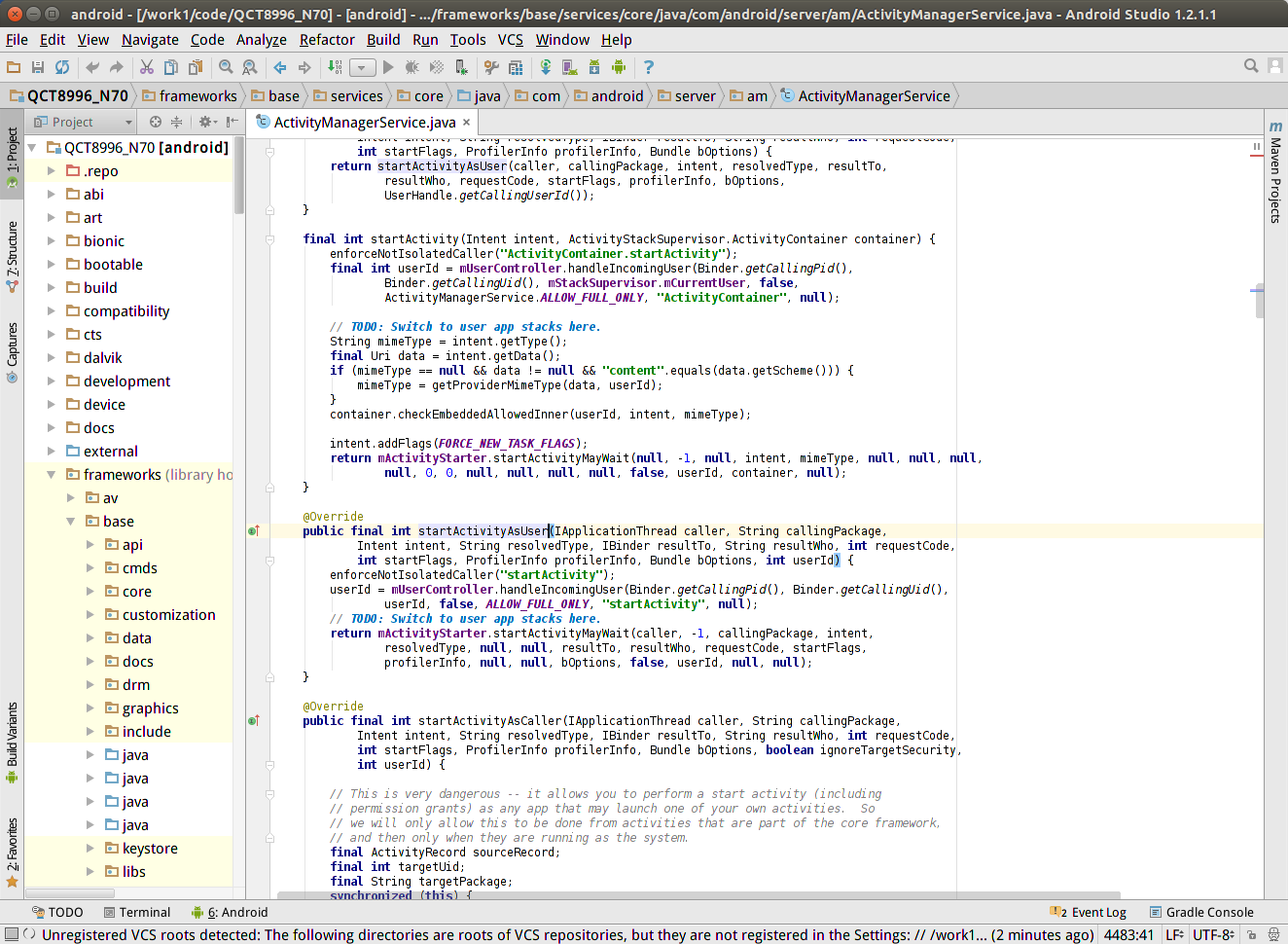
## 準備工作

本文檔示例使用的操作系統是Ubuntu 14.04, 使用到以下文件和工具:  
1. Android source code (本例使用的是QCT8996的master-n-rel分支code)  
2. Android Studio  
3. Android SDK  
4. 手機(ROM與code適配), 通過USB與PC連接  
  
將Android source code編譯完成(配置編譯環境需要有JDK等工具, 一般Ubuntu自帶), 再編譯idegen工具, 在code的root目錄下使用命令”make idegen”:  
  
編譯完成後會在out目錄下生成idegen相關的jar文件

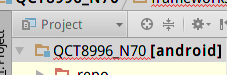
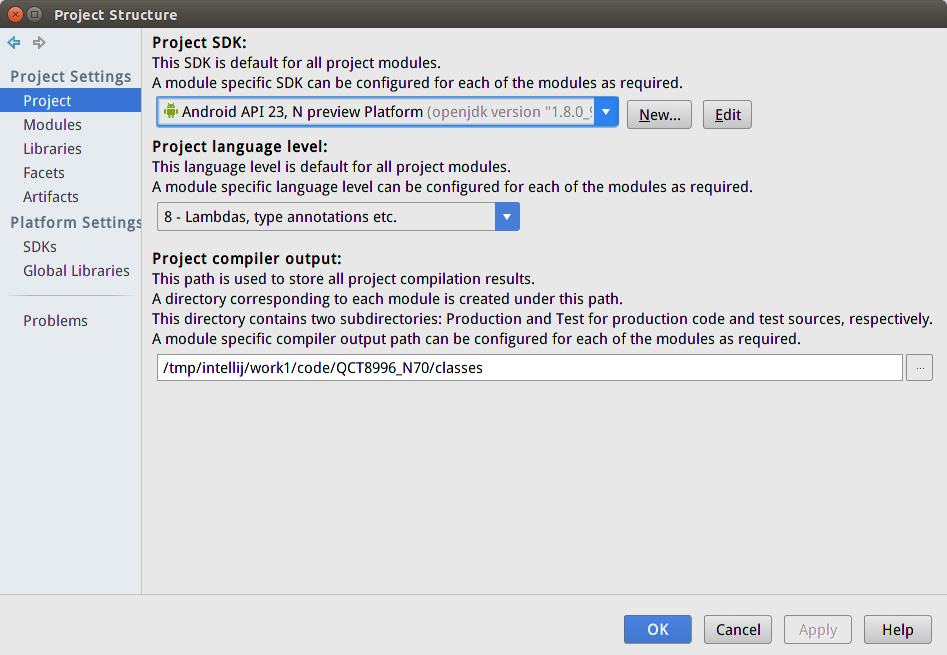
## 生成Project文件

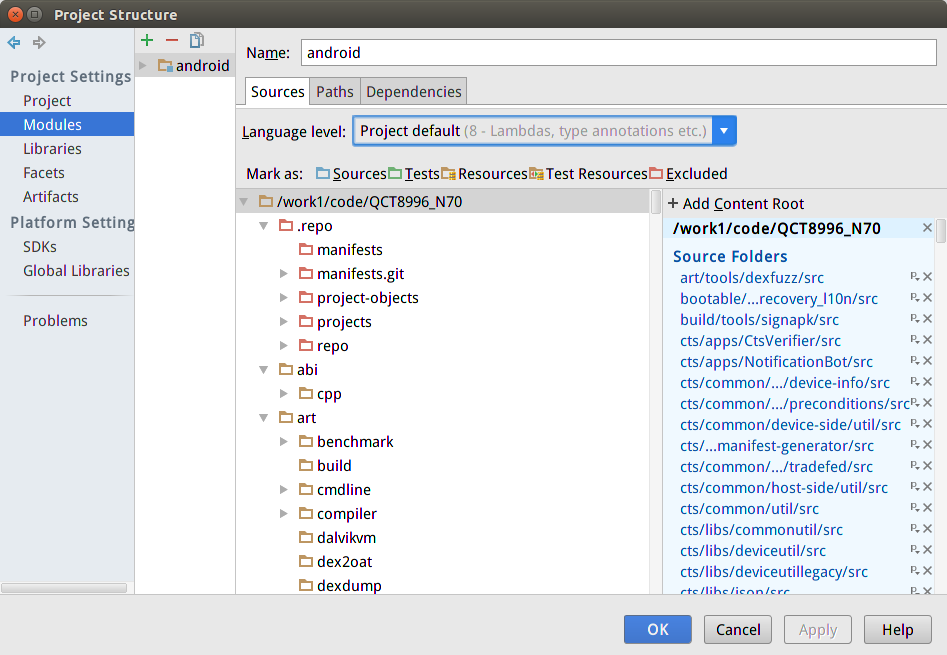
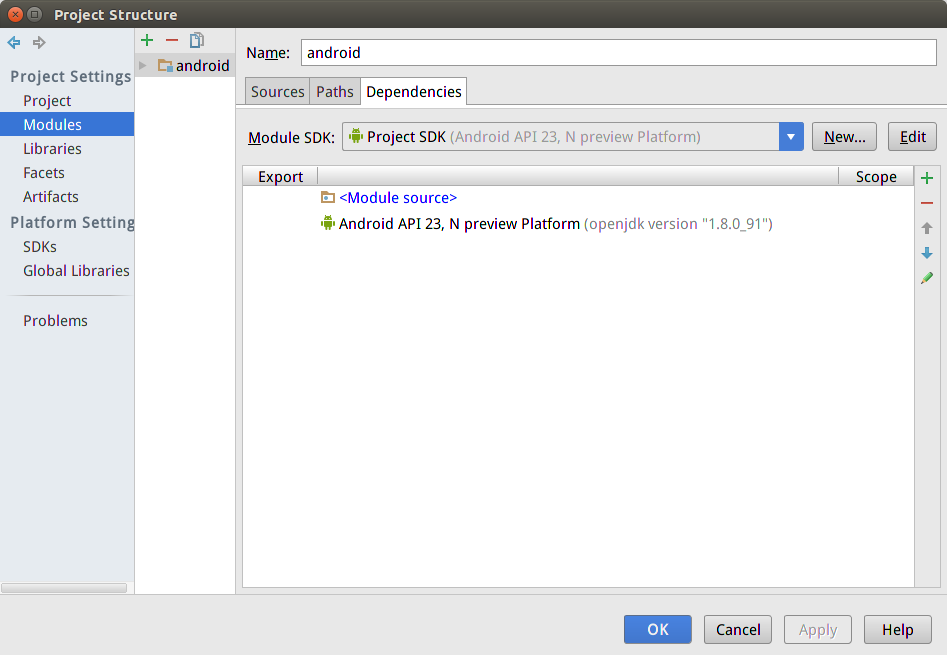
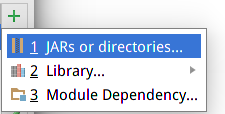
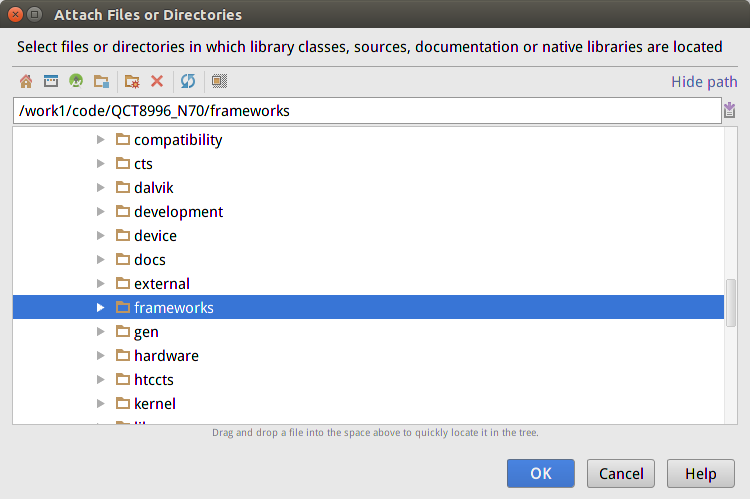
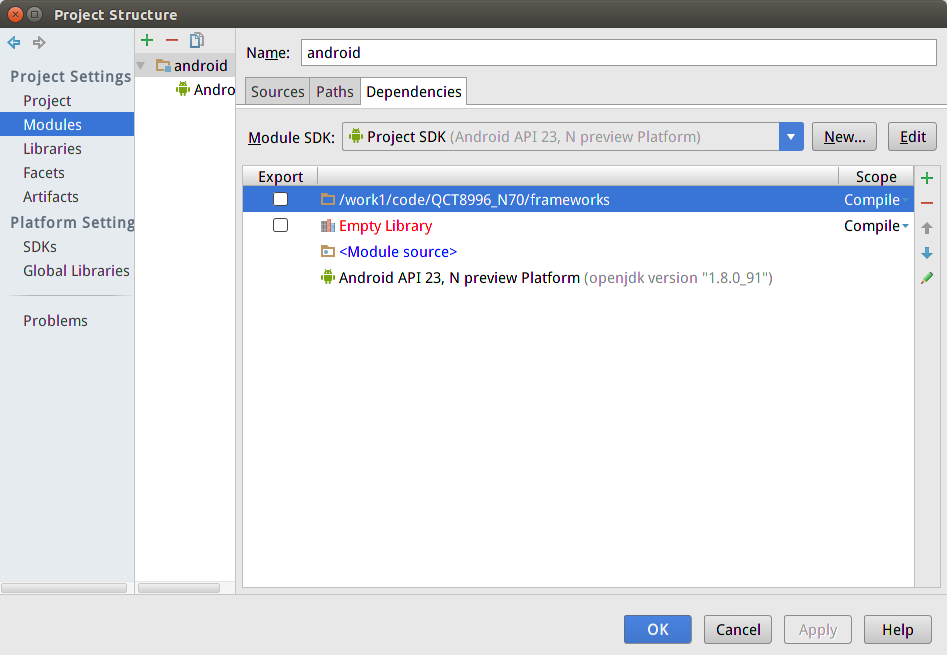
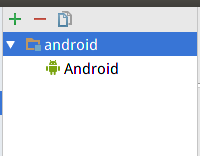
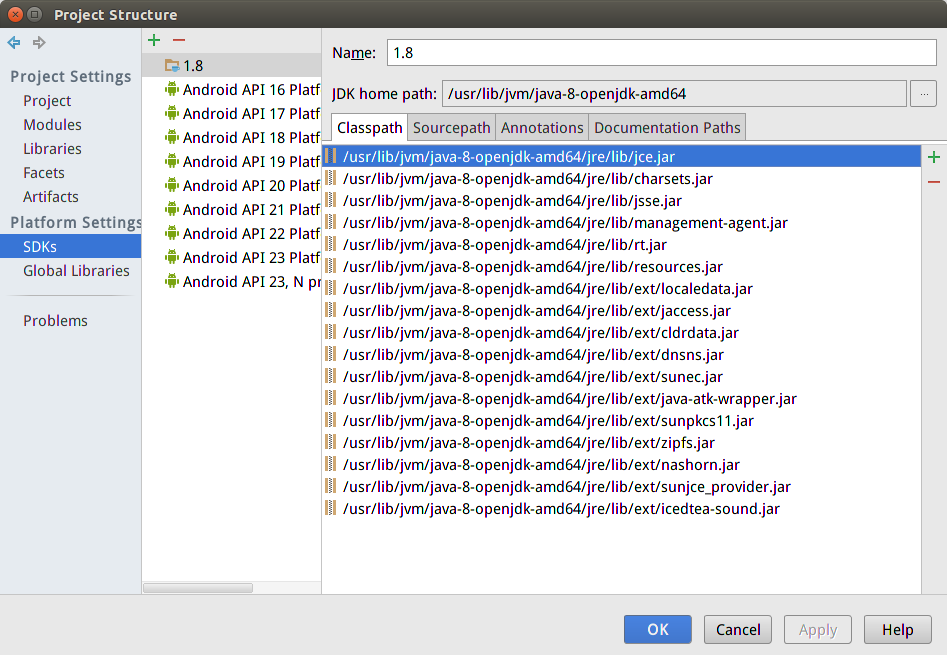
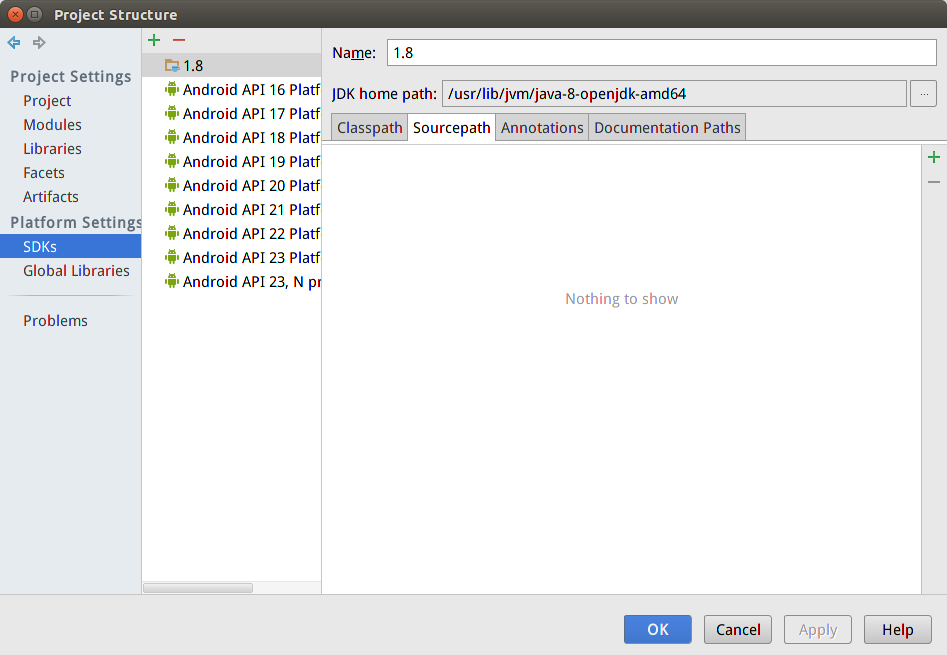
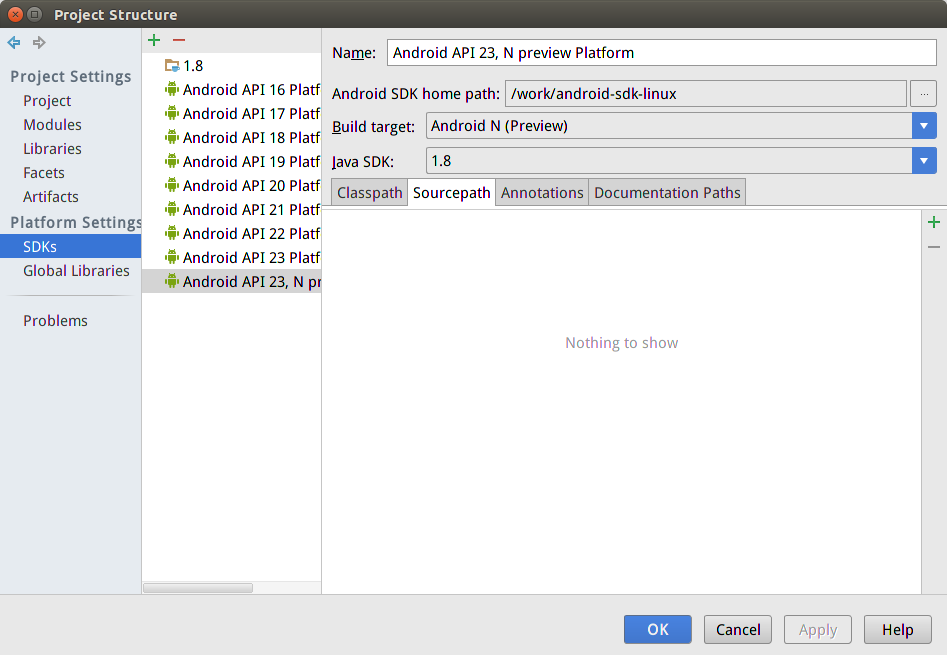
執行idegen.sh腳本來生成Android Studio使用的project文件, 在code的root目錄下使用命令” development/tools/idegen/idegen.sh”:  
  
在執行過程中可能會有些出錯信息, 但是對生成文件沒有影響:  
  
完成後會有如下信息出現:  
  
在code root目錄下可以找到idegen生成的project文件:  


## 打開Project文件

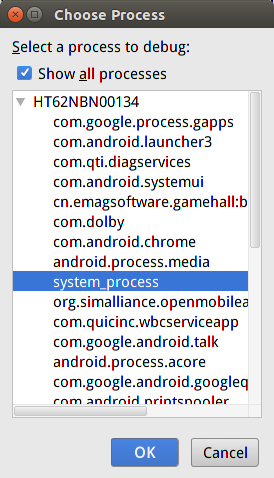
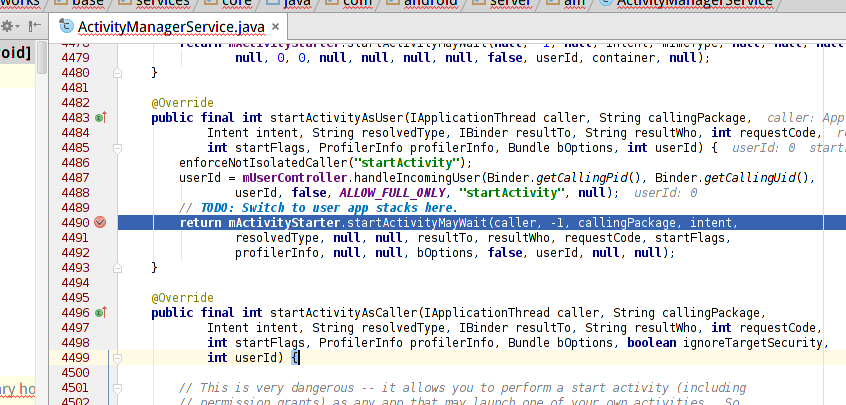
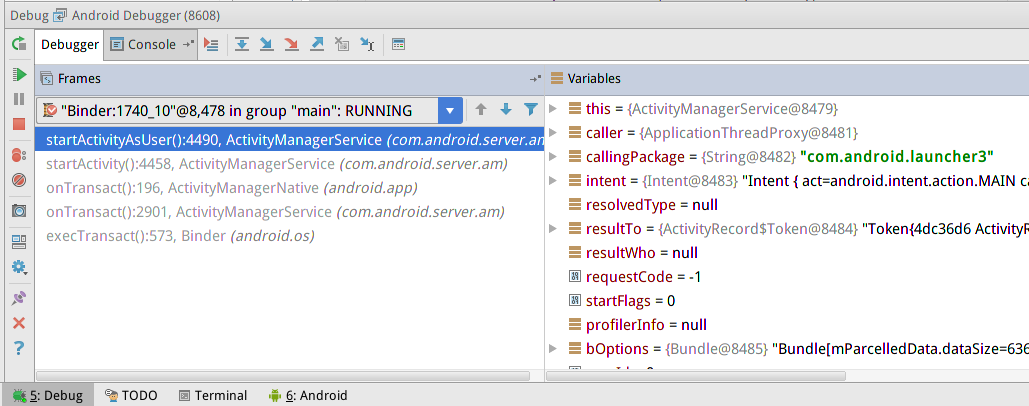
運行Android Studio, 第一次運行會進入設置嚮導, 需要手動指定Android SDK的路徑, 配置完成後會看到如下界面:  
  
選擇第二項”Open an existing Android Studio project”, 在彈出的窗口中找到Android source code路徑, 選擇android.ipr文件, 並點擊OK:  
  
第一次打開會loading較長時間, 完成後就可以看到所有的source code都在project中了, 可以很方便的查看(打開project時Android Studio可能會有一些提示, 可以忽略):  
  
第一次打開project會創建code索引, 可能花費較長的時間, 索引完成後就可以在code中進行跳轉了.

## 配置Project

右鍵點擊project的名稱”QCT8996\_N70”, 選擇”Open Module Settings”:  
  
可以看到打開了此project的Setting界面, 下面開始逐一設置:  


如上圖所示, 首先需要指定此project對應的SDK version和language level, 本示例用的是Android N的project, 所以分別選擇Android N API 23和Java 1.8.  
然後點擊左側的”Modules”:  
  
如上圖, 在這裡可以看到被加入此project的source code, 然後切換到”Dependencie” tab, 可以在列表中看到很多jar文件, 為了DEBUG能正確的跳轉到code而非jar文件, 將所有的jar文件從列表中移除(如需要可以保留所需的jar文件或手動添加), 選中需要刪除的jar文件, 點擊右側的減號-圖標刪除, 完成後如下圖:  
  
然後點擊右側的加號+圖標, 選擇”JARs or directories…”  
  
然後在彈出的窗口中選擇需要加入到依賴關係的code並點擊OK, 用來在code跳轉時使用:  
  
這裡示例中分兩次添加了frameworks和vendor兩個目錄, 可以按照需求添加更多的目錄, 完成後如下圖:  
  
這裡vendor目錄顯示為”Empty Library”, 不過沒有影響.  
另外在中間一欄, 本例的project打開時就有一個名為”android”的子module, 如果沒有這個子module, 可以點擊綠色的加號+添加, 告知IDE此project是Android project, 用來debug. 如下圖:  
  
接下來在左側點擊”SDKs” tab進入到SDK設置頁面:  
  
可以看到JDK和Android SDK在中間的列表中, 為了編譯Android N, 這邊用的JDK是1.8版本, 為了在DEBUG 時能正確進入Android source code中的Java lib而非JDK的Java lib, 這裡在第二個”Sourcepath” tab中, 把JDK的source路徑刪除掉, 如下圖:  
  
另外為了在DEBUG時能正確進入Android source code中的method而非SDK的method, 我們選擇對應的SDK版本(本例用的是Android API 23, N), 將”Sourcepath”中的路徑刪除, 效果如下圖:  
  
至此Project的設置完成, 點擊OK關閉.

## 進行DEBUG

之前的準備工作完成並且project的code索引也完成的話, DEBUG功能應該是可用的, 下圖所示的圖標會亮起並可以點擊:  
  
通過USB連接手機, 在ADB可以正常進入手機的情況下, 點擊DEBUG圖標, Android Studio會通過adb連接手機, 出現如下process選擇框:  
  
這裡選擇system\_process以DEBUG system\_server, 點擊OK後, 開始DEBUG.  
這時可以在code上添加斷點(break point), 在code的前面空白部分點擊, 生成一個紅色圓形標記, 表明成功的添加了一個斷點, 如下圖:  
  
這裡示例DEBUG Activity的啟動流程, 在startActivityAsUser的mActivityStarter.startActivityMayWait處加入一個斷點, 然後在手機上操作, 從Launcher上點擊phone icon, 嘗試進入Phone app, 這時會命中(hit)斷點, 此行code變成高亮, 手機會在此流程上暫停, 無法進行其他操作, 在DEBUG界面的下方可以看到當前的call stack和variables信息:  
  
這時可以點擊中間的按鈕(或者使用快捷鍵)進行單步(Step)調試, 以跟踪code流程:  
  
常用的功能有:  
Step Over(F8): 單步執行到下一條Code  
Step Into(F7): 單步執行進入本條Code內部(不包含Java lib的Method)  
Force Step Into(Alt + Shift + F7): 單步執行進入本條Code內部(進入所有Method)  
Step Out(F8): 執行到跳出本Method  
Run to Cursor(Alt + F9): 執行到Cursor處  
Resume Program(F9): 不再單步執行, 恢復program(直到下一次命中斷點)

## 總結

**以上為使用Android Studio在Android source code中進行runtime debug的基本方法, 使用這種工具對於分析issue和了解code邏輯有很大的幫助.**