

Ex4: Remote Debugging

Overview

Often we need or want to debug software on a remote host via a network debugger. Android (and Reptilian) has a built-in remote debugging system to facilitate development at the Java and native (C/C++) layers. In this project, you will use the network debugging system to push and test a simple Android application on Reptilian. The skills learned will be used in upcoming projects. You will take screenshots of app's results. You'll submit screenshots as well your source file on Canvas.

Structure

The exercise can be completed as follows:

- 1) Install Android Studio 4.0.1 (or later), SDK, and NDK
- 2) Add functionality to the skeleton GUI project provided (see specification below)
- 3) Build and remotely test (via Android remote debugging / LLDB) application
- 4) Submit your source file and your screenshots on Canvas.

Reptilian has remote debugging built in, but in order to connect to this system, the VM must port-forward to the local host. In the latest version of Reptilian, this forwarding is already set up for you.

Android Studio

Android Studio is set up to quickly and easily connect to the remote debugging system on Android devices. In this exercise, you'll install Android Studio 4.0.1, the Software Development Kit (SDK), the Native Development Kit (NDK), and several tools.

You can download Android Studio from here: <https://developer.android.com/studio/index.html>. You do not need to install the "Android Virtual Device" (though you may if you'd like). On the first run, you'll be prompted to download and install elements of the SDK and support tools. **This will take some time!**

Once Android Studio starts, in top ribbon menu, do the following to install the rest of the necessary software:

- 1) Click Tools → SDK Manager
- 2) Under "SDK Platforms", add "Android 9.0 (Pie)"
- 3) Under "SDK Tools", add "CMake", and "NDK".
- 4) Click "Apply" to install the added software.

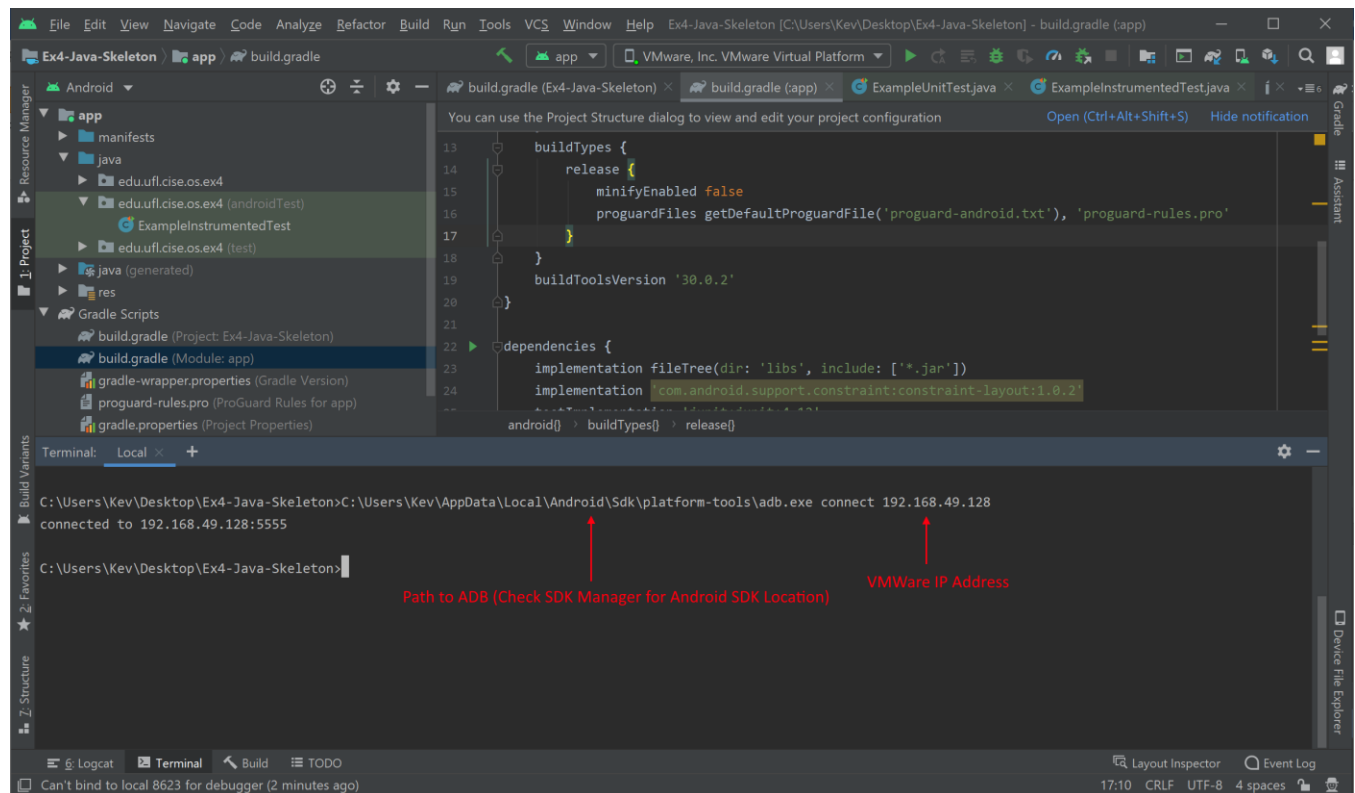
To connect Android Studio to Reptilian, you will need to connect VMware to Android Studio's ADB instance. Navigate to Android Studio's terminal and run the following command:

```
$ [Path to ADB] connect [IP Address]
```

For example, your path to ADB may look like the following:

```
$ C:\Users\[Username]\AppData\Local\Android\Sdk\platform-tools\adb.exe
```

The ADB executable can generally be found in the path to the Android SDK Location, under “platform-tools”.



Afterwards, your Reptilian instance will appear as a device labeled as “VMWare, Inc. VMWare Virtual Platform”.

Application

A skeleton Java project is provided for you to work with. We have also included a Kotlin project (a new “blessed” Android language), but please know that this is just for fun – it **is not supported**, so TAs and the instructor may not be able to help you with problems. (We’ll try though!)

The skeleton project includes two text boxes – **leftText** and **rightText** – and two buttons – **leftButton** (labelled “Copy to Right”) and **rightButton** (labelled “Copy to Left”).

You must add code to the skeleton project so that, when **leftButton** is clicked, the text in **leftText** is copied to **rightText**, and when **rightButton** is clicked, **rightText** is copied to **leftText**. You should then

test your application on Reptilian (via the Android Studio debugger) and take a screenshot of the application running in debug mode with both the Reptilian GUI and Android Studio debugger running.

Submissions

You will submit the following at the end of this exercise on Canvas:

- Kotlin or Java source file for Ex4Activity
- Screenshot of Ex4 running on Reptilian GUI & the Android Studio debugger stopped at a breakpoint set somewhere in your code.

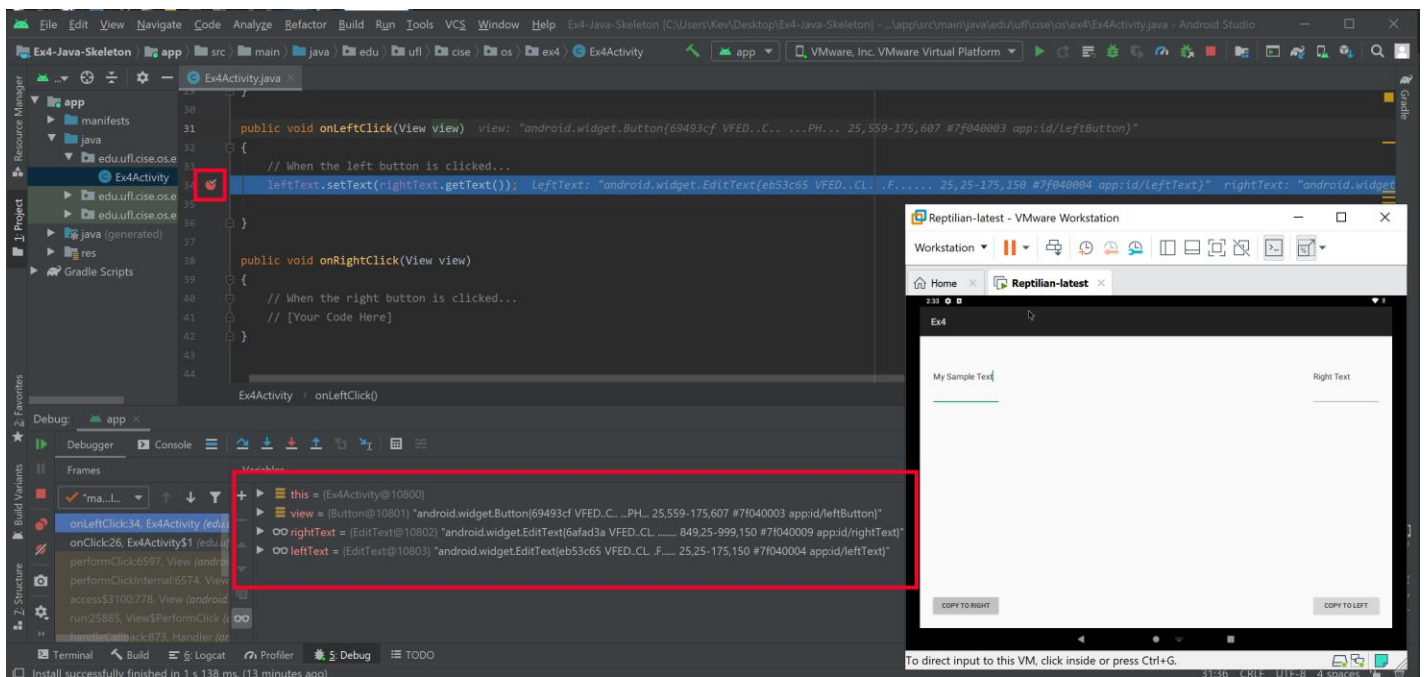


Figure 1: An example screenshot depicting everything that should be present in a suitable submission.