

How to make a Release Android App debuggable

Nicholas Starke | <https://twitter.com/nstarke> | <https://github.com/nstarke>

Let's say you want to access the application shared preferences in `/data/data/com.mypackage`. You could try to run `adb shell` and then `run-as com.mypackage` (or `adb shell run-as com.mypackage ls /data/data/com.mypackage/shared_prefs`), but on a production release app downloaded from an app store you're most likely to see:

```
run-as: Package 'com.mypackage' is not debuggable
```

While ADB's `adb backup` command will pull many of the files from the application's data directory, the files can be inconsistent with what is actually on the device. If you want to see what is actually on the device, you need to make the application debuggable.

For this task, we'll need `apktool` (<http://ibotpeaches.github.io/Apktool/>). Once you have it setup, you'll need to find the path to the APK to pull it off the device. Run:

```
$ adb shell pm list packages -f -3
```

Find the package in the list (it should look something like `/data/app/com.mypackage.apk=com.mypackage`), and pull it off the device:

```
$ adb pull /data/app/com.mypackage.apk
```

Next, we need to disassemble the apk using `apktool`:

```
$ apktool d -o output-dir com.mypackage.apk
```

In `output-dir`, find `AndroidManifest.xml` and open it up in the text editor of your choice. In the `application` xml node, add the following xml attribute:

```
android:debuggable="true"
```

Now we need to reassemble the application. We do this by running:

```
$ apktool b -o com.mypackage.apk output-dir
```

Finally, we need to resign the APK so that the device will accept it:

```
$ keytool -genkey -v -keystore resign.keystore -alias alias_name -keyalg  
RSA -keysize 2048 -validity 10000  
$ jarsigner -verbose -sigalg SHA1withRSA -digestalg SHA1 -keystore  
resign.keystore com.mypackage.apk alias_name
```

That's it! You should now be able to transfer the new com.mypackage.apk back to the device and run it. Now you should be able to run `adb shell run-as ls /data/data/com.mypackage/` without getting the debuggable error.