**Android Studio ADB Commands**

To see if device or how many are connected to Android studio

adb devices

*if device connected but you see* "Unauthorized"

GO to Developers Options

"Revoke USB Authorization" -->tap once-->confirm

*Turn off Developers Options Menu*

*Turn ON* (don’t forget to enable USB Debugging Box again.

Sometimes you need USB out/in

Trust this computer if prompted, enable box

Take your USB cable out, and put it back right away

run

adb devices  -l ( with all info)

**You may use ( -e, -d, -s whenever you feel ) to separate devices in your command line.**

**Examples are strictly my choice**

emulator-5554 ( -e )

emulator-5556 ( -s )

adb –s emulator-5556

2424242 real device ( -d)

2425252 real device ( -s)

adb -s 2425252

if more than 2

Capture a bug report using adb

If you have just one device connected, you can get a bugreport using [adb](https://developer.android.com/studio/command-line/adb.html) as follows:

$ adb bugreport E:\Reports\MyBugReports ( or any other directory on your PC/Laptop )

If you do not specify a path for the bugreport, it is saved to the local directory.

If you have multiple devices connected, you must specify the device with the -s option. Run the following adbcommands to get the device serial number and generate the bug report.

$ adb devices  
List of devices attached  
emulator-5554      device (-e)  
8XV7N15C31003476 device (-d)

$ adb -d bugreport (or)  
$ adb -s 8XV7N15C31003476 bugreport

To send this report to your PC/Laptop

$ adb -d bugreport C:\a\

or

$ adb -s 8XV7N15C31003476 bugreport C:\a\

Check the directory and open Bug report

Getting Bug Report from an emulator.

*Be aware that you will be able to run adb –e bugreport , but you wouldn’t be able to send it to PC/Laptop directory unless*

*you are using OS 7.0 or higher*

$ adb -e bugreport C:\a\

Inspect the bug report ZIP file

By default the ZIP file is called bugreport-***BUILD\_ID***-***DATE***.zip and it may contain multiple files, but the most important file is bugreport-***BUILD\_ID***-***DATE***.txt. This is the bug report and it contains diagnostic output for system services (dumpsys), error logs (dumpstate), and system message logs (logcat). The system messages include stack traces when the device throws an error, and messages written from all apps with the [Log](https://developer.android.com/reference/android/util/Log.html) class.

The ZIP file contains a version.txt metadata file that contains the Android release letter, and when systrace is enabled, the ZIP file also contains a systrace.txt file. The [Systrace tool](https://developer.android.com/studio/profile/systrace-commandline.html) helps analyze the performance of your application by capturing and displaying execution times of your application processes and other Android system processes.

Capture a bug report from the Android Emulator

From the Android Emulator, you can use the **File a bug** feature in the extended controls:

1. Click **More** https://developer.android.com/studio/images/buttons/emulator-extended-controls.png in the emulator panel. ( VERY COOL TOOLS are placed in here ☺ )
2. In the **Extended controls** window, select **Bug Report** on the left.

This opens a screen where you can see the bug report details such as the screenshot, the AVD configuration info, and the bug report log. You can also type a message with reproduction steps to save with the report.

1. Wait for the bug report to finish collecting, and then click **Save Report**.

Let’s play with Other TOOLS

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

Get info about your device:

adb –e shell getprop ro.product.model *( for real device –d )*

adb –e shell getprop ro.product.device

adb –e shell getprop ro.build.version.release

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

Lets install manually an app that is not in a Market Place (from APK folder on your Google Drive ( see instructions and task )

adb –d install “com.united.mobile.apk”

 ( or to -e –emulator )this name is ONLY valid during an installation....Forget about this . *You can also rename it to anything like baby.apk - doesn’t matter. Because app information is stored in .apk and not in the name itself)*

Install APP if multiple devices attached: you may use -d or -e

In this document some of the devices are separated by serial number or -d, or -e.

adb -s emulator-5554 install com.united.mobile.apk ( or “com.united.mobile.apk” )

adb -s yourdevice#  install com.united.mobile.apk ( or “com.united.mobile.apk” )

*How to get package info:*

adb shell ( or adb –d or adb –e, or adb –s yourdevice# )

adb –d shell ( on my side, because I have two devices – real and emulator )

pm list packages  ( list all packages on your device)

pm list packages -f  ( will give you a full path where all the above packages being located)

pm list packages -f | grep united    ( you will get a package name for United APP )

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

now you have two devices .

adb devices

emulator-5554   device

6703cab3        device ( this is serial# for my device – yours going to be different.

adb -s emulator-5554 shell getprop ro.build.version.release

adb -s 6703cab3 shell getprop ro.build.version.release

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

adb -s emulator-5554 shell

pm list packages -f | grep united    ( you will get a package name for United APP )

E:\AndroidStudio\android-sdk-windows\platform-tools>adb -s emulator-5554 shell

root@generic\_x86:/ # pm list packages -f | grep united

package:/data/app/com.united.mobile.android-1/base.apk=com.united.mobile.android

for real device :

adb -s 6703cab3 shell

shell@d2spr:/ $ pm list packages -f | grep united

/data/app/com.united.mobile.android-1.apk=com.united.mobile.android

How to make your device REMOTE (real device)?

1. Set your Device and Laptop/PC to the SAME network
2. Check on your Device : Settings🡪about device🡪status ( write down )

This is a manual way to figure out what your IP is

But you may use adb commands to do the same.

adb -d shell netcfg

*( mine is 192.168.4.238 – please note your IP addr is different from mine, but I’m using my own numbers. Please adjust your command line to match your IP )*

or

adb -d shell ip addr

1. adb -d usb ( restarting in usb mode)

2. adb -d tcpip 5556 ( assigning port 5556 )

( checking if this port is available to use. Receive message "restarting in TCP mode port: 5556")

3. adb -d connect 192.168.4.243:5556  ( "marry” your IP address to the port 5556)

After receiving a message "connected to 192.168.4.243:5556"

Disconnect USB cable from computer and type:  adb devices

Your Device is now listed:

192.168.4.243:5556      device

Lets uninstall our APP from remote device ( you may use any APP )

adb -s 192.168.4.243:5556 uninstall “com.united.mobile.android”

and Install it back:

adb -s 192.168.4.243:5556 install “com.united.mobile.apk”

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

adb -s 192.168.4.243:5556 reboot

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

How to make a device to be real again

adb -s 192.168.4.243:5556 reboot

Connect device back to the PC/Laptop and run

adb devices

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx