ADB Survival Guide

This ADB Survival Guide is offered by http://www.all4android.net/ .

Android Debug Brigde (ADB) is a command line tool, coming with Android SDK provided by Google, that lets developers to communicate with an emulator or a connected Android device directly from the command line. To find adb tool, you must go on <android_sdk_path>/platform-tools directory on your computer.

In this article, you're going to learn the handy adb commands which are essential to know when you want to develop Android applications. So, you can consider this as your ADB Survival Guide !

Summary

List connected device(s)

First thing to do is to list connected device(s) by launching following command :

adb devices

Result shows us that an emulator is actually running which has emulator-5554 as id.

```
D:\tmp\personnel\mobile\android-sdk-windows\platform-tools>adb devices
List of devices attached
emulator-5554   device
```

Upload a specified file on device

To move a file from your computer to a running Android device via the command line, adb offers you the push command that must be used like that :

adb push myMusicFile.mp4 /sdcard/music

Download a specified file from your device

As push adb command works to upload a specified file, pull adb command must be used to download a specified file from your device to your computer :

adb pull /sdcard/music/myMusicFile.mp4

Apply command to a specific device

In our example, just one Android device is connected. But, in some cases, if you have several connected devices, you could define the device that you want to use to execute the adb command. Enter the following command :

adb -s emulator-5554 push myMusicFile.mp4 /sdcard/music

Reboot a device

You can even reboot a device directly from the command line like that :

adb reboot

Reboot a device to the bootloader

Even better, you can also reboot a device to the bootloader :

adb reboot-bootloader

Install an application

Once you have generated an application APK, you can install this on the Android device of your choice thanks to the install adb command. Note that you can use the optional -r argument to reinstall and kept any data if application is already installed on the device.

adb install -r myApplication.apk

Uninstall an application

To uninstall an application, you must use the uninstall adb command like that :

adb uninstall com.ssaurel.myapplication

Note here that com.ssaurel.myapplication is the package name of application you want to uninstall from device.

Start an activity

To start an activity, you must use the shell adb command that lets you to execute specific command on the device. Imagine that you want start main activity of the application com.ssaurel.myapplication. You must enter following command line :

adb shell am start -n com.ssaurel.myapplication/.MainActivity

Here, you enter first the package name of the application and then the activity to start. Note that you can use a fully qualified activity like that :

adb shell am start -n com.ssaurel.myapplication/com.ssaurel.myapplication.MainActivity

Entering in device's shell

To execute specific commands on your device, you can use adh shell followed by the

command name and wait for the result. Second solution, that you should use if you have several commands to execute, is to enter directly in device's shell with that command :

adb shell

Take a screenshot

Another great feature offered by adb is that it's able to take screenshot of your device's display. Great to automate screenshots via scripts. To take a simple screenshot, execute following command :

adb shell screencap /sdcard/myScreenshot.png

To download the screenshot on your computer, you can use the following command :

adb pull /sdcard/myScreenshot.png

Record device display

Even better, since Android KitKat 4.4 you can record device display. Great to make demo video of an application for example. To achieve that, you must use screenrecord adb shell command. It accepts a lot of options like possibility to define a specific size, a bitrate or a time limit (default value is 120 seconds).

To record a 2 minutes device display, you can execute following command:

adb shell screenrecord --time-limit 120 /sdcard/myVideo.mp4

Then, get the recorded video thanks to adb pull command.

Power button

You can also emulate input on your device. For example, you can thus emulate power button event to turn the device on or off like that:

adb shell input keyevent 26

Unlock screen

With previous command, you can then unlock screen when you know keyevent number :

adb shell input keyevent 82

Besides, you can combine these 2 key events actions to turn off a device then to turn on and unlock screen.

List installed packages

Other possibility is to list all installed packages on a device :

adb shell pm list packages -f

Logging

To know what your device is making, you can print log data to the screen :

adb logcat

Content is automatically refreshed. To stop monitoring, press CTRL-C on your keyboard. Some options are available to filter content by priority level for example :

adb logcat ":W"*

Here, we choose to display warning priority level. You can also filter by tagname and priority like that :

adb logcat -s MY_TAG: W

You can also search specific content in logcat buffer thanks to -b option :

adb logcat -b radio

Don’t forget that you can use grep command to filter content also :

adb logcat | grep "com.ssaurel"

Finally, you can clear the logcat buffer :

adb logcat -c

Dump system data

To analyze some speficic problems, you can dump system data. Directly on the screen or in a defined file. To dump on the screen, enter following command line :

adb shell dumpsys

Besides, you can dump data for a specific service like battery :

adb shell dumpsys battery

Result can be seen below :

If you prefer dump system data in a file, enter following command :

adb shell dumpstate -o /sdcard/myDump.txt

Execute classic Linux / Unix commands

As Android is based on Linux system, you can also take profit to execute classic commands like ls, cat, etc … Thus, you can get CPU info by displaying content of /proc/cpuinfo :

adb shell cat /proc/cpuinfo

This ADB Survival Guide is over. ADB Tool offers you a lof of possibilities. Don't hesitate to go further by reading official ADB documentation here :

http://developer.android.com/tools/help/adb.html .

This ADB Survival Guide is offered by www.all4android.net . Get more articles and tutorials on Android on All 4 Android.