Is it possible to USB tether an android device using adb through the terminal?

Asked 6 years, 11 months ago Active 1 month ago Viewed 21k times

12

6

I'm setting up some tests and it will require a decent number of phones to be usb tethered and configured. I've been successful in configuring them the way I want to once they have been tethered, however it would be quite tedious to tether the phones through navigating the menus, each and every time I (re)start my computer or move the test bank. I am currently using Nexus S phones running cyanogenmod v10.1.0, however the test bank will likely be Samsung Galaxy S4's possibly mixed with the few Nexus S phones I have on hand.

I want to do this as a bash script, but I'm trying to get it work at the command line (Ubuntu 13.04) first so as to remove issues that could come from scripting. I should be able to handle making it into a script myself, but if it's simple to provide an answer as bash script, please do. I tried shelling into the device (`adb -s $deviceID shell`) and running:

```
setprop sys.usb.config rndis,adb
```

This promptly kicks me out of the device shell and the device is no longer accessible. If I run an `adb devices` I see the phone as "?????????? No Permissions" at which point I have to take the USB cable out and then plug it in again, and also restart the adb server with `adb kill-server` `adb start-server` . This will not work because I cannot access the phone to make the configuration changes that I need.

I've googled around but have been unable to find anything fruitful. Any suggestions?

android bash ubuntu adb tethering

asked Nov 26 '13 at 20:07

turbo

1,847 2 21 35

duplicate of android.stackexchange.com/questions/29954/… – eadmaster Feb 12 '19 at 0:37

7 Answers

Active | Oldest | Votes

19

Must have root to change values with `setprop` , and I am on a Mac OS without a rndis driver so I could not test your method of USB tethering. Another way, if you have the connectivity service (`adb shell service list`):

The following commands call `ConnectivityManager.setUsbTethering(boolean enable)` in Android 4.3:

`adb shell su -c service call connectivity 34 i32 1` turns on USB tethering.

`adb shell su -c service call connectivity 34 i32 0` turns off USB tethering.

For other Android versions replace `34` with the following `setUsbTethering` calling codes per Android version:

```
4.4.4: 34
5.1.0: 30
6.0.1: 30
7.0.0: 33
```

edited Oct 31 '16 at 13:06
Alex P.
25.8k 16 98 150

answered Jun 21 '14 at 21:40
PRNDL Development Studios
3,202 1 31 49

2 For reference, this only works on Android 4.x (more likely, only specific versions). The number "34" is the method number in the list of methods in `IConnectivityManager.aidl` – Steve Pomeroy Jun 29 '14 at 16:35

For reference, RNDIS seems to be the same as USB, but it depends on what device you are using. It was RNDIS for the nexus s and USB for the galaxy s4. Unfortunately I no longer work on this project so I can't test your solution, but it seems that it would work so I have accepted your answer. – turbo Sep 2 '14 at 14:36

2 This only works on Android 4.3 and 4.4. – Léo Lam Feb 14 '15 at 14:37

6

For Android 5.0+ (Lollipop, Marshmallow) use:

`adb shell su -c service call connectivity 30 i32 1` to turn USB Tethering ON

`adb shell su -c service call connectivity 30 i32 0` to turn USB Tethering OFF

Keep in mind that this requires root.

answered Dec 22 '15 at 15:50

Where is it documented? – deFreitas Sep 11 '16 at 3:31

1 Use "service list" and "dumpsys activity services" in adb to get a overview of services your device is aware of. From here on its either Google or browsing through the (AOSP) source code since non of this is "officially" documented as far as i know. – user1972814 Sep 15 '16 at 14:19

6

Commands in accepted answer not work on Oreo because now should be additional parameter `callerPkg` and if put there some random text it works.

```
int setUsbTethering(boolean enable, String callerPkg);
```

So, for 8.0 / 8.1 Oreo:

`service call connectivity 34 i32 1 s16 text` - turn USB tethering ON

`service call connectivity 34 i32 0 s16 text` - turn USB tethering OFF

It works for me Android Pie with

`service call connectivity 33 i32 1 s16 text` - turn USB tethering ON

`service call connectivity 33 i32 0 s16 text` - turn USB tethering OFF

edited Feb 18 '19 at 7:10
user1868097
3 2

answered Aug 29 '18 at 10:01
James Hunt
61 1 3

5

You can also script the inputs to start the Settings app and tick the checkbox, like in https://github.com/medvid/android-tether/blob/master/tether#L83.

Here's my script (pretty much the same as in the link, but slightly adapted):

```
adb shell am force-stop com.android.settings
adb shell input keyevent 3 # Home
sleep 2
adb shell am start -a android.intent.action.MAIN -n
com.android.settings/.TetherSettings
sleep 2
adb shell input keyevent 19 # Up
adb shell input keyevent 20 # Down
adb shell input keyevent 66 # Enter
sleep 2
adb shell input keyevent 3 # Home
```

For Windows, just replace `sleep` with `timeout -t`.

Works fine for my OnePlus 3T running Android Pie (9) (with Google's Settings app (running the Pixel Experience ROM); can't verify if it works with other Settings apps or not)

edited Dec 24 '19 at 5:35
Community ♦
1 1

answered Nov 24 '18 at 15:15
22samuelk
658 1 7 18

1

The `service` method did not work for me on my Samsung device. I figured out how to do it by configuring the network interface directly, though. Here is a script that sets up a Linux machine and a USB-connected rooted Android device for USB tethering. This does not set up DNS or NAT masquerading, but is sufficient to make the device accessible at 192.168.42.129:

```
#!/bin/bash
set -euo pipefail

# Set up USB tethering for an Android device.
# Usage: adb-usb-tether [USB-VENDOR USB-PRODUCT]
# If USB vendor/product is unspecified, use first USB network interface.
# On the Android side, tethering is enabled via adb shell.

if [[ $# -eq 2 ]]
then
    any=false
    vendor=$1
    product=$2
else
    any=true
fi

function find_if() {
    local path if
    for path in /sys/class/net/*
    do
        if=$(basename "$path")
        if [[ "$(readlink "$path")" == */usb* ]]
        then
            local ifproduct ifvendor
            ifproduct=$(cat "$(realpath "$path")/../../../idProduct")
            ifvendor=$(cat "$(realpath "$path")/../../../idVendor")
            if $any || [[ "$ifproduct" == "$product" && "$ifvendor" == "$vendor" ]]
            then
                echo "Found interface: $if" 1>&2
                echo "$if"
                return
            fi
        fi
    done
}

function adb_shell() {
    adb shell "$(printf " %q" "$@")"
}

function adb_su() {
    local quoted
    quoted="$(printf " %q" "$@")"
    adb shell su -c "$(printf %q "$quoted")"
}

if=$(find_if)
if [[ -z "$if" ]]
```

0

For Fairphone 2 with Fairphone Open OS (the "Android without Google" version, which is not installed by default) you need to:

- Enable Developer mode (may be actived by default)
- Search setting for "root" and enable root access for ADB
- Enter bash command in quotes and use service code 31:
  - Enable: `adb shell su -c "service call connectivity 31 i32 1"`
  - Disable: `adb shell su -c "service call connectivity 31 i32 0"`

answered Jun 1 '17 at 6:05

Jack Miller
4,013 1 30 41

0

Android 4.2 Jelly bean:

```
adb shell su -c service call connectivity 33 i32 1
```

answered Sep 18 at 5:23

Rayan
1