ADB Server

Restart ADB server

```
adb kill-server
adb start-server
```

Device Rebooting

Reboot the device

```
adb reboot
```

Reboot the device into recovery mode (for upgrading or downgrading devices)

```
adb reboot recovery
```

Device Management

```
adb shell input keyevent <key_code>
```

Where is a number or text, see below:

0 –> “KEYCODE_UNKNOWN”
1 –> “KEYCODE_MENU”
2 –> “KEYCODE_SOFT_RIGHT”
3 –> “KEYCODE_HOME”
4 –> “KEYCODE_BACK”
5 –> “KEYCODE_CALL”
6 –> “KEYCODE_ENDCALL”
7 –> “KEYCODE_0”
8 –> “KEYCODE_1”
9 –> “KEYCODE_2”
10 –> “KEYCODE_3”
11 –> “KEYCODE_4”
12 –> “KEYCODE_5”
13 –> “KEYCODE_6”
14 –> “KEYCODE_7”

15 –> “KEYCODE_8”
16 –> “KEYCODE_9”
17 –> “KEYCODE_STAR”
18 –> “KEYCODE_POUND”
19 –> “KEYCODE_DPAD_UP”
20 –> “KEYCODE_DPAD_DOWN”
21 –> “KEYCODE_DPAD_LEFT”
22 –> “KEYCODE_DPAD_RIGHT”
23 –> “KEYCODE_DPAD_CENTER”
24 –> “KEYCODE_VOLUME_UP”
25 –> “KEYCODE_VOLUME_DOWN”
26 –> “KEYCODE_POWER”
27 –> “KEYCODE_CAMERA”
28 –> “KEYCODE_CLEAR”
29 –> “KEYCODE_A”
30 –> “KEYCODE_B”
31 –> “KEYCODE_C”
32 –> “KEYCODE_D”
33 –> “KEYCODE_E”
34 –> “KEYCODE_F”
35 –> “KEYCODE_G”
36 –> “KEYCODE_H”
37 –> “KEYCODE_I”
38 –> “KEYCODE_J”
39 –> “KEYCODE_K”
40 –> “KEYCODE_L”
41 –> “KEYCODE_M”
42 –> “KEYCODE_N”
43 –> “KEYCODE_O”
44 –> “KEYCODE_P”
45 –> “KEYCODE_Q”
46 –> “KEYCODE_R”
47 –> “KEYCODE_S”
48 –> “KEYCODE_T”
49 –> “KEYCODE_U”
50 –> “KEYCODE_V”
51 –> “KEYCODE_W”
52 –> “KEYCODE_X”
53 –> “KEYCODE_Y”
54 –> “KEYCODE_Z”
55 –> “KEYCODE_COMMA”
56 –> “KEYCODE_PERIOD”

57 –> “KEYCODE_ALT_LEFT”
58 –> “KEYCODE_ALT_RIGHT”
59 –> “KEYCODE_SHIFT_LEFT”
60 –> “KEYCODE_SHIFT_RIGHT”
61 –> “KEYCODE_TAB”
62 –> “KEYCODE_SPACE”
63 –> “KEYCODE_SYM”
64 –> “KEYCODE_EXPLORER”
65 –> “KEYCODE_ENVELOPE”
66 –> “KEYCODE_ENTER”
67 –> “KEYCODE_DEL”
68 –> “KEYCODE_GRAVE”
69 –> “KEYCODE_MINUS”
70 –> “KEYCODE_EQUALS”
71 –> “KEYCODE_LEFT_BRACKET”
72 –> “KEYCODE_RIGHT_BRACKET”
73 –> “KEYCODE_BACKSLASH”
74 –> “KEYCODE_SEMICOLON”
75 –> “KEYCODE_APOSTROPHE”
76 –> “KEYCODE_SLASH”
77 –> “KEYCODE_AT”
78 –> “KEYCODE_NUM”
79 –> “KEYCODE_HEADSETHOOK”
80 –> “KEYCODE_FOCUS”
81 –> “KEYCODE_PLUS” 82 –> “KEYCODE_MENU”
83 –> “KEYCODE_NOTIFICATION”
84 –> “KEYCODE_SEARCH”
85 –> “TAG_LAST_KEYCODE”

Wake up the device

```
adb shell input keyevent KEYCODE_WAKEUP
```

Swipe (unlock screen)

```
adb shell input keyevent 82
```

Home button (KEYCODE_HOME)

```
adb shell input keyevent 3
```

Send a string of characters

Insert the text into a view with a focus (if it supports text input)

```
adb shell input text 'Paste text on Android device'
```

Simulate a physical action

Get Xpoint and Ypoint by enabling pointer location in the Developer Options – Input – Pointer location.

TAP

adb shell input tap Xpoint Ypoint

Example – Tap over position (487,428):

```
> adb shell input tap 487 428
```

SWIPE

adb shell input swipe Xpoint1 Ypoint1 Xpoint2 Ypoint2 [DURATION*]

*DURATION is optional, default=300ms

Example – Swipe from position (356,1257) to (356,100):

```
> adb shell input swipe 356 1257 356 100
```

Setting Device Configuration

Open Settings

```
adb shell am start -a android.settings.SETTINGS
```

Get/Put the device_name

```
adb shell settings get global device_name
db shell settings put global device_name "Device_XXX"
```

Enable/Disable Stay awake

```
adb shell settings put global stay_on_while_plugged_in 3 (enable)
adb shell settings put global stay_on_while_plugged_in 0 (disable)
```

Setting the device brightness

```
adb shell settings put system screen_brightness 125
```

Setting the sound levels

```
adb shell settings put system volume_music_speaker 0
adb shell settings put system volume_notification_speaker 0
adb shell settings put system volume_ring_speaker 0
adb shell settings put system volume_system_speaker 0
```

Setting the screen rotation

```
adb shell settings put system accelerometer_rotation 0
```

accelerometer_rotation: auto-rotation, 0 disable, 1 enable

```
adb shell settings put system user_rotation 3
```

user_rotation: actual rotation, clockwise, 0 0°, 1 90°, 2 180°, 3 270°

Getting/Setting the language

```
adb shell settings get system system_locales
adb shell settings put system system_locales fr-FR
```

A device reboot is needed

Allow applications from unknown sources

```
adb shell settings put global install_non_market_apps 1
adb shell settings put secure install_non_market_apps 1 (Oreo+)
```

Disable USB Debugging mode

The enable command has no sense. You cannot send ADB commands to the terminal if the Debug mode is disabled

```
adb shell settings put global adb_enabled 0
```

Configure the default HOME (for MDM)

The “home app” must be installed and active

```
adb shell cmd package set-home-activity
com.telelogos.mediacontact/com.telelogos.mckiosk.McKioskActivity
```

Get the current keyboard

```
adb shell settings get secure default_input_method
```

Get the list of available keyboards

```
adb shell ime list -s
```

Enable/Disable/Set a keyboard

Each platform has a specific name of the keyboard

```
adb shell ime enable com.android.inputmethod.latin/.LatinIME
adb shell ime set com.android.inputmethod.latin/.LatinIME
adb shell ime disable com.android.inputmethod.latin/.LatinIME
```

Block OS and security updates

```
adb shell settings put system security_update_db 0
adb shell settings put system SOFTWARE_UPDATE_WIFI_ONLY2 0
```

Set NTP server

```
adb shell settings put global ntp_server ntp1.ggl.inet
```

NTP server is not present on all platforms

Enable/Disable WiFi connection

```
adb shell "dumpsys wifi | grep 'Wi-Fi is'" (get current status)
adb shell "svc wifi enable" (enable)
adb shell "svc wifi disable" (disable)
```

Open WiFi settings

```
adb shell am start -a android.settings.WIRELESS_SETTINGS
```

WiFi detailed status

```
adb shell dumpsys wifi | grep "mNetworkInfo"
```

If connected to a valid network, it shows “CONNECTED/CONNECTED“. It also shows the name of the connected network.
If wifi has been enabled but it is yet to connect to a network, it shows “DISCONNECTED/SCANNING“.
If wifi is disabled, it shows “DISCONNECTED/DISCONNECTED“.

Enable/Disable Bluetooth connection

```
adb shell settings get global bluetooth_on (get current status)
adb shell settings put global bluetooth_on 1 (enable)
adb shell settings put global bluetooth_on 1 (disable)
```

A device reboot is needed

```
adb shell am start -a android.bluetooth.adapter.action.REQUEST_ENABLE
```

```
adb shell am start -a android.bluetooth.adapter.action.REQUEST_DISABLE
```

It prompts the user to ‘allow’ or ‘deny’

Enable/Disable NFC connection

```
adb shell dumpsys nfc (get current status)
adb shell svc nfc enable / adb shell service call nfc 7 (enable)
adb shell svc nfc disable / adb shell service call nfc 6 (disable)
```

Enable/Disable Google Play Protect

```
adb shell settings get global package_verifier_user_consent (get current status)
adb shell settings put global package_verifier_user_consent 1 (enable)
adb shell settings put global package_verifier_user_consent -1 (disable)
```

Getting Device Information

Get Android version

```
adb shell getprop ro.build.version.release
```

Get Build number

```
adb shell getprop ro.build.display.id
```

Get Patch number

```
adb shell getprop ro.device.patch.version
```

Get BT MAC address

```
adb shell settings get secure bluetooth_address
```


Get NTP server

```
adb shell settings get global ntp_server
```

Get IP address

```
adb shell ip addr show wlan0  | grep 'inet ' | cut -d ' ' -f 6 | cut -d / -f 1
```

File Management

Copy from the local source to the device

```
adb push <local path> <device path>
```

Copy from the device to the local source

```
adb pull <device path> <local path>
```

Delete a file

```
adb shell rm "/sdcard/file.ini"
```

Create/Delete a directory

```
adb shell mkdir "/sdcard/WsmHost/" > NUL
adb shell rmdir "/sdcard/WsmHost/"
```

See private directory where an app is installed

```
adb shell
run-as <package_name>
```

<package_name>: The installed app must be in debug version, not release or logcat or installed from Google Play

Example:

```
> adb shell
> run-as com.tellnext.emulator
```

Logcat

Clear logcat

```
adb logcat -c
```

Print log messages to stdout

```
adb logcat <text>:V *:S
```

<text>: BTPairing, TellNext, Calibrate, SpeakerNext, EVAPlayer

Example – Print BTPairing and TellNext messages (Verbose) and hide the rest of the messages (Silent):

```
> adb logcat BTPairing:V TellNext:V *:S
```

Example – Print VoiXtreme Engine messages (Verbose) and hide the rest of the messages (Silent):

```
> adb logcat VoiXtreme:V NUANCE:V *:S
```

Set Logcat buffer size (K or M)

```
adb logcat -G16M
adb logcat -g
```

Dump log messages to a specified file

```
adb logcat -f <filename>
```

Activate log messages (if logs not visible – property reset after rebooting the device)

```
adb shell setprop log.tag.<MyAppTag> DEBUG
```

Example – Activate logs for SpeakerNext:

```
> adb shell setprop log.tag.SpeakerNext DEBUG
```

Devices

Bonded devices

Find your Bluetooth service:

```
adb shell "dumpsys -l | grep bluetooth"
```

Get bonded devices:

```
adb shell "dumpsys bluetooth_manager | sed -n '/Bonded devices:/,/^$/p'"
```

Apps Management

Install an application

```
adb shell pm install -g "/sdcard/application.apk" --user all
```

List installed packages

```
adb shell pm list packages
```

List installed 3rd-party packages

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

Launch an app

Get the corresponding ActivityName:

```
adb shell dumpsys package <packageName> | findstr Activity
```

Launch the app:

```
adb shell am start -n <ActivityName>
```

Example – Launch BT Pairing:

```
> adb shell dumpsys package com.tellnext.bluetooth_pairing | findstr Activity
> adb shell am start -n com.tellnext.bluetooth_pairing/com.tellnext.StarterActivity
```

Enable/Disable Accessibility Services for an App

Get Accessibility Service Name:

```
adb shell dumpsys package <packageName>
```

Enable the Service:

```
adb shell settings put secure enabled_accessibility_services <serviceName>
```

Example – Enable BT Pairing Accessibility Service:

```
> adb shell dumpsys package com.tellnext.bluetooth_pairing | findstr Accessibility
> adb shell settings put secure enabled_accessibility_services
com.tellnext.bluetooth_pairing/com.tellnext.accessibility_service.BTAccessibilitySer
vice
```

This action can also disable other services. So, it's better to use ":" to join services:

```
adb shell settings put secure enabled_accessibility_services
<serviceName_1>:<serviceName_2>
```

Example – Enable BT Pairing Accessibility Service and EVA Player service:

```
> adb shell settings put secure enabled_accessibility_services
com.tellnext.bluetooth_pairing/com.tellnext.accessibility_service.BTAccessibilitySer
vice:com.eva.player/.service.PlayerAccessibilityService
```

Disable accessibility services for one app:

```
adb shell am force-stop <packageName>
```

Finding the name of the current foreground app

```
adb shell dumpsys window windows | grep mCurrentFocus
```

Kill app

```
adb shell am force-stop <packageName>
```

Example – Kill BT Pairing App:

```
> adb shell am force-stop com.tellnext.bluetooth_pairing
```

Enable/Disable apps notifications

```
adb shell cmd appops set <packageName> POST_NOTIFICATION ignore
adb shell cmd appops set <packageName> POST_NOTIFICATION allow
```

Example – Disable Hangouts (Google Talk) notifications:

```
> adb shell cmd appops set com.google.android.talk POST_NOTIFICATION ignore
```

Give specific authorization to an app

```
adb shell cmd appops set <packageName> <permission> allow
```

Example:

```
> adb shell cmd appops set com.telelogos.mediacontact WRITE_EXTERNAL_STORAGE allow
```

Give the rights for MDM app (device owner/admin)

dpm set-active-admin: Sets the given component as active admin for an existing user.
dpm set-device-owner: Sets the given component as active admin, and its package as the device owner.
dpm set-profile-owner: Sets the given component as active admin and profile owner for an existing user.

```
adb shell dpm set-active-admin com.telelogos.mediacontact/.McDeviceAdminReceiver >
NUL
adb shell dpm set-device-owner com.telelogos.mediacontact/.McDeviceAdminReceiver >
NUL
```

Wait for an activity to appear (specific screen)

Example:

```
:WAITING_FOR_KLMS
SET KMLS_ACTION=0
echo "---- PENDING ACTION BEFORE KLMS SCREEN APPEARS -----"
timeout /t 1 > NUL
for /f "delims=" %%i in ('%adb% shell "dumpsys activity activities | grep -c
\".activities.ConfirmDialog\" "') do Set KLMS_ACTION=%%i
if %KLMS_ACTION% EQU 0 (goto WAITING_FOR_KLMS)
```

Zebra devices

Identify MC33 keypad model

```
adb shell getprop ro.config.device.keyboard
```

Possible responses: 7 (29-key device) / 6 (38-key device) / 5 (47-key device)

Honeywell devices

Identify CK65 keypad model

```
adb shell getprop ro.hon.plat.keypad
```

Example:

```
> adb shell getprop ro.hon.plat.keypad
> qwerty
```

Possible responses: qwerty (51-key alpha) / numeric (38-key numeric)