Continue



```
If you're an Android enthusiast like me, you've probably heard about the Android Debug Bridge (ADB) - a versatile command-line tool that lets you communicate with a device. Whether you're a developer looking to debug an app, or a power user wanting to tinker with their device, ADB is an essential tool. You can download it as part of the Android
SDK Platform Tools package. You can use ADB to perform various tasks, such as installing apps, debugging, accessing hidden features, and more. In this article, we will show you a detailed list of all ADB commands for Android, along with a brief description and an example for each one. adb provides access to a Unix shell on your Windows PC or Mac
that you can use to run a variety of commands. The client runs on your development machine. You can invoke a client from a command-line terminal by issuing an adb command. Daemon (adbd), which runs commands on a device. The
daemon runs as a background process on each device. Server, which manages communication between the client and the daemon. The server runs as a background process on your development machine. Android Debug Bridge There are only a handful of commands listed in the official adb documentation, but with additional arguments and
parameters, that number could go very high. You can save this ADB cheat sheet guide for whenever you want to perform any debugging on your Android USB drivers, enable USB debugging on your Android device and install the ADB tools on your computer. Once you have set up
everything, you can open a terminal or command provides a list of devices that are currently connected to your computer. Example: Running this command will give you an output similar to this: List of devices attached emulator-5554 device With
adb push, you can transfer a file or directory from your computer and you want to transfer it to your device's /sdcard/ directory. adb push sample.txt /sdcard/ The opposite of adb push, the adb pull command lets you copy a file or directory from your
Android device to your computer. Example:Let's say you want to fetch a file named "sample.txt" from the /sdcard/ directory of your device. To make this
work, you need to copy the APK file into the platforms-tools folder on your computer. Example: To install an APK named "app.apk": adb install an app without losing its data, you should use the following command: adb install -r
app.apk And for those who want to install APK files directly to the SD card, you can use the following two commands so the app is not installed in your Android phone or tablet using ADB, the adb uninstall command is what you'll use. It requires the package
name of the app. Example: To uninstall an app with the package name com. example. myapp if you need to restart your device and it's not at hand, you can easily do so with the adb reboot command. Example: Just run: adb reboot Your device will promptly restart. For developers or those troubleshooting their device,
the adb logcat command displays the device log in real-time. This log provides insights into what's happening behind the scenes. Example:To view your device's logs: adb logcat This command also comes with some important parameters and variations that may be useful when debugging. You should use the following command when you want to clear
any existing logs on your device. adb logcat -c // Clear // And if you want to save the log contents / logcat data to your PC, you should use the following command: adb logcat -d > // The adb backup command allows you to create a backup named
 "backup.ab": adb backup -f backup.ab -all The adb backup -apk command is used to create a backup of your device's installed apps with their APKs: adb backup apk -f backup.ab This command is an extensive version of the adb backup command,
allowing you to create a complete backup of your device, including apps, app data, and system data. Example: To create a comprehensive backup app data, you can restore it using the adb restore command. Example: To create a comprehensive backup app data, and system data, you can restore it using the adb restore command. Example: To create a comprehensive backup app data, and system data, you can restore it using the adb restore command. Example: To create a comprehensive backup app.
backup.ab The adb sideload command is useful for flashing zip files, especially OTA update named "update.zip": adb sideload update.zip If you're working with system files, you may need to remount the /system partition in read-write mode. The adb remount
command lets you do just that. Example: To remount the system partition: adb remount Using the adb shell command, you can start a remote command prompt where you can run commands directly on your device. Want to take a
screenshot directly from the command line? The adb shell screencap command lets you capture the device's current screen. Example: To save a screenshot as "screenshot as "screenshot as gem for recording the screen of your device's storage: adb shell screencap /sdcard/screenshot.png" on your device's storage: adb shell screenshot.png" on your device's storage: adb shell screenshot.png" on your device's storage: adb shell screenshot.png" on your device's storage: ada shell screenshot.png" on your device's stor
helpful for developers or content creators. Example: To start recording mp4": adb shell dumpsys command is a powerful one. It provides comprehensive information about system services. Developers often use this to extract in-depth details for troubleshooting.
Example:To get information about the battery: adb shell dumpsys battery If you want to change system properties, adb shell setprop is the command you need. Remember, tweaking these properties can affect system behavior, so proceed with caution. Example:To change the device's density: adb shell setprop ro.sf.lcd_density 240 The adb shell wm
command allows you to interact with the window manager, which controls elements like screen density and screen size. Example: To set the device's screen density to 240: adb shell wm density 240 The adb shell wm size command lets you change the screen density and screen density a
Example: To set the screen size to 1080×1920: adb shell wm size 1080x1920 The adb shell pm command is used for packages or uninstalled packages: adb shell pm list packages. The adb shell input command allows you to simulate user input events like taps, swipes, or
key presses. Example:To simulate a tap at coordinates (500, 700): adb shell input tap 500 700 The adb shell am command is a versatile tool for interacting with Android's Activity Manager, enabling activities and broadcasting intents. Example:To start an activity: adb shell am start -n com.example.myapp/.MainActivity The adb
shell dumpstate command generates a bug report of the device's current state, which can be useful for diagnosing issues. Example: To create a bug report txt The adb shell dumpstate > bugreport.txt The adb shell dumpstate > bugreport.txt The adb shell pm list users command displays a list of user accounts on the Android device. This can be useful for managing user profiles on a multi-user
device. Example: To list the user accounts on the device: adb shell wm overscan settings, which can be helpful for adjusting the display on certain devices or screens. Example: To reduce overscan by 5% in all directions: adb shell wm overscan 5,5,5,5 The adb shell wm density command
allows you to change the screen density of your device temporarily, which can be useful for testing app layouts on different screen density 320 The adb shell input text command allows you to simulate text input on your Android device. This can be useful for automated testing or filling
out forms. Example:To input the text "Hello, Android!": adb shell input text "Hello, Android!" With the adb shell input swipe command, you can simulate a swipe from (100, 500) to (300, 500): adb shell input swipe 100 500 300 500 The adb shell input keyevent
command allows you to simulate key presses on the device, such as volume control or media playback. Example: To simulate the volume up key press: adb shell input keyevent 24 The adb shell dumpsys battery-related issues.
Example: To get battery information: adb shell dumpsys batteryinfo The adb shell cat command allows you to view the contents of a text file named "example.txt": adb shell cat /path/to/example.txt The adb shell pm hide command lets you hide a specific package on
your Android device. This can be useful for disabling pre-installed apps. Example: adb get-serialno command. Example: adb get-serialno This command shows the ADB status of the connected Android
phone, tablet, watch or an emulator. If you're running a script and wait-for-device was mentioned earlier, combining it with adb shell can be
valuable when you want to execute shell commands only when the device is ready. Example: To wait for the device and then run a shell command sets up port forward requests
from port 5000 on your computer to port 6000 if you have multiple APKs: adb install-multiple APKs: adb install-multiple app1.apk app2.apk app3.apk The adb tcpip command allows you to switch from USB to
TCP/IP mode for ADB connections, which can be useful when you want to connect wirelessly. Example: To switch to TCP/IP mode on port 5555: adb tcpip 5555 You can use the adb start-server command if it stops working or
become unresponsive, you can easily kill the ADB server on your Android device with the adb kill-server command. The adb connect ip_address_of_device and connect it to the PC. You can use the adb jdwp command to see the list of all the JDWP processes on your PC. JDWP is
short for Iava Debug Wire Protocol. I hope this list gives you a solid introduction to some of the most common ADB commands out there. Remember, while ADB is a powerful tool, always ensure you know the consequences of the commands you're running - especially if you're new to ADB. I'd advise you to bookmark this page as I will be updating it
with newer commands (as I learn them) over time. If there are any more commands that you think should be included in the list above, drops us a comment below and lets us know. You can also visit the ADB documentation page for more information about this tool. The adb devices command is a widely used tool in Android development and
debugging. It is part of the Android Debug Bridge (ADB), a versatile command-line tool that allows communication with an emulator instance or connected Android devices or emulators directly from their development environment. It can list all the connected
devices or emulators along with essential information related to those devices, which facilitates seamless management and devices or emulators along the adb devices command is to identify the Android devices or emulators
currently connected to your host machine. This is particularly useful in a development and testing environment where multiple devices might be connected simultaneously. By listing all the devices might be connected simultaneously. By listing all the devices might be connected simultaneously. By listing all the devices might be connected simultaneously. By listing all the devices might be connected simultaneously. By listing all the devices might be connected simultaneously. By listing all the devices might be connected simultaneously. By listing all the devices might be connected simultaneously. By listing all the devices might be connected simultaneously. By listing all the devices might be connected simultaneously. By listing all the devices might be connected simultaneously. By listing all the devices might be connected simultaneously. By listing all the devices might be connected simultaneously. By listing all the devices might be connected simultaneously. By listing all the devices might be connected simultaneously. By listing all the devices might be connected simultaneously. By listing all the devices might be connected simultaneously. By listing all the devices might be connected simultaneously. By listing all the devices might be connected simultaneously. By listing all the devices might be connected simultaneously. By listing all the devices might be connected simultaneously. By listing all the devices might be connected simultaneously. By listing all the devices might be connected simultaneously. By listing all the devices might be connected simultaneously. By listing all the devices might be connected simultaneously. By listing all the devices might be connected simultaneously. By listing all the devices might be connected simultaneously. By listing all the devices might be connected simultaneously 
device is properly connected and recognized by the ADB service. Explanation: adb: This is the Android Debug Bridge command-line tool. devices attached 192.168.1.2:5555 device emulator-5556 device The output shows a list of devices.
identifiers and their connection statuses. In this example, two devices are connected: one with a specific IP address over a network and another is an emulator running locally on the machine. Code: Motivation: Building upon the basic functionality of listing connected Android devices, the adb devices -l command extends the information by providing
detailed system information about each connected device. This additional information, such as the devices model, product name, and transport ID, can be crucial when dealing with multiple devices that might share similar identifiers but have different configurations. This ensures developers can quickly differentiate between devices, aiding in more
targeted testing and debugging. Explanation: adb: This is the core command of the Android Debug Bridge. devices attached to the ADB server.-l: This flag requests more detailed output, which includes additional information about each device, such as the product, model, and transport id. Example
output:List of devices attached 192.168.1.2:5555 device product:sdk gphone x86 model:Pixel 4 XL transport id:1 emulator-5556 device product:sdk gphone x86 fevice product:sdk gphone x86 fevice product:sdk gphone x86 model:Pixel 3a transport id:2 In this example, you can see not only the device identifiers but also specific attributes that provide more context about each device. Here, devices are
distinguished by their model names like Pixel 4 XL and Pixel 3a, making it easier for developers to effectively manage multiple connected Android devices and emulators. Whether you simply want to verify connectivity or need
detailed information about each device, this command offers a straightforward solution. Overall, streamlined device management enhances the development workflow, leading to better productivity and more reliable applications. Here's a list of some useful ADB and Fastboot commands that might come in handy in different situations. These ADB
commands and ADB Shell commands work on all Android devices or customizing it will be well aware of ADB and fastboot. These are system utilities that provide access to the backend of Android when it is connected to a PC. These are also one of
the constants in the Android ecosystem. Same commands will produce the same results regardless of the device is turned on while Fastboot can be used when the Android Debug Bridge can be used when the Android device is turned on while Fastboot can be used when the Android Debug Bridge can be used when the Android Debug Brid
Fastboot commands. Before you can make use of ADB and Fastboot drivers on your Android device. The option is found in the Developer options but if you don't know how to get there, you can follow the tutorial linked below. Besides, you'll also need the ADB and Fastboot drivers on your Windows, Mac, or
Linux computer, ADB Commands List As mentioned earlier, ADB commands are used when the device is powered on and you have access to the device and grant ADB commands that you may need at some point. You
can even use your Android phone like a PC to execute ADB and Fastboot commands. adb devices Shows all the devices connected to ADB. Can be used to make sure your device is connected properly before performing any other operations. 'adb devices' command adb usb It shows all
Android devices and emulators connected to your Computer via a USB cable. adb devices attached to your PC. adb connect ip address of device using the 'adb connect to your computer. adb reboot ADB
can be used to reboot your device, useful when your hardware buttons aren't working or if you're already using ADB. You can also reboot bootloader Reboots the device into
recovery mode. adb install It can be used to install an APK file on your Android device from your computer. It is more convenient than the alternative, which is to transfer the APK file to install in the 'platform-tools'
folder, otherwise you'll also have to type the location of the APK. adb install or LiveWallpapers.apk Use the following command to re-install or update an existing app on your device and keep its data. adb install it to the SD storage using the
installed app, which is pretty clear from the command itself. adb uninstall com.facebook.katana Use the following command if you want to uninstall an app package but keep its data and cache directories. adb uninstall com.facebook.katana Use the following command if you want to uninstall an app package but keep its data and cache directories. adb uninstall an app package but keep its data and cache directories around after package removal. adb logcat Displays the log data onto the
screen. You can use this command with the following parameters for different purposes as shown below, adb logicat -c // clear // By adding the '-c' parameter to the 'adb logicat output to a file on
your computer. adb bugreport Displays the Android device information such as dumpsys, dumpstate and logcat data on the screen. adb jdwp Lists the JDWP (Java Debug Wire Protocol) processes on the device. if you're not already aware of it, chances are you don't have to worry about it either. Don't Miss: 50 Best Apps for Rooted Android Devices adb
get-serialno Displays the adb instance serial number string with the device. adb get-state Shows the status of the device adb wait-for-device This command is issued. It executes when the device adb get-state Shows the status of the device adb get-state Shows the status of the device.
server process. adb kill-server Stops the adb server process (terminal adb.exe process). adb sideload Just as you can flash zip packages from a custom recovery which is the most common custom recovery for all devices does support it. This way you can flash
packages directly from your computer without having to transfer them first to your device or any external storage. To sideload a mod or update.zip file successfully, you must copy it first to the SDK folder, adb sideload sample.zip and pull This command can be used to pull any files from your device and save it on your computer. To download or pull any files from your device and save it on your device and save it on your device or any external storage.
 file from your Android device to the SDK platform-tools directory, use adb pull /sdcard/screenrecord.mp4 If you want to download a file from your phone's storage to a specific drive on your computer, execute the following command adb pull /sdcard/screenrecord.mp4 e:\ adb push Similarly, this command can be used to push a file from your computer
to your device. If the file to be pushed it save in the SDK folder, use adb push OPLiveWallpapers.apk /sdcard Having given you the above ADB commands list, let's check out the ADB Shell commands list below. adb backup //
By using this ADB command, you can create a full backup of your Android device and save to your computer. adb shell Commands List Below is the list of some really useful ADB shell commands. adb shell to your device and save to your device and save to your device and save to your device.
command console in the device and lets you control the device through it. adb shell pm uninstall -k -user 0 You can uninstall or remove any system app installed on your Android device. This is the easiest way to uninstall bloatware. Just use the following command followed by the app package name. pm uninstall -k -user 0 com.facebook.katana Learn
more about debloating Android devices without root in our dedicated article, adb shell dumpsys You can use this ADB shell dumpsys You can also use it to get information about specific components of your device, such as display, battery,
etc. adb shell dumpsys display adb shell dumpsys battery adb shell dum
following command: adb shell dumpsys display Issuing the above command for the Galaxy S9+ gave me the following info about my phone's display resolution and density. mDisplayInfos= PhysicalDisplayInfos= PhysicalDisplayI
following commands. This value can vary per phone, you'll know it's right when you can read text and apps don't render too small value is from 120 up to 640 for 720p (HD) adb shell wm size 720x1560 adb shell wm density 350 adb shell pm list packages The following
commands print the names of all app packages installed on your Android device. You can add additional conditions or filters to list specific packages only. To list all packages adb shell pm list packages adb sh
disabled app packages: adb shell pm list packages -u ADB Shell command to Send SMS screen By sung this command, you can send the text message screen with the message content and phone number
/sdcard/screenshot.png adb shell screenrecord On Android devices running Android 4.4 KitKat and above, you can even record your phone or tablet's screen and download the recorded video to your computer using the ADB shell commands. Besides, you can also set conditions like video duration, resolution in pixels and video bitrate, etc. You need to
press Ctrl+C to stop recording manually. adb shell screenrecord /sdcard/screenrecord /sdcard/screenrecord manually. adb shell screenrecord manually. Add screenrecord manually manua
minutes). You can decrease this time limit according to your needs (180 seconds is the maximum limit). adb shell screenrecord --time-limit 120 /sdcard/screenrecord maximum limit). abs shell screenrecord --time-limit 120 /sdcard/screenrecord maximum limit).
/sdcard/screenrecord.mp4 adb shell cd Change ADB shell directory using 'cd ' adb shell (Hit Enter then the following command) cd /system adb shell rm By using this ADB shell command first and hit the Enter key. After that, you can use
one of the following commands followed by the file or directory name as shown below. Delete a file: rm -f /sdcard/OPWallpaperResources.apk Delete a directory or folder: rm -d /sdcard/ZooperWidget You can also use 'rmdir' in place of 'rm -d' to remove a directory. adb shell mkdir This ADB shell command is used to create a new directory or
directories under an existing directory. You can also set permission for the directory too. Execute 'adb shell' and then the following commands: mkdir -p /sdcard/NewFolder mkdir -p /sdcard/NewFolder mkdir -p /sdcard/NewFolder mkdir -m 644 /sdcard/N
Again, you need to start with the 'adb shell' command first. To copy files and then paste them, by mentioning the source and destination locations as shown below: cp /sdcard/OPWallpaperResources.apk /sdcard/LiveWallpaperResources.apk /sdcard/LiveWallpaperResources.apk /sdcard/LiveWallpaperResources.apk /sdcard/DPWallpaperResources.apk /sdcard/DPWallpaperResourc
/sdcard/OPWallpaperResources.apk /system/app If you want to move a file to a different location with a new name, my /sdcard/OPWallpaperResources.apk /system/app If you want to move a file to a different location with a new name, my /sdcard/OPWallpaperResources.apk /system/app If you want to move a file to a different location with a new name, my /sdcard/OPWallpaperResources.apk /sdcard/OPWallpaperResources.apk /system/app If you want to move a file to a different location with a new name, my /sdcard/OPWallpaperResources.apk /sdcard/OPWallpaperResources.apk /system/app If you want to move a file to a different location with a new name, my /sdcard/OPWallpaperResources.apk /sdcard/OPWall
your phone's Wi-Fi IP address. Execute 'adb shell' in the command window and then issue the following command after executing 'adb shell': top If you want to stop CPU processes monitor
press Ctrl+C on your keyboard. adb shell getprop & adb shell getprop & adb shell getprop window, hit the Enter key and then issue the following
command: getprop Below are some more examples: getprop ro.build.version.sdk getprop ro.chipname Now, to set the examples below: setprop net.dns1 1.2.3.4 setprop net.dns2 1.2.3.5 Similarly, you can also set a custom VMHeap size: setprop
dalvik.vm.heapsize 40m ADB Shell KeyEvent commands By using the following ADB Shell keyevent 3 // Home btn adb shell input keyevent 4 // Back btn adb shell input keyevent 5 // Call adb shell input
keyevent 6 // End call adb shell input keyevent 26 // Turn Android device ON and OFF. It will toggle device to on/off status. adb shell input keyevent 67 // Delete (backspace) adb shell input keyevent 207 // Contacts adb shell input keyevent 27 // Contacts adb shell input keyevent 27 // Contacts adb shell input keyevent 28 // Turn Android device ON and OFF. It will toggle device to on/off status.
use the ADB command mentioned above. adb reboot bootloader Once in Fastboot mode, you can use a number of Fastboot command to make sure the devices is properly connected. fastboot devices fastboot oem unlock
Unlocks the bootloader on some Android devices such as Pixel or OnePlus. Most other Android devices require you to get an unlock key from the fastboot mode back into fastboot mode. Sometimes this may be necessary when
image file.img This command can be used to boot your device using a certain image file without having to flash it first. Can be useful to test an image before flashing it. Should you have any questions about Fastboot commands, please let us know. So, here end our list of useful ADB and Fastboot commands that can help Android users perform a
plethora of commands using cmd. Let's know if we forgot to mention any command that should have been our list. Read Next Hackbench - Find Best Performing Kernel for Android If you are a pro-Android user and uses their phone for customizations, rooting or unlocking bootloaders, then you must be aware of the term ADB and fastboot. Basically
 ADB and fastboot is a command-line tool through which you can modify a lot of things on your Android phones. For instance, you can give or revoke read and write permission for a package, you can modify a lot of things on your Android phones. For instance, you can give or revoke read and write permission for a package, you can give or revoke read and write permission for a package, you can use ADB commands through which you will actually
be able to explore the full potential of Android OS and your smartphones. Most of the users who use ADB knowingly or unknowingly only uses a limited set of commands and does not use the full potential of the ADB tool. ADB stands for Android debug bridge and it consists of 3 components. Client- It is the laptop or your PC to which the Android
device or the emulator is connected to.Daemon - It is a service that runs on an Android device and your computer as well, it is used to facilitate command execution and acceptance between the devices. It is also known as 'adb'. Server - Server is the interface to handle the communication between Client and Daemon. Do note that before using ADE
and fastboot tools, make sure that you have the latest version of the driver for your Android device and the latest version of SDK Platform-tools are installed on your computer. 1. ADB Commands List Directory Make sure before you try these ADB commands on your Android device, you have allowed the USB debugging option for Developers options
from settings. Also, you have given authorization access to your computer from your phone. To enable USB debugging you need to go to the developer's options from settings which can be enabled by tapping and
open ADB command-line. Write 'adb devices you must see a pop-up on your phone asking for permission click on allow and that's it. Also, make sure you do all these experiments on a backup phone and not on a daily driver, as one wrong code or command can cause your phone to brick or malfunction. Now you are good to go and can test ADB
commands. Below we have some ADB commands through which you can reboot your phone, sideload APKS, flash flashable zips, push or pull files, uninstall apps, debloat your phone and many more such operations. SEE ALSO Setup System-Wide ADB and Fastboot on Windows This is a very basic command and writing will show you the ADB versions.
and all the possible commands associated with ADB. This is one of the very first commands which we all use while using ADB tools. By writing adb devices we can check the list of devices connected properly and all the necessary
Use the adb version command to check the version installed or not. One of the very common commands of adb tool is the adb reboot it is used when you want to reboot your device. With the help of this command, you can even reboot
your device into fastboot, bootloader or even recovery. This command to directly reboot your device into bootloader, make sure your device is connected to your PC. Just type adb reboot bootloader and your
 device will reboot into bootloader. use this command to directly reboot your device into recovery, make sure your device is connected to your PC. Just type adb reboot recovery and your device will reboot into recovery. Android is an Open Source OS and is very vast. Until now you might know only two methods of installing Android apps on your
Android device. You either download it from Google Play Store or galaxy Store, or else you download the APK and manually install it. However, you can directly install the APK from your PC to your Android device, without copying or doing anything else. Just
copy the APK file to the SDK platform-tools folder and then type the adb install package name and that's it. Below is an example of how to write code, it can vary from app to app and package to package. adb install com.facebook.katana.apk You can also re-install an APK on your Android device without deleting the data of the App. Just write the below
command, and your app will be uninstalled. adb uninstall com.facebook.katana However, if you want to delete the app but wants to keep the data of the app, then simply use the adb uninstall -k com.facebook.katana To know about the Android device or the emulator
connected to your Windows, Mac or Linux computer, simply use the adb usb command and it will get you all of the info. SEE ALSO Minimal ADB and Fastboot | Download adb logicat command, you will be able to see the log data of your
device on your PC. There are a few parameters as well which can be used like 'adb logcat -c // clear // with this you will be able to clear all the existing logs of your Android phone or tabler or the emulator. To save the logcat data of your Android device or emulator on your PC use the following command. adb logcat -d > [path_to_file] // The adb start-
server command is a very useful command and it is used to start the adb server is not responding or by chance, it has stopped, you can use this command will work properly, so in case the server again. if the ADB command line is not working properly, and even after restarting the terminal
doesn't help you. Then you must try to stop the adb server and start it again, so just use the command adb kill-server and then use the above-mentioned command. It can be used to sideload software and OS update.zip files using a computer. If
you have downloaded a flashable update.zip file then you need to copy it to the platform-tools folder and then execute the command as written below. For ease, rename the downloaded flashable zip file into an update.zip file adb sideload update.zip file on your Android device to your computer, simply use the adb pull
command. It can be helpful to pull any files from your device and save them to the platform-tools folder on your computer to your device. Do note that whichever file you need to transfer to your Android device from your
PC, you need to copy it to the platform-tools folder first. You can also push the files from your PC to the sd card of your Android device directly. Below is the command to push a file from your PC to the sd card of your Android device directly.
Android device to your computer. SEE ALSO Allow USB Debugging | Authorize ADB Commands on Android Similar to the above-mentioned command, with the help of the adb backup command. Since Android OS uses a lot of codes to run,
often we cross through a lot of bugs. So with the help of this adb bugreport command, you will be able to see the log data, dump state and dumpsys from your Android device to your computer. Often OEMs asks for bug reports, and this how you can send them or show them the bug you are facing on your Android device. The term JDWP stands for Java
Debug Wire protocol. With the help of this ADB command, you will be able to see the list of all the JDWP processes on your PC. To get the ADB instance serial number, use the ADB status of a connected device or emulator. The adb wait-for-device is a command that
tells ADB that it has to wait and keep the connection on hold until the next command is being issued or executed. That was an extensive cheat sheet regarding the ADB commands and, we hope it was a helpful article for you, as ADB commands come in handy a lot of times while experimenting with something with our devices, or while restoring our
devices. We will keep on updating ADB commands of ADB. Read next: Find out MD5 Checksum and SHA Hash of a File (Windows, Mac, and Linux) In this article, you will find the list of all ADB and Fastboot Commands for Android devices. You can execute these commands on any computer regardless of its
operating system (Windows, Mac, or Linux). The commands are universal, which means the same commands will work on any Android device, regardless of its manufacturer or device variant. Android enthusiasts, familiar with unlocking bootloader, rooting, or installing a custom recovery on their device, must be well aware of ADB and fastboot. These
are system utilities that provide backend of Android when it is connected to PC on fastboot or bootloader mode. There are some specific codes that can be executed using a computer to perform a function on Android devices. Mostly, people use fastboot to
unlock the bootloader of a phone. Sometime, you may need to unbrick the device (in case of a hard brick) using fastboot. Most of the manufacturer or variant. If you are stuck somewhere, where you need fastboot / ADB command to assist, this post is for you. I have
listed all the important fastboot, and ADB commands in the table given below. Codes, along with its property and functions, are also appropriately mentioned. Use Ctrl + F to find your requirement. Before going to the fastboot & ADB commands, you need certain tools and settings to be enabled on your device as well as the computer. Otherwise, you
can not make use of the fastboot or ADB commands. The tools are needed to be installed on the computer. On the other hand, the necessary settings are to be made on the phone. Methods are the same for all Android smartphones. So, don't worry about the OEM or variant. Consider the following checkpoints: Here is the complete list of all the ADB
commands that you can execute using the command prompt or terminal while connecting the Android devices that are correctly connected above: adb helpDisplays the help documentation on ADB commands devices that are correctly connected
to the PC via ADBadb usbIt will show you all the connected Android to systemadb reboot bootloader modeadb reboot wour PC using USBadb reboot footloader modeadb reboot would not evice into fastboot or bootloader modeadb reboot would not evice into fastboot or bootloader modeadb reboot would not evice or Emulators connected to your PC using USBadb reboot would not evice into fastboot or bootloader modeadb reboot would not evice into fastboot or bootloader modeadb reboot would not evice into fastboot or bootloader modeadb reboot would not evice into fastboot or bootloader modeadb reboot would not evice into fastboot or bootloader modeadb reboot would not evice into fastboot or bootloader modeadb reboot would not evice into fastboot or bootloader modeadb reboot would not evice into fastboot or bootloader modeadb reboot would not evice into fastboot or bootloader modeadb reboot would not evice into fastboot or bootloader modeadb reboot would not evice into fastboot or bootloader modeadb reboot would not evice into fastboot or bootloader modeadb reboot would not evice into fastboot or bootloader modeadb reboot would not evice into fastboot or bootloader modeadb reboot would not evice into fastboot or bootloader modeadb reboot would not evice into fastboot or bootloader modeadb reboot would not evice into fastboot would not evice into fastbo
updates using recovery mode (when the ZIP file is on the computer, not in mobile). Read more about it here!adb install APK files manually on your Android using ADBadb install or update an APK to a newer version on Androiderical APK files manually on your Android using ADBadb install apk (replacing the file name.apkInstall APK files manually on your Android using ADBadb install apk (replacing the file name).
using ADBadb install -s filename.apkInstall APK in the external SD Card (if supported by the application) adb uninstall Uninstall any app using ADB command (see next) adb uninstall -k com.tencent.igUninstall the app keeping the data and cache
for some time until another process is doneadb get-stateShows the device status in the command prompt or terminaladb start-serverStarts the ADB server processadb backup //Using this command, you can create a full backup of your Android device and save it to the computeradb restore //Use this
command to restore the backupadb connect ip address of deviceUsing this command, you can connect the IP address of your Android device to the computer or vice versa. To pull a file from the Android device to the
from computer to Android using ADB, you can execute the following command: adb push /local/path/CartoonHD.apk /sdcard/apps/ Here is the complete list of all the ADB Shell commands that you can execute the full path with extension and the destination path as well. Check out the example below: adb push /local/path/CartoonHD.apk /sdcard/apps/ Here is the complete list of all the ADB Shell commands that you can execute the full path with extension and the destination path as well.
using the command prompt or terminal while connecting the Android device to the PC with all required settings. ADB Shell CommandsFunctionsadb shellStarts the remote shell command console in the device and lets you control the Android through itadb shell with device to the PC with all required settings. ADB Shell CommandsFunctionsadb shellStarts the remote shell command console in the device and lets you control the Android device (details given
below)adb shell pm list packages Shows all the installed on your Androidadb shell pm list packages apps installed on your Androidadb shell pm list packages apps list on your Androidadb shell pm list packages apps list on your Androidadb shell pm list packages.
Androidadb shell pm list packages -eEnabled (active) apps list on your Android with installed from your Android with installed from your Android with packages -uList of all the apps that you uninstalled from your Android with installed from your Android with installed from your Android with packages -uList of all the apps that you uninstalled from your Android with installed from your Android 
 following:rm -f /sdcard/whatsapp.apkDelete a file: replace whatsapp.apk to your desired file name with extensionadb shellHit enter and execute the following:rm -d /sdcard/DCIMDelate a file: replace whatsapp.apk to your desired file name with extensionadb shellHit enter and execute the following:mkdir /sdcard/Folder1Create a new folder "Folder1" under /sdcardad
shellHit enter and execute the following:netstatCheck the network statics on your Android device using ADBadb shellHit enter and execute the following:topUse this Coe to monitor the running CPU process on your Android deviceadb shellHit enter and execute the following:netstatCheck the network statics on your Android deviceadb shellHit enter and execute the following:netstatCheck the network statics on your Android deviceadb shellHit enter and execute the following:netstatCheck the network statics on your Android deviceadb shellHit enter and execute the following:netstatCheck the network statics on your Android deviceadb shellHit enter and execute the following:netstatCheck the network statics on your Android deviceadb shellHit enter and execute the following:netstatCheck the network statics on your Android deviceadb shellHit enter and execute the following:netstatCheck the network statics on your Android deviceadb shellHit enter and execute the following:netstatCheck the network statics on your Android deviceadb shellHit enter and execute the following:netstatCheck the network statics on your Android deviceadb shellHit enter and execute the following:netstatCheck the network statics on your Android deviceadb shellHit enter and execute the following:netstatCheck the network statics on your Android deviceable shell and the properties of the network statics on your Android deviceable shell and the network statics of the network statics and the network statics are not all the network statics and the network statics are not all the network statics and the network statics are not all the network statics are n
enter and execute the following:getprop ro.build.version.sdkGet the property of Android's build.prop configuration and change the properties If you want to uninstall a system app via ADB Shell command, you can execute the following:
code: adb shell pm uninstall -k -user 0 package.name.com Replace package.name.com with the actual package name that you want to uninstall bloatware which comes inbuilt with the system that can not be uninstalled generally from the app menu. For example, if Facebook comes
as a system app, and you want to remove it, execute the code in the following way: pm uninstall -k --user 0 com.facebook.katana adb shell dumpsys Using the dumpsys command, you can also use it to get information about your Android device's specific
elements, such as the battery, display, and battery stats. adb shell dumpsys display adb shell dumpsys battery stats You can easily change the Pixel density of your Android device's screen using adb shell dumpsys battery stats. In order to do that, you need to know the actual Pixel density of your device. Let's take an
example with the Galaxy S9 Plus. At first, you need to know the original resolution executing the adb shell dumpsys command. Following are the info about phone's display for Galaxy S9 Plus (after executing about phone's display for Galaxy S9 Plus (after executing about phone's display for Galaxy S9 Plus (after executing about phone's display for Galaxy S9 Plus (after executing about phone's display for Galaxy S9 Plus (after executing about phone's display for Galaxy S9 Plus (after executing about phone's display for Galaxy S9 Plus (after executing about phone's display for Galaxy S9 Plus (after executing about phone's display for Galaxy S9 Plus (after executing about phone's display for Galaxy S9 Plus (after executing about phone's display for Galaxy S9 Plus (after executing about phone's display for Galaxy S9 Plus (after executing about phone's display for Galaxy S9 Plus (after executing about phone's display for Galaxy S9 Plus (after executing about phone's display for Galaxy S9 Plus (after executing about phone's display for Galaxy S9 Plus (after executing about phone's display for Galaxy S9 Plus (after executing about phone's display for Galaxy S9 Plus (after executing about phone's display for Galaxy S9 Plus (after executing about phone's display for Galaxy S9 Plus (after executing about phone's display for Galaxy S9 Plus (after executing about phone's display for Galaxy S9 Plus (after executing about phone's display for Galaxy S9 Plus (after executing about phone's display for Galaxy S9 Plus (after executing about phone's display for Galaxy S9 Plus (after executing about phone's display for Galaxy S9 Plus (after executing about phone's display for Galaxy S9 Plus (after executing about phone's display for Galaxy S9 Plus (after executing about phone's display for Galaxy S9 Plus (after executing about phone's display for Galaxy S9 Plus (after executing about phone's display for Galaxy S9 Plus (after executing about phone's display for Galaxy S9 Plus (after executing about phone's display for Galaxy S9 Plus (
dpi If you want to set a lower resolution, you can check out the following examples (remember, the values may vary from device to device): for 1080x2220 adb shell wm size 1080x220 adb shell wm size 1080x220 adb shell wm size 1080x20 adb shell wm size
excellent for you. With this ADB command, you can send a text message from your Android phone while connected to the computer; adb shell am start -a android intent, action. SENDTO -d sms; CCXXXXXXXXXX --es sms body "SMS BODY GOES HERE" --ez exit on sent true adb shell input keyevent 22 adb shell input keyevent 66 Using this code, you
can capture a screenshot on your Android phone and pull the snapped screen to your computer instantly: adb shell screenshot.png Hit enter, and then: adb pull the video to the computer. Execute the following commands: adb shell
screenrecord /sdcard/movie.mp4 (Press Ctrl+C or Command+C to stop the recording) adb pull /sdcard/movie.mp4 By default, the screen recording is limited to 3 minutes only. If you want to set a condition like clip duration, resolution in pixels, and video bitrate, you can use the following commands: adb shell screenrecord --size 1920x1080
/sdcard/movie.mp4 I used a 1920x1080p resolution here for an example. You can change the video output. For example, I am setting the bitrate to 4MBPS.
The code would be as follows: adb shell screenrecord --bit-rate 4000000 /sdcard/movie.mp4 Execute the following command to copy and paste files/folders: adb shell and destination value as per your requirements.) Next, to move a file
or a folder from one location to another, you can execute the following: my /sdcard/facebook.apk /sdcard/facebook.apk /sdcard/facebook.apk Using the
Shell KeyEvent commands, you can trigger certain functions perform by hardware buttons and UI options on your Android device. Execute these codes only if you know about tweaking VM heap to improve performance. adb shell input keyevent 3 // Home btn adb shell input keyevent 4 // Back btn adb shell input keyevent 5 // Call adb shell input
keyevent 6 // End call adb shell input keyevent 26 // Turn Android device ON and OFF. It will toggle device to on/off status. adb shell input keyevent 67 // Delete (backspace) adb shell input keyevent 27 // Contacts adb shell input keyevent 27 // Contacts adb shell input keyevent 27 // Contacts adb shell input keyevent 28 // Turn Android device ON and OFF. It will toggle device to on/off status.
keyevent 220 / 221 // Brightness down/up adb shell input keyevent 277 / 278 /279 // Cut/Copy/Paste Here is the command prompt or terminal while connecting the Android device to the PC with all required settings. Perform the following if you had fulfilled the prerequisites
mentioned above: Fastboot commandsFunctionsfastboot devices Executing this command will show you the list of Android devices connected to the PC on Fastboot devices c
Or Browse bootloader unlocking guide for your device here! fastboot oem lockRelock the bootloader if you want to go back to stock position on your Android device fastboot Mode again from the Fastboot Mode again from the Fastboot Mode itself. (necessary sometimes while flashing some image files) fastboot flashThis is the most
important command for Android enthusiasts. Using this command, you can flash anything to your Android, like an OTA ZIP or a TWRP Image fastboot flash boot image to root your phone fastboot flash recovery images like
TWRP on Android devices. Read more about flashing TWRP recovery here!fastboot boot filename.imgThis command is used to boot your device using a particular image file without flashing it. This can be useful for testing purposes before a permanent flash. Read mode! So, that's pretty much on our blog for fastboot, and ADB commands. If I missed
anything, please do let me know in the comment section below. I hope this post is helpful for you to perform a plethora of commands using cmd or terminal. If you have any questions about ADB or fastboot controls, do let me know by dropping a comment below. I will reply you back as soon as possible. Like this post? Please do share the post link on
social media to thank me Was this article helpful?YesNo ADB commands provide access to a Unix Shell that runs a command device. An 'adb shell' command console. Thus, ADB Shell commands let you control your Android device. In this
article, we've compiled a huge list of ADB Shell commands that can perform amazing tasks like managing apps and files, debugging your device, enabling features, and tweaking it to get the most out of it. Please note that there are 2 prerequisites before you can utilize ADB and Fastboot commands. In this ADB Shell commands cheat
sheet, I'll try to explain the function of all commands with examples. adb shell This command activates the remote shell command console on the connected Android smartphone or tablet. Before you can execute an ADB Shell command window, press Enter, and type or paste the rest after the '$' sign. adbbe.
shell pm uninstall Using this command, followed by the app data and cache after package removal. If you want the app data to be cleared as well, use the following adb shell pm uninstall --user 0
com.android.chrome If you don't know the app package name for the apps to remove, use adb shell pm list packages to find it. This command can help you if you want to remove bloatware from your Android phone. Please note that most system apps don't have the 'Uninstall' option on the device but this command works magically.
Android", "Tutorials"]}]} data-page=1 data-max-pages=635 data-start=1 data-end=1> Using the above command, you can reinstall an uninstalled system app on your Android device, you can execute the above command
followed by the app package name adb shell pm clear -user 0 com.google.ar.core adb shell pm clear -user 0 com.goog
followed by the app package name. adb shell pm hide --user 0 com.whatsapp adb shell pm list packages. For instance, if you can print the list of app package names for all apps installed on your Android device. You can use this command with different parameters to get a more specific list of app packages. For instance, if you
want to list the system apps only, use adb shell pm list packages -r To list all third-party apps installed on your Android phone or tablet, you issue the following command. adb shell pm list packages -r To list all third-party apps installed on your Android phone or tablet, you issue the following command. adb shell pm list packages -r To list all third-party apps installed on your Android phone or tablet, you issue the following command.
the list of all enabled or disabled apps on your device? Try the command with parameters like '-d' (for enabled apps), and '-u' (for uninstalled apps), and '-u' (for uninstalled apps), and 'shell pm list packages -d adb shell pm list packages -d 
packages -i To list app packages with specific keyword filters. adb shell pm list packages -f You can easily get a list of group packages by a certain manufacturer or a common term. For instance, if you want to list all apps by Google,
you can use the following command. adb shell pm list packages | grep 'google' You can replace 'google' with 'samsung', 'huawei', 'xiaomi', 'miui', 'evenwell', 'android adb shell pm path This command displays the APK path on the
device's file system. adb shell pm create-user Use this command to create a new user on your Android device, you can use the above command followed by the user id. adb shell pm remove-user user 1 adb shell pm get-max-users
With this command, you can print the maximum number of users supported on an Android device, optionally only those in group. You can
use it with the following parameters. -g: Organize permissions -d: List the permissions -d: List the permissions only -u: List the permissions -d group adb shell pm list permissions -d -g adb shell pm path Get the path of a given app package. adb shell
pm path adb shell settings You can use this command to print information about specific settings on your Android device. By adding different parameters, you can find the Android settings provider, current wi-Fi status, etc. adb shell settings list
system adb shell settings get system volume system adb shell settings get global mobile data adb shell settings get global wifi on adb shell settings get secure adb shell settings get global wifi on adb shell settings get 
very flexible command that can be used standalone or with various parameters to get data related to battery, display, CPU, RAM, storage, etc. The execution of this command will give you detailed information about the Android
manifest automatically android.permission.DUMP adb shell dumpsys Other variations of the command are as follows: adb shell dumpsys display (get details about the display) adb shell dumpsys battery (get detailed info about your device's battery and
status) adb shell dumpsys batterystats (battery usage statistics) adb shell dumpsys cpuinfo (get a complete list of all ongoing activities on your device) adb shell dumpsys cpuinfo (get detailed about CPU usage by the running processes and apps on your
Android device, adb shell wm density The above command will print detailed info like pixel resolution, FPS, and DPI of your phone's display. adb shell wm size You can find out the display resolution of your phone with this
command. PS C:\Users\Technastic\Desktop> adb shell wm size Physical size: 1440x3040 Override size: 1080x2280 If you want to modify the screen resolution, execute the command given below. Suppose your phone's display resolution is
QHD+; you can easily change it to Full HD+ or HD+. adb shell wm density 360 adb shell wm density 420 adb shell wm density 420 adb shell wm density 360 adb shell wm density 360 adb shell wm density 420 adb shell wm density 420 adb shell wm density 360 adb shell wm density 360 adb shell wm density 420 adb shell wm density 360 adb shell wm density
01.png adb shell screenrecord ADB also lets you record your phone or tablet's screen and download the recorded video to your computer. Besides, you can also set conditions like video duration, resolution in pixels, video bitrate, etc. adb shell screenrecord /sdcard/screenrecord /sd
screen recording using Ctrl + C. If you want to record the screen in a specific resolution, the following command lets you set custom width and height in pixels. adb shell screenrecord --size 1920x1080 /sdcard/screenrecord -o1.mp4 Android's screen recorder's duration is set to 180 seconds (3 minutes) by default. You can decrease this time limit
according to your needs (180 seconds is the maximum limit). adb shell screenrecord --time-limit 120 /sdcard/screenrecord --time-limit 120 /sdcard/
getprop & adb shell setprop The 'getprop' and 'setprop' commands can be used to view and set or change the configuration of the 'build.prop' file on Android devices. The following command, for example, displays the Android system information. adb shell getprop Below are some more examples: getprop ro.build.version.sdk getprop ro.chipname In
case you want to change the value of an entry in the build.prop, you can use the adb shell setprop net.dns1 1.2.3.4 setprop net.dns1 1.3.4.5 getprop net.dns2 1.1.2.3 setprop net.dns2 1.2.3.4 In the same way, if you want to change the configuration of the VM heap size on your Android device, you can use
```

the following command. setprop dalvik.vm.heapsize 60m The adb shell getprop commands fetch information about Android system properties, SDK API level, Android security patch version, Soc, Android security patch version, device model, device

getprop ro.bootimage.build.fingerprint adb shell getprop ro.boot.wifimacaddr adb -s shell getprop To check the full configuration, running services, and information about your Android phone or tablet, use this command. First off, run the adb devices and copy the alpha-numeric value of your device ID from the output. PS C:\Users\Technastic\Desktop> adb devices List of devices attached RZ8M810BARJ device Then execute the following command. Don't forget to replace the device ID highlighted in blue with the ID of your device. adb -s RZ8M810BARJ shell getprop adb shell cat /proc/cpuinfo Use the above command to get complete information about the CPU on your phone or tablet. Manage App Permissions ADB makes managing app permissions on Android a breeze. Below are some examples. Reset Permissions adb shell pm revoke Get Device Properties By running the following command, you can see the system properties. adb shell getprop | grep -e 'model' -e 'version.sdk' -e 'manufacturer' -e 'platform' -e 'revision' -e 'serialno' -e 'product.name' -e 'product.n Android device's storage. Launch the command window, execute the 'adb shell' command, and then try the following command with '-f' (to delete a file) and '-d' (to remove a directory) parameters. rm -f /sdcard/WhatsApp Note: Instead of 'rm-d', you can also use 'rmdir'. adb shell mkdir Besides deleting an existing directory or folder, ADB Shell lets you create a new directory or subdirectory. Not only that, you can set permissions for the newly created folder. mkdir -m 644 /sdcard/NewFolder adb shell cp 'cp' stands for 'copy and allows you to copy files and directories on your Android device. To copy files and then paste them, mention the source and destination locations as shown below: cp /sdcard/OPWallpaperResources.apk /sdcard/DCIM/Camera adb shell mv 'mv' stands for 'move'. This command moves a file stored on your device from the source to a destination location. mv /sdcard/livewallpapers.apk /system/app The following command will allow you to move a file with a new name. mv /sdcard/livewallpapers.apk /sdcard/Wallpapers adb shell top This command displays the list of top CPU processes on an Android phone or tablet. The CPU processes on an Android phone or tablet. The CPU process monitor can be stopped using Ctrl + C. adb shell ip Find out the WiFi IP address of an Android phone or tablet. The CPU processes on an Android phone or tablet. adb shell netstat Displays the network statistics of Android phones. adb shell am start -a android.intent.action.CALL -d tel:+19797220011 Send SMS screen If you want to send a text message using a command, try the following code. adb shell am start -a android.intent.action.SENDTO -d sms:+19797220011 --es sms body "Test Message" --ez exit on sent false You can input or print text on your phone using the command. I tested this command in Messages, WhatsApp, Facebook, etc. If a messaging app is not open, the text will open in Google Search. adb shell input text 'I love this adb command will open the Gallery app. adb shell am start -t image/* -a android intent.action.VIEW ADB Shell Key Event Commands to Toggle and Trigger Functions Android devices support KeyEvent commands to Toggle and Trigger Functions Android devices support KeyEvent commands to Toggle and Trigger Functions Android phone or tablet device and even launch apps by using these KeyEvent commands. These commands might come in handy if the hardware keys on your device are dysfunctional due to physical damage. Turn Android device ON or OFF: adb shell input keyevent 2 Press App Switcher Button: adb shell input keyevent KEYCODE APP SWITCH Press Home button: adb shell input keyevent 3 Press Back button: adb shell input keyevent 4 Press Call button: adb shell input keyevent 5 End a call: adb shell input keyevent 5 End a call: adb shell input keyevent 6 Press Power Button to wake up screen: adb shell input keyevent 6 Press Power Button to wake up screen: adb shell input keyevent 6 Press Power Button to wake up screen: adb shell input keyevent 6 Press Power Button to wake up screen: adb shell input keyevent 6 Press Power Button to wake up screen: adb shell input keyevent 6 Press Power Button to wake up screen: adb shell input keyevent 6 Press Power Button to wake up screen: adb shell input keyevent 6 Press Power Button to wake up screen: adb shell input keyevent 6 Press Power Button to wake up screen: adb shell input keyevent 6 Press Power Button to wake up screen: adb shell input keyevent 6 Press Power Button to wake up screen: adb shell input keyevent 6 Press Power Button to wake up screen: adb shell input keyevent 6 Press Power Button to wake up screen: adb shell input keyevent 6 Press Power Button to wake up screen: adb shell input keyevent 6 Press Power Button to wake up screen: adb shell input keyevent 6 Press Power Button to wake up screen: adb shell input keyevent 6 Press Power Button to wake up screen: adb shell input keyevent 6 Press Power Button to wake up screen: adb shell input keyevent 6 Press Power Button to wake up screen: adb shell input keyevent 6 Press Power Button to wake up screen: adb shell input keyevent 6 Press Power Button to wake up screen: adb shell input keyevent 6 Press Power Button to wake up screen: adb shell input keyevent 6 Press Power Button to wake up screen: adb shell input keyevent 6 Press Power Button to wake up screen: adb shell input keyevent 6 Press Power Button to wake up screen: adb shell input keyevent 6 Press Power Button to wake up screen: adb shell input keyevent 6 Press Power Button to wake up screen: ada screen button to wake up screen button button input keyevent 66 Press Backspace button: adb shell input keyevent 27 Copy text: adb shell input keyevent 270 Decrease display brightness: adb shell input keyevent 270 Increase Display brightness: adb shell input keyevent 271 Copy text: adb shell input keyevent 278 Paste text: adb shell input keyevent 278 Paste text: adb shell input keyevent 279 Increase Display brightness: adb shell input keyevent 270 Increase Display brightness Display brightness Display brightness Display brightness Display brightness Display brightness Display b keyevent 279 Make the device sleep: adb shell input keyevent KEYCODE SLEEP Make device wakeup: adb shell input keyevent KEYCODE WAKEUP Toggle Power menu: adb shell input keyevent XEYCODE WAKEUP Toggle Power menu: adb shell input keyevent XEYCODE was also shell keyevent 168 Trigger Zoom out: adb shell input keyevent 82 Open Notifications: adb shell input keyevent 83 Launch Search: adb shell input keyevent 84 Play or pause media: adb shell input keyevent 85 Mute audio: adb shell input keyevent 91 Page up: adb shell input keyevent 92 Page down: adb shell input keyevent 84 Play or pause media: adb shell input keyevent 85 Mute audio: adb shell input keyevent 86 Mute audio: adb shell input keyevent 87 Page up: adb shell input keyevent 88 Play or pause media: adb shell input keyevent 89 Page down: adb shell input keyevent 80 Page up: adb shell input keyevent keyevent 93 Open Calendar; adb shell input keyevent 24 Make the lock screen sleep; adb shell input keyevent 224 Make device wakeup; adb shell input keyevent 224 Make device wakeup; adb shell input keyevent 227 Wakeup the lock screen; adb shell input keyevent 228 Wakeup the lock screen; adb shell input keyevent 228 Wakeup the lock screen sleep; adb shell input keyevent 229 Wakeup the lock screen sleep; adb shell input keyevent 229 Wakeup the lock screen sleep; adb shell input keyevent 229 Wakeup the lock screen sleep; adb shell input keyevent 220 Wakeup the lock scr KEYCODE_WAKEUP Toggle Power menu: adb shell input keyevent KEYCODE_POWER Android Keyboard-related Key Event Commands you can use the following key events commands to print letters, numbers, and symbols on your Android device right from your computer. ADB can help you wake up the device and unlock the screen PIN or pattern without touch input. Clear: adb shell input keyevent 28 Caps lock: adb shell input keyevent 11 Number '2': adb shell input keyevent 12 Number '6': adb shell input keyevent 11 Number '1': adb shell input keyevent 12 Number '1': adb shell input keyevent 12 Number '1': adb shell input keyevent 13 Number '1': adb shell input keyevent 14 Number '1': adb shell input keyevent 15 Number '1': adb shell input keyevent 16 Number '1': adb shell input keyevent 17 Number '1': adb shell input keyevent 18 Number '1': adb shell input keyevent 18 Number '1': adb shell input keyevent 19 Number '1': adb shell input keyevent 19 Number '1': adb shell input keyevent 10 Number '1': adb shell input key shell input keyevent 13 Number '7': adb shell input keyevent 14 Number '8': adb shell input keyevent 30 Letter 'b': adb shell input keyevent 32 Letter 'b': adb shell input keyevent 33 Letter 'f': adb shell input keyevent 36 Letter 'b': adb shell input keyevent 37 Letter 'b': adb shell input keyevent 38 Letter 'b': adb shell input keyevent 38 Letter 'b': adb shell input keyevent 39 Letter 'b': adb shell input keyevent 30 Letter 'b': adb shell shell input keyevent 34 Letter 'g': adb shell input keyevent 35 Letter 'h': adb shell input keyevent 36 Letter 'l': adb shell input keyevent 37 Letter 'g': adb shell input keyevent 41 Letter 'h': adb shell input keyevent 42 Letter 'o': adb shell input keyevent 38 Letter 'l': adb shell input keyevent 40 Letter 'g': adb shell input keyevent 40 Letter 'g': adb shell input keyevent 40 Letter 'h': adb shell input keyevent 43 Letter 'p': adb shell input keyevent 44 Letter 'q': adb shell input keyevent 45 Letter 'r': adb shell input keyevent 45 Letter 'r': adb shell input keyevent 46 Letter 'r': adb shell input keyevent 47 Key 't': adb shell input keyevent 48 Letter 'r': adb shell input key keyevent 52 Letter 'y': adb shell input keyevent 53 Key 'shift left': adb shell input keyevent 56 Key 'shift right': adb shell input keyevent 57 Key 'alt right': adb shell input keyevent 58 Key 'shift right': adb shell input keyevent 59 Key 'shift right': adb shell input keyevent 59 Key 'shift right': adb shell input keyevent 50 Key 'shift right': adb 'tab': adb shell input keyevent 61 Key 'space': adb shell input keyevent 62 Key 'symbols': adb shell input keyevent 70 Key 'left bracket': adb shell input keyevent 71 Key 'right bracket': adb shell input keyevent 72 Key 'backslash': adb shell input keyevent 73 Key 'left bracket': adb shell input keyevent 70 Key 'left 'semicolon': adb shell input keyevent 74 Key 'apostrophe': adb shell input keyevent 75 Key 'slash': adb shell input keyevent 76 Key '@': adb shell input keyevent 74 Key 'apostrophe': adb shell input keyevent 75 Key 'slash': adb shell input keyevent 76 Key '@': adb shell input keyevent 76 Key 'apostrophe': adb shell input keyevent 77 The following key event 76 Key 'apostrophe': adb shell input keyevent 76 Key 'apostrophe': adb shell input keyevent 77 The following key event 76 Key 'apostrophe': adb shell input keyevent 77 The following key event 78 Key 'apostrophe': adb shell input keyevent 78 Key 'apostrophe': adb shell input keyevent 78 Key 'apostrophe': adb shell input keyevent 79 Key 'apostrophe': adb shell input keyevent 70 Key 'apostrophe': ad keyevent 146 Number '3': adb shell input keyevent 147 Number '6': adb shell input keyevent 150 Number '6': adb shell input keyevent 151 Number '6': adb shell input keyevent 153 Symbol 'divide': adb shell input keyevent 154 Number '6': adb shell input keyevent 150 Number '6': adb she Symbol 'multiply': adb shell input keyevent 155 Symbol 'subtract': adb shell input keyevent 156 Symbol 'equals': adb shell input keyevent 157 Symbol 'dot': adb shell input keyevent 158 Symbol 'equals': adb shell input keyevent 157 Symbol 'dot': adb shell input keyevent 158 Symbol 'equals': adb she Commands List PDF Read Next: ADB Commands for Battery Optimization on Android Source: Google Developers Was this Article helpful?YesNo