
目录

前言	1.1
adb概述	1.2
adb shell	1.3
adb常用命令	1.4
概述	1.4.1
Fastboot	1.4.2
adb操作app	1.4.3
adb安装apk	1.4.3.1
adb卸载app	1.4.3.2
adb的help语法	1.5
附录	1.6
文档和资料	1.6.1
参考资料	1.6.2

安卓自动化工具：adb

- 最新版本： v0.4.0
- 更新时间： 20241129

简介

整理安卓自动化工具adb的各种相关内容。

源码+浏览+下载

本书的各种源码、在线浏览地址、多种格式文件下载如下：

HonKit源码

- [crifan/android_auto_tool_adb](#): 安卓自动化工具：adb

如何使用此HonKit源码去生成发布为电子书

详见：[crifan/honkit_template: demo how to use crifan honkit template and demo](#)

在线浏览

- 安卓自动化工具：[adb book.crifan.org](#)
- 安卓自动化工具：[adb crifan.github.io](#)

离线下载阅读

- 安卓自动化工具：[adb PDF](#)
- 安卓自动化工具：[adb ePub](#)
- 安卓自动化工具：[adb Mobi](#)

版权和用途说明

此电子书教程的全部内容，如无特别说明，均为本人原创。其中部分内容参考自网络，均已备注了出处。如发现有侵权，请通过邮箱联系我 [admin](#) 艾特 [crifan.com](#) ，我会尽快删除。谢谢合作。

各种技术类教程，仅作为学习和研究使用。请勿用于任何非法用途。如有非法用途，均与本人无关。

鸣谢

感谢我的老婆陈雪的包容理解和悉心照料，才使得我 [crifan](#) 有更多精力去专注技术专研和整理归纳出这些电子书和技术教程，特此鸣谢。

其他

作者的其他电子书

本人 [crifan](#) 还写了其他 [150+](#) 本电子书教程，感兴趣可移步至：

[crifan/crifan_ebook_readme](#): Crifan的电子书的使用说明

关于作者

关于作者更多介绍，详见：

[关于CrifanLi李茂 – 在路上](#)

crifan.org，使用[署名4.0国际\(CC BY 4.0\)](#)协议发布 all right reserved, powered by Gitbook最后更新：
2024-12-06 22:21:11

adb概述

TODO:

- 【整理】 adb命令语法帮助信息
- 【已解决】 Mac中运行adb shell无需su超级用户即可正常运行输出结果
- 【未解决】 尝试解决adb只有su超级用户权限才能工作： adb版本
- 【已解决】 Android Studio和安卓设备Google Pixel3重启后adb shell再次异常： 只有su超级用户权限才能正常运行命令
- 【已解决】 尝试解决adb没有su超级用户权限无法运行子命令： ps进程相关
- 【已解决】 Android Studio中对于ADB： Use existing manually managed server且指定adb端口
- 【已解决】 Mac中给adb start-server指定端口
- 【已解决】 Mac中adb shell子命令运行卡死无响应
- 【已解决】 安卓手机adb shell报错： Can't find service
- 【已解决】 安卓手机Google Pixel3中adb命令pm运行报错： cmd Can't find service: package
- 【已解决】 安卓手机adb shell service list报错： Found 0 services
- 【已解决】 安卓手机adb shell无法直接运行命令但是进入shell后su后却可以
- 【已解决】 adb shell带su超级用户权限去运行子命令
- 【未解决】 pm安装重签名的apk报错： Failure INSTALL_FAILED_UPDATE_INCOMPATIBLE
- 【已解决】 pm安装apktool重新打包的apk报错： Failure
INSTALL_PARSE_FAILED_NO_CERTIFICATES
- 【无需解决】 安卓手机pm卸载app报错： Failure DELETE_FAILED_INTERNAL_ERROR
- 【已解决】 安卓手机中用adb shell中的pm去安装YouTube的apk
- 【已解决】 安卓手机Google Pixel3中用pm安装YouTube的apk报错： Failure
INSTALL_FAILED_VERSION_DOWNGRADE
- 【整理】 adb命令语法帮助信息

-
- adb
 - 原本作用： 安卓正向开发期间， 自动化操作安卓设备
 - 典型用途
 - 安卓正向开发（期间， 自动化操作安卓设备）
 - 其他用途
 - 安卓逆向开发
 - 操作安卓设备
 - 安卓设备自动化操作
 - 群控

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新:
2024-12-06 22:03:36

adb shell

命令提示符

adb shell 进入shell后:

- 命令行提示符
 - # = 井号 : root 用户
 - \$ = 美元 符号: 普通 用户

另外, 也可以通过:

```
whoami
```

查看当前用户是什么

举例:

- root用户

```
blueline:/ # whoami  
root
```

- 普通用户: shell

```
13 blueline:/ $ whoami  
shell
```

adb shell命令行前面的数字

正常情况, shell前面是没有数字的:

```
└─ adb shell  
blueline:/ # pwd  
/
```

但是, 如果前面出现一个数字加上竖杠, 则表示: 前一次命令执行的返回值, 前一个命令运行出错了的出错码

比如:

```
blueline:/ # pm --help  
cmd: Can't find service: package  
20 blueline:/ #
```

此处的 20| 就是 前一个命令运行出错的返回值 = 出错码

而继续运行，如果后续命令正常运行，则出错码就消失了：

```
20 blueline:/ # which pm
/system/bin/pm
blueline:/ #
```

其实表示的是：

- 上一个命令运行结果=返回值
 - 0：表示没有出错
 - 所以就不显示出错码
 - 非0：就显示，提示你出错了

```
crifan@licrifandeMacBook-Pro ~/dev/dev_tool/android/reverse_engineering/adb_root adb shell
blueline:/ # pwd
/
blueline:/ # pm --help
cmd: Can't find service: package
20|blueline:/ # which pm
/system/bin/pm
blueline:/ #
```

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新:
2024-11-29 18:05:24

adb常用命令

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新:
2024-11-29 17:59:08

adb常用命令概述

命令	说明
adb devices	显示所有连接的 adb 兼容设备
adb push	将文件从 电脑端 复制=下载到Android手机的位置
adb pull	将文件从 Android 复制到电脑端的系统
adb install	将应用程序从系统的 apk 文件位置安装到 Android 设备上
adb backup	备份 Android 设备
adb connect	通过 WiFi 网络使用adb命令
adb shell screencap	获取设备的屏幕截图
adb reboot	在正常模式下，重新启动 Android 手机
adb reboot bootloader	启动设备（以）进入 Fastboot 模式

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新:
2024-11-29 18:10:21

Fastboot

- 常用 Fastboot 命令

◦

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新:
2024-11-29 18:11:19

adb操作app

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新:
2024-11-29 17:55:14

adb安装apk

- 语法

```
adb install xxx.apk
```

- == `adb push xxx.apk somePath + pm install /somePath/xxx.apk`

举例

```
crifan@licrifandeMacBook-Pro ~/dev/dev_tool/android/EdXposed` pwd
/Users/crifan/dev/dev_tool/android/EdXposed
crifan@licrifandeMacBook-Pro ~/dev/dev_tool/android/EdXposed` ll
total 8224
crifan@licrifandeMacBook-Pro ~/dev/dev_tool/android/EdXposed` adb install EdXposedManager-4.6.2-46200-org.meowcat.edxposed.manager-release.apk
Performing Streamed Install
Success
```

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新:
2024-11-29 17:56:40

adb卸载app

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新:
2024-11-29 18:00:12

adb的help语法

adb语法

```

> adb --help
Android Debug Bridge version 1.0.41
Version 33.0.2-8557947
Installed as /Users/crifan/dev/dev_tool/android/AndroidSDK/platform-tools/adb

global options:
-a                listen on all network interfaces, not just localhost
-d                use USB device (error if multiple devices connected)
-e                use TCP/IP device (error if multiple TCP/IP devices available)

-s SERIAL        use device with given serial (overrides $ANDROID_SERIAL)
-t ID            use device with given transport id
-H               name of adb server host [default localhost]
-P               port of adb server [default=5037]
-L SOCKET        listen on given socket for adb server [default tcp:localhost:
5037]
--one-device SERIAL USB only allowed with 'start-server' or 'server nodaemon', server
will only connect to one USB device, specified by a serial number or USB device address.
--exit-on-write-error exit if stdout is closed

general commands:
devices [-l]     list connected devices (-l for long output)
help            show this help message
version         show version num

networking:
connect HOST[:PORT] connect to a device via TCP/IP [default port=5555]
disconnect [HOST[:PORT]]
disconnect from given TCP/IP device [default port=5555], or all
pair HOST[:PORT] [PAIRING CODE]
pair with a device for secure TCP/IP communication
forward --list   list all forward socket connections
forward [--no-rebind] LOCAL REMOTE
forward socket connection using:
tcp: port> ( local> may be "tcp:0" to pick any open port)
localabstract: unix domain socket name
localreserved: unix domain socket name
localfilesystem: unix domain socket name>
jdpw: process pid> (remote only)
vsoc: CID: port> (remote only)
acceptfd: fd> (listen only)
forward --remove LOCAL remove specific forward socket connection
forward --remove-all remove all forward socket connections
ppp TTY [PARAMETER...] run PPP over USB

```

```
reverse --list          list all reverse socket connections from device
reverse [--no-rebind] REMOTE LOCAL
    reverse socket connection using:
        tcp: port> ( remote> may be "tcp:0" to pick any open port)
        localabstract: unix domain socket name
        localreserved: unix domain socket name
        localfilesystem:<unix domain socket name>
reverse --remove REMOTE remove specific reverse socket connection
reverse --remove-all   remove all reverse socket connections from device
mdns check              check if mdns discovery is available
mdns services           list all discovered services
```

file transfer:

```
push [--sync] [-z ALGORITHM] [-Z] LOCAL... REMOTE
    copy local files/directories to device
    --sync: only push files that are newer on the host than the device
    -n: dry run: push files to device without storing to the filesystem
    -z: enable compression with a specified algorithm (any/none/brotli/lz4/zstd)
    -Z: disable compression
pull [-a] [-z ALGORITHM] [-Z] REMOTE... LOCAL
    copy files/dirs from device
    -a: preserve file timestamp and mode
    -z: enable compression with a specified algorithm (any/none/brotli/lz4/zstd)
    -Z: disable compression
sync [-l] [-z ALGORITHM] [-Z] [all data odm oem product system system_ext vendor]
    sync a local build from $ANDROID_PRODUCT_OUT to the device (default all)
    -n: dry run: push files to device without storing to the filesystem
    -l: list files that would be copied, but don't copy them
    -z: enable compression with a specified algorithm (any/none/brotli/lz4/zstd)
    -Z: disable compression
```

shell:

```
shell [-e ESCAPE] [-n] [-Tt] [-x] [COMMAND...]
    run remote shell command (interactive shell if no command given)
    -e: choose escape character, or "none"; default '~'
    -n: don't read from stdin
    -T: disable pty allocation
    -t: allocate a pty if on a tty (-tt: force pty allocation)
    -x: disable remote exit codes and stdout/stderr separation
emu COMMAND          run emulator console command
```

app installation (see also `adb shell cmd package help`):

```
install [-lrtsdg] [--instant] PACKAGE
    push a single package to the device and install it
install-multiple [-lrtsdpg] [--instant] PACKAGE...
    push multiple APKs to the device for a single package and install them
install-multi-package [-lrtsdpg] [--instant] PACKAGE...
    push one or more packages to the device and install them atomically
    -r: replace existing application
    -t: allow test packages
    -d: allow version code downgrade (debuggable packages only)
    -p: partial application install (install-multiple only)
    -g: grant all runtime permissions
```

```

--abi ABI: override platform's default ABI
--instant: cause the app to be installed as an ephemeral install app
--no-streaming: always push APK to device and invoke Package Manager as separate steps
--streaming: force streaming APK directly into Package Manager
--fastdeploy: use fast deploy
--no-fastdeploy: prevent use of fast deploy
--force-agent: force update of deployment agent when using fast deploy
--date-check-agent: update deployment agent when local version is newer and using fast deploy
--version-check-agent: update deployment agent when local version has different version code and using fast deploy
--local-agent: locate agent files from local source build (instead of SDK location)

(See also `adb shell pm help` for more options.)
uninstall [-k] PACKAGE
remove this app package from the device
'-k': keep the data and cache directories

debugging:
bugreport [PATH]
write bugreport to given PATH [default=bugreport.zip];
if PATH is a directory, the bug report is saved in that directory.
devices that don't support zipped bug reports output to stdout.
jdpw                list pids of processes hosting a JDWP transport
logcat              show device log (logcat --help for more)

security:
disable-verity      disable dm-verity checking on userdebug builds
enable-verity       re-enable dm-verity checking on userdebug builds
keygen FILE
generate adb public/private key; private key stored in FILE,

scripting:
wait-for[-TRANSPORT]-STATE...
wait for device to be in a given state
STATE: device, recovery, rescue, sideload, bootloader, or disconnect
TRANSPORT: usb, local, or any [default any]
get-state           print offline | bootloader | device
get-serialno        print serial-number>
get-devpath         print device-path
remount [-R]
remount partitions read-write. if a reboot is required, -R will
will automatically reboot the device.
reboot [bootloader recovery sideload sideload-auto-reboot]
reboot the device; defaults to booting system image but
supports bootloader and recovery too. sideload reboots
into recovery and automatically starts sideload mode,
sideload-auto-reboot is the same but reboots after sideloading.
sideload OTAPACKAGE sideload the given full OTA package
root               restart adbd with root permissions
unroot             restart adbd without root permissions
usb               restart adbd listening on USB

```

```
tcpip PORT                restart adbd listening on TCP on PORT

internal debugging:
start-server              ensure that there is a server running
kill-server              kill the server if it is running
reconnect                kick connection from host side to force reconnect
reconnect device         kick connection from device side to force reconnect
reconnect offline        reset offline/unauthorized devices to force reconnect

usb:
attach                   attach a detached USB device
detach                   detach from a USB device to allow use by other processes

environment variables:
$ADB_TRACE               comma-separated list of debug info to log:
                        all,adb,sockets,packets,rwx,usb,sync,sysdeps,transport,jdwp
$ADB_VENDOR_KEYS         colon-separated list of keys (files or directories)
$ANDROID_SERIAL          serial number to connect to (see -s)
$ANDROID_LOG_TAGS        tags to be used by logcat (see logcat --help)
$ADB_LOCAL_TRANSPORT_MAX_PORT max emulator scan port (default 5585, 16 emus)
$ADB_MDNS_AUTO_CONNECT   comma-separated list of mdns services to allow auto-connect (
default adb-tls-connect)
```

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新:
2024-11-29 17:58:04

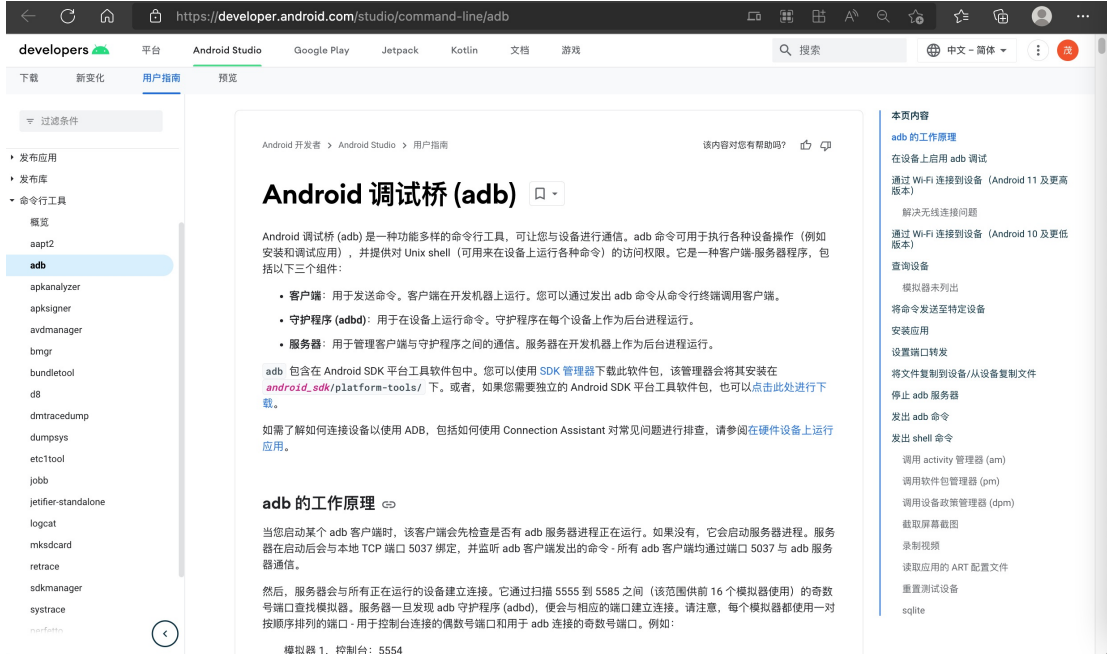
附录

下面列出相关参考资料。

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新:
2022-03-17 20:39:28

adb文档和资料

- 官网文档
 - [Android 调试桥 \(adb\) | Android 开发者 | Android Developers](https://developer.android.com/studio/command-line/adb)



- 其他资料
 - [Home | Android Debug Bridge \(ADB\) Commands Manual \(adbcommand.com\)](https://adbcommand.com/)
- Listing of adb Commands
 - [Android Debug Bridge | Android Developers \(oschina.net\)](https://oschina.net/)

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新:
2024-11-29 18:12:19

参考资料

-

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新:
2024-11-29 17:51:39