HOW TO PORT CUSTOM ROMS/RECOVERIES CUSTOM ROM RELATED STUFFS

Credits
smily9000

What You Need

- You need CRB Kitchen (download) (search as crb xda)
  - Windows 10/11 need to use wls2
  - 8GB ram
- Linux arch For unpack F2FS or donated version of CRB Kitchen
- Notepad ++ or xml/prop editor
- Winmerge
- 7Zip ZS
- Android Image Kitchen
- Mixlporer

Downloads

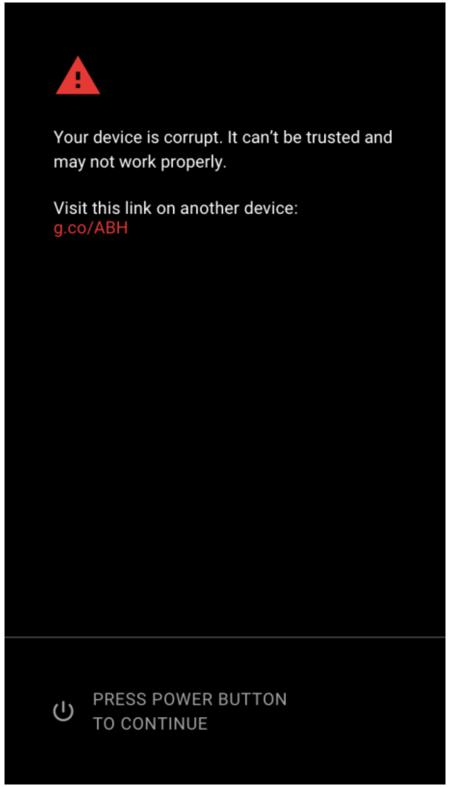
https://drive.google.com/drive/folders/1aQFhMCZt0XTEjgw1dn1DCf6pmaqtvxP6?usp=drive_link

Main Topics

Main Topics

- Identify Your Device
    - Arm32/64 VNDK A/B Super
    - File Systems and which FS you can boot
    - How to Choose Vendor

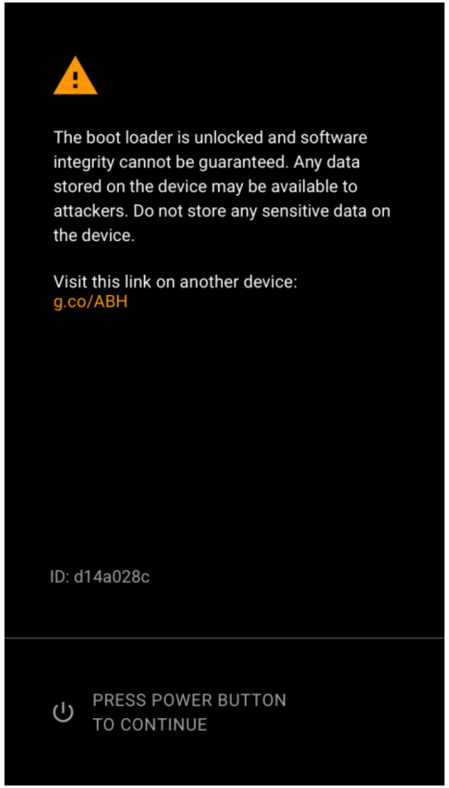
Three Stages


Your device is corrupt. It can't be trusted and may not work properly.

Visit this link on another device:
g.co/ABH

PRESS POWER BUTTON TO CONTINUE

Red State

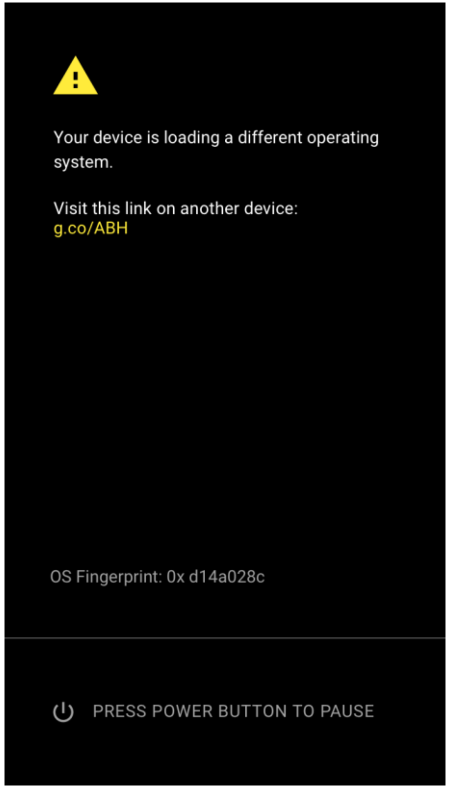

The boot loader is unlocked and software integrity cannot be guaranteed. Any data stored on the device may be available to attackers. Do not store any sensitive data on the device.

Visit this link on another device:
g.co/ABH

ID: d14a028c

PRESS POWER BUTTON TO CONTINUE

Orange State

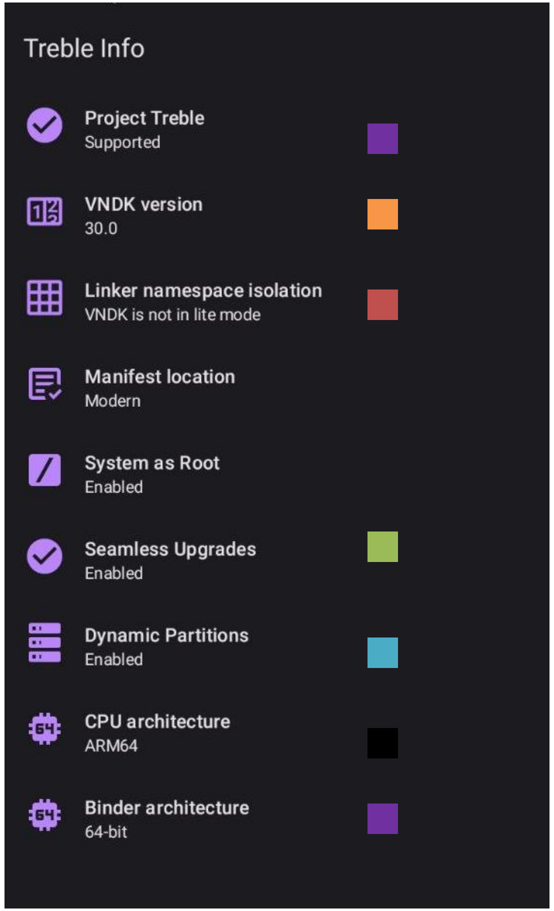

Your device is loading a different operating system.

Visit this link on another device:
g.co/ABH

OS Fingerprint: 0x d14a028c

PRESS POWER BUTTON TO PAUSE

Yellow State


Treble Info

Project Treble	Supported	🟪
VNDK version	30.0	🟧
Linker namespace isolation	VNDK is not in lite mode	🟥
Manifest location	Modern	
System as Root	Enabled	
Seamless Upgrades	Enabled	🟩
Dynamic Partitions	Enabled	🟦
CPU architecture	ARM64	⬛
Binder architecture	64-bit	🟪

Identify Your device

🟪 You can boot gsis

🟧 You have android 11 vendor (30)

- 33 android 11 (you can boot only android 13+)
- 32 android 12.1 (L) (you can boot only android 12.1+)
- 31 android 12.0 (you can boot only android 12.0+)
- 30 android 11 (you can boot only android 11+)
- 29 android 10 (you can boot only android 10+)
- 28 android 9 (you can boot only android 9+)
- 27 android 8.1 (you can boot only android 8.1+)
- 26 android 8.0 (you can boot only android 8.0+)

🟥 Full VNDK (vendor native development kit)

🟩 You have a/b device (boot_a/boot_b)(slot select)

🟦 You have super partition

⬛ You have arm64 cpu

🟪 Now you are on arm64 rom

File systems used by OEMS

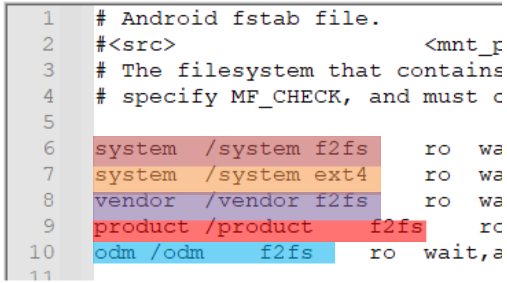
- Ext4 (read and write or readonly file system)
  - most of devices shipped with android below android 12.1
  - You can unpack it via 7zip or ext4 explore like disk repairing/partitioning tool (disk digger) and with free version of crb

- F2FS (readonly file system)
  - most of device shipped with android android 12.1 and some devices updated to android 12.1 via ota
  - You need linux arch to unpack that or donated version of crb

- ERFOS (readonly file system)
  - Huawei devices above emui 10
  - Some xiaomi devies (redmi A1)
  - Some Samsung Devices
  - You need donated version on crb to unpack and repack

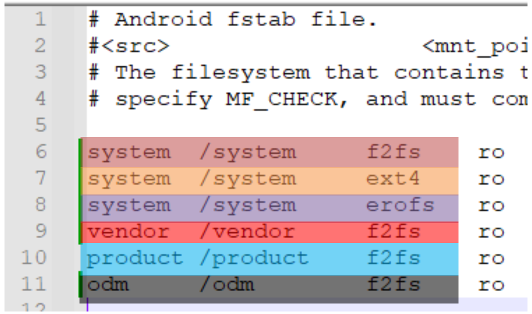
How to identify what file systems can you boot

- You have to check fstab from vendor/etc.



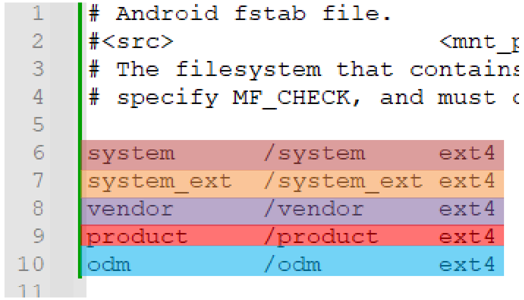
```
# Android fstab file.
#<src>                  <mnt_p
# The filesystem that contains
# specify MF_CHECK, and must c

system  /system f2fs     ro   wa
system  /system ext4     ro   wa
vendor  /vendor f2fs     ro   wa
product /product   f2fs       rc
odm /odm     f2fs    ro   wait,a
```

- 🟥 You can boot f2fs system images
- 🟧 You can boot ext4 system images
- 🟪 You can boot only f2fs vendor images
- 🟥 You can boot only f2fs product images
- 🟦 You can boot only f2fs odm images



```
# Android fstab file.
#<src>                  <mnt_poi
# The filesystem that contains t
# specify MF_CHECK, and must con

system   /system      f2fs    ro
system   /system      ext4    ro
system   /system      erofs   ro
vendor   /vendor      f2fs    ro
product  /product     f2fs    ro
odm      /odm         f2fs    ro
```

- 🟥 You can boot f2fs system images
- 🟧 You can boot ext4 system images
- 🟪 You can boot erofs system images
- 🟥 You can boot only f2fs vendor images
- 🟦 You can boot only f2fs product images
- ⬛ You can boot only f2fs odm images



```
# Android fstab file.
#<src>                  <mnt_p
# The filesystem that contains
# specify MF_CHECK, and must c

system       /system       ext4
system_ext   /system_ext   ext4
vendor       /vendor       ext4
product      /product      ext4
odm          /odm          ext4
```

- 🟥 You can boot only ext4 system images
- 🟧 You can boot only ext4 system_ext images
- 🟪 You can boot only ext4 vendor images
- 🟥 You can boot only ext4 product images
- 🟦 You can boot only ext4 odm images

How to Choose Vendor

How to Choose Vendor

You have to select a device with,

Same android os
Same kernel version
Same family soc.
Same hardware(compass/gravity/proximity/nfc/gyro/)
Wifi only/LTE

If your device dont have nfc/gyro/compass like feature but you can use device which has nfc/gyro/compass for port.

If your device do have nfc/gyro/compass like feature but you used device which has no nfc/gyro/compass for port. Then in that vendor there is no nfc/gyro/compass drives so they will not work on your device and fixing might be very hard

If you have samsung you have to use samsung device if you use ril (netowrk/calls/mobile data/sms)

How to Choose Vendor

My Device	Devices can used for port vendors				
Samsung Galaxy M01/A01 Android 10/11/12 kernel 4.9 sdm439	Samsung Galaxy A02s/M02s Android 10/11/12 kernel 4.9 sdm450	Samsung Galaxy A20s Android 10/11 kernel 4.9 sdm450	Samsung Galaxy A11/M11 Android 10/11/12 kernel 4.9 sdm450	Samsung Galaxy Tab A 8.0 Android 10/11 kernel 4.9 sdm429	Redmi 8a/7a Android 10/11/12 kernel 4.9 Sdm439 (if i use this ril is much hard to fix)

Mediatek P22 to P22
Mediatek P22 to P35
Exyons 850 to 850
MT6572 to MT6582/92/80
SC8830 to SC7731

Devices cannot used for port vendors	
Samsung Galaxy J8 Android 10 kernel 3.18 sdm450	Samsung Galaxy Jene Android 10 kernel 3.18 sdm450

Flashing Custom Images

- Vbmeta

Flashing Custom Images

- First you have to flash empty vbmeta partitions (vbmeta_system/vbmeta/vbmeta_vendor) . You can use odin or fastboot or fastbootd or mtk client or edl or any flash tool

- You have to use fastbootd for flash custom system.img/vendor.im etc.
- You need Custom recovery or stock recovery with fastbootd. But Some OEMs they dont include fastbootd (samsung)
- Reboot to fastbootd and install latest sdk drivers to use fastbootd
- You can only resize system.img/vendor.img/etc via fastbootd
- TWRP cannot resize system.img/vendor.img/etc

- But in some samsung devices you can enable fastbootd by this tool

VBMETA

- Most of time you need empty vbmeta for boot custom images (gsis/custom recoveries)
- You have to patch your all vbmeta (vbmeta , vbmeta_system, vbmeta_vendor)files via crb and flash them to your device
- If not you have to use magisk patched boot.img

How to Unpack Super.img

How to Unpack super.img.lz4

- Unpack super.img and get vendor.img odm.img product.img system.img etc
- If you have super.img.lz4 , then unpack it
- You can use simg2img.exe to convert super.img to raw image
  - Open cmd where simgimg.exe and super.img located
  - Then excute simg2img.exe super.img super.raw
- if you dont have super.img just unpack vendor.img
  - Then excute simg2img.exe vendor.img vendor.raw
- Then open it in # in 7z Ez
- There will be all partitions
- Extract them and rename partitions according to partition name
- if your device is ext4 you can open them in 7zip

How to Unpack Super or System/Vendor/Product (EXT4)

https://www.youtube.com/watch?v=1lH7AvpwjCw

How to Convert F2FS to EXT4

Convert F2FS to Ext4

- I used Fedroa
- Duel boot or install it on vbox or any
- Open Terminal
  - Open teminal form where located raw_f2fs_system.img (system/vendor/product/odm/…)
  - sudo apt install f2fs-tools
  - mkdir system
  - sudo mount -o ro -t auto name_of_raw_f2fs_system.img system
  - sudo dd if=/dev/zero of=system_new.img bs=6k count=1048576 (bs=6k 6gb)
  - sudo mkfs.ext4 system_new.img
  - sudo tune2fs -c0 -i0 system_new.img
  - mkdir new
  - sudo mount -o rw,sync,loop system_new.img new
  - sudo cp -fva system/* new/

You will get a new 6GB ext4 image

Credits
@nnippon

How to Port Vendor

How to Port Vendor

- unpack both vendors (stock and vendor used to be port)
- you can use CRB kitchen if your device is ext4
- you have to replace whole firmware folder and make fstab for basically boot that vendor
- replace all vendor/etc/fstab.xx with stock
- you have to replace mounting fstab lines with stock lines
  - open vendor/etc/init/hw/init_target.rc and compare it with winmerge
  - Replace all fstab related line (mount_all /vendor/etc/)
  - Replace executing moudules line with stock.
  - exec u:r:vendor_modprobe:s0 -- /vendor/bin/modprobe -a -d /vendor/lib/modules ...
- check vendor/lib/modules/*.ko
- they are the kernel drivers (bluetooth/wifi/touch/graphic/fm_radio/gps/etc)
- if there is no modules ignore this.then you kernel dont use any modules
- replace those .ko files with stock files
- Resize Vendor if needed
- repack vendor and flash and try

- You can use this method to port arm64 vendor to arm32 device but you need a arm64 kernel. Most of time stock kernel is arm64. you can check it by kernel config

- If you port originaly relased vendor from oem, (oneui/miui/...)

- if you ported android 10 vendor you must use your stock android 10 boot.img
- if you ported android 11 vendor you must use your stock android 11 boot.img
- if you ported android 12 vendor you must use your stock android 12 boot.img
- if you ported android 13 vendor you must use your stock android 13 boot.img

- If you port custom vendor from lineageos or anyother,

- if you ported android 10 vendor you have to check with your stock android 10 boot.img
- if you ported android 11 vendor you have to check with your stock android 10/11 boot.img
- if you ported android 12 vendor you have to check with your stock android 10/11/12 boot.img
- if you ported android 12 vendor you have to check with your stock android 10/11/12 boot.img
- if you ported android 13 vendor you have to check with your stock android 10/11/12/13 boot.img

How to Port Basic Vendor

https://www.youtube.com/watch?v=bsXy7sxz218

How to Check Kernel is arm64 or Arm32

- You need you stock boot.img or recovery.img
- Open it using 7zip #
- Then open 2.gz (gz may be different) file then reopen 2 file again via #
- Then open 4.gz (gz may be different) then open 4 file as a text .
- Then check if it has these configs, (search for that ctrl+f)
  - CONFIG_ARM64=y
  - CONFIG_64BIT=y
- If not your kernel is arm32 . Then you have to rebuild arm64 kernel for boot arm64

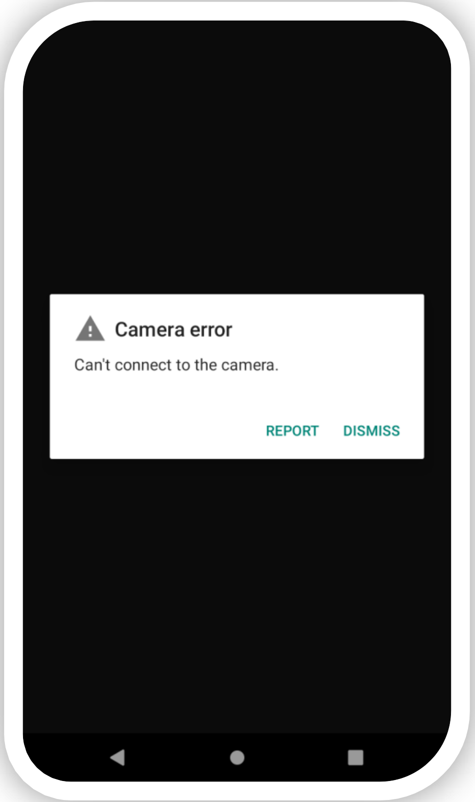
How to Check Kernel is arm64 or Arm32

https://www.youtube.com/watch?v=80VaUlMsxvk

Bug Fixing

Bug Fixing

- To fix camera you have to use correct camera sensnors lib files (depend on device)
- Replace with stock
- camera configs are on,
  - vendor/etc/camera
- camera libs on (lib(64) or lib(64)/camera)
  - dualcali_golden_*.bin
  - libactuator_*.so
  - libarcsoft_*.so
  - libchromatix_*.so
  - libmmcamera_*.so
  - libmtkcamera_*.so
  - libcamera*.so
  - camera*.so
- Maybe some lines in init/hw/init*.rc
- they are depend on soc and camera sensors


Camera error

Can't connect to the camera.

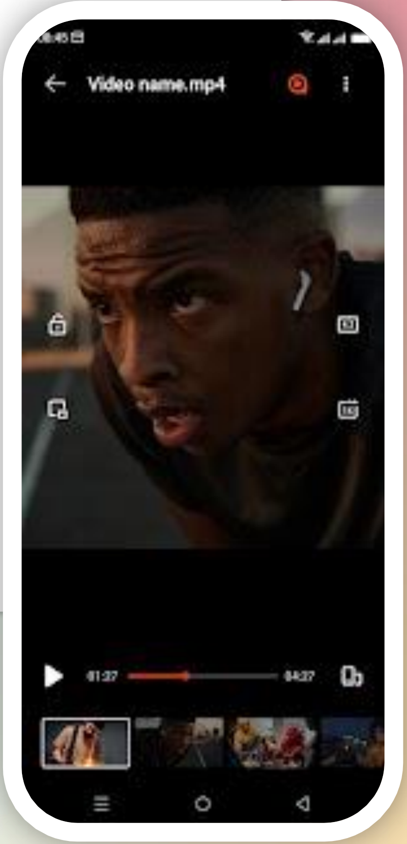
Bug Fixing

- To fix Codecs (vedio playing issues) (Replace with stock)

- vendor/libs(64)
  - libstagefright_*.so
  - libOmx*.so

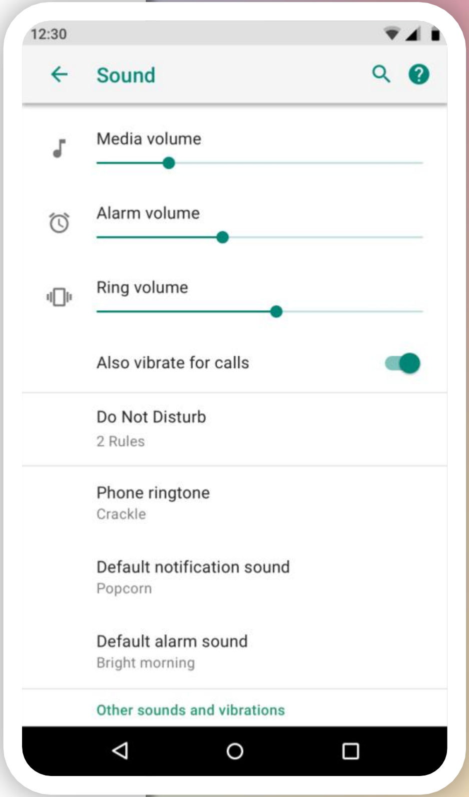
- vendor/etc
  - media_*.xml

- they are depend on soc and oem



Bug Fixing

- To fix Audio (Replace with stock)
- lib(64)/
  - audio.primary.*.so
  - audio.*.so
- vendor/etc
  - audio_*.conf
  - audio_*.xml
  - audio_*.txt
  - default_volume_tables.xml
  - dax3_media_codecs_dolby_audio.xml
  - hearing_aid_audio_policy_configuration.xml (bt audio and earphone)
  - mixer_*.xml
  - sound_*.xml
  - usb_audio_policy_configuration.xml
  - playback_record_audio_policy_configuration
- vendor/etc/audio/*
- whole acdbdata folder in vendor/etc/
- they are depend on soc and oem


12:30

Sound

Media volume

Alarm volume

Ring volume

Also vibrate for calls

Do Not Disturb
2 Rules

Phone ringtone
Crackle

Default notification sound
Popcorn

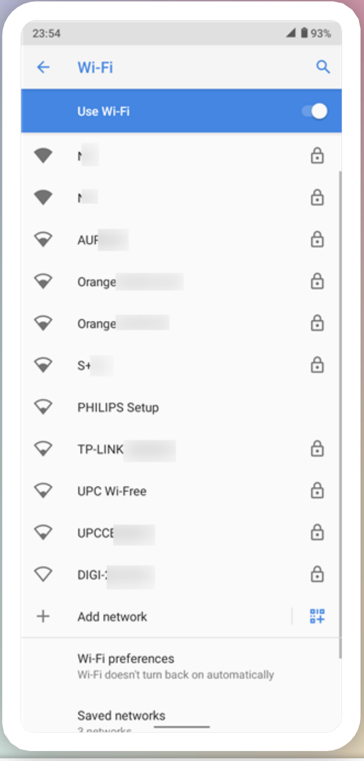
Default alarm sound
Bright morning

Other sounds and vibrations

Bug Fixing

- To Fix Graphics (Replace with stock)
- lib(64)/hw
  - gralloc.*.so
- lib(64)/egl/*
- lib(64)
  - egl*.so

- To Fix Wifi (Replace with stock)
- vendor/etc/wifi/*
- and lib/module/.ko (wlan driver) (depend on wlan ic)
- lib(64)
  - vendor.oem.hardware.wifi*.so
- bin
  - vendor.oem.hardware.wifi*
  - wpa_supplicant



Maybe some lines in init/hw/init*.rc
they are depend on soc and network ic

Bug Fixing

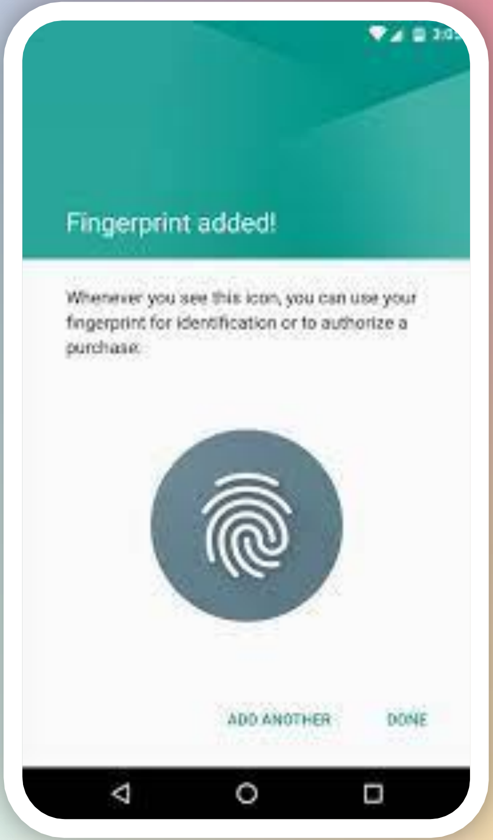
- To Fix Sensors (Replace with stock)
- lib(64)
  - sensors*.so
- bin
  - sensors*
- etc/
  - sensors*

- Maybe also some lines in init/hw
- They are depend on soc and OEM

Bug Fixing

- To Fix Fingerprint (Replace with stock)
- lib(64)
  - *biometric*.so
  - fingerprint*.so
- bin/
  - *biometric*
- lib(64)/hw/
  - *biometric*.so
  - fingerprint*.so

- Maybe also some lines in init/hw
- They are depend on soc and OEM


Fingerprint added!

Whenever you see this icon, you can use your fingerprint for identification or to authorize a purchase.

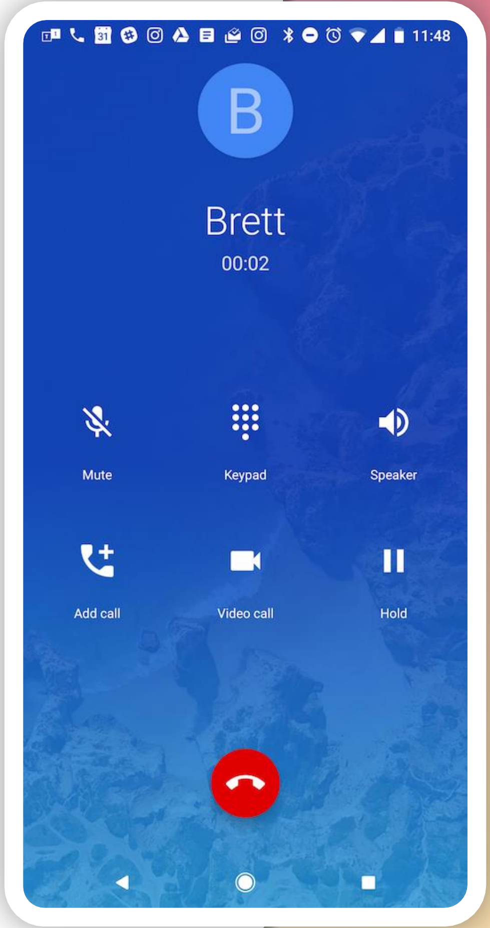
Bug Fixing

RIL

- There are some kind of ril
- Mediatek MTK_RIL
  - (used by mtk devices,redmi/xaiomi/huweai/reamle/vivo/oppo/nokia...)
- Qualcomm QCOM_RIL (QCRIL)
  - (used by qcom devices - redmi/xaiomi/huweai/reamle/vivo/oppo/nokia...)
- Unisoc- (idk the name of .so files)
  - (used by unisoc devices - redmi/xaiomi/huweai/reamle/vivo/oppo/nokia...)
- Samsung RIL (SEC_RIL)
  - (used by mtk/qcom/unisoc/exyons samsung devices)
- LG RIL - LGE_RIL
  - (used by mtk/qcom/unisoc/exyons lg devices)

Bug Fixing

- To Fix RIL (Replace with stock)
- lib(64)/
  - ril*.so
  - radio*.so
- bin/
  - ril*
  - radio*
  - modem*
  - baseband*.
- vendor/radio/*
- vendor/etc/init/
  - ril*.rc
  - modem*.rc
  - baseband.*.rc
- build.prop (ril*/radio* related line)
- Must be some lines in init/hw/init*.rc
- Highly depend on soc and oem (samsung is very diffrenet)



How to Port Recovery

How to port recovery (bugs depend on oem)

- Decryption is most common bug (data partition is not showing any files or folders with random letters)
- Touch (lack of firmware file)

What you need,

- you have to use recovery with same family soc,
- if your device has super, you have to use super partitioned device
- if your device doesnt have super, dont use recovery.img from a super partitioned device

- If you ignore the above 2 lines , you cannot mount system product etc in recovery

- Take 2 recovery images (stock and recovery you want to port)
  - eg – stock recovery i renamed as stock.img
  - recovery need to port i renamed as port.img
- Copy both images to root of Android Image Kitchen
- Then first drag stock.img to unpack.bat then you will get 2 folders as ramdisk and split_image rename ramdisk to stock_ram and split_image to stock_split_image.
- Then delete stock.img and rename port.img to stock.img
- Then drag stock.img (renamed port.img) to unpack.bat then you will get 2 folders as ramdisk and split_image
- Open stock_split_image and delete stock.img-ramdisk.cpio.gz (.gz extension may different)
- Then replace all files in split_image with stock_split_image.
- Then replace recovery.fstab with stock fstab (ramdisk/etc or ramdisk/system/etc)
- If you have twrp.fstab or twrp.flags also replace them with stock recovery.fstab
- Click on repack.bat

Bug Fixing

- Touch
- You have to add vendor/frimware
- You have to add touch driver lib/modules/.ko (if your device use touch driver)

How to port recovery

https://www.youtube.com/watch?v=ZJc_SKpJD9o

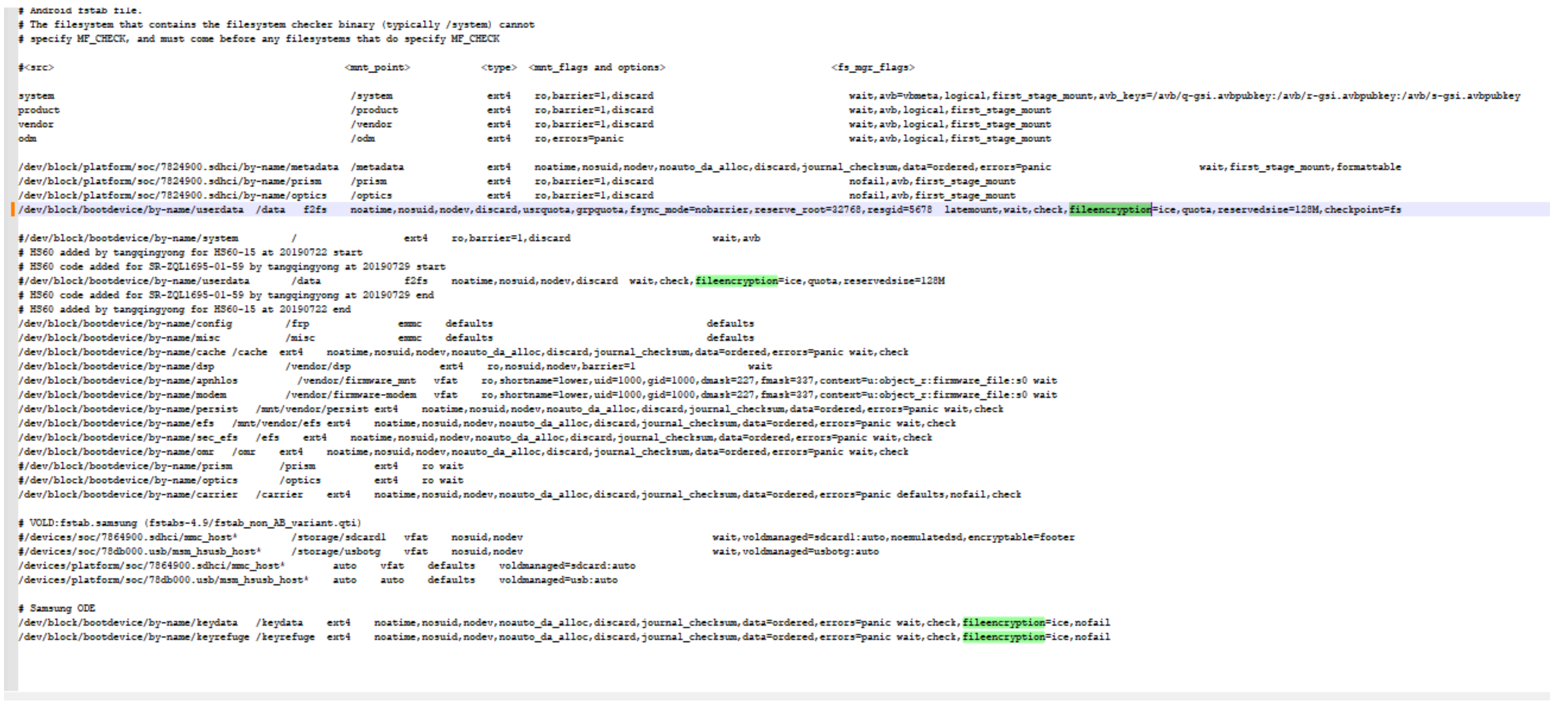
Encryption/Decryption

Decryption/Encyption

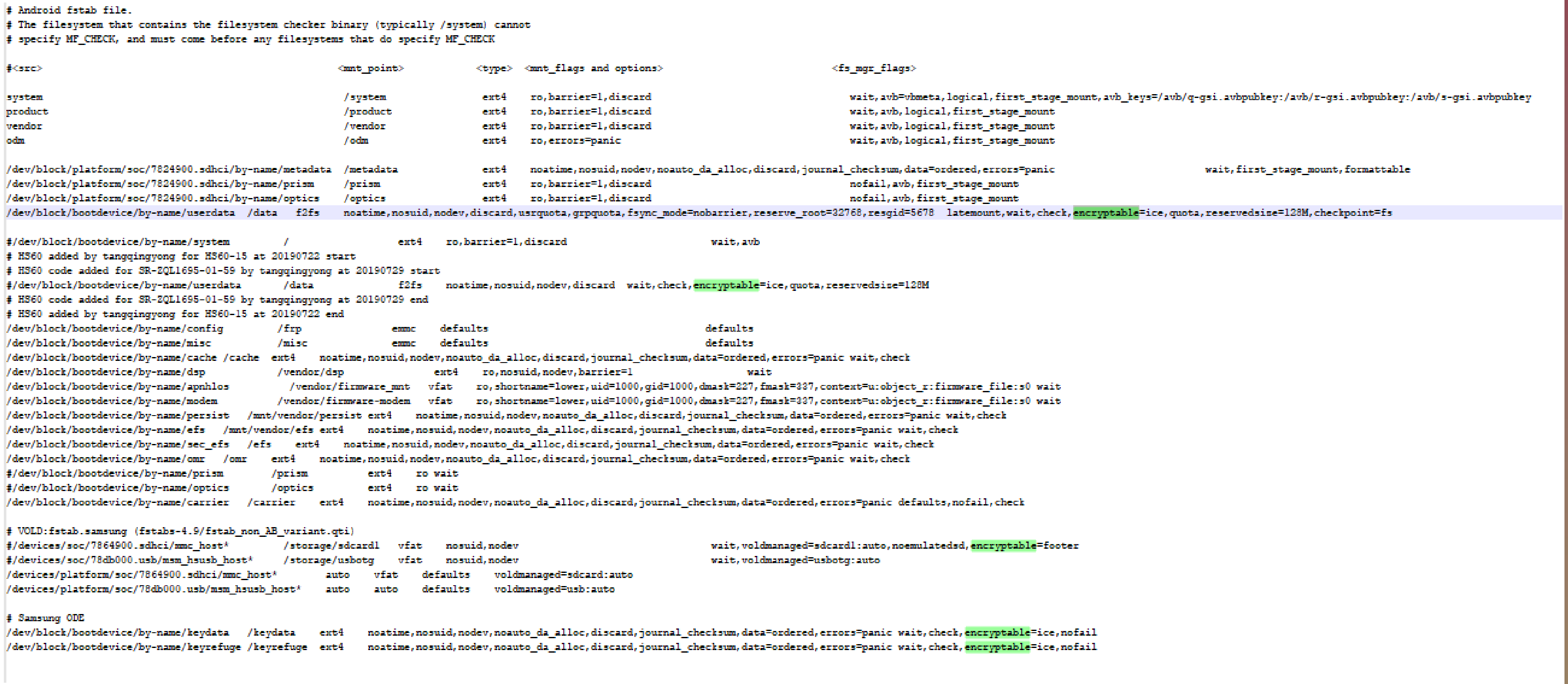
- After android 7 android uses FED (Full disk encryption)
- After android 9 they use FBE (File Based encryption)
- You need to add keymaster/gatekeeper libs and other encyption related libs and bin file to make encryption decryption working TWRP/OFOX/SHRP/PBR
- If decryption encryption not work in custom recovery,
    - You cannot backup/restore data partiton via custom recovery
    - You cannot use internal storage in TWRP
        - Maybe data cannot mount
        - Maybe data partition has 0 byte
        - Maybe you can see folders random letters
- Some OEM Gsis cannot boot with encrypted data partition
    - So you have to disable data encryption
- When decrypted some brands disable some funtions (depend on oem)
    - Lock Screen (Fingerprint/pin/pattern/face unlock/..) (M01/A02s/M11)
    - Call sound (Honor 7x)
- You have to encrypt for fix those issues

Remove Encryption

- Replace fileencryption with encryptable in vendor/etc/fstab



```
# Android fstab file.
# The filesystem that contains the filesystem checker binary (typically /system) cannot
# specify MF_CHECK, and must come before any filesystems that do specify MF_CHECK

#<src>                                                  <mnt_point>          <type>  <mnt_flags and options>                              <fs_mgr_flags>

system                                                   /system              ext4    ro,barrier=1,discard                                  wait,avb=vbmeta,logical,first_stage_mount,avb_keys=/avb/q-gsi.avbpubkey:/avb/r-gsi.avbpubkey:/avb/s-gsi.avbpubkey
product                                                  /product             ext4    ro,barrier=1,discard                                  wait,avb,logical,first_stage_mount
vendor                                                   /vendor              ext4    ro,barrier=1,discard                                  wait,avb,logical,first_stage_mount
odm                                                      /odm                 ext4    ro,errors=panic                                       wait,avb,logical,first_stage_mount

/dev/block/platform/soc/7824900.sdhci/by-name/metadata  /metadata            ext4    noatime,nosuid,nodev,noauto_da_alloc,discard,journal_checksum,data=ordered,errors=panic                         wait,first_stage_mount,formattable
/dev/block/platform/soc/7824900.sdhci/by-name/prism     /prism               ext4    ro,barrier=1,discard                                  nofail,avb,first_stage_mount
/dev/block/platform/soc/7824900.sdhci/by-name/optics    /optics              ext4    ro,barrier=1,discard                                  nofail,avb,first_stage_mount
/dev/block/bootdevice/by-name/userdata  /data   f2fs    noatime,nosuid,nodev,discard,usrquota,grpquota,fsync_mode=nobarrier,reserve_root=32768,resgid=5678  latemount,wait,check,fileencryption=ice,quota,reservedsize=128M,checkpoint=fs

#/dev/block/bootdevice/by-name/system         /                  ext4    ro,barrier=1,discard                        wait,avb
# HS60 added by tangqingyong for HS60-15 at 20190722 start
# HS60 code added for SR-ZQL1695-01-59 by tangqingyong at 20190729 start
#/dev/block/bootdevice/by-name/userdata       /data              f2fs    noatime,nosuid,nodev,discard  wait,check,fileencryption=ice,quota,reservedsize=128M
# HS60 code added for SR-ZQL1695-01-59 by tangqingyong at 20190729 end
# HS60 added by tangqingyong for HS60-15 at 20190722 end
/dev/block/bootdevice/by-name/config         /frp               emmc    defaults                                    defaults
/dev/block/bootdevice/by-name/misc           /misc              emmc    defaults                                    defaults
/dev/block/bootdevice/by-name/cache /cache  ext4    noatime,nosuid,nodev,noauto_da_alloc,discard,journal_checksum,data=ordered,errors=panic wait,check
/dev/block/bootdevice/by-name/dsp            /vendor/dsp            ext4    ro,nosuid,nodev,barrier=1                       wait
/dev/block/bootdevice/by-name/apnhlos          /vendor/firmware_mnt   vfat    ro,shortname=lower,uid=1000,gid=1000,dmask=227,fmask=337,context=u:object_r:firmware_file:s0 wait
/dev/block/bootdevice/by-name/modem          /vendor/firmware-modem   vfat    ro,shortname=lower,uid=1000,gid=1000,dmask=227,fmask=337,context=u:object_r:firmware_file:s0 wait
/dev/block/bootdevice/by-name/persist   /mnt/vendor/persist ext4    noatime,nosuid,nodev,noauto_da_alloc,discard,journal_checksum,data=ordered,errors=panic wait,check
/dev/block/bootdevice/by-name/efs   /mnt/vendor/efs ext4    noatime,nosuid,nodev,noauto_da_alloc,discard,journal_checksum,data=ordered,errors=panic wait,check
/dev/block/bootdevice/by-name/sec_efs   /efs    ext4    noatime,nosuid,nodev,noauto_da_alloc,discard,journal_checksum,data=ordered,errors=panic wait,check
/dev/block/bootdevice/by-name/omr   /omr    ext4    noatime,nosuid,nodev,noauto_da_alloc,discard,journal_checksum,data=ordered,errors=panic wait,check
#/dev/block/bootdevice/by-name/prism        /prism          ext4    ro wait
#/dev/block/bootdevice/by-name/optics       /optics         ext4    ro wait
/dev/block/bootdevice/by-name/carrier   /carrier    ext4    noatime,nosuid,nodev,noauto_da_alloc,discard,journal_checksum,data=ordered,errors=panic defaults,nofail,check

# VOLD:fstab.samsung (fstabs-4.9/fstab_non_AB_variant.qti)
#/devices/soc/7864900.sdhci/mmc_host*         /storage/sdcard1   vfat    nosuid,nodev                                wait,voldmanaged=sdcard1:auto,noemulatedsd,encryptable=footer
#/devices/soc/78db000.usb/msm_hsusb_host*     /storage/usbotg    vfat    nosuid,nodev                                wait,voldmanaged=usbotg:auto
/devices/platform/soc/7864900.sdhci/mmc_host*        auto    vfat    defaults    voldmanaged=sdcard:auto
/devices/platform/soc/78db000.usb/msm_hsusb_host*    auto    auto    defaults    voldmanaged=usb:auto

# Samsung ODE
/dev/block/bootdevice/by-name/keydata   /keydata    ext4    noatime,nosuid,nodev,noauto_da_alloc,discard,journal_checksum,data=ordered,errors=panic wait,check,fileencryption=ice,nofail
/dev/block/bootdevice/by-name/keyrefuge /keyrefuge  ext4    noatime,nosuid,nodev,noauto_da_alloc,discard,journal_checksum,data=ordered,errors=panic wait,check,fileencryption=ice,nofail
```



```
# Android fstab file.
# The filesystem that contains the filesystem checker binary (typically /system) cannot
# specify MF_CHECK, and must come before any filesystems that do specify MF_CHECK

#<src>                                                 <mnt_point>           <type>  <mnt_flags and options>                              <fs_mgr_flags>

system                                                  /system               ext4    ro,barrier=1,discard                                  wait,avb=vbmeta,logical,first_stage_mount,avb_keys=/avb/q-gsi.avbpubkey:/avb/r-gsi.avbpubkey:/avb/s-gsi.avbpubkey
product                                                 /product              ext4    ro,barrier=1,discard                                  wait,avb,logical,first_stage_mount
vendor                                                  /vendor               ext4    ro,barrier=1,discard                                  wait,avb,logical,first_stage_mount
odm                                                     /odm                  ext4    ro,errors=panic                                       wait,avb,logical,first_stage_mount

/dev/block/platform/soc/7824900.sdhci/by-name/metadata  /metadata             ext4    noatime,nosuid,nodev,noauto_da_alloc,discard,journal_checksum,data=ordered,errors=panic                     wait,first_stage_mount,formattable
/dev/block/platform/soc/7824900.sdhci/by-name/prism     /prism                ext4    ro,barrier=1,discard                                  nofail,avb,first_stage_mount
/dev/block/platform/soc/7824900.sdhci/by-name/optics    /optics               ext4    ro,barrier=1,discard                                  nofail,avb,first_stage_mount
/dev/block/bootdevice/by-name/userdata  /data   f2fs    noatime,nosuid,nodev,discard,usrquota,grpquota,fsync_mode=nobarrier,reserve_root=32768,resgid=5678  latemount,wait,check,encryptable=ice,quota,reservedsize=128M,checkpoint=fs

#/dev/block/bootdevice/by-name/system         /                  ext4    ro,barrier=1,discard                        wait,avb
# HS60 added by tangqingyong for HS60-15 at 20190722 start
# HS60 code added for SR-ZQL1695-01-59 by tangqingyong at 20190729 start
#/dev/block/bootdevice/by-name/userdata       /data              f2fs    noatime,nosuid,nodev,discard  wait,check,encryptable=ice,quota,reservedsize=128M
# HS60 code added for SR-ZQL1695-01-59 by tangqingyong at 20190729 end
# HS60 added by tangqingyong for HS60-15 at 20190722 end
/dev/block/bootdevice/by-name/config         /frp               emmc    defaults                                    defaults
/dev/block/bootdevice/by-name/misc           /misc              emmc    defaults                                    defaults
/dev/block/bootdevice/by-name/cache /cache  ext4    noatime,nosuid,nodev,noauto_da_alloc,discard,journal_checksum,data=ordered,errors=panic wait,check
/dev/block/bootdevice/by-name/dsp            /vendor/dsp            ext4    ro,nosuid,nodev,barrier=1                   wait
/dev/block/bootdevice/by-name/apnhlos          /vendor/firmware_mnt   vfat    ro,shortname=lower,uid=1000,gid=1000,dmask=227,fmask=337,context=u:object_r:firmware_file:s0 wait
/dev/block/bootdevice/by-name/modem          /vendor/firmware-modem   vfat    ro,shortname=lower,uid=1000,gid=1000,dmask=227,fmask=337,context=u:object_r:firmware_file:s0 wait
/dev/block/bootdevice/by-name/persist   /mnt/vendor/persist ext4    noatime,nosuid,nodev,noauto_da_alloc,discard,journal_checksum,data=ordered,errors=panic wait,check
/dev/block/bootdevice/by-name/efs   /mnt/vendor/efs ext4    noatime,nosuid,nodev,noauto_da_alloc,discard,journal_checksum,data=ordered,errors=panic wait,check
/dev/block/bootdevice/by-name/sec_efs   /efs    ext4    noatime,nosuid,nodev,noauto_da_alloc,discard,journal_checksum,data=ordered,errors=panic wait,check
/dev/block/bootdevice/by-name/omr   /omr    ext4    noatime,nosuid,nodev,noauto_da_alloc,discard,journal_checksum,data=ordered,errors=panic wait,check
#/dev/block/bootdevice/by-name/prism        /prism          ext4    ro wait
#/dev/block/bootdevice/by-name/optics       /optics         ext4    ro wait
/dev/block/bootdevice/by-name/carrier   /carrier    ext4    noatime,nosuid,nodev,noauto_da_alloc,discard,journal_checksum,data=ordered,errors=panic defaults,nofail,check

# VOLD:fstab.samsung (fstabs-4.9/fstab_non_AB_variant.qti)
#/devices/soc/7864900.sdhci/mmc_host*         /storage/sdcard1    vfat    nosuid,nodev                                wait,voldmanaged=sdcard1:auto,noemulatedsd,encryptable=footer
#/devices/soc/78db000.usb/msm_hsusb_host*     /storage/usbotg     vfat    nosuid,nodev                                wait,voldmanaged=usbotg:auto
/devices/platform/soc/7864900.sdhci/mmc_host*        auto    vfat    defaults    voldmanaged=sdcard:auto
/devices/platform/soc/78db000.usb/msm_hsusb_host*    auto    auto    defaults    voldmanaged=usb:auto

# Samsung ODE
/dev/block/bootdevice/by-name/keydata   /keydata    ext4    noatime,nosuid,nodev,noauto_da_alloc,discard,journal_checksum,data=ordered,errors=panic wait,check,encryptable=ice,nofail
/dev/block/bootdevice/by-name/keyrefuge /keyrefuge  ext4    noatime,nosuid,nodev,noauto_da_alloc,discard,journal_checksum,data=ordered,errors=panic wait,check,encryptable=ice,nofail
```

How to Build TWRP

How to Build TWRP from Source

- This guide is for basic twrp and you have to fix bugs and bugs are depend on oem and soc
- If your device is a/b device (if you dont have recovery.img or you have boot_a boot_b like partitions your device is a/b) you have to build bootctrl to build fully funtional TWRP

Bug Fixing

- Touch
- You have to add vendor/frimware
- You have to add touch driver lib/modules/.ko (if your device use touch driver)

How to build TWRP

https://www.youtube.com/watch?app=desktop&v=ZYU5xJ2we1o

How to Port System

How to port system

Most of time you can boot system.img from another device with same soc family without modifications

In Samsung Galaxy M01,

- You can boot redmi 8a/7a stock rom (miui) by just flashing redmi 7a/8a system.img via fastbootd (bugs depend on oem)

But,

- You have to use same architechture
- if system.img is arm64. you must use arm64 vendor
- if system.img is arm32 you must use arm64 vendor
- Also you must you same android version of system.img vendor.img and boot.img
- Eg – if I want to boot miui 11 (android 10) original redmi 7a stock rom to m01. i have to use android 10 vendor.img and boot.img. If i use android 11 or 12 or 9 vendor.img and boot.img it wont boot.

- You can boot a11/a02s/a20s system images also
- a20s and redmi 8a/7a dont have super. Then you just need to flash system.img only
- But in a02s/a11 they have super. So they have product.img and system.img inside of super.img So it need flash both system image and product image.
- But some OEM uses custom partitons,
  - oem
  - my_product
  - cust
  - hw_cust
  - hw_product
  - system_ext
- So you have to unpack all partitions (system/product/system_ext/etc) and then merge custom partitions to system and create new system.img with partitions
- Then you need to flash only system.img

- You have to unpack all partitions in super.img via CRB
- Then delete system links of custom partitions like product system_ext cust …. In system/
- Then copy all partitions (product system_ext …) to /system.
- And check is there other partitions in root (oem/preload) if they are paste them also
- Then resize it and repack system image

How to port system

https://www.youtube.com/watch?v=TdW-gh8Ywts

I tested,

a11 rom on m01 (android 12) [i used android 12 vendor and boot.img]

miui form redmi 7a on m01 (android 10) [i used android 10 vendor and boot.img]

a20s rom on m01 (android 11) [i used android 11 vendor and boot.img]

i think thses are possible

m01 core can boot a02/m02 oneui core 2.1 (same soc)

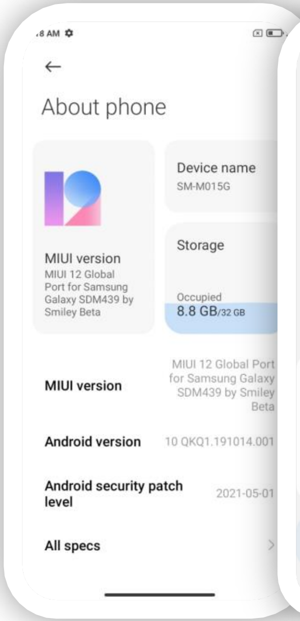
oneplus nord n100 with moto e7 plus

redmi 9c with samsung a04e/oneplus nord n20 se

samsung galaxy a12 with samsung m12/a13/.. (exyons 850)

samsung galaxy m21 with m31/a51

lenovo tab m7 3rd gen with redmi a1



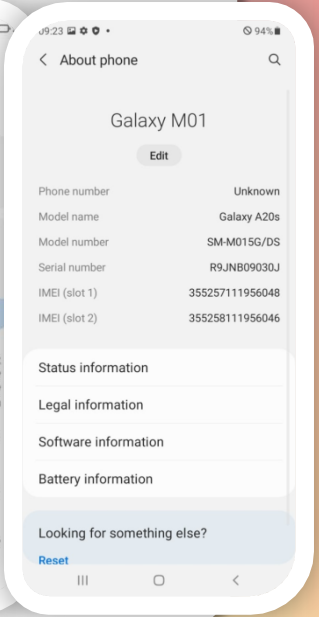
About phone

Device name
SM-M015G

Storage
Occupied
8.8 GB/32 GB

MIUI version
MIUI 12 Global Port for Samsung Galaxy SDM439 by Smiley Beta

MIUI version	MIUI 12 Global Port for Samsung Galaxy SDM439 by Smiley Beta
Android version	10 QKQ1.191014.001
Android security patch level	2021-05-01
All specs	



About phone

Galaxy M01

Edit

Phone number	Unknown
Model name	Galaxy A20s
Model number	SM-M015G/DS
Serial number	R9JNB09030J
IMEI (slot 1)	355257111956048
IMEI (slot 2)	355258111956046

Status information

Legal information

Software information

Battery information

Looking for something else?

Reset

Building AOSP form Source

Build from source

- you need ,
- kernel source
- device tree
- vendor tree
- Linux machine or server
  - 16/32 gb ram
  - more powerfull processor (8 cores)
  - 256gb ssd
  - high speed internet connection for download source code (around 100gb-200gb)

- if you dont want to build oss vendor
- you only need device tree you dont need kernel source or vendor tree. you can use prebuilt kernel

- check about more device trees in github according to your soc and oem

Selinux Premissive Boot Image

How to make premissive boot

- Copy boot.img to root of android image kitchen
- Drag and drop boot.img to upack.bat
- Open split_img/boot.img-cmdline as text
- And add androidboot.selinux=permissive after last word
- Then repack using repack.bat

- Some devices need to rebuild kernel with remove selinux enforcing

How to make premissive boot

https://www.youtube.com/watch?v=RYkSvB8l1KE

How to convert read only super to read and write (ext4/erofs/f2fs)

How to convert read only super to read and write

- You can you this tool with magisk or with twrp
- This will backup your super.img and create new rw super.img
- Then you can flash new super.img via fastbootd or twrp
- More info and Download Link

https://forum.xda-developers.com/t/magisk-twrp-arm32-64-a8-universal-read-only-to-read-write-for-android-ro2rw-auto-converting-super-system-partitions-to-read-write-mode.4521131/

Booting GSIs

Boot GSIs

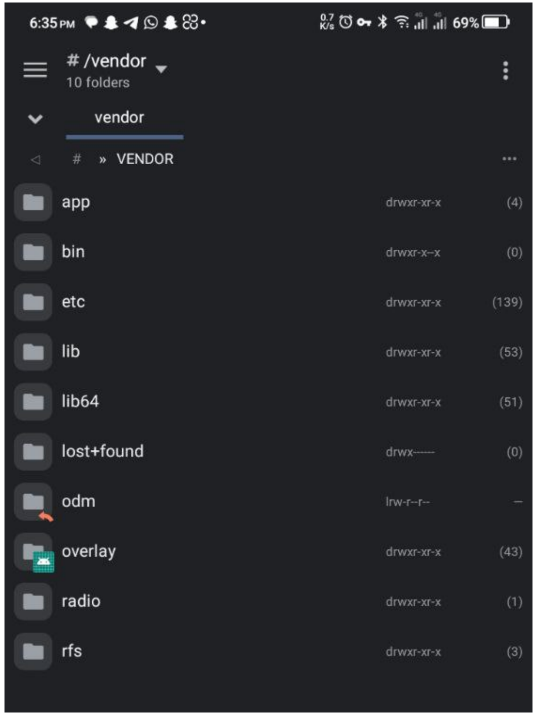
- To boot gsis you only need vendor image and system linked images by vendor (odm like)
- You can use mixplorer and see it

super	3672	3672
product	450	5
odm	4	5
Vendor	- 582	- 582
Free space	2630	3080

- In M01,
- It uses only odm in vendor
- Super.img has 3672MB (it has only system vendor odm product in super)
- vendor use 582mb , odm use 4mb , product use 450 mb
- you can flash only gsis around 2.6gb
- But if you flashed product image with 5mb you can flash gsi around 3.0GB
- So you have to flash empty images for all unsasory partitions like product system_ext oem my_product etc

How to Know essential paritions for boot

- Use Mixplorer app



```
6:35 PM                                  69%
# /vendor
10 folders
vendor
# » VENDOR
app          drwxr-xr-x   (4)
bin          drwxr-x--x   (0)
etc          drwxr-xr-x   (139)
lib          drwxr-xr-x   (53)
lib64        drwxr-xr-x   (51)
lost+found   drwx------   (0)
odm          lrw-r--r--   --
overlay      drwxr-xr-x   (43)
radio        drwxr-xr-x   (1)
rfs          drwxr-xr-x   (3)
```

Open Vendor with mixplorer
You can see odm with red arrow icon. It is a nessory parition for boot

If there is no folder with red arrow icon you there is nessory parition for boot except vendor and system

You can flash empty for others

How to Repack Super.img

How to make super.img (This method can add a gsi to super)

- You need crb 3.13v. If not you need donated version
- Convert your spersed super.img and to raw and extract all partitions in it
- Extract super.img via crb

I think this is the best method,

- Then unpack images you want modify (system.img/vendor.img/..)
- Then repack them as raw images
- Then copy all partitions with your modified files to build folder of your unpacked super
    - If I want to add gsi to super.img i have to unpack gsi.img and repack as it as raw image. Then move new system.img and stock vendor and other partitions (odm/product/..) to build folder of your unpacked super
- Then Repack super.img as raw or spersed or lz4

How to Repack Super

https://www.youtube.com/watch?v=ZncYVg39vc4

How to add gsi to super.img

How to add a gsi to super

- Convert your spersed super.img and to raw and extract all partitions in it
- Extract super.img via crb
- Then unpack gsi.xy and rename the image to system.img.
- Then convert it to raw with simg2img.exe
- Then copy all partitions with your raw_gsi.img to build folder of your unpacked super
- Then Repack super.img as raw or spersed or lz4

How to add GSI to Super

https://www.youtube.com/watch?v=ZncYVg39vc4

How to Add Gapps to Vanilla

How to Add GAPPS to Vanilla

- Extract gsi.xy and rename it to system.img
- Extract system.img via crb
- Dowload gapps accroding to architure and android version (android 12.1 arm64)
- Open Gapps.zip and extract files accoring to folder names
- Resize system.img and Repack

How to Add Gapps to Vanilla

https://youtu.be/vTTN9B8Bsnl

How to Make OEM GSIs (oneui myui miui)

How to Make OEM GSIs (oneui myui miui)

- There is tool called UKA TOOL
- You can use that tool in android device with magisk
- You need stock rom of the device you need to make gsi
- Then follow instructions
- Download link
- How to use (https://t.me/g20gsi/897)

- Most of time you need to Decrypt and Disable Dm-verty to boot GSIs

TWRP for Huawei

TWRP for Huawei

- In some Huawei Devices you can boot TWRP without unlocking bootloader (emui 10)
- Huawei uses erecovery_ramdisk and recovery_ramdisk instead of recovery partitions
- You have to flash twrp for both erecovery_ramdisk and recovery_ramdisk partitions
- You have to use brom/edl mode to flash it. So you need to short testpoints
- You have to use MTK Cliend or SP Flash tool. (MTK)

- I booted TWRP on Huawei y5p and y6p (mtk) locked bootloader

- But you cannot flash any custom roms or flash magisk,
    - If you you flash them your device will boot to red state

How to enable disabled features in android go

How to enable disabled features in android go

Draw over other apps

Picture in Picture mode

Add below lines to vendor/etc/permission/handheld_core_hardware.xml

- <feature name="android.software.managed_users" notLowRam="false"/>
- <feature name="android.software.activities_on_secondary_displays" notLowRam="false" />
- <feature name="android.software.picture_in_picture" notLowRam="false" />
- <feature name="android.software.voice_recognizers" notLowRam="false" />
- <feature name="android.software.app_widgets" />
- <feature name="android.software.home_screen" />

Add Features for Samsung (ONEUI 2.0+)

Add Features for Samsung (ONEUI 2.0+)

- Enable Multiwindow tray
- Enable Flagship Launcher Animations
- Enable High Performance Mode
- Enable Flagship Edge Ligthining+ Animations
- Enable Spotify as added alarm
- Enable Side Key Function
- Enable AOD Clock Transition Animation
- Enable Dolby Atmos without Headsets
- Enable Music Information in AOD
- Enable Dolby Atmos in Games
- Enable Advanced Screen Capture and Screen Record
- Enable Wireless PowerShare
- Enable Ultra Power Saving
- Enable Secure Wifi
- 60Hz Default
- Disable Smart Switch
- Enable Drawer Clearer Contrast
- Enable 3 Resolution options in Settings
- Enable Smooth Scroll of Surface Flinger
- Disable Samsung Ads
- Enable Live Clock in Launcher
- Enable Game Launcher in Drawer
- Enable Blur
- Enable Google Discover in Drawer (Needs OneUI 3+))

Add lines to system/etc/floationg_fetatures.xml and vendor etc/floating_features.xml
Downlad the xml file
(ONEUI_Features.xml)

How to Convert OneUI core to OneUI

How to Convert OneUI core to OneUI

- Change sep_lite to sep_basic in system/etc/floationg_fetatures.xml and vendor etc/floating_features.xml

```
<SEC_FLOATING_FEATURE_COMMON_CONFIG_SEP_CATEGORY>sep_lite</SEC_FLOATING_FEATURE_COMMON_CONFIG_SEP_CATEGORY>
```

to

```
<SEC_FLOATING_FEATURE_COMMON_CONFIG_SEP_CATEGORY>sep_basic</SEC_FLOATING_FEATURE_COMMON_CONFIG_SEP_CATEGORY>
```

- Rename,

- system/etc/permissions/com.samsung.device.lite.xml to com.samsung.device.xml
- system/etc/permissions/com.samsung.feature.samsung_experience_mobile_lite.xml to com.samsung.feature.samsung_experience_mobile.xml
- system/framework/com.samsung.device.lite.jar to com.samsung.device.jar
- system/framework/oat/arm(64)/com.samsung.device.lite.odex to com.samsung.device.odex
- system/framework/oat/arm(64)/com.samsung.device.lite.vdex to com.samsung.device.vdex

How to Meke Flashable Zips

How to Meke Flashable Zips

- You can make flashable custom images like boot.img recovery.img super.img system.img etc
- You can make flashable zip for add some files to vendor or system or etc
- And more
- Check here for more info

https://forum.xda-developers.com/t/zip-dual-installer-dynamic-installer-stable-4-7-b3-android-10-or-earlier.4279541/

Backup IMEI

Backup IMEI

- You have to backup imei and network configureations before doing any thing
- Backup IMEI related partition
  - Nvdata
  - Nvram
  - Proinfo
  - EFS
  - SEC_EFS
  - MODEMST1
  - MODEMST2
  - FSG
  - QCN (for qualcomm imei and network configureations)

- You can backup qcn using qualcomm flash image loader
- In samsung devices if you loss the imei certificate . You cannot fix your device without paying around 50usd. You have to cpid imei repair. You can check imei certificate by dialling *#*#0011#*#*#

THANKS

Credits
smily9000