Android build system

Nguyen Tran (Nguyen.TranLeHoang@vn.bosch.com)

Oct-2024

BOSCH

Day 7

(New product creation)

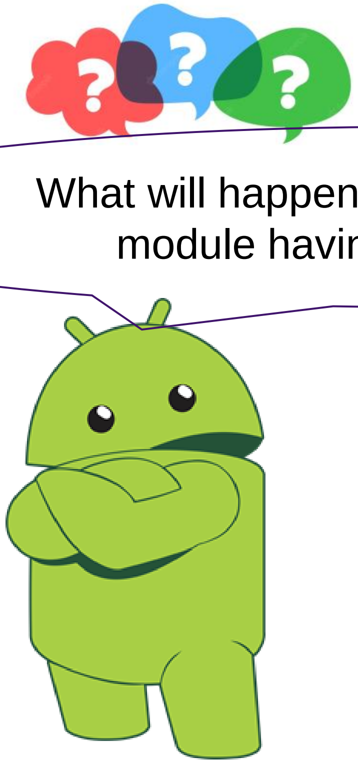
QUIZ (1/12)

Why did Google decide to use **Android.bp**,
instead of **Android.mk**?

QUIZ (2/12)

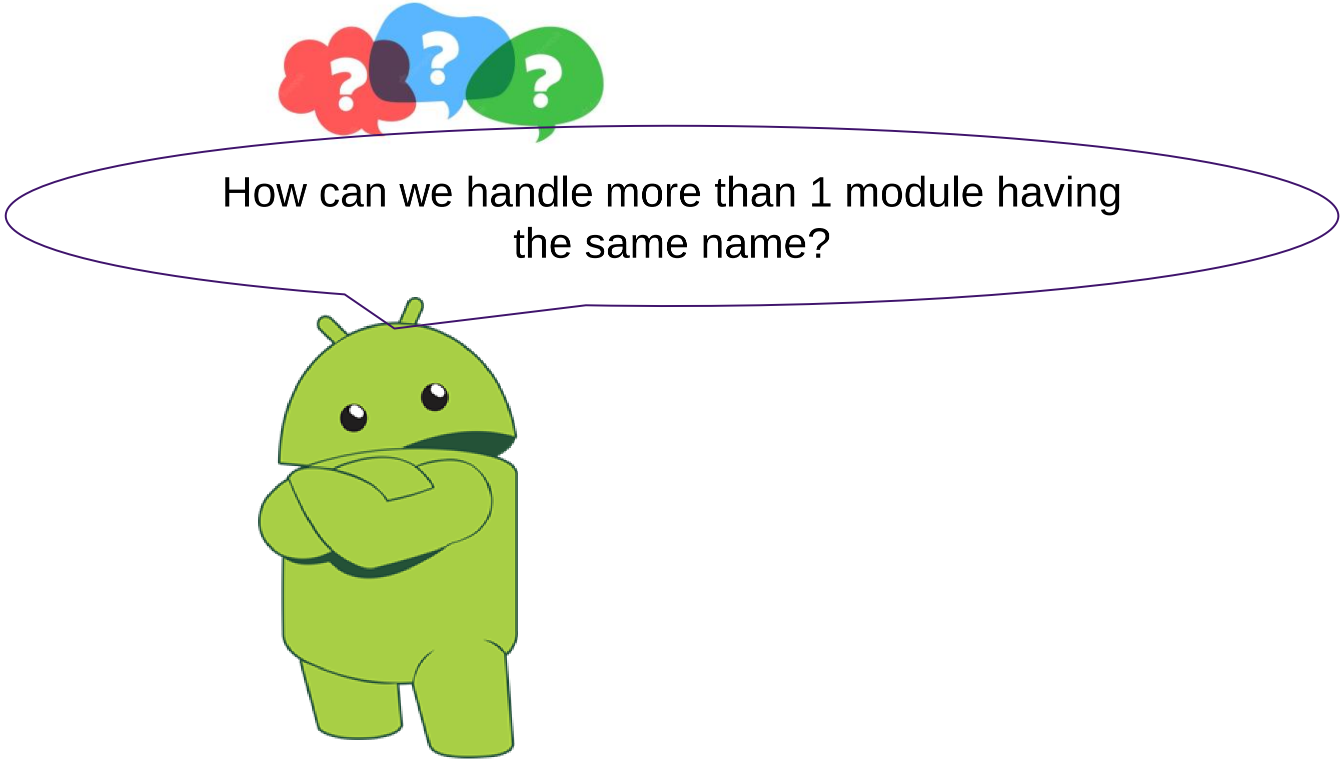
How do we handle conditionals in Android.bp?
Why do we do that?

QUIZ (3/12)



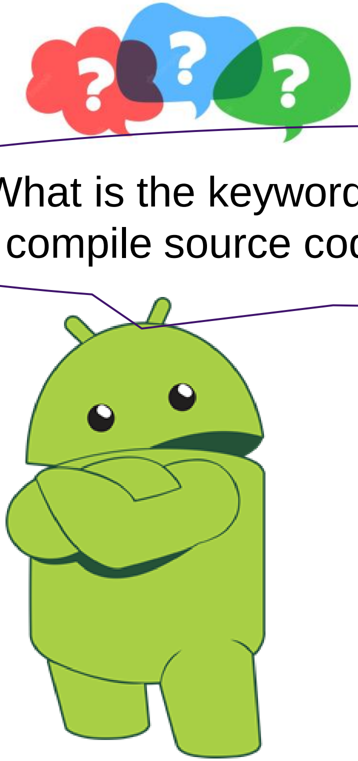
What will happen if we have more than 1 module having the same name?

QUIZ (4/12)



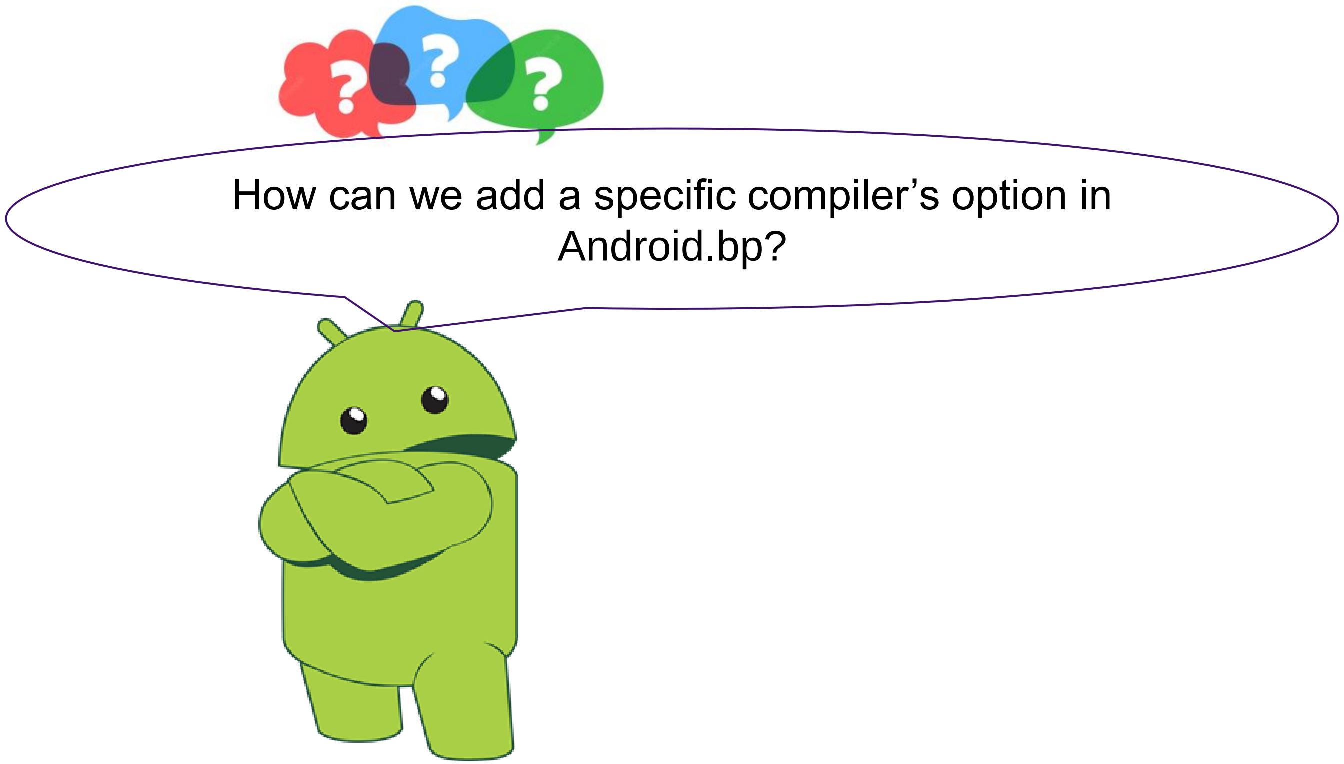
How can we handle more than 1 module having the same name?

QUIZ (5/12)



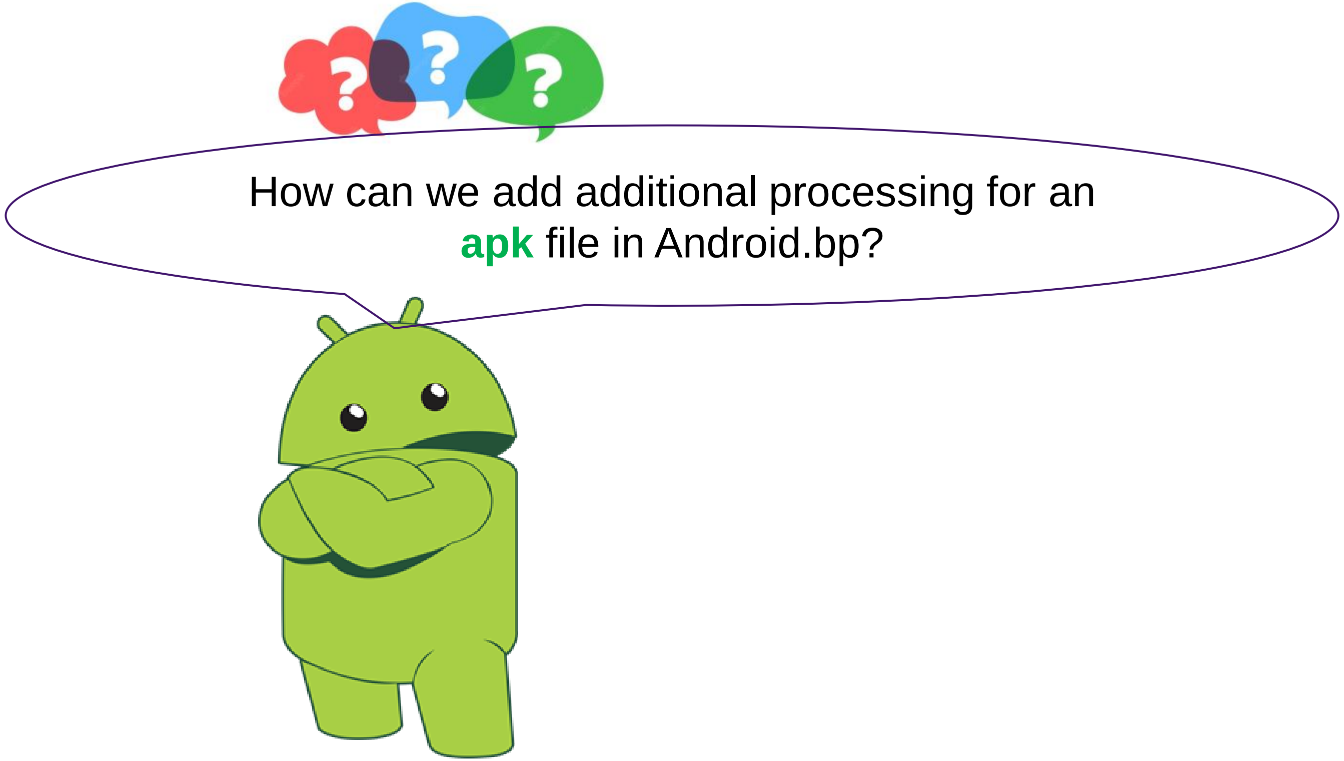
What is the keyword in Android.bp that we can compile source code into a dynamic library?

QUIZ (6/12)



How can we add a specific compiler’s option in Android.bp?

QUIZ (7/12)

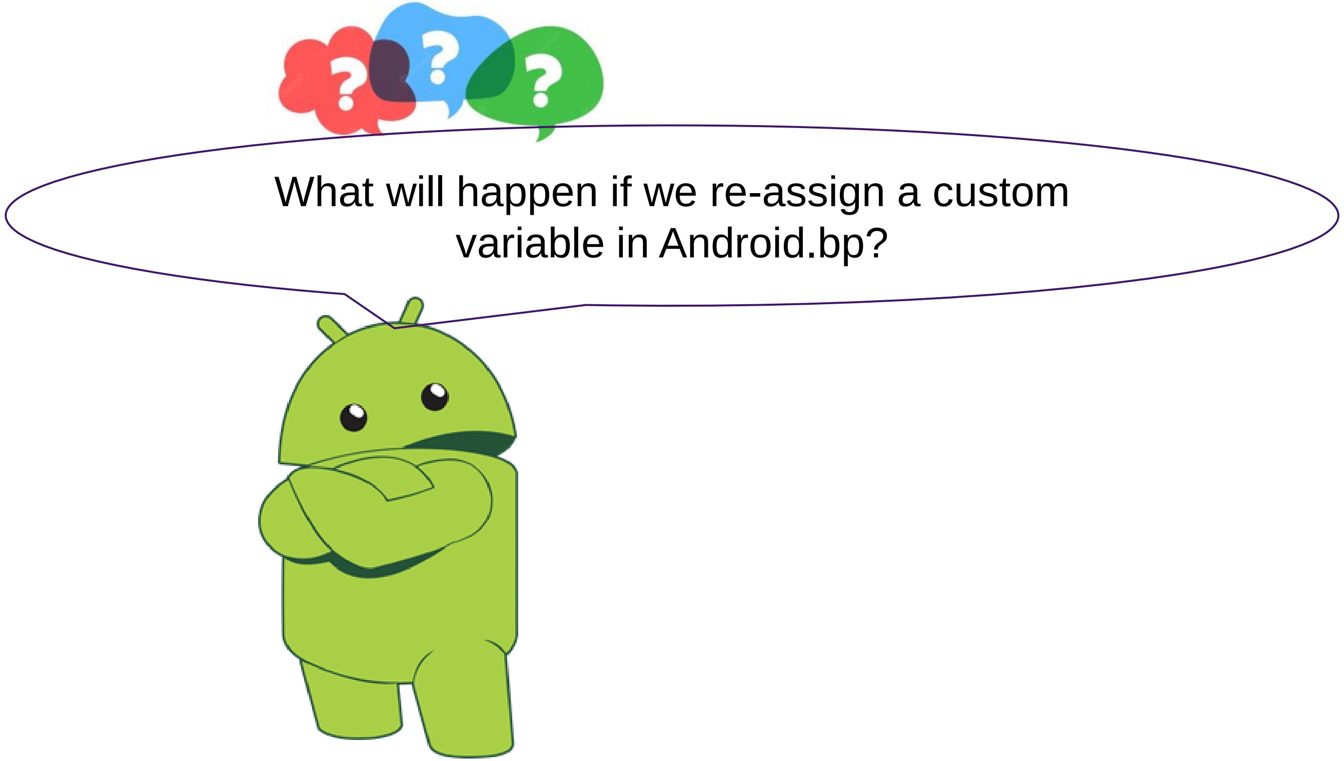


How can we add additional processing for an **apk** file in Android.bp?

QUIZ (8/12)

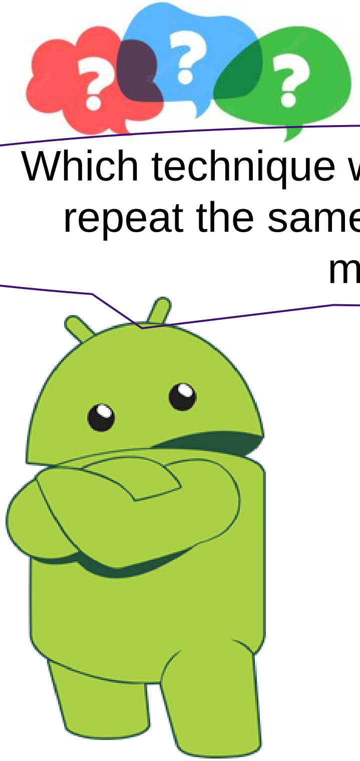
Which patterns will be matched with **“java/**/*.java”** in Android.bp?

QUIZ (9/12)



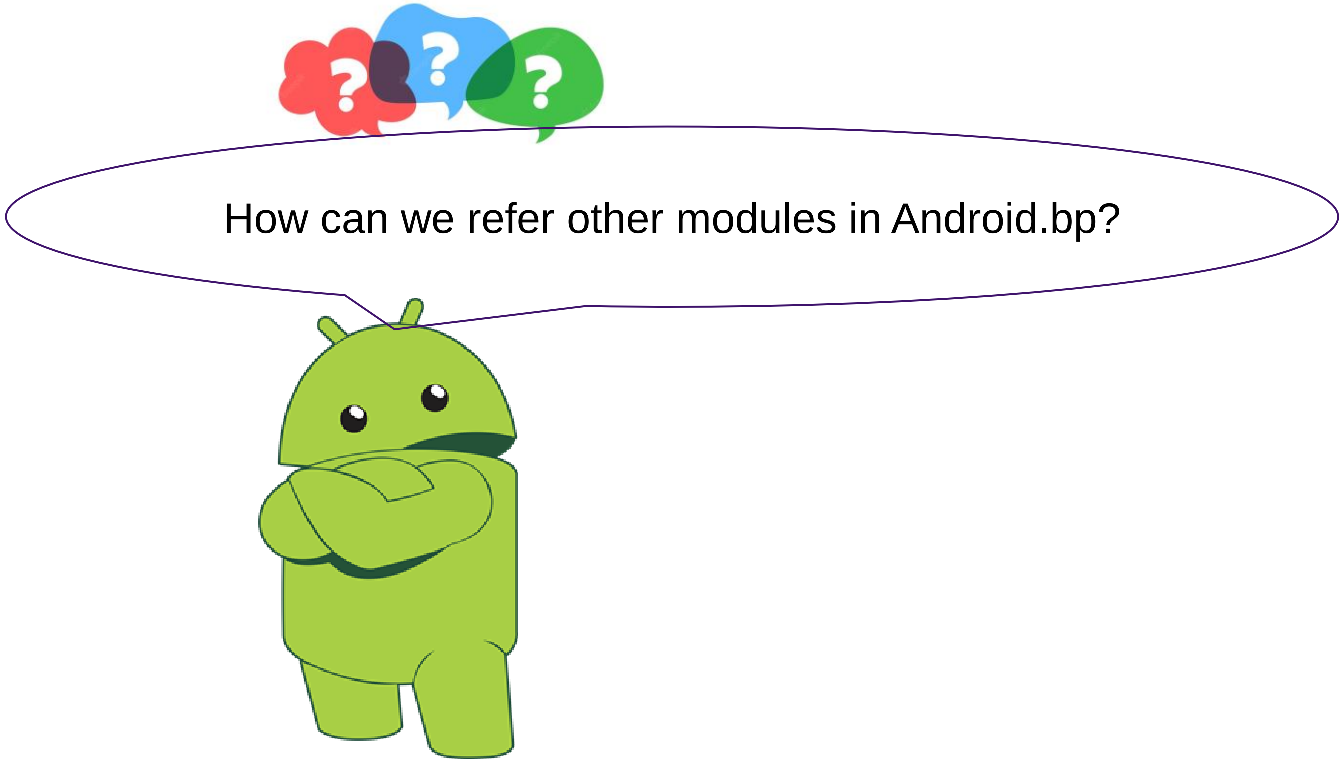
What will happen if we re-assign a custom variable in Android.bp?

QUIZ (10/12)



Which technique will be used if we want to repeat the same properties in multiple modules?

QUIZ (11/12)



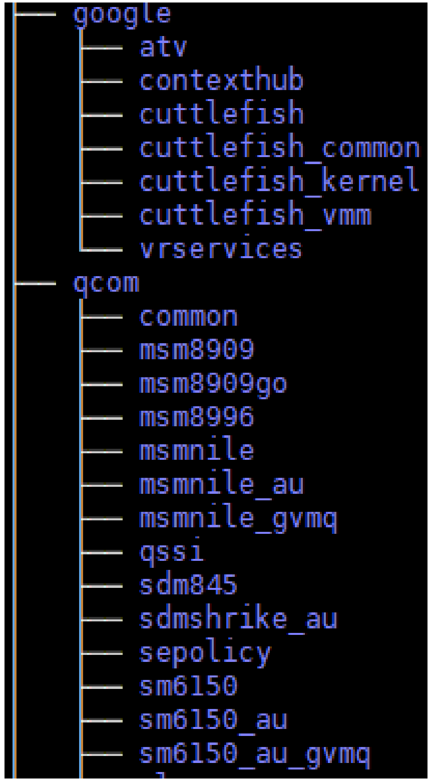
How can we refer other modules in Android.bp?

QUIZ (12/12)

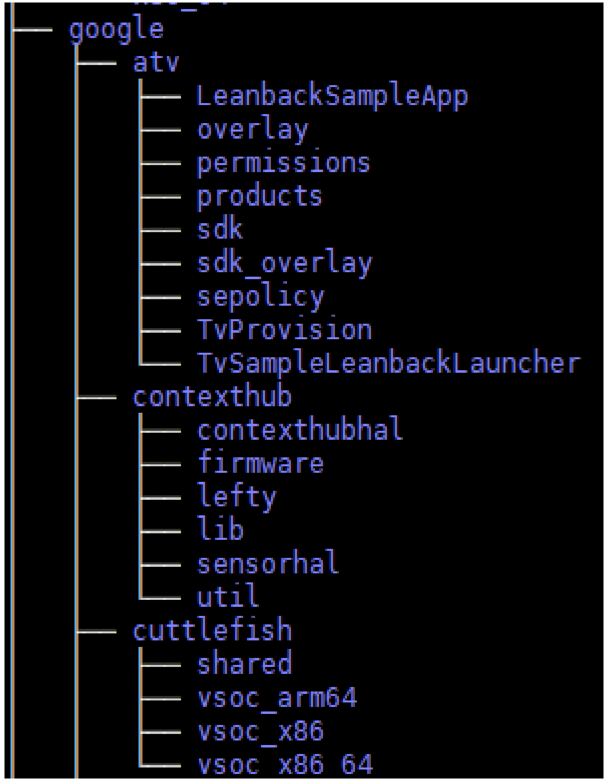
Which variable types are supported in Android.bp?

Group working (10 mins)

The pictures showed some Android products. What do you think about it?



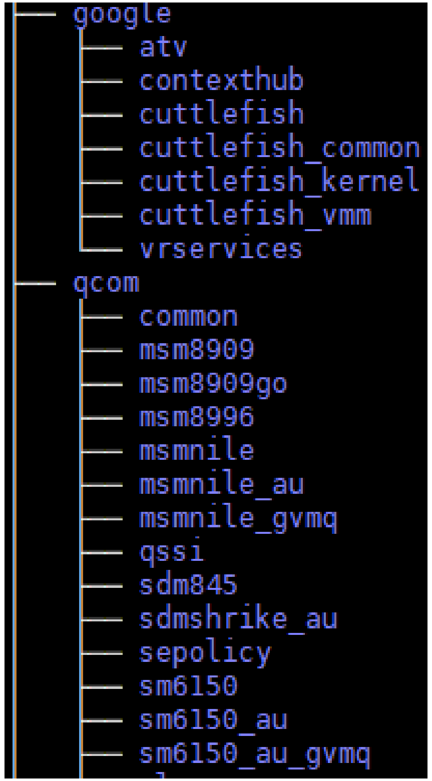
```
├── google
│   ├── atv
│   ├── contexthub
│   ├── cuttlefish
│   ├── cuttlefish_common
│   ├── cuttlefish_kernel
│   ├── cuttlefish_vmm
│   └── vrservices
├── qcom
│   ├── common
│   ├── msm8909
│   ├── msm8909go
│   ├── msm8996
│   ├── msmnile
│   ├── msmnile_au
│   ├── msmnile_gvmq
│   ├── qssi
│   ├── sdm845
│   ├── sdmshrike_au
│   ├── sepolicy
│   ├── sm6150
│   ├── sm6150_au
│   ├── sm6150_au_gvmq
```



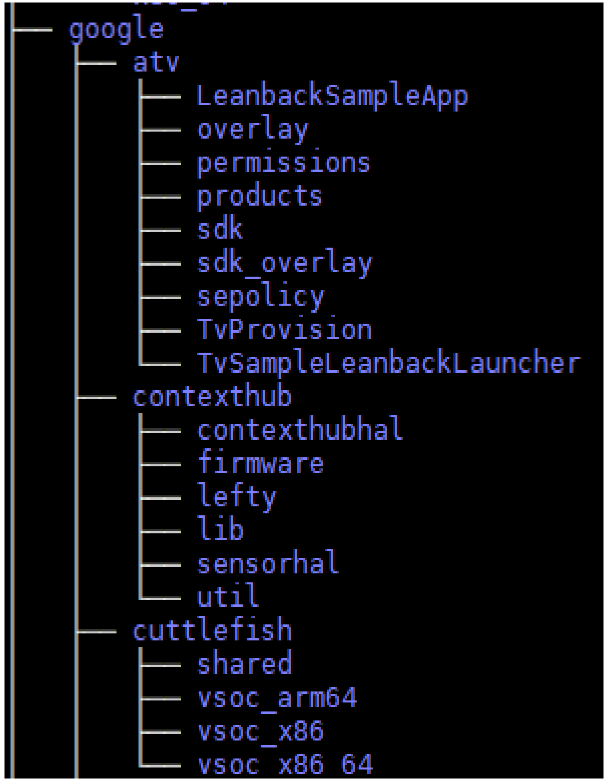
```
├── google
│   ├── atv
│   │   ├── LeanbackSampleApp
│   │   ├── overlay
│   │   ├── permissions
│   │   ├── products
│   │   ├── sdk
│   │   ├── sdk_overlay
│   │   ├── sepolicy
│   │   ├── TvProvision
│   │   └── TvSampleLeanbackLauncher
│   ├── contexthub
│   │   ├── contexthubhal
│   │   ├── firmware
│   │   ├── lefty
│   │   ├── lib
│   │   ├── sensorhal
│   │   └── util
│   ├── cuttlefish
│   │   ├── shared
│   │   ├── vsoc_arm64
│   │   ├── vsoc_x86
│   │   └── vsoc_x86_64
```

Structure

- The **<build root>/device** folder contains product customizations:
  - Building procedures and extensions for the targeted “**Android based product**” of this build.
- Devices are **grouped by vendor**.
- Each device have **one or more products and boards**.



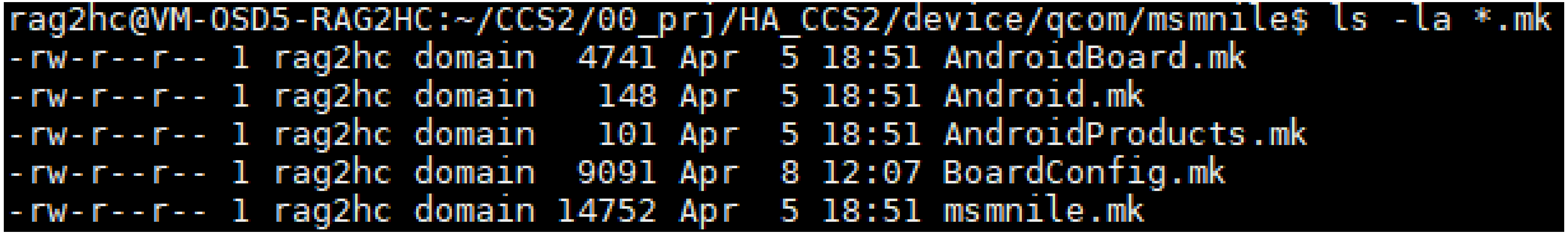
```
├── google
│   ├── atv
│   ├── contexthub
│   ├── cuttlefish
│   ├── cuttlefish_common
│   ├── cuttlefish_kernel
│   ├── cuttlefish_vmm
│   └── vrservices
├── qcom
│   ├── common
│   ├── msm8909
│   ├── msm8909go
│   ├── msm8996
│   ├── msmnile
│   ├── msmnile_au
│   ├── msmnile_gvmq
│   ├── qssi
│   ├── sdm845
│   ├── sdmshrike_au
│   ├── sepolicy
│   ├── sm6150
│   ├── sm6150_au
│   ├── sm6150_au_gvmq
```



```
├── google
│   ├── atv
│   │   ├── LeanbackSampleApp
│   │   ├── overlay
│   │   ├── permissions
│   │   ├── products
│   │   ├── sdk
│   │   ├── sdk_overlay
│   │   ├── sepolicy
│   │   ├── TvProvision
│   │   └── TvSampleLeanbackLauncher
│   ├── contexthub
│   │   ├── contexthubhal
│   │   ├── firmware
│   │   ├── lefty
│   │   ├── lib
│   │   ├── sensorhal
│   │   └── util
│   ├── cuttlefish
│   │   ├── shared
│   │   ├── vsoc_arm64
│   │   ├── vsoc_x86
│   │   └── vsoc_x86_64
```

Structure

- Product main **makefiles**:
  - AndroidProducts.mk
  - Android.mk
  - AndroidBoard.mk
  - BoardConfig.mk
  - <device/board_name>.mk

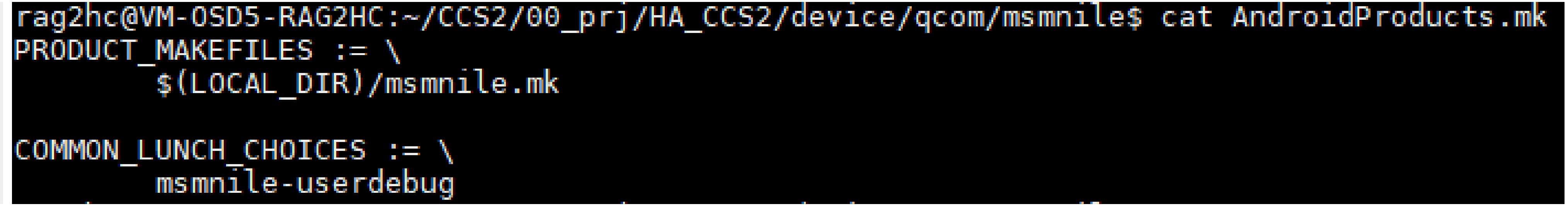


```
rag2hc@VM-OSD5-RAG2HC:~/CCS2/00_prj/HA_CCS2/device/qcom/msmnile$ ls -la *.mk
-rw-r--r-- 1 rag2hc domain  4741 Apr  5 18:51 AndroidBoard.mk
-rw-r--r-- 1 rag2hc domain   148 Apr  5 18:51 Android.mk
-rw-r--r-- 1 rag2hc domain   101 Apr  5 18:51 AndroidProducts.mk
-rw-r--r-- 1 rag2hc domain  9091 Apr  8 12:07 BoardConfig.mk
-rw-r--r-- 1 rag2hc domain 14752 Apr  5 18:51 msmnile.mk
```

AndroidProducts.mk

- **AndroidProducts.mk** lists **the products of this vendor** setting the **PRODUCT_MAKEFILES** build variable.

- For instance: **device/qcom/msmnile/AndroidProducts.mk**

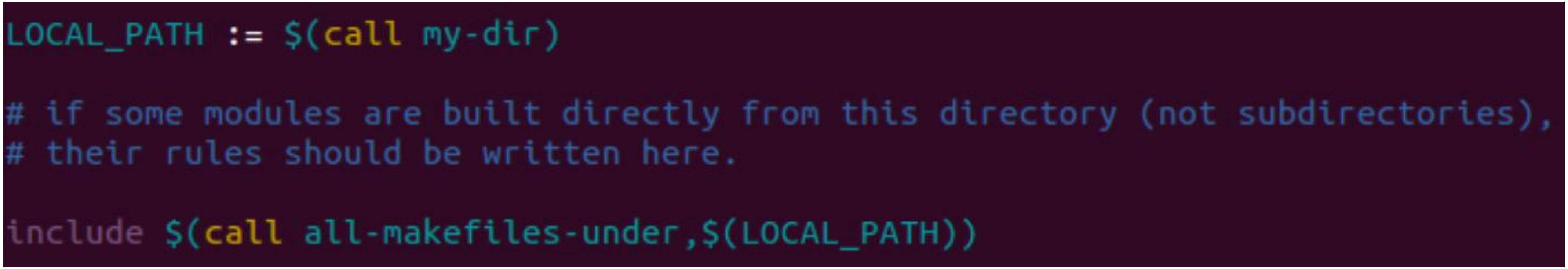


```
rag2hc@VM-OSD5-RAG2HC:~/CCS2/00_prj/HA_CCS2/device/qcom/msmnile$ cat AndroidProducts.mk
PRODUCT_MAKEFILES := \
        $(LOCAL_DIR)/msmnile.mk

COMMON_LUNCH_CHOICES := \
        msmnile-userdebug
```

- A good **reference for this kind of makefile** is at **build/target/product/AndroidProducts.mk**

Android.mk

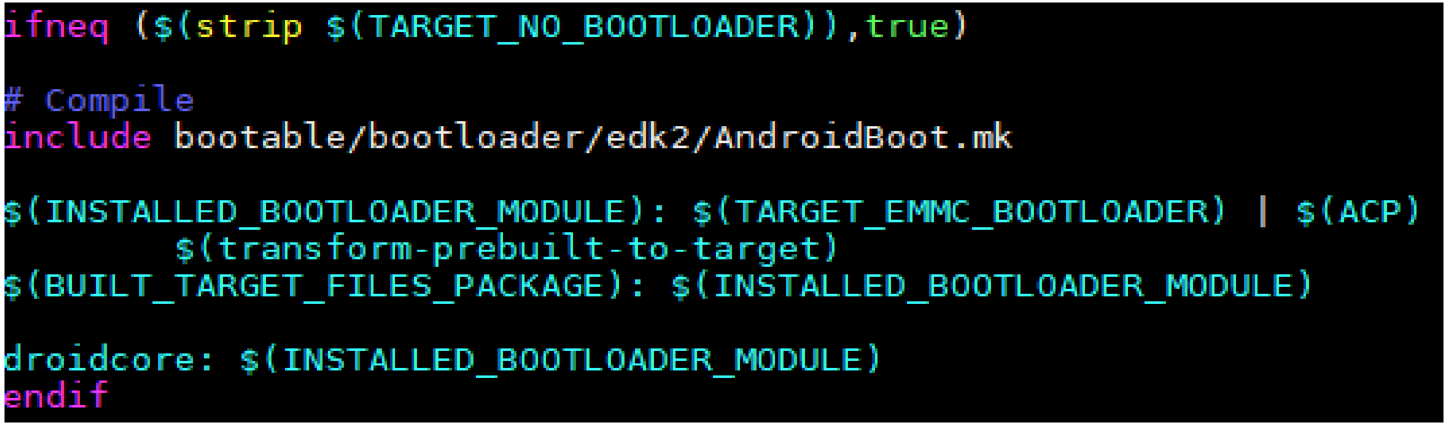
- If there is any module is defined under **devices/<my_company>/<my_product>** folder to be built, an **Android.mk** file is needed to call the **build on the sub folders**.



```
LOCAL_PATH := $(call my-dir)

# if some modules are built directly from this directory (not subdirectories),
# their rules should be written here.

include $(call all-makefiles-under,$(LOCAL_PATH))
```

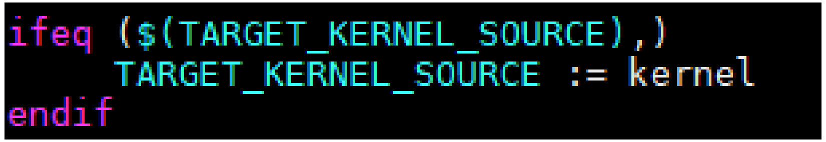
AndroidBoard.mk

- Compile **(L)ittle (K)ernel** bootloader and the **nandwrite** utility.

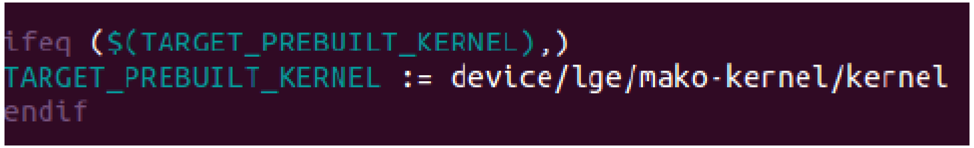


```
ifneq ($(strip $(TARGET_NO_BOOTLOADER)),true)

# Compile
include bootable/bootloader/edk2/AndroidBoot.mk

$(INSTALLED_BOOTLOADER_MODULE): $(TARGET_EMMC_BOOTLOADER) | $(ACP)
        $(transform-prebuilt-to-target)
$(BUILT_TARGET_FILES_PACKAGE): $(INSTALLED_BOOTLOADER_MODULE)

droidcore: $(INSTALLED_BOOTLOADER_MODULE)
endif
```

- Defines the **kernel source** or **kernel prebuilt binaries** to use



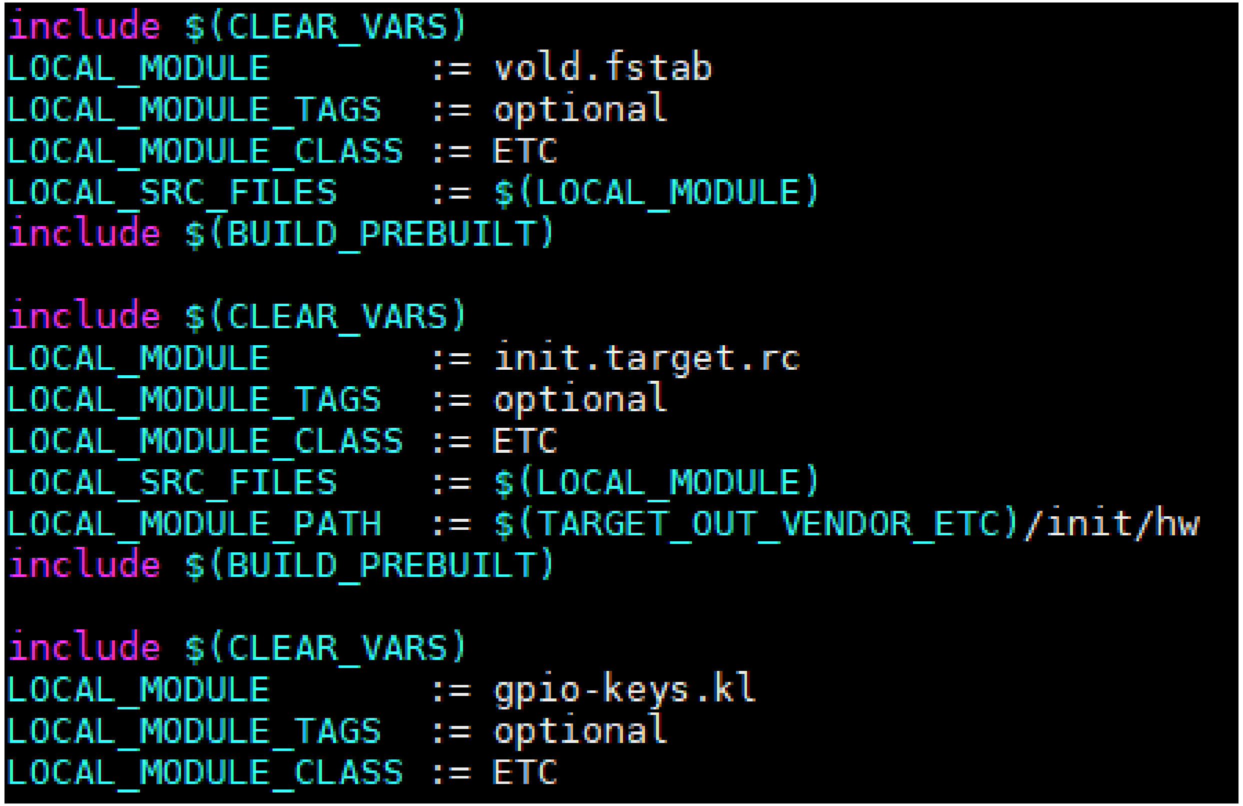
```
ifeq ($(TARGET_KERNEL_SOURCE),)
     TARGET_KERNEL_SOURCE := kernel
endif
```



```
ifeq ($(TARGET_PREBUILT_KERNEL),)
TARGET_PREBUILT_KERNEL := device/lge/mako-kernel/kernel
endif
```

AndroidBoard.mk

- Copy additional target-specific files.



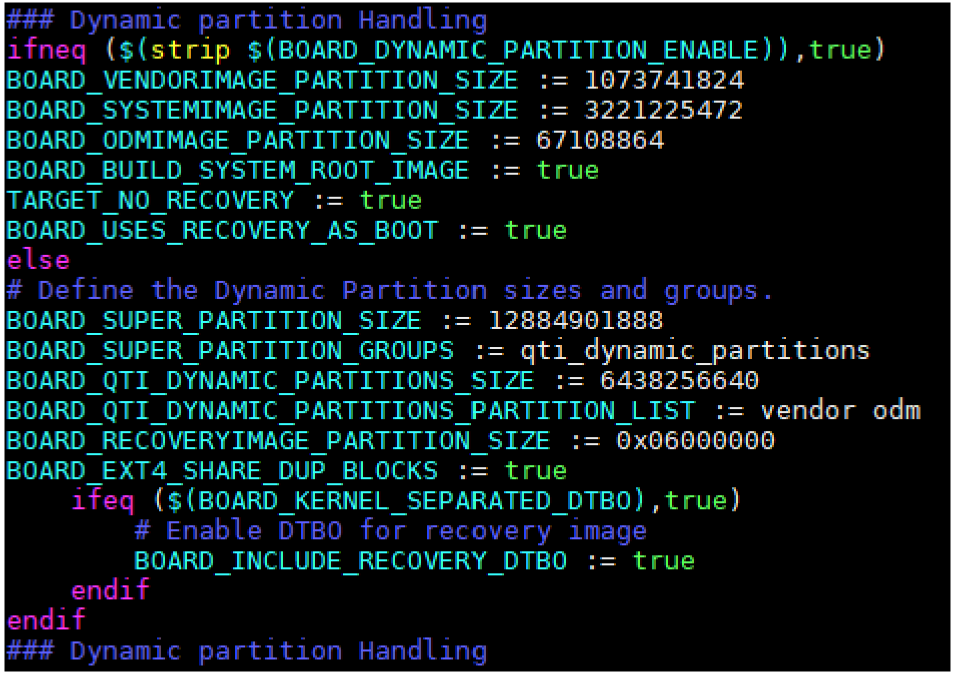
```
include $(CLEAR_VARS)
LOCAL_MODULE       := vold.fstab
LOCAL_MODULE_TAGS  := optional
LOCAL_MODULE_CLASS := ETC
LOCAL_SRC_FILES    := $(LOCAL_MODULE)
include $(BUILD_PREBUILT)

include $(CLEAR_VARS)
LOCAL_MODULE       := init.target.rc
LOCAL_MODULE_TAGS  := optional
LOCAL_MODULE_CLASS := ETC
LOCAL_SRC_FILES    := $(LOCAL_MODULE)
LOCAL_MODULE_PATH  := $(TARGET_OUT_VENDOR_ETC)/init/hw
include $(BUILD_PREBUILT)

include $(CLEAR_VARS)
LOCAL_MODULE       := gpio-keys.kl
LOCAL_MODULE_TAGS  := optional
LOCAL_MODULE_CLASS := ETC
```

BoardConfig.mk

- Configures the board (hardware) related parameters:
  - Usually in the same product folder, but could be separated:

    device/<my_company>/<my_board>/

  - Things are defined here:
    - Partitions sizes



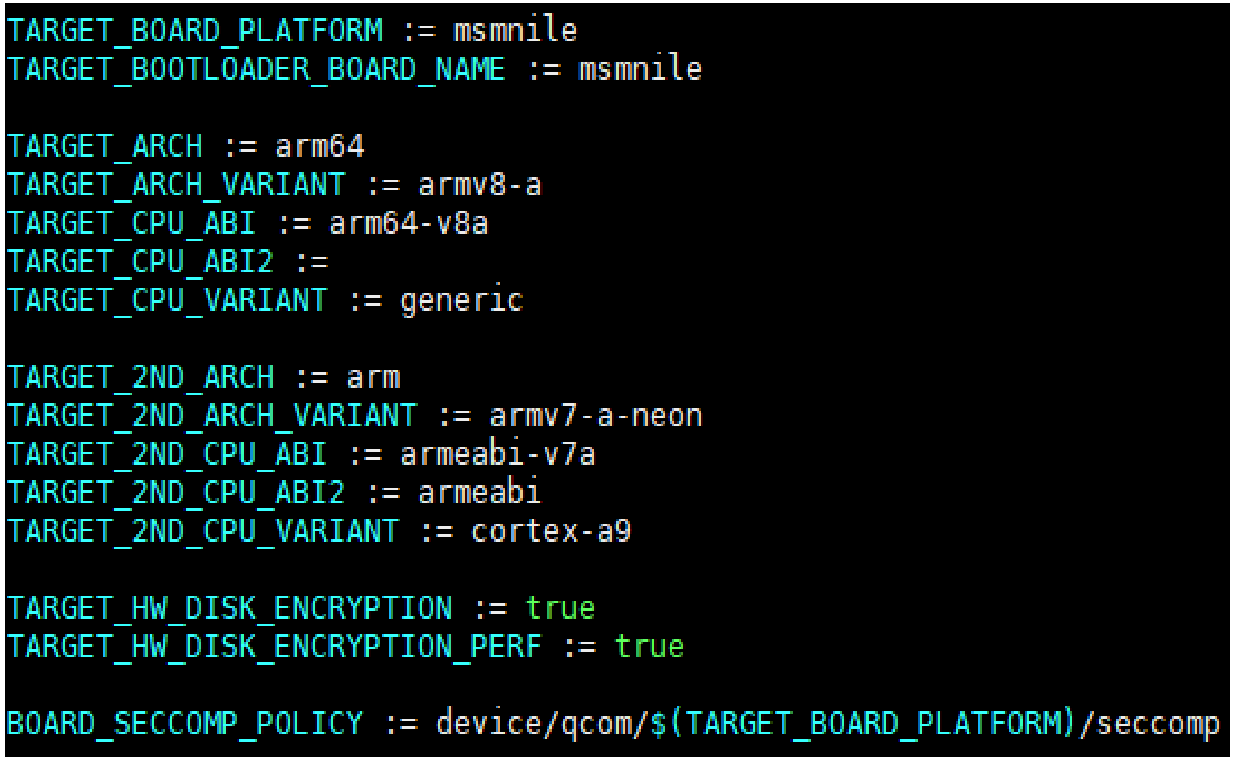
```
### Dynamic partition Handling
ifneq ($(strip $(BOARD_DYNAMIC_PARTITION_ENABLE)),true)
BOARD_VENDORIMAGE_PARTITION_SIZE := 1073741824
BOARD_SYSTEMIMAGE_PARTITION_SIZE := 3221225472
BOARD_ODMIMAGE_PARTITION_SIZE := 67108864
BOARD_BUILD_SYSTEM_ROOT_IMAGE := true
TARGET_NO_RECOVERY := true
BOARD_USES_RECOVERY_AS_BOOT := true
else
# Define the Dynamic Partition sizes and groups.
BOARD_SUPER_PARTITION_SIZE := 12884901888
BOARD_SUPER_PARTITION_GROUPS := qti_dynamic_partitions
BOARD_QTI_DYNAMIC_PARTITIONS_SIZE := 6438256640
BOARD_QTI_DYNAMIC_PARTITIONS_PARTITION_LIST := vendor odm
BOARD_RECOVERYIMAGE_PARTITION_SIZE := 0x06000000
BOARD_EXT4_SHARE_DUP_BLOCKS := true
    ifeq ($(BOARD_KERNEL_SEPARATED_DTBO),true)
        # Enable DTBO for recovery image
        BOARD_INCLUDE_RECOVERY_DTBO := true
    endif
endif
### Dynamic partition Handling
```

BoardConfig.mk

- Configures the board (hardware) related parameters:
  - Usually in the same product folder, but could be separated:

    device/<my_company>/<my_board>/

  - Things are defined here:
    - Target CPU, CPU variant



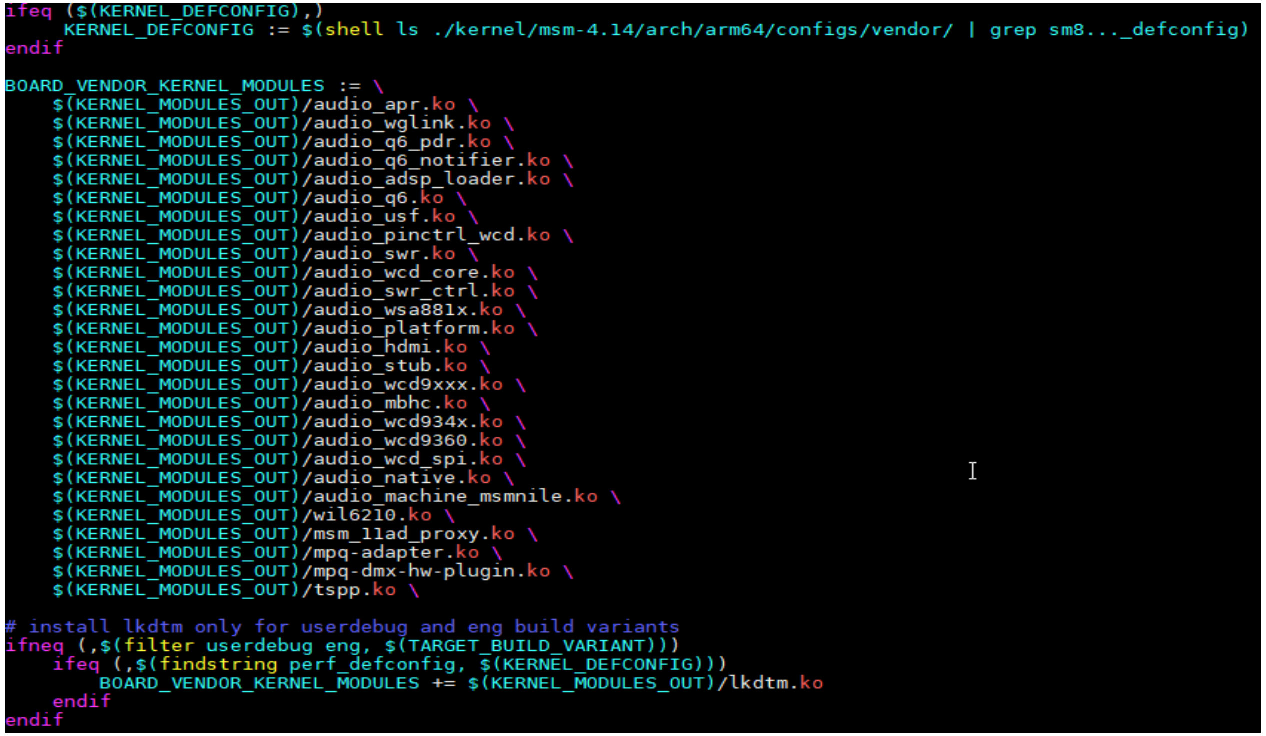
```
TARGET_BOARD_PLATFORM := msmnile
TARGET_BOOTLOADER_BOARD_NAME := msmnile

TARGET_ARCH := arm64
TARGET_ARCH_VARIANT := armv8-a
TARGET_CPU_ABI := arm64-v8a
TARGET_CPU_ABI2 :=
TARGET_CPU_VARIANT := generic

TARGET_2ND_ARCH := arm
TARGET_2ND_ARCH_VARIANT := armv7-a-neon
TARGET_2ND_CPU_ABI := armeabi-v7a
TARGET_2ND_CPU_ABI2 := armeabi
TARGET_2ND_CPU_VARIANT := cortex-a9

TARGET_HW_DISK_ENCRYPTION := true
TARGET_HW_DISK_ENCRYPTION_PERF := true

BOARD_SECCOMP_POLICY := device/qcom/$(TARGET_BOARD_PLATFORM)/seccomp
```

BoardConfig.mk

- Configures the board (hardware) related parameters:
  - Usually in the same product folder, but could be separated:

    device/<my_company>/<my_board>/

  - Things are defined here:
    - Kernel build compilation flags



```
ifeq ($(KERNEL_DEFCONFIG),)
     KERNEL_DEFCONFIG := $(shell ls ./kernel/msm-4.14/arch/arm64/configs/vendor/ | grep sm8..._defconfig)
endif

BOARD_VENDOR_KERNEL_MODULES := \
    $(KERNEL_MODULES_OUT)/audio_apr.ko \
    $(KERNEL_MODULES_OUT)/audio_wglink.ko \
    $(KERNEL_MODULES_OUT)/audio_q6_pdr.ko \
    $(KERNEL_MODULES_OUT)/audio_q6_notifier.ko \
    $(KERNEL_MODULES_OUT)/audio_adsp_loader.ko \
    $(KERNEL_MODULES_OUT)/audio_q6.ko \
    $(KERNEL_MODULES_OUT)/audio_usf.ko \
    $(KERNEL_MODULES_OUT)/audio_pinctrl_wcd.ko \
    $(KERNEL_MODULES_OUT)/audio_swr.ko \
    $(KERNEL_MODULES_OUT)/audio_wcd_core.ko \
    $(KERNEL_MODULES_OUT)/audio_swr_ctrl.ko \
    $(KERNEL_MODULES_OUT)/audio_wsa881x.ko \
    $(KERNEL_MODULES_OUT)/audio_platform.ko \
    $(KERNEL_MODULES_OUT)/audio_hdmi.ko \
    $(KERNEL_MODULES_OUT)/audio_stub.ko \
    $(KERNEL_MODULES_OUT)/audio_wcd9xxx.ko \
    $(KERNEL_MODULES_OUT)/audio_mbhc.ko \
    $(KERNEL_MODULES_OUT)/audio_wcd934x.ko \
    $(KERNEL_MODULES_OUT)/audio_wcd9360.ko \
    $(KERNEL_MODULES_OUT)/audio_wcd_spi.ko \
    $(KERNEL_MODULES_OUT)/audio_native.ko \
    $(KERNEL_MODULES_OUT)/audio_machine_msmnile.ko \
    $(KERNEL_MODULES_OUT)/wil6210.ko \
    $(KERNEL_MODULES_OUT)/msm_11ad_proxy.ko \
    $(KERNEL_MODULES_OUT)/mpq-adapter.ko \
    $(KERNEL_MODULES_OUT)/mpq-dmx-hw-plugin.ko \
    $(KERNEL_MODULES_OUT)/tspp.ko \

# install lkdtm only for userdebug and eng build variants
ifneq (,$(filter userdebug eng, $(TARGET_BUILD_VARIANT)))
    ifeq (,$(findstring perf_defconfig, $(KERNEL_DEFCONFIG)))
        BOARD_VENDOR_KERNEL_MODULES += $(KERNEL_MODULES_OUT)/lkdtm.ko
    endif
endif
```

<device/board_name>.mk

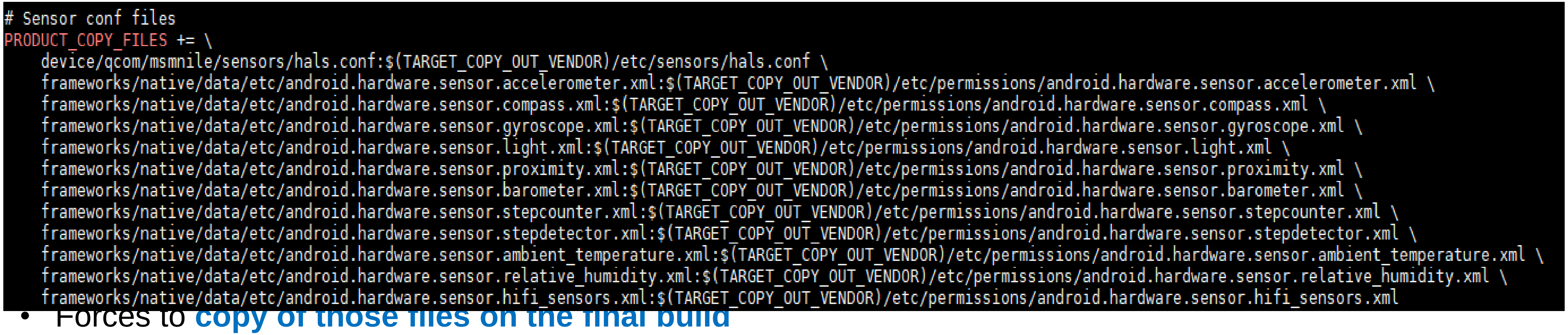
- **<device/board_name>.mk** contains the **product properties** (**name**, **version**, etc.) and extras like **modules/programs** or **prebuilt files** to be included in the build.
- It could **include/inherit** from other **predefined mk files** from **build/target/product/**
- A good reference for this kind of makefile is at **build/target/product/aosp_x86.mk**

<device/board_name>.mk

- Product properties:
  - PRODUCT_NAME := msmnile
    - This is the name that will **appear in the lunch combo** option.
    - This must **match this product folder** under devices folder.
  - PRODUCT_DEVICE := msmnile
    - This must match the device’s sub directory.
    - **TARGET_DEVICE** derives from this variable.
  - PRODUCT_MODEL := msmnile for arm64
    - The end-user-visible name for the end product.

<device/board_name>.mk

- Product files to copy:

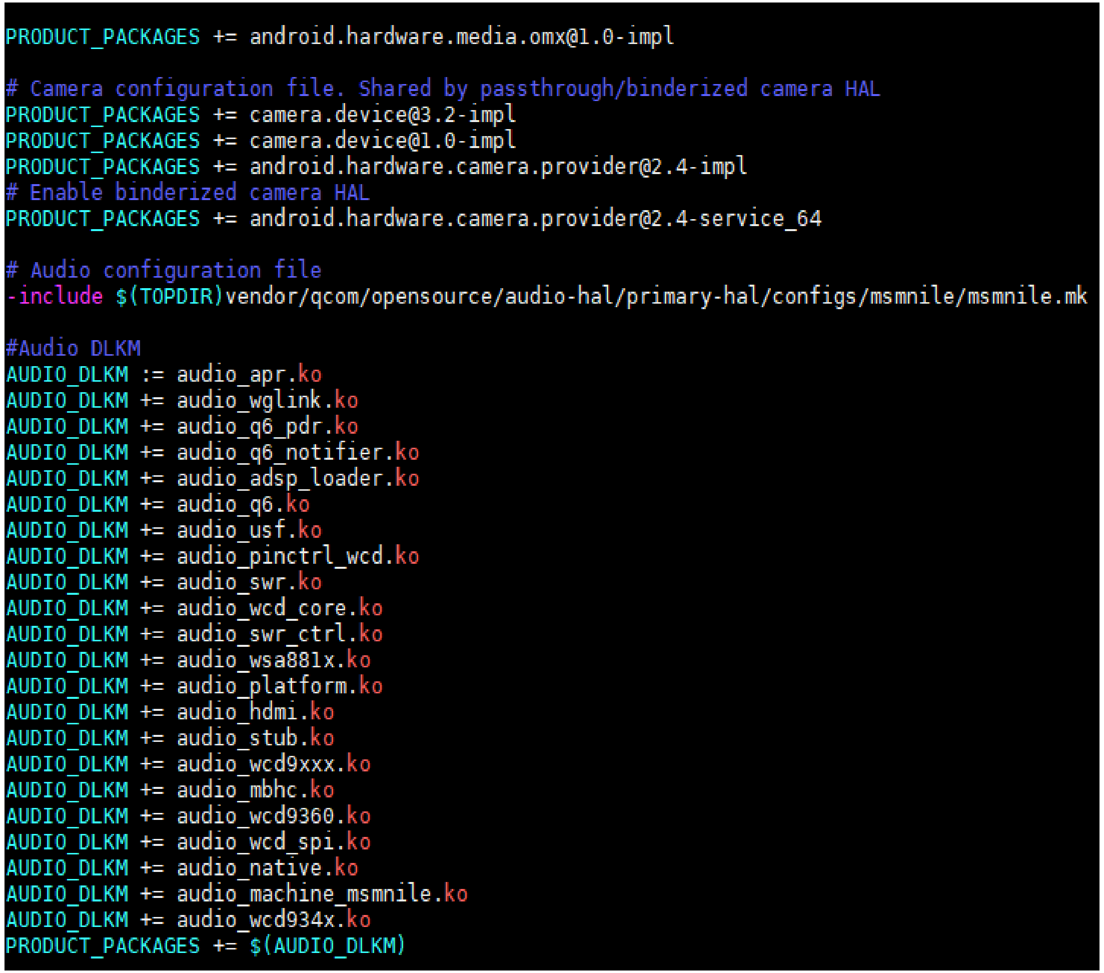


```
# Sensor conf files
PRODUCT_COPY_FILES += \
    device/qcom/msmnile/sensors/hals.conf:$(TARGET_COPY_OUT_VENDOR)/etc/sensors/hals.conf \
    frameworks/native/data/etc/android.hardware.sensor.accelerometer.xml:$(TARGET_COPY_OUT_VENDOR)/etc/permissions/android.hardware.sensor.accelerometer.xml \
    frameworks/native/data/etc/android.hardware.sensor.compass.xml:$(TARGET_COPY_OUT_VENDOR)/etc/permissions/android.hardware.sensor.compass.xml \
    frameworks/native/data/etc/android.hardware.sensor.gyroscope.xml:$(TARGET_COPY_OUT_VENDOR)/etc/permissions/android.hardware.sensor.gyroscope.xml \
    frameworks/native/data/etc/android.hardware.sensor.light.xml:$(TARGET_COPY_OUT_VENDOR)/etc/permissions/android.hardware.sensor.light.xml \
    frameworks/native/data/etc/android.hardware.sensor.proximity.xml:$(TARGET_COPY_OUT_VENDOR)/etc/permissions/android.hardware.sensor.proximity.xml \
    frameworks/native/data/etc/android.hardware.sensor.barometer.xml:$(TARGET_COPY_OUT_VENDOR)/etc/permissions/android.hardware.sensor.barometer.xml \
    frameworks/native/data/etc/android.hardware.sensor.stepcounter.xml:$(TARGET_COPY_OUT_VENDOR)/etc/permissions/android.hardware.sensor.stepcounter.xml \
    frameworks/native/data/etc/android.hardware.sensor.stepdetector.xml:$(TARGET_COPY_OUT_VENDOR)/etc/permissions/android.hardware.sensor.stepdetector.xml \
    frameworks/native/data/etc/android.hardware.sensor.ambient_temperature.xml:$(TARGET_COPY_OUT_VENDOR)/etc/permissions/android.hardware.sensor.ambient_temperature.xml \
    frameworks/native/data/etc/android.hardware.sensor.relative_humidity.xml:$(TARGET_COPY_OUT_VENDOR)/etc/permissions/android.hardware.sensor.relative_humidity.xml \
    frameworks/native/data/etc/android.hardware.sensor.hifi_sensors.xml:$(TARGET_COPY_OUT_VENDOR)/etc/permissions/android.hardware.sensor.hifi_sensors.xml
```

  - Forces to **copy of those files on the final build**

<device/board_name>.mk

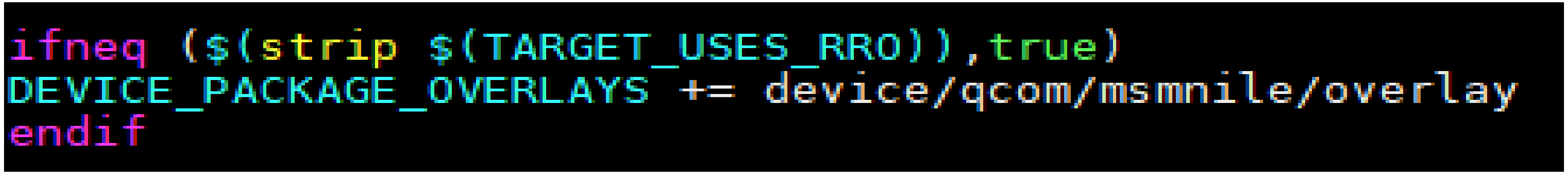
- Modules to be included:
  - Defines which modules we want to include on the build.
  - It could include **libs/apps** that are only defined under **device/<my_company>/<my_product>**.



```
PRODUCT_PACKAGES += android.hardware.media.omx@1.0-impl

# Camera configuration file. Shared by passthrough/binderized camera HAL
PRODUCT_PACKAGES += camera.device@3.2-impl
PRODUCT_PACKAGES += camera.device@1.0-impl
PRODUCT_PACKAGES += android.hardware.camera.provider@2.4-impl
# Enable binderized camera HAL
PRODUCT_PACKAGES += android.hardware.camera.provider@2.4-service_64

# Audio configuration file
-include $(TOPDIR)vendor/qcom/opensource/audio-hal/primary-hal/configs/msmnile/msmnile.mk

#Audio DLKM
AUDIO_DLKM := audio_apr.ko
AUDIO_DLKM += audio_wglink.ko
AUDIO_DLKM += audio_q6_pdr.ko
AUDIO_DLKM += audio_q6_notifier.ko
AUDIO_DLKM += audio_adsp_loader.ko
AUDIO_DLKM += audio_q6.ko
AUDIO_DLKM += audio_usf.ko
AUDIO_DLKM += audio_pinctrl_wcd.ko
AUDIO_DLKM += audio_swr.ko
AUDIO_DLKM += audio_wcd_core.ko
AUDIO_DLKM += audio_swr_ctrl.ko
AUDIO_DLKM += audio_wsa881x.ko
AUDIO_DLKM += audio_platform.ko
AUDIO_DLKM += audio_hdmi.ko
AUDIO_DLKM += audio_stub.ko
AUDIO_DLKM += audio_wcd9xxx.ko
AUDIO_DLKM += audio_mbhc.ko
AUDIO_DLKM += audio_wcd9360.ko
AUDIO_DLKM += audio_wcd_spi.ko
AUDIO_DLKM += audio_native.ko
AUDIO_DLKM += audio_machine_msmnile.ko
AUDIO_DLKM += audio_wcd934x.ko
PRODUCT_PACKAGES += $(AUDIO_DLKM)
```

<device/board_name>.mk

- Overriding **frameworks/packages config/layout** files



```
ifneq ($(strip $(TARGET_USES_RRO)),true)
DEVICE_PACKAGE_OVERLAYS += device/qcom/msmnile/overlay
endif
```

  - Defines a directory that will **override the AOSP sources**.
  - **Avoid changing the frameworks folder directly**
  - The sub folders must have the same **<build root>** structure:
    - device/<my_company>/<my_product>/overlay/frameworks/base/core/res/res/values/config.xml

<device/board_name>.mk

- Common overlayed files:

frameworks/base/core/res/res/values/config.xml

  - **config_supportAutoRotation**: Enables auto rotation support
  - **config_longPressOnPowerBehavior**: defines if pressing power button show a global actions menu, only power off or do nothing.
  - **config_shortPressOnPowerBehavior**: Similar to above but with other options
  - “Documented” here: **https://github.com/android/platform_frameworks_base/blob/master/core/res/res/values/config.xml**

<device/board_name>.mk

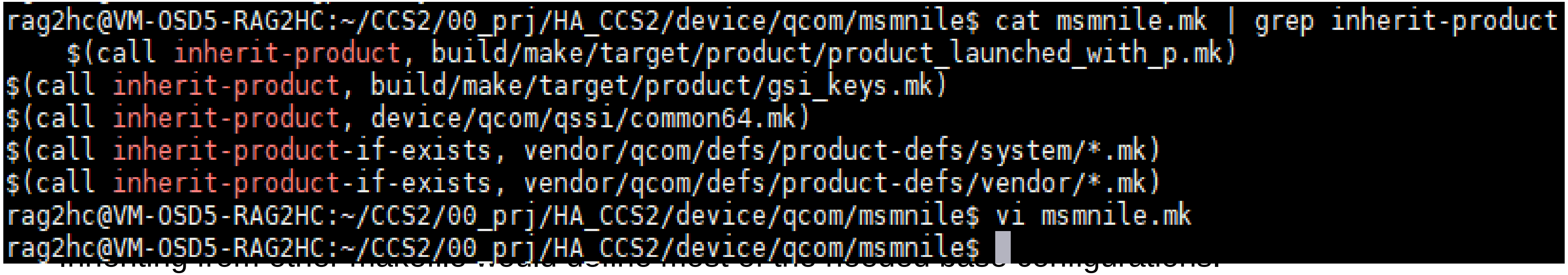
- Common overlayed files

frameworks/base/core/res/res/drawable-nodpi/default_wallpaper.jpg

  - Replaces the default wallpaper with no Wallpaper service customization

<device/board_name>.mk

- Inherit to reuse



```
rag2hc@VM-OSD5-RAG2HC:~/CCS2/00_prj/HA_CCS2/device/qcom/msmnile$ cat msmnile.mk | grep inherit-product
    $(call inherit-product, build/make/target/product/product_launched_with_p.mk)
$(call inherit-product, build/make/target/product/gsi_keys.mk)
$(call inherit-product, device/qcom/qssi/common64.mk)
$(call inherit-product-if-exists, vendor/qcom/defs/product-defs/system/*.mk)
$(call inherit-product-if-exists, vendor/qcom/defs/product-defs/vendor/*.mk)
rag2hc@VM-OSD5-RAG2HC:~/CCS2/00_prj/HA_CCS2/device/qcom/msmnile$ vi msmnile.mk
rag2hc@VM-OSD5-RAG2HC:~/CCS2/00_prj/HA_CCS2/device/qcom/msmnile$ 
```

  - Inheriting from other makefile would define most of the needed base configurations.

  - This will help us to **reuse the configuration** and to **have the minimal configuration** we need

Exercises

Exercises

1) Create new product name da3_<group_acc>_<your_NTID> (eg: **da3_cmtraning01_lug1hc**) with 3 build types: **user**, **userdebug**, **eng**

2) Add your new created product in **lunch** menu

3) New product can be **selectable** and **buildable**

4) For building the whole SW which can be flashed to DA3 target, please refer the document.

5) Flash your binary to DA3 target and get some properties below:

- ro.build.product
- ro.product.model
- ro.product.device
- ro.build.fingerprint

Thank for your listening!