



Genymobile

| Build AOSP

# AGENDA



01

Fetch sources



02

Building  
Configuration



03

Build & flash



04

AOSP puzzle

# 01 FETCH SOURCES

# FETCH SOURCES

Get materials



Genymobile

s.android.com

Sources  
(git)

Doc &  
tutorials

CDD, CTS

# FETCH SOURCES

Repo them all



repo



fetch

status

cmd

# FETCH SOURCES

fetch & coffee



Genymobile

```
# repo init
-u https://android.googlesource.com/platform/manifest
-b    android-6.0.0_r3

# repo sync -j8
```

# SOURCES DETAILS

Folders 1/5



.repo

repo configuration & data

bionic

Google Android libc implementation

bootable

Recovery

build

Build System

# SOURCES DETAILS

Folders 2/5



cts

Compatibility Test Suite

dalvik | art

Virtual Machine

device

Configuration for each target device



# SOURCES DETAILS

Folders 3/5



docs

[source.android.com](https://source.android.com)

external

boringssl, sqlite, bzip2, ...

frameworks

Android Framework -> API

hardware

Hardware Abstraction Layer

# SOURCES DETAILS

Folders 4/5



libcore

Java implementation (Apache Harmony)

ndk

Native Development Kit Sources

out

Results of YOUR compilations

packages

base apps/services (sms, photo, Settings, NFC ...)

# SOURCES DETAILS

Folders 5/5



prebuilts

Binaries (toolchain, gdbserver...)

sdk

Standard Development Kit Sources

system

Base : init, netd, vold, toolbox, logcat...

vendor

Vendors directories (libs, apps, overlays)

# LINUX KERNEL

Need a pingouin?



Genymobile

**As binary (default)**

/vendor | /device

**As sources (if lucky)**

/kernel



## 02 BUILDING CONFIGURATION

# HOWTO BUILD

3 basic cmds



Genymobile

```
# source build/envsetup.sh
```

```
# lunch
```

```
# make
```

# PREPARE TO BUILD



Set env

```
# source ./build/envsetup.sh
```

```
// include all existing vendorsetup.sh in ./device
// and define cmds
```

```
...
- lunch:    lunch <product_name>-<build_variant>
- croot:    Changes directory to the top of the tree.
- m:        Makes from the top of the tree.
- mm* :     Builds all of the modules in the current/suplied directory, with,
/without their dependencies.
- cgrep:    Greps on all local C/C++ files.
- jgrep:    Greps on all local Java files.
- resgrep:  Greps on all local res/*.xml files.
- sepgrep:  Greps on all loacal sepolicy files
...
```

# PREPARE TO BUILD



Prepare ingredient

```
add_lunch_combo TARGETNAME-TYPE
```

**vendorsetup.sh**

Lunch entry point  
in device/manufacturer/device

TYPE = eng | userdebug | release

```
PRODUCT_MAKEFILES = TARGETNAME.mk
```

**AndroidProducts.mk**

The first makefile



# BUILD BASICS



## Nexus 9 example

```
# tail -n1 device/htc/flounder/vendorsetup.sh  
add_lunch_combo aosp_flounder-userdebug
```

```
# tail -n3 device/htc/flounder/AndroidProducts.mk  
PRODUCT_MAKEFILES := \  
    $(LOCAL_DIR)/aosp_flounder.mk \  
    $(LOCAL_DIR)/aosp_flounder64.mk \  
    $(LOCAL_DIR)/aosp_flounder32.mk \  
    $(LOCAL_DIR)/aosp_flounder_64_only.mk
```



## device.mk files family

Define your  
device  
software

Heavy use  
of  
inheritance

No specific  
naming

# BUILD BASICS

## Reading recipes



Genymobile

```
# cat device/htc/flounder/aosp_flounder.mk | magic_summarize
```

```
PRODUCT_PACKAGES += \  
    rild \  
    Launcher3
```

```
# Inherit from those products. Most specific first.
```

```
$(call inherit-product, device/htc/flounder/product.mk)
```

```
$(call inherit-product, device/htc/flounder/device-lte.mk)
```

```
$(call inherit-product-if-exists, vendor/htc/flounder_lte/device-vendor.mk)
```

```
$(call inherit-product, $(SRC_TARGET_DIR)/product/aosp_base.mk)
```

```
PRODUCT_DEVICE := flounder
```

```
PRODUCT_MODEL := AOSP on Flounder
```

```
PRODUCT_MANUFACTURER := HTC
```

# BUILD BASICS



Genymobile

## Reading recipes

```
# cat device/htc/flounder/device.mk | magic_summarize
```

```
LOCAL_KERNEL := $(TARGET_PREBUILT_KERNEL)
```

```
PRODUCT_COPY_FILES := \  
    $(LOCAL_KERNEL):kernel
```

```
PRODUCT_COPY_FILES += \  
    $(LOCAL_PATH)/init.flounder.rc:root/init.flounder.rc \
```

```
PRODUCT_CHARACTERISTICS := tablet, nosdcard
```

```
PRODUCT_PACKAGES += \  
    nfc_nci.bcm2079x.default \  
    NfcNci
```

```
PRODUCT_PROPERTY_OVERRIDES := \  
    wifi.interface=wlan0 \  
    ro.opengles.version=196609 \
```



# BoardConfig.mk

Define your  
device  
hardware

Can include  
other  
BoardConfig.mk

A bunch of  
specific  
defines

# BUILD BASICS



Genymobile

## Reading recipes

```
# cat device/htc/flounder/BoardConfig.mk | magic_summarize

TARGET_CPU_ABI := arm64-v8a
TARGET_ARCH := arm64
TARGET_ARCH_VARIANT := armv7-a-neon
TARGET_CPU_VARIANT := denver
TARGET_NO_BOOTLOADER := true
BOARD_KERNEL_CMDLINE := androidboot.hardware=flounder

BOARD_USES_ALSA_AUDIO := true
BOARD_HAVE_BLUETOOTH := true
BOARD_HAVE_BLUETOOTH_BCM := true

BOARD_SEPOLICY_DIRS += device/htc/flounder/sepolicy
```



# BUILD

# BUILD SYSTEM



Set env

```
# source ./build/envsetup.sh
```

```
# lunch
```

You're building on Linux

Lunch menu... pick a combo:

1. aosp\_arm-eng
2. aosp\_x86-eng
3. aosp\_mips-eng
4. vbox\_x86-eng
5. mini\_armv7a\_neon-userdebug
6. mini\_x86-userdebug
7. aosp\_hammerhead-userdebug #n5
8. aosp\_mako-userdebug #n4
9. aosp\_flo-userdebug
10. aosp\_manta-userdebug #n10
- [...]



# BUILD SYSTEM

May the sources be with you



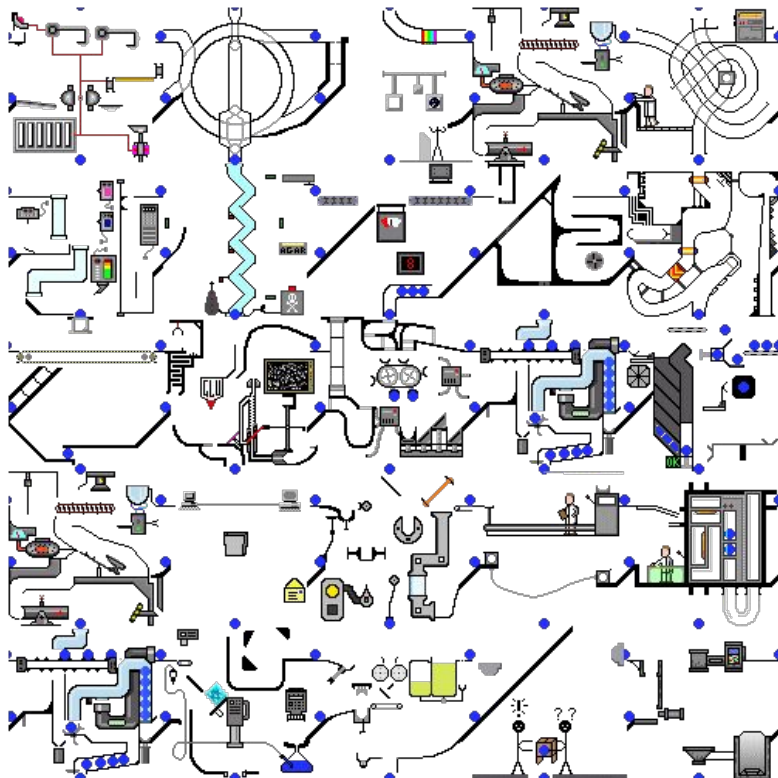
ONLY  
NEXUS  
BUILD  
YOU CAN

And maybe Sony ...



# BUILD SYSTEM

Let's compile



```
# make -j42
```

It's coffee time !

# BUILD SYSTEM

AOSP makefiles



Genymobile

**Android.mk, Android.mk everywhere**

Define a component : binary, apk, lib, ...

AOSP 6.0 : ~ 2900 Android.mk

# BUILD SYSTEM



## AOSP makefiles : Settings app

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

LOCAL_JAVA_LIBRARIES := bouncycastle conscrypt telephony-common ims-common
LOCAL_STATIC_JAVA_LIBRARIES := android-support-v4 android-support-v13 jsr305

LOCAL_SRC_FILES := $(call all-java-files-under, src)
LOCAL_RESOURCE_DIR := $(LOCAL_PATH)/res
LOCAL_PACKAGE_NAME := Settings
LOCAL_CERTIFICATE := platform
LOCAL_PRIVILEGED_MODULE := true
LOCAL_PROGUARD_FLAG_FILES := proguard.flags
include $(BUILD_PACKAGE)
```

**learn more in [build/core/clear\\_vars.mk](#)**

# BUILD SYSTEM

End of coffee time



Genymobile

[...]

#### make completed **successfully** (04:11:12 (hh:mm:ss)) ####

# BUILD SYSTEM

Output



Genymobile

Compilation  
produce  
plenty of  
images

One **.img** for each partition:  
system, boot, userdata,  
recovery, ...

# FLASH SYSTEM

Ready?



Genymobile

```
# adb reboot bootloader
```

```
# fastboot flashall -w
```



```
# make adb fastboot
```

# FLASH SYSTEM

Please, let me brick my device



Genymobile

```
# adb reboot bootloader
```

```
# fastboot oem unlock
```



May wipe your data



# FLASH SYSTEM

Last call



Genymobile

```
# fastboot flashall -w
```

or

```
# fastboot flashall ./system.img
```

```
# fastboot reboot
```

# FLASH SYSTEM

It's good to fail



Genymobile

Nice try!

But fails to boot / radio doesn't work  
/ there is no sound / ...

04

# AOSP PUZZLE

Missing pieces

# AOSP PUZZLE

About licensing

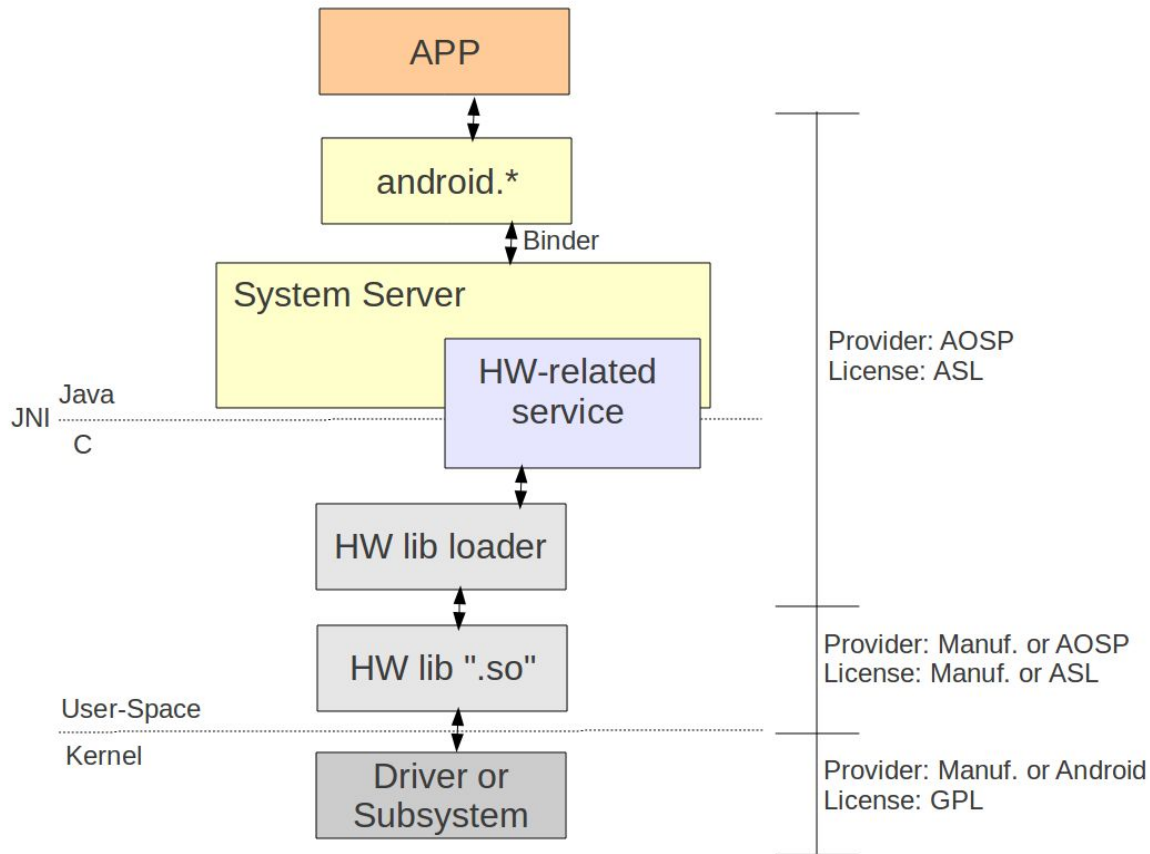


Kernel

Framework & App

# AOSP PUZZLE

## The full Stack



# AOSP PUZZLE

Were are missing blobs?



Genymobile

**Download them from Google/Sony/...**

<https://developers.google.com/android/nexus/drivers>

<http://developer.sonymobile.com/knowledge-base/open-source/>

**Or get them from official working image**

Distributed blobs are sometimes buggy

# AOSP PUZZLE

Were are missing blobs?



Genymobile

**Copy in /vendor + include in build system**

inherited in device.mk

**NEW :** in vendor partition for some devices

# AOSP PUZZLE

Try again



Genymobile

Delete ./out

Rebuild

Reflash



# COMPILATION TIME

Tired of coffee break



Genymobile

*Android support*

# CCACHE

# AOSP PUZZLE

Base apps looks bad!



Genymobile

The  
Gapps  
issue

## NOT Open source

- gapps not distributed for everybody
- aosp apps != gapps
- Google Mobile Services are getting fat

# ANDROID CERTIFICATION

Google approve this build



CDD

software &  
hardware  
requirements

CTS

API &  
function  
(auto)

CTS  
verifier

API &  
function  
(manual)

Google

final tests

# SUMMARY

What did he said?



Genymobile

```
# repo init ...  
# repo sync  
# [get binaries]  
  
# source ./build/envsetup.sh  
# lunch  
# make -j42  
  
# [unlock bootloader]  
# fastboot flashall
```

# REFERENCES



## Links

<https://source.android.com>

[http://elinux.org/Android\\_Portal](http://elinux.org/Android_Portal)

## Books

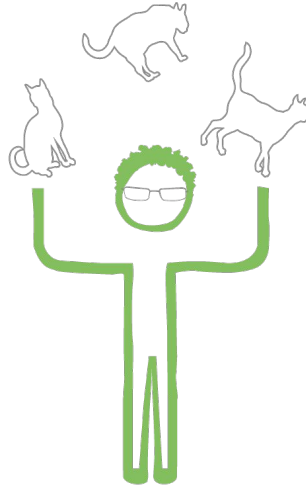
Embedded Android, K. Yaghmour ISBN: 978-1-4493-0829-2

The following copyright have been stripped from all the examples :

```
#  
# Copyright (C) 2013 The Android Open-Source Project  
#  
# Licensed under the Apache License, Version 2.0 (the "License");  
# you may not use this file except in compliance with the License.  
# You may obtain a copy of the License at  
#  
#     http://www.apache.org/licenses/LICENSE-2.0  
#  
# Unless required by applicable law or agreed to in writing, software  
# distributed under the License is distributed on an "AS IS" BASIS,  
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
# See the License for the specific language governing permissions and  
# limitations under the License.  
#
```



Genymobile



Thank you !

Vercoutère Guillaume  
[gvercoutere@gmail.com](mailto:gvercoutere@gmail.com)  
[www.genymobile.com](http://www.genymobile.com)