Security Class: Top-Secret( ) Secret( ) Internal( ) Public(√)

Rockchip_Android_GMS_Configuration_Guide

Status: [ ] Draft [√] Released [ ] Modifying	File No.:	RK-YH-YF-251
	Current Version:	V1.1
	Author:	Bian Jinchen
	Finish Date:	2020-06-30
	Auditor:	Chen Haiyan
	Finish Date:	2020-06-30

Version no.	Author	Revision Date	Revision Description	Remark
V1.0	Bian Jinchen	2020-02-10	Initial version release	
V1.1	Bian Jinchen	2020-06-30	Add some description for user build	

If there is any question about the document, please send email to: kenjc.bian@rock-chips.com

Terminology

- GMS (Google Mobile Service)
  Google Mobile Service, including Google service framework, Play Store, Chrome browser and other applications.
- 3PL
  The third party lab, it will assist Google to do the product certification in different regions of the world according to Google CDD requirements and GMS Requirments. There are some famous labs such as Foxconn/Haman/Windriver/Pegatron and so on. Customers can contact the lab by themselves, or ask sales to introduce.

- Google Partner/MADA (Mobile Application Distribution Agreement)
  After signing MADA with Google, customers can access Google Partner document and contact with 3PL to do GMS test and certification.
- EEA (European Economic Area)
  Device sold to EU countries needs to pass the GMS certification of the EEA, and the EEA type configured in the code needs to be configured for different situations and signed agreements. For instructions on EEA, please refer to the Google Partner documentation: https://support.google.com/androidpartners_gms/answer/9071728?hl=en
- xTS
  After executing the binary in the corresponding directory, you will enter the xTS-tradefed command line, and then execute the following test command or retry command.
  Please refer to：https://source.android.com/compatibility/cts/run
- retry
  It means to continue the test based on the existing results. After entering the tradefed, execute `l r` to view the previous test results. The first column is `session id`, and the parameter `--retry` when executing retry `session_id` to continue testing the specified result.
- Single test
  Refer to test certain test items separately for debugging. After the test, test results and logs are generated, which are located in the `results` and `logs` directories respectively.

Please know before testing

If you want to use Google mobile services on your device, you must pass GMS certification. After you pass the certification, you can enjoy Google's mobile services. The security and compatibility of your device can be greatly guaranteed.
To pass GMS certification, you must have qualifications. Generally speaking, there are two ways to pass GMS certification:

1. You have MADA
   In this case, there will be a Google contact for you, and you can consult with he/she about document permission issues and various account issues as Partner. To get GMS certification, you need to contact with Google or contact 3PL to apply. After passing the detailed configuration and self-tests of this document and passing all xTS tests in 3PL, it can be equipped with GMS services.
2. You do not have MADA
   Only the company with MADA qualification can apply for and pass GMS certification. If you do not have MADA qualification, please contact the company with ODM MADA qualification first. You can contact yourself or ask sales to introduce.

List of all GMS Tests

1. CTS
2. CTS verifier
3. GTS
4. VTS
5. CTS (ReferencePlan, CTS-ON-GSI)
6. STS
7. Performance Test (Only for Go Edition)

GMS SDK parameter settings

- Kernel compilation
  When compiling the kernel, please make sure to compile with Clang. If your product is build with Android Go, please replace android-10.config with android-10-go.config, for example:
  `make ARCH=arm64 rockchip_defconfig android-10-go.config rk3326.config`
- Android compilation
  Please check the following configurations located in:
  `device/rockchip/rkxxxx/BoardConfig.mk`:

1. Widevine
   Note that only RK3288 and RK3399 already support Widevine L1. For tablet devices without a radio, generally Widevine L3 can meet the requirements.

```
BOARD_WIDEVINE_OEMCRYPTO_LEVEL := 3
```

2. EEA，Change these macros only if the device is equipped with EEA

```
BUILD_WITH_EEA := true
BUILD_WITH_EEA_TYPE := type1 (Please config according to your EEA type)
```

3. Enable GMS bundle

```
BUILD_WITH_GOOGLE_MARKET := true
# Only change the following marco when your product has radio
BUILD_WITH_GOOGLE_MARKET_ALL := true
# When creating the user build, please set this marco to false to delete the
Rockchip StressTest tool, otherwise the device may not start normally.
PRODUCT_HAVE_RKAPPS := false
```

4. Keymaster & Optee
   Please pay attention to that when you apply for sublincense, only RK3326's optee version is V2, while others' are V1.

```
PRODUCT_HAVE_OPTEE := true
```

5. Hardware Features
   Please double check with your hardware and kernel driver colleagues the hardware supported by the device. If it is not supported (such as gyroscope, BLE, etc.), you must remove it from the software. It can be controlled by the following macros. If there is no macro, please go to frameworks/native/data/etc/ and delete the corresponding feature.xml:

```
BOARD_GRAVITY_SENSOR_SUPPORT := true
BOARD_COMPASS_SENSOR_SUPPORT := true
```

For example, remove BLE:

```
BOARD_BLUETOOTH_LE_SUPPORT := false
```

6. FRP (Factory Reset Protection)
   GMS requires FRP to be enabled. For more details on this function, please refer to the cooresponding document in SDK:
   `Rockchip_Introduction_Android_Factory_Reset_Protection_CN&EN.pdf`

```
BUILD_WITH_GOOGLE_FRP := true
```

7. AVB (Android Verified Boot)
   GMS requires AVB to be enabled. For more details on this function, please refer to the cooresponding document in SDK:
   `Rockchip_Introduction_Android_Verify_Boot_CN&EN.pdf`

```
BOARD_AVB_ENABLE := true
```

To facilitate development and debugging, the SDK will not open AVB by default, nor will it lock the bootloader. This will result in the failure to show that the certification has been made after passing the GMS certification. Before mass production, please ensure that the following patches are incorporated in uboot to ensure that the bootloader can be locked at the same time when the certification key is written and the certification is displayed normally.

```
RKDocs/android/patches/gms/0001-libavb-Lock-the-device-when-the-device-init-or-write.patch
```

8. SELinux
   GMS requires SELinux to be enabled. After enabling SELinux, some features may not work properly. For more details on this function, please refer to the cooresponding document in SDK:
   `Rockchip_Developer_Guide_Android_SELinux(Sepolicy)_CN.pdf`

```
device/rockchip/common/BoardConfig.mk:
BOARD_SELINUX_ENFORCING ?= true
```

9. Fixed fingerprint (optional)
   During testing, it may be necessary to adjust the firmware. It is recommended to keep fingerprint unchanged with the following patches before testing to avoid being unable to retry the xTS after updating the firmware.

```
diff --git a/prebuild.mk b/prebuild.mk
index 28391f6..7f38922 100644
--- a/prebuild.mk
+++ b/prebuild.mk
@@ -4,3 +4,4 @@ $(warning You can disable this by removing this and setting
BOARD_RECORD_COMMIT_
 $(shell test -d .repo && .repo/repo/repo manifest -r -o
$(OUT_DIR)/commit_id.xml)
 -include $(TARGET_DEVICE_DIR)/prebuild.mk

+ROCKCHIP_BUILD_NUMBER := 202001 (Some numbers)
```

10. Security Patch level
    Please refer to the cooresponding document in SDK to get and learn about the security patches:
    `Rockchip_Introduction_Android_Security_Patch_CN.pdf`
    When passing the GMS, please pay attention to the requirements of the GMS certification window. Generally speaking, the security patch is valid for three months, and the security patch of the firmware must be not too old when sending the device for xTS testing.

```
build/make:
core/version_defaults.mk
PLATFORM_SECURITY_PATCH := 2020-03-05
```

11. Attestation key
    Please write the attestation key before performing the GMS test. First you need to apply for a keybox from Google through 3PL. After getting the keybox, use the keybox packaging tool provided by Rockchip to package it. Package it into the programming format, and then use the programming tool to write to the device. The related tools are in the project directory:
    `RKTools/linux/Linux_AttestationKeyboxPack_Tool.rar`
    `RKTools/windows/KeyBoxWrite_v1.51_0109.zip`
    For detailed steps, please refer to the documentation provided along with the SDK:
    `Rockchip_User_Guide_KeyWrite_CN.pdf`
    If you want to apply for a keybox, please provide the following materials (for the specific requirements, consult with 3pl):

```
1. A single testcase report from CTS.
2. Device information (Ask 3pl for a template).
3. A file end with .txt, include Device ID.

Google Request to use Device ID to apply the key:
Device ID within the file should meet the following properties:
    1. Unique and cannot be duplicate of another Device IDs within the file
    2. Must be between 1-32 characters in length
    3. Only following characters allowed [a-z][A-Z][0-9][_][-][.]
    4. No whitespaces allowed

Device ID files have the following requirements:
    1. ASCII text file in unix format.
    2. File name should be created with the following characters [a-z][A-Z][0-
9][_][-][.] in a meaningful way (e.g. Make_Model_Date_Quantity.txt)
    3. Must only contain Device IDs. No comments, headers, or other
information
    4. One Device ID per line
    5. No duplicate Device IDs within file
    6. No blank lines
    7. No white spaces
```

12. What to check before sending to 3PL for testing
    Before sending the device to the 3PL for testing, you must carefully check according to GMS Requirements to save time, and the inspection methods are given for some requirements:

- Safety mode
  Press and hold `Volume-` during the boot animation or `Hold the restart icon when restarting`, then it will enter safety mode.
- Lockdown mode

1. Set lock screen password
2. Open the app `Settings`
3. Scroll down and click `Secure and Location`
4. Click `Lock Screen Preferences` under Device Security
5. Click `Show lockdown`

- FDE (Full Disk Encryption)/FBE (File Base Encryption)
  Check the encryption states on:
  `settings->security & location->Encryption & credentials->Encrypt tablet`
  Or:
  `adb shell getprop|grep crypt`

Preparation before test

1. Please follow the flashing method in the Rockchip Android SDK release notes to program the firmware for the test device;
2. Because the GMS bundle is integrated, the first startup after programming is slow. Please wait patiently. After the startup, complete the settings in the GMS Setup Wizard. Set the default language to United States English. Skip the wifi part first. Select the United States, set user information, and enter the Home screen;
3. Make sure the machine is configured as follows, set-> Wi-Fi to connect to wifi.
4. Before you start testing CTS or GTS, remember not to log in to your GMS account, otherwise there will be some failures;
5. Set `Settings->Security->Screenlock` to `None` ;
6. If the product (such as a laptop product) has a physical keyboard, `Languages & input-> Physical keyboard-> Show virtual keyboard`, check this option;
7. Click Settings-> About tablet (phone)-> Build Number continuously to make the hidden Developer Options appear;
8. Enable `Settings->Developeroptions->Stayawake` ;
9. Enable `Settings->Location` (Default is on, don't close) ;
10. Go to `Settings->Display->Sleep`, then set the time to the longest and adjust the brightness to the darkest (in order to save power during the long time testing) ;
11. Check the sensor calibration status. The calibration status is permanently valid. If the machine has been calibrated, you do not need to do it again. Be sure to confirm the calibration status before testing VTS. View method: `cat /sys/class/sensor_class/accel_calibration`. If there is a value to print, similar to `accel calibration: -604, 131, 535`, the calibration was successful. The uncalibrated machine is placed horizontally and stationary. Enter the command `echo 1> /sys/class/sensor_class/accel_calibration` to calibrate. Please confirm whether the calibration is successful after calibration;
12. The machine with the physical vertical screen should be placed in the vertical screen, and the machine with the physical horizontal screen should be placed in the horizontal screen;
13. Each time you retest, configure the machine as above.

Host configuration

You need to use Ubuntu and configure java, Python, adb, fastboot, aapt environment yourself. It is recommended to use Android_O_cts_env.tar.gz in RK-FTP, and configure environment variables after decompression:

```
In Ubuntu, you can add the following content to ~/.bashrc or edit the
following into a file such as env, then source env before testing:

export JAVA_HOME=/home/Your_Name/Software/jdk1.8.0_77
export JRE_HOME=${JAVA_HOME}/jre
export CLASSPATH=.:${JAVA_HOME}/lib:${JRE_HOME}/lib
export PATH=${JAVA_HOME}/bin:$PATH
export PATH=/home/Your_Name/Software/android-sdk-linux/tools:$PATH
export PATH=/home/Your_Name/Software/android-sdk-linux/platform-tools:$PATH
export PATH=/home/Your_Name/Software/android-sdk-linux/build-
tools/19.0.0:$PATH
```

If aapt is not available, please install the C ++ compatible library:

```
sudo apt-get install lib32stdc++6 lib32z1
```

1. Install Python development kit:

```
$ sudo apt-get install python-dev
```

2. Install Protocol Buffer tools (for Python):

```
$ sudo apt-get install python-protobuf
$ sudo apt-get install protobuf-compiler
```

3. Install Python virtual environment related tools:

```
$ sudo apt-get install python-virtualenv
$ sudo apt-get install python-pip
```

4. If you are using python3, you may need to install the following separately:

```
$ sudo apt install virtualenv
```

Q & A

1. Fail to create virtual environment
   If the virtual environment cannot be created when the VTS test fails, you can try to force pip to pip2 and install virtualenv through pip:

```
sudo apt autoremove python-virtualenv
sudo apt autoremove virtualenv
sudo ln -sf ~/.local/bin/pip2 /usr/bin/pip
pip install --user virtualenv
```

2. Upgrading adb and fastboot
   Replace with the adb and fastboot tools provided in the SDK. Refer to the documentation provided along with the SDK:
   `Rockchip_Android_10_development_guide_V1.2_CN.pdf`

```
Q: Replace your adb/fastboot
A: Take adb as an example, type in the terminal:
whereis adb
$ adb: /home/rockchip/Software/android-sdk-linux/platform-tools/adb
adb kill-server
After confirming the location of adb/fastboot, replace the adb/fastboot binary
```

VTS

Requires additional programming of GSI (i.e. system-xxx-signed.img of AOSP, available from Google official website/3PL or Rockchip security patch FTP, using signed image) and boot-debug.img (packaged firmware will be packaged after AVB is enabled in rockdev/Image-xxx). Please refer to the fastboot section of the SDK documentation for the programming method: `Rockchip_Android_10_development_guide_V1.2_CN.pdf`
Or refer to the Google Partner documentation:
https://support.google.com/androidpartners_gms/answer/9380762?hl=en

Test Suite	Test Command	Retry Command	Single Test Command
android-vts-xxx-arm_64.zip	`run vts`	`run vts --retry 0`	`run vts -m module_name -t case_name`

Cooperative testing of multiple devices

Most of the following tests support collaborative testing. The command is:
`--shard-count`

- For example, 3 devices run cts at the same time:

```
run cts --shard-count 3 -s SN1 -s SN2 -s SN3
```

CTS-ON-GSI

Need to flash GSI (ie AOSP's system-xxx-signed.img, which can be obtained from Google's official website/3PL or Rockchip security patch FTP, using a signed image). The flashing method is the same as above.

Test Suite	Test Command	Retry Command	Single Test Command
android-vts-xxx-arm_64.zip	`run cts-on-gsi`	`run retry --retry 0`	`run cts-on-gsi -m module_name -t case_name`

CTS

Use the fully compiled user firmware for testing. Please check and configure the firmware configuration as described in the document.

Test Suite	Test Command	Retry Command	Single Test Command
android-cts-10_xx-linux_x86-arm.zip	`run cts`	`run retry --retry 0`	`run cts -m module_name -t case_name`

CTS-Verifier

Use the fully compiled user firmware for testing. Please check and configure the firmware configuration as described in the document.

Test Suite	Test Command	Retry Command	Single Test Command
android-cts-verifier-10_xx-linux_x86-arm.zip	manual	manual	manual

GTS

Use the fully compiled user firmware for testing. Please check and configure the firmware configuration as described in the document. Before testing, be sure to confirm the host environment and whether the gts key is configured or not. You can check it through `echo $APE_API_KEY`.

- GTS key configuration method
  After getting it from 3PL, put the key and add it to the environment variables, such as:

```
$ vi .bashrc
export APE_API_KEY=/path/to/key.json
```

Test Suite	Test Command	Retry Command	Single Test Command
android-gts-xx.zip	`run gts`	`run retry --retry 0`	`run gts -m module_name -t case_name`

STS

Use the fully compiled userdebug firmware for testing. Please check and configure the firmware configuration as described in the document. For the details about dealing with the fail items, please refer to the SDK documentation `Rockchip_Introduction_Android_Security_Patch.pdf`

Test Suite	Test Command	Retry Command	Single Test Command
android-sts-xx.zip	`run sts-engbuild`	`run retry --retry 0`	`run sts -m module_name -t case_name`

GoTS (Performance Test)

Performance test uses the fully compiled userdebug firmware for testing. Please check and configure the firmware configuration as described in the document.

BTS

Since April 1, 2018, all devices that have been GMS certified must undergo Android BTS (Build Test Suite) testing. For details, see:
Google Partner documentation：
https://support.google.com/androidpartners_gms/answer/9027630?hl=en
When uploading firmware to Google, **Do not upload the packaged** `update.img`, otherwise Google will not be able to parse the image. For details, please check the documentation and ask 3pl. The BTS packaging method suggested by RK is: put the following images into the folder named fingerprint (all ":", "/" are changed to "~"), and then compress it to zip format.

```
rockchip~rk3326_qgo~rk3326_qgo~10~QD1A.190821.014.C2~201911~user~release-keys$
ls
boot-debug.img  boot.img  dtbo.img  misc.img  recovery.img  super.img
vbmeta.img
```

About PHA: Some customers have once passed all GMS tests, but finally were rejected because PHA (potential harmful applications) was detected in the BTS. According to the security meeting at the Google Hong Kong Summit, you can first upload the app that needs to be preset to Google Play to perform the test, and specifically consults with 3PL or TAM to avoid similar situations in BTS.

GMS Express

In order to establish a good Android ecosystem, so that all Andrid devices in the market can timely apply the security update released by Google monthly, Google launched the GMS Express Program.
Rockchip full range of Android 10.0 tablet platforms will support GMS Express. As shown in the following table, this document introduces Rockchip GMS Express technology related contents.

Soc Platforms	Go GMS Express	Regular Express
RK3126C	Ready	Ready
RK3326	Ready	Ready
RK3368	Ready	Ready
RK3288	N/A	WIP
RK3399	N/A	Ready

1. Precautions

1. Customers using Rockchip GMS Express Basline must ensure that they have signed a MADA agreement with Google. If they integrate Android Go, they must ensure that they have Go's supplementary MADA, otherwise all legal risks will be taken by themselves.

2. The Rockchip GMS Express baseline will be continuously updated every month (including security updates, GMS bundle updates, and AOSP important patches provided by Google, etc.) as required by Google. It will be officially released after the monthly baseline is available, so make sure to catch up with the update and push the update to the end user by OTA. If it is not updated in time, our company will not provide GMS related technical support.

2. Project download and configuration

Please contact RK FAE and the external SDK related contact to obtain the repo download address of GMS Express Baseline and download related permissions (Note: **This is not a SDK released by RK regularly**), otherwise regular security updates and GMS bundle updates will not be obtained. The SDK code for all platforms with Android 10 are unified. After you obtaining the Rockchip SDK, the method to update to the Express baseline is as follows, with no need to download it again:

```
repo init -m Android10_Express.xml
```

To get Express baselinue directly:

```
repo init --repo-url=ssh://git@www.rockchip.com.cn:2222/repo-
release/tools/repo.git  -u
ssh://git@www.rockchip.com.cn:2222/Android_Qt/manifests.git -m
Android10_Express.xml
```

2. Please ensure the following configuration in the Makefile of the product directory:

```
BUILD_WITH_GOOGLE_GMS_EXPRESS := true
```

After this macro configuration is opened, **we do not preset any RK applications, you need to add it by yourself.** After compilation, the com.google.android.feature.GMSEXPRESS_BUILD flag will be configured by default. Our baseline meets Express Plus Claim.
Google encourages Express Plus products to receive relevant subsidy support after passing GMS certification. Please consult with 3PL for more details. After the secondary development of a specific project, please refer to the following document to confirm whether it meets the requirements of Express Plus or not. If yes, modify the following files:
vendor/rockchip/common/gms-express.xml

```
diff --git a/gms-express.xml b/gms-express.xml
index 78f4d99..de93557 100644
--- a/gms-express.xml
+++ b/gms-express.xml
@@ -16,5 +16,5 @@
  -->
  <!-- These are configurations that should exist on GMS Express devices. -->
  <config>
-     <feature name="com.google.android.feature.GMSEXPRESS_BUILD " />
+     <feature name="com.google.android.feature.GMSEXPRESS_PLUS_BUILD" />
  </config>
```

Bootup wizard cannot skip Wi-Fi setting with probability

Poor performance leads to this issue. Patch the following CL:

```
patch/frameworks/base/increase_waiting_time_for_setup_wizard.diff
```

After passing the certification, cannot display the `certified`

There are generally two cases:

1. The device doesn't flash the attestation key
2. The device is unlocked
   To facilitate development and debugging, the state of the device is set to `unlocked` by default. Here is a CL to make the device locked when the attestation key is programmed. Before mass production, this patch **MUST** be applied to ensure that the device can be locked when the attestation key is programmed.

```
patches/gms/0001-libavb-Lock-the-device-when-the-device-init-or-write.patch
```