

Android RIL Driver

User Guide

Version: 2.0

Date: 2021-04-19

Status: Released



Our aim is to provide customers with timely and comprehensive service. For any assistance, please contact our company headquarters:

Quectel Wireless Solutions Co., Ltd.

Building 5, Shanghai Business Park Phase III (Area B), No.1016 Tianlin Road, Minhang District, Shanghai 200233, China

Tel: +86 21 5108 6236

Email: info@quectel.com

Or our local office. For more information, please visit:

<http://www.quectel.com/support/sales.htm>.

For technical support, or to report documentation errors, please visit:

<http://www.quectel.com/support/technical.htm>

Or email to support@quectel.com.

General Notes

Quectel offers the information as a service to its customers. The information provided is based upon customers' requirements. Quectel makes every effort to ensure the quality of the information it makes available. Quectel does not make any warranty as to the information contained herein, and does not accept any liability for any injury, loss or damage of any kind incurred by use of or reliance upon the information. All information supplied herein is subject to change without prior notice.

Disclaimer

While Quectel has made efforts to ensure that the functions and features under development are free from errors, it is possible that these functions and features could contain errors, inaccuracies and omissions. Unless otherwise provided by valid agreement, Quectel makes no warranties of any kind, implied or express, with respect to the use of features and functions under development. To the maximum extent permitted by law, Quectel excludes all liability for any loss or damage suffered in connection with the use of the functions and features under development, regardless of whether such loss or damage may have been foreseeable.

Duty of Confidentiality

The Receiving Party shall keep confidential all documentation and information provided by Quectel, except when the specific permission has been granted by Quectel. The Receiving Party shall not access or use Quectel's documentation and information for any purpose except as expressly provided herein. Furthermore, the Receiving Party shall not disclose any of the Quectel's documentation and information to any third party without the prior written consent by Quectel. For any noncompliance to the above requirements, unauthorized use, or other illegal or malicious use of the documentation and information, Quectel will reserve the right to take legal action.

Copyright

The information contained here is proprietary technical information of Quectel. Transmitting, reproducing, disseminating and editing this document as well as using the content without permission are forbidden. Offenders will be held liable for payment of damages. All rights are reserved in the event of a patent grant or registration of a utility model or design.

Copyright © Quectel Wireless Solutions Co., Ltd. 2021. All rights reserved.

About the Document

Revision History

Version	Date	Author	Description
1.0	2015-02-27	Carl YIN	Initial
1.1	2015-03-25	Carl YIN	Updated supported products
1.2	2015-04-07	Kent XU	Added zero packet feature in Section 3.3.3.
1.3	2015-07-10	Kent XU	<ol style="list-style-type: none"> Added GSM modules in supported products Added Android 5.x in supported Android versions
1.4	2016-06-21	Carl YIN/ Neo HOU	<ol style="list-style-type: none"> Added EC21&EC25 in supported products Added FAQ
1.5	2016-08-23	Kent XU	Added EC20 R2.0 in supported products
1.6	2017-01-05	Macro GAO	<ol style="list-style-type: none"> Added RIL driver integration by library in Section 3.5. Deleted how to modify the right of RILD. Added how to configure device permissions in Section 3.6.2. Added how to enable SELinux in Section 3.6.3.
1.7	2018-07-13	Macro GAO	Added Android 8.x in supported Android versions
1.8	2019-03-07	Macro GAO	<ol style="list-style-type: none"> Added Android 9.x in supported Android versions. Added EC21, EC200T, EM06, EM12, EG12, AG36, UC200T& MC90 in supported products. Updated the description of system configuration in Chapter 3. Added how to configure SELinux in Chapter 4.
1.9	2020-05-11	Marco GAO	<ol style="list-style-type: none"> Updated the applicable modules in Chapter 1.1. Updated the supported Android versions in Chapter 2.3. Added the supported IRadio versions in Chapter 2.4. Updated the description in Chapter 3.2. Updated reference documents for USB driver installation

			<p>of different modules in Chapter 3.3.</p> <ol style="list-style-type: none"> 6. Updated Quectel RIL driver version for customers using Android 8.0 or later versions in Chapter 3.4. 7. Updated the description of service ril-daemon configuration in Chapter 3.5.1. 8. Updated HIDL description in Chapter 3.5.3. 9. Updated how to configure SELinux in Chapter 4. 10. Updated how to catch logs in Chapter 5. 11. Removed the section of Why Short Messages cannot be Sent or Received and added Why Phone Process Does not Work in Chapter 6.
2.0	2021-04-19	Marco GAO	<ol style="list-style-type: none"> 1. Added EC200S to the applicable modules in Chapter 1.1. 2. Updated the file structure of Quectel RIL driver package in Chapter 2.1. 3. Added Android 11.x to the supported Android versions in Chapter 2.3. 4. Added IRadio 1.3 and 1.4 to the supported IRadio versions in Chapter 2.4. 5. Updated the path of RIL library files in Android system for Android 10.x/11.x (IRadio 1.0) and for IRadio 1.4 in Chapter 3.4. 6. Added support for PCIe access in Chapter 3.5.1.1 & 4.3. 7. Updated the HIDL description to be added for Android 10.x/11.x (IRadio 1.0) and for IRadio 1.4 in Chapter 3.5.3. 8. Updated the procedures of catching logs automatically in Chapter 5.1. 9. Added the command to check the definition of service ril-daemon in <i>*rild.rc</i> in Chapter 6.2.

Contents

About the Document	3
Contents	5
Table Index	6
Figure Index	7
1 Introduction	8
1.1. Applicable Modules.....	8
2 Overview of Android RIL Driver	10
2.1. Directory Structure.....	10
2.2. Supported Functions.....	11
2.3. Supported Android Versions.....	11
2.4. Supported IRadio Versions.....	12
3 RIL Integration	13
3.1. Android RIL Architecture.....	13
3.2. PPP Configuration in Linux Kernel.....	14
3.3. USB Driver Integration in Linux Kernel.....	14
3.4. RIL Driver Integration by Library.....	14
3.5. System Configuration.....	15
3.5.1. Configure Service ril-daemon.....	15
3.5.1.1. Modules Accessed via USB/PCIe Interface.....	16
3.5.1.2. Modules Accessed via UART Interface.....	16
3.5.2. Modify rild.c (for Android 4.x–7.x).....	17
3.5.3. Add HIDL Description (for Android 8.x–11.x).....	17
4 SELinux Configuration	19
4.1. Modify Service ril-daemon Configuration.....	19
4.2. Uncomment the switchUser() Function.....	19
4.3. Configure SELinux Rule for RIL.....	19
5 Debugging Methods	21
5.1. Catch Logs Automatically (Recommended).....	21
5.2. Catch Logs Manually.....	21
5.3. Common Log Tags.....	22
6 FAQs	23
6.1. How to Set the APN.....	23
6.2. Why Quectel RIL Driver Does Not Work.....	24
6.3. Why Phone Process Does Not Work.....	25
7 Appendix A References	26

Table Index

Table 1: Applicable Modules.....	8
Table 2: Supported Functions	11
Table 3: Supported Android Versions.....	11
Table 4: Supported IRadio Versions.....	12
Table 5: Common Log Tags	22
Table 6: Related Documents	26
Table 7: Terms and Abbreviations	26

Figure Index

Figure 1: File Structure of RIL Driver Package	10
Figure 2: Android RIL Architecture	13
Figure 3: Edit Access Point	23

1 Introduction

This document mainly introduces how to integrate RIL (Radio Interface Layer) driver into the Android OS of your target devices as well as how to modify the configuration files for starting the RIL service.

1.1. Applicable Modules

The document is applicable to the following Quectel modules.

Table 1: Applicable Modules

LTE Standard Module Series	EC2x: EC25/ EC20 R2.1/ EC21
	EG9x: EG91/ EG95
	EG2x-G: EG21-G/ EG25-G
	EM05
	EC200x: EC200T/ EC200S
LTE-A Module Series	Ex06: EM06/ EP06/ EG06
	Ex12: EM12-G/ EG12
Automotive Module Series	AGxx: AG35
LPWA Module Series	BGxx: BG95/ BG96
UMTS/HSPA(+) Module Series	UCxx: UC15/ UC200T
	UGxx: UG95/ UG96
GSM/GPRS/GNSS Module Series	Mxx: M66/ M72/ M95
	MCxx: MC60/ MC90
5G Module Series	RM5xxQ: RM500Q/ RM510Q-GL
	RG50xQ: RG500Q

NOTE

Quectel modules listed above may include multiple models. Please refer to the corresponding module specifications for details.

2 Overview of Android RIL Driver

2.1. Directory Structure

The file structure of Quectel RIL driver package is shown as below.

```

├── arm64-v8a
│   ├── chat
│   ├── ip-down
│   ├── ip-up
│   └── libreference-ril.so
├── armeabi
│   ├── chat
│   ├── ip-down
│   ├── ip-up
│   └── libreference-ril.so
├── doc
│   ├── Quectel_Android_RIL_Driver_User_Guide_V2.0.pdf
│   └── ReleaseNote.txt
├── libril
│   ├── arm64-v8a
│   │   └── libril.so
│   └── armeabi
│       └── libril.so
└── ql-ril.conf
    
```

Figure 1: File Structure of RIL Driver Package

2.2. Supported Functions

Quectel RIL driver supports the following functions.

Table 2: Supported Functions

Functions	Supported or Not
SMS	Yes
Voice Call	Yes
Data Service	Yes
(U)SIM Tool Kit	No
Phonebook	Yes

2.3. Supported Android Versions

Presently, Quectel RIL driver supports the following Android versions.

Table 3: Supported Android Versions

Versions	Supported or Not
Android 4.x	Yes
Android 5.x	Yes
Android 6.x	Yes
Android 7.x	Yes
Android 8.x	Yes
Android 9.x	Yes
Android 10.x	Yes
Android 11.x	Yes

2.4. Supported IRadio Versions

Quectel RIL driver supports the following IRadio versions.

Table 4: Supported IRadio Versions

Versions	Supported or Not
IRadio 1.0	Yes
IRadio 1.1	Yes
IRadio 1.2	Yes
IRadio 1.3	Yes
IRadio 1.4	Yes

3 RIL Integration

The chapter mainly describes the Android RIL architecture and procedures of setting up an Android system with the RIL driver.

3.1. Android RIL Architecture

Android RIL provides the abstract layer between Android telephony services and the radio hardware. The following figure illustrates the RIL in the context of Android telephony architecture.

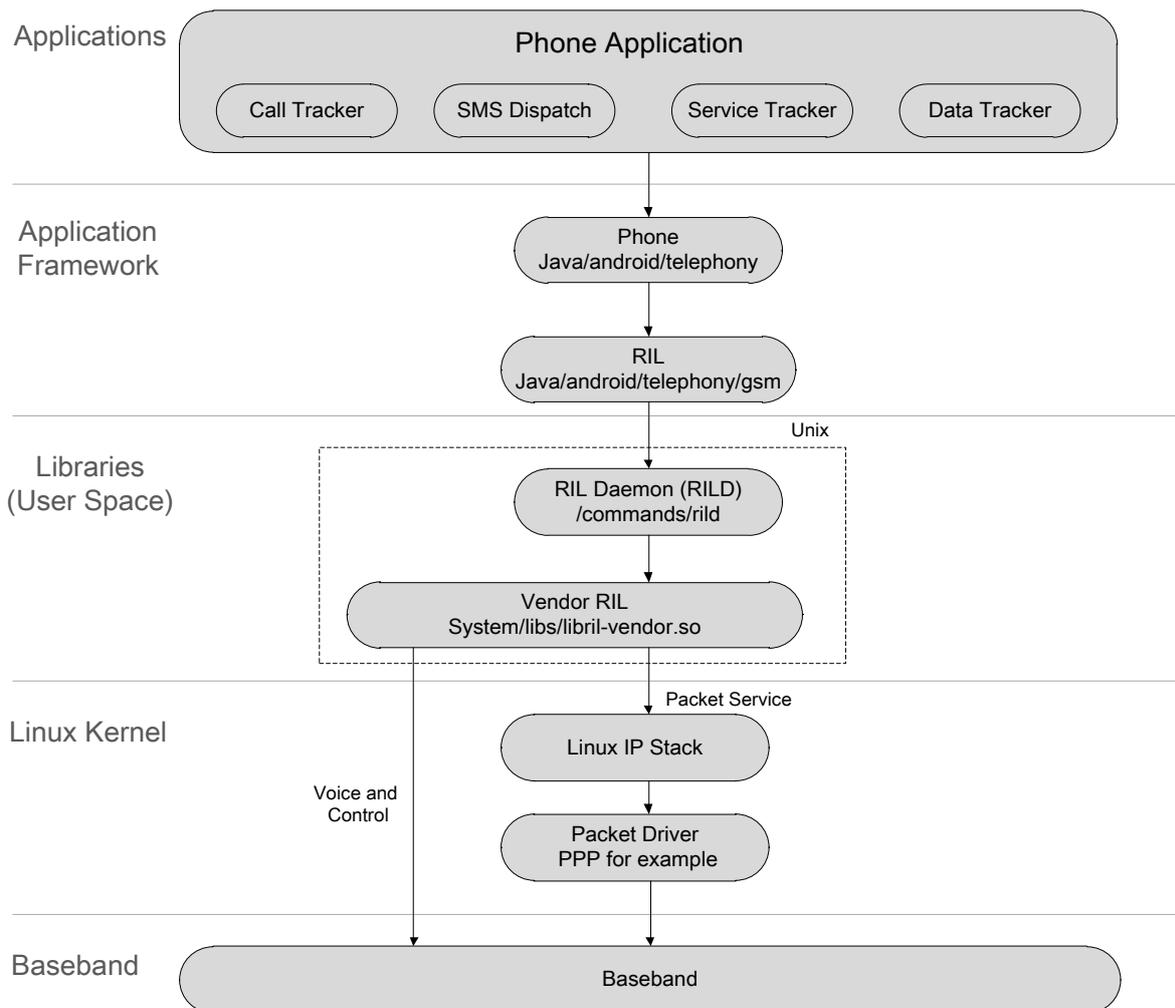


Figure 2: Android RIL Architecture

Located between the kernel and application framework, the Android RIL consists of two parts: RILD and Vendor RIL. RILD is responsible for communications between socket and application framework, while Vendor RIL is responsible for radio communication via AT command channel and data communication via packet data channel (PDCH).

Additionally, the java framework of RIL consists of two parts as well: RIL module and phone module. The RIL module communicates with the lower RILD, while the phone module directly provides phone function interfaces to applications.

3.2. PPP Configuration in Linux Kernel

If PPP dial-up function is needed, you need to configure the kernel to support the function. Otherwise, skip this step. For detailed procedures, see the reference documents in **Table 6**.

3.3. USB Driver Integration in Linux Kernel

For modules accessed via USB interface, integrate the USB drivers in Linux kernel. Otherwise, skip this step. For detailed procedures, see the reference documents in **Table 6**.

3.4. RIL Driver Integration by Library

Put the corresponding RIL library files provided by Quectel to the following path of Android system.

1. IRadio 1.0 (Default)

Android Version	Path (32-bit)	Path (64-bit)
4.x-7.x	/system/lib/libreferece-ril.so	/system/lib64/libreferece-ril.so
8.x/9.x	/vendor/lib/libreferece-ril.so	/vendor/lib64/libreferece-ril.so
10.x/11.x	/vendor/lib/hw/libreferece-ril.so	/vendor/lib64/hw/libreferece-ril.so

2. IRadio 1.4

Android Version	Path (32-bit)	Path (64-bit)
10.x/11.x	/vendor/lib/hw/libreferece-ril.so /vendor/lib/libril.so	/vendor/lib64/hw/libreferece-ril.so /vendor/lib64/libril.so

3. To Support PPP (Optional)

```
/system/bin/chat
/system/etc/ppp/ip-down
/system/etc/ppp/ip-up
```

NOTE

For Android 11.x or IRadio 1.4, Quectel RIL driver V3.3.40 or later versions should be used.

3.5. System Configuration

In order to use the RIL driver normally, some configuration files in Android system should be modified.

3.5.1. Configure Service ril-daemon

Service ril-daemon¹⁾ can be configured by adding the following lines to *init*.rc* or **rild.rc*. The relevant lines vary depending on the accessing interface and the Android version being used.

The location of *init*.rc* varies according to your project settings. The following gives a non-exhaustive list of file paths which may contain the *init*.rc* file.

- *device/fsl/imx6dq/sabresd_6dq/init.rc*
- *device/ti/am335xevm_sk/init.am335xevm.rc*
- *device/rockchip/rk3399/init.rk3399.rc*
- *device/samsung/smdkv210/init.smdkv210_sdmmc.rc*

**rild.rc* is only available for Android 7.x or later versions.

NOTE

¹⁾ The name of RILD service might also be vendor.ril-daemon. In this document, we will only use service ril-daemon for illustration.

3.5.1.1. Modules Accessed via USB/PCIe Interface

For modules accessed via USB/PCIe interface, add the following lines to *init*.rc* or **rild.rc*.

- **For Android 4.x–7.x**

```
service ril-daemon /system/bin/rild -l <libreference-ril path>
  class main
  socket rild stream 660 root radio
  socket rild-debug stream 660 radio system
  user root
  group radio cache inet misc audio sdcard_rw log
```

- **For Android 8.x–11.x**

```
service ril-daemon /vendor/bin/hw/rild -l <libreference-ril path>
  class main
  user root
  group radio cache inet misc audio sdcard_rw log
  capabilities BLOCK_SUSPEND NET_ADMIN NET_RAW
```

- **Parameter**

Parameter	Type	Description
-l <libreference-ril path>	Required parameter	Path of <i>libreference-ril.so</i> .

3.5.1.2. Modules Accessed via UART Interface

For modules accessed via UART interface, add the following lines to *init*.rc* or **rild.rc*.

- **For Android 4.x–7.x**

```
service ril-daemon /system/bin/rild -l <libreference-ril path> -- -d <UART port name> -B <baud rate> -C
<hardware flow control>
  class main
  socket rild stream 660 root radio
  socket rild-debug stream 660 radio system
  user root
  group radio cache inet misc audio sdcard_rw log
```

- **For Android 8.x–11.x**

```
service ril-daemon /vendor/bin/hw/rild -l <libreference-ril path> -- -d <UART port name> -B <baud rate>
-C <hardware flow control>
```

```
class main
user root
group radio cache inet misc audio sdcard_rw log
capabilities BLOCK_SUSPEND NET_ADMIN NET_RAW
```

- **Parameter**

Parameter	Type	Description
-l <libreference-ril path>	Required parameter	Path of <i>libreferece-ril.so</i> .
-d <UART port name>	Required parameter	Name of the UART port being used. For example: <i>/dev/ttyS1</i> .
-B <baud rate>	Optional parameter	Data rate of the UART port. For example: 115200, 230400 or 460800. Default: 115200. Unit: bps.
-C <hardware flow control>	Optional parameter	Enable/Disable the hardware flow control function. 1: Enable 0: Disable (default)

3.5.2. Modify rild.c (for Android 4.x–7.x)

For Android 4.x–7.x, comment the *switchUser()* function in the file *(\$Android_src)/hardware/ril/rild/rild.c* to get root privilege for RILD (ril-daemon).

```
OpenLib:
#endif
//switchUser();

dlHandle = dlopen(rilLibPath, RTLD_NOW);
```

3.5.3. Add HIDL Description (for Android 8.x–11.x)

For Android 8.x–11.x, the communication interface between Android phone framework and service *ril-daemon* has changed from socket to HIDL. Therefore, the following HIDL description needs to be added to *manifest.xml*.

- **IRadio 1.0 (Default)**

For Android 8.x/9.x:

```
<hal format="hidl">
  <name>android.hardware.radio.deprecated</name>
  <transport>hwbinder</transport>
  <version>1.0</version>
```

```

        <interface>
            <name>IOemHook</name>
            <instance>slot1</instance>
        </interface>
</hal>
<hal format="hidl">
    <name>android.hardware.radio</name>
    <transport>hwbinder</transport>
    <version>1.0</version>
    <interface>
        <name>IRadio</name>
        <instance>slot1</instance>
    </interface>
</hal>
    
```

For Android 10.x/11.x:

```

<hal format="hidl">
    <name>android.hardware.radio</name>
    <transport>hwbinder</transport>
    <version>1.0</version>
    <interface>
        <name>IRadio</name>
        <instance>slot1</instance>
    </interface>
</hal>
    
```

- **IRadio 1.4**

For Android 10.x/11.x:

```

<hal format="hidl">
    <name>android.hardware.radio</name>
    <transport>hwbinder</transport>
    <fqname>@1.4::IRadio/slot1</fqname>
    <fqname>@1.2::ISap/slot1</fqname>
</hal>
    
```

The location of *manifest.xml* file varies according to your project settings. For example:

- *device/rockchip/rk3399/manifest.xml*
- *device/fsl/imx6dq/sabresd_6dq/manifest.xml*

4 SELinux Configuration

If the SELinux installed in your Android devices is enabled (i.e. in enforcing mode), follow the procedures below to make sure that Quectel RIL has full access to SELinux privileges. Otherwise, you can skip this chapter.

4.1. Modify Service ril-daemon Configuration

To make sure Quectel RIL has full access to SELinux privileges, the user of service ril-daemon should be radio. Therefore, the user of service ril-daemon in the lines illustrated in **Chapter 3.5.1** should be changed from root to radio. An example with Android 8.x–11.x for modules accessed via USB/PCIe interface is shown below.

```
service ril-daemon /vendor/bin/hw/rild -l <libreference-ril path>
    class main
    user radio
    group radio cache inet misc audio sdcard_rw log
    capabilities BLOCK_SUSPEND NET_ADMIN NET_RAW
```

4.2. Uncomment the switchUser() Function

To make sure Quectel RIL has full access to SELinux privileges, the *switchUser()* function mentioned in **Chapter 3.5.2** should be uncommented.

4.3. Configure SELinux Rule for RIL

The following definition should be added to *ueventd.rc*:

```
#quectel port
/dev/ttyUSB*          0660      radio    radio
/dev/ttyACM*          0660      radio    radio
/dev/cdc-wdm*          0660      radio    radio
/dev/qcqm*             0660      radio    radio
```

```

/dev/cdc-acm*          0660      radio   radio
#PCIe
/dev/mhi_DUN          0660      radio   radio
/dev/mhi_DIAG         0660      radio   radio
/dev/mhi_BHI          0660      radio   radio
/dev/mhi_LOOPBACK     0660      radio   radio
/dev/mhi_QMI0         0660      radio   radio
    
```

The following definition should be added to *file_contexts*:

```

/dev/ttyUSB[0-9]      u:object_r:radio_device:s0
/dev/ttyACM[0-9]      u:object_r:radio_device:s0
/dev/cdc-wdm[0-9]     u:object_r:radio_device:s0
/dev/qcqm[0-9]        u:object_r:radio_device:s0
/vendor/bin/hw/rild    u:object_r:rild_exec:s0
/dev/socket/rildOemHook u:object_r:rild_socket:s0
#PCIe
/dev/mhi_DUN          u:object_r:radio_device:s0
/dev/mhi_DIAG         u:object_r:radio_device:s0
/dev/mhi_BHI          u:object_r:radio_device:s0
/dev/mhi_LOOPBACK     u:object_r:radio_device:s0
/dev/mhi_QMI0         u:object_r:radio_device:s0
    
```

The following definition should be added to *rild.te*:

```
allow rild self:packet_socket { create bind write read };
```

The locations of *ueventd.rc*, *file_contexts* and *rild.te* may vary according to your project settings. But in general, these files are located in the following paths respectively by default.

- *device/fsl/imx6dq/sabresd_6dq/ueventd.freescale.rc*
- *device/fsl/imx6dq/sabresd_6dq/sepolicy/file_contexts*
- *device/fsl/imx6dq/sabresd_6dq/sepolicy/rild.te*
- *device/rockchip/common/ueventd.rockchip.rc*
- *device/rockchip/common/sepolicy/file_contexts*

5 Debugging Methods

5.1. Catch Logs Automatically (Recommended)

Catching logs automatically is recommended for Quectel RIL driver and you should fulfil two prerequisites: SELinux is disabled and the library files applied are provided by Quectel. The detailed procedures are as below:

- 1) Create a folder `/data/quectel_debug_log` in the Android system and then restart Android.

```
adb root
adb shell mkdir /data/quectel_debug_log
adb shell chmod 777 /data/quectel_debug_log
adb reboot
```

- 2) If SELinux is in enforcing mode, set it to permissive mode and remember not to restart Android. (If SELinux is not in enforcing mode, skip this step.)

```
adb root
adb shell setenforce 0
adb shell killall rild
```

- 3) Get logs to local.

```
adb pull /data/quectel_debug_log
```

5.2. Catch Logs Manually

Quectel RIL driver also supports catching logs manually. The detailed procedures are as below:

- 1) Catch the logs of RIL module by typing the following command in Window's CMD tool:

```
adb logcat -b radio -v time
```

- 2) Catch the logs of Android system by typing the following command in Window's CMD tool:

```
adb logcat -v time
```

5.3. Common Log Tags

The following table lists some log tags that are commonly applied.

Table 5: Common Log Tags

RIL	<i>/hardware/ril/reference-ril/refereince-ril.c</i>
AT	<i>/hardware/ril/reference-ril/atchannel.c</i>
RILD	<i>/hardware/ril/rild/rild.c</i>
RILC	<i>/hardware/ril/libril/ril.cpp</i>
RILB	<i>/frameworks/base/telephony/java/com/android/internal/telephony/BaseCommands.java</i>
RILJ	<i>/frameworks/base/telephony/java/com/android/internal/telephony/gsm/RIL.java</i>
GSM	<i>/frameworks/base/telephony/java/com/android/internal/telephony/gsm/GSMPhone.java</i>

6 FAQs

6.1. How to Set the APN

If the dialling process is interrupted, it is quite possible that the APN has not been set yet. Check the APN in Android UI: “Settings” → “WIRELESS & NETWORKS” → “...” → “Mobile Networks” → “Access Point Names”. If nothing is found in “Access Point Names”, it indicates that the APN has not been set. In such a case, you need to add a new APN to the system. The following figure shows an example of the access point editing interface. Please note that the Access Point Name varies according to operators and (U)SIM cards.

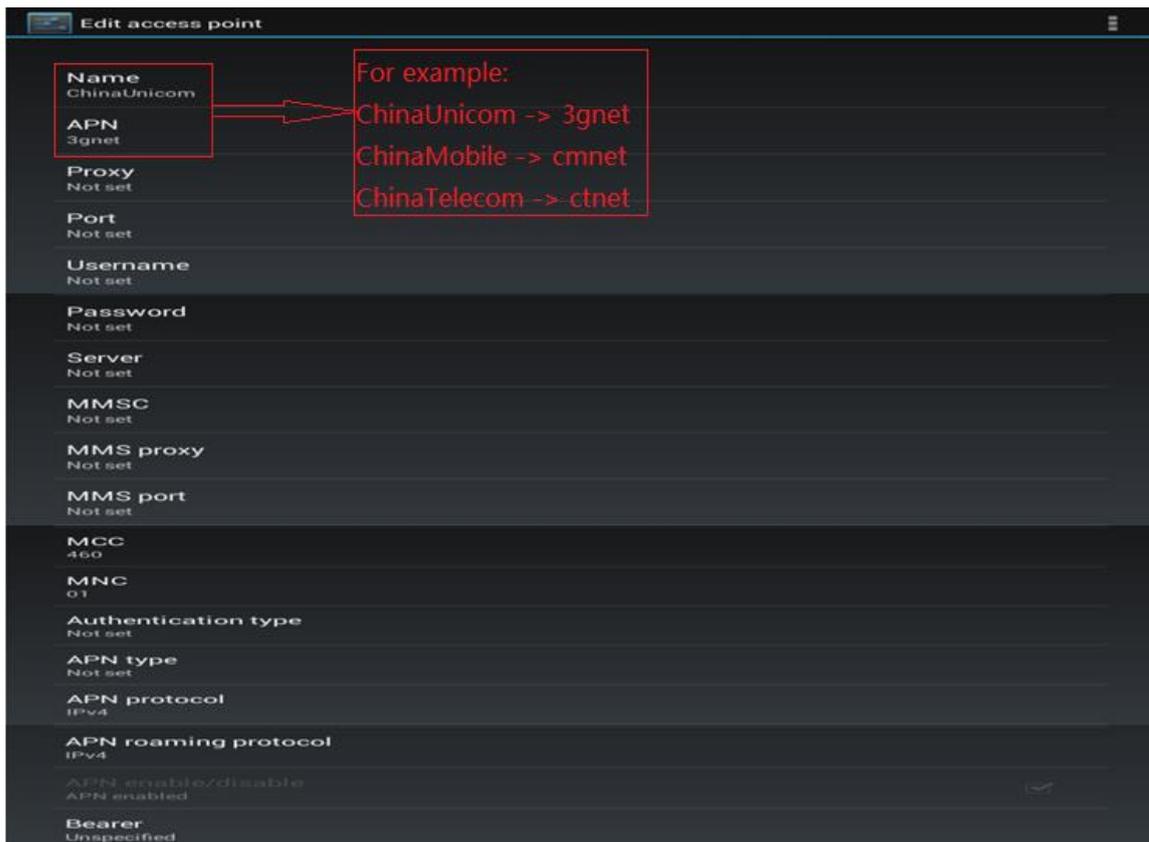


Figure 3: Edit Access Point

6.2. Why Quectel RIL Driver Does Not Work

There are many reasons that may cause the failed operation of Quectel RIL. Some common causes are listed as below for troubleshooting.

1. RIL daemon is not running.

Use **getprop init.svc.ril-daemon** command to check the status of RIL daemon. If no value is returned or values like **Stopped** or **Restarting** is returned instead of **Running**, it indicates that RIL daemon is not running.

2. RIL library is not loaded correctly.

Check the definition of service ril-daemon in *init*.rc* through **cat /init*.rc | grep ril-daemon** or in **rild.rc* through **cat /vendor/etc/init/*rild.rc | grep ril-daemon**. The expected result should be one of the following:

- service ril-daemon /system/bin/rild -l /system/*/libreference-ril.so
- service ril-daemon /system/bin/rild -l /vendor/*/libreference-ril.so
- service ril-daemon /vendor/bin/hw/rild -l /vendor/*/libreference-ril.so

Check the arguments, word spelling, blank space, etc. to make sure the RIL library is loaded correctly.

3. Failed to access USB serial port device file.

- Use **ls -l /dev/ttyUSB*** command to check the access right of the device file.
- Use **getenforce** command to check whether SELinux is enabled. If yes, use **setenforce 0** command to disable SELinux first and then check whether the access becomes normal.

4. The RIL library is not provided by Quectel.

Use **getprop gsm.version.ril-impl** command to check the Quectel RIL version, and the returned value should start with **Quectel_Android_RIL_Driver_V**. If not, it indicates the RIL library is not provided by Quectel.

6.3. Why Phone Process Does Not Work

Android system determines whether the system supports data access, phone, SMS and other features by configuring the items. The items to be configured and files to be installed are listed below. Please check whether the configuration is correct. Take *imx6q* as an example:

- If data access is required, the attribute *networkAttributes* must contain the following items:

```
<string-array translatable="false" name="networkAttributes">
  <item>"mobile,0,0,0,-1,true"</item>
  <item>"mobile_mms,2,0,4,60000,true"</item>
  <item>"mobile_supl,3,0,2,60000,true"</item>
  <item>"mobile_dun,4,0,2,60000,true"</item>
```

- If the phone feature is required, configure it as below:

```
<bool name="config_voice_capable">true</bool>
```

- If SMS is required, configure it as below:

```
<bool name="config_sms_capable">true</bool>
```

The above three configurations are in the file *config.xml*. The path can be, for example, *device/fsl/imx6dq/sabresd_6dq/overlay/frameworks/base/core/res/res/values/config.xml*.

- The following files must be installed on the Android device.

```
/vendor/bin/hw/rild
/vendor/lib/libril.so
/system/priv-app/TeleService/TeleService.apk
/system/priv-app/TelephonyProvider/TelephonyProvider.apk
/system/framework/telephony-common.jar
```

7 Appendix A References

Table 6: Related Documents

SN	Document name	Description
[1]	Quectel_WCDMA<E_Linux_USB_Driver_User_Guide	Linux USB driver user guide for UCxx/UGxx series
[2]	Quectel_LTE&5G_Linux_USB_Driver_User_Guide	Linux USB driver user guide for EC2x/EG9x/EG2x-G/EM05/Ex06/Ex12/AGxx/BGxx/RM5xxQ/RG50xQ series
[3]	Quectel_EC200T_Linux_USB_Driver_User_Guide	Linux USB driver user guide for EC200T series
[4]	Quectel_UC200T_Linux_USB_Driver_User_Guide	Linux USB driver user guide for UC200T series

Table 7: Terms and Abbreviations

Abbreviation	Description
APK	Android Package
APN	Access Point Name
bps	Bit Per Second
GSM	Global System for Mobile Communications
HIDL	Hardware Interface Definition Language
IP	Internet Protocol
IRadio	Interfaces Radio (hardware/interfaces/radio)
PDCH	Packet Data Channel
PPP	Point-to-Point Protocol
RIL	Radio Interface Layer
RILD	RIL Daemon

OS	Operating System
SELinux	Security-Enhanced Linux
SMS	Short Message Service
UART	Universal Asynchronous Receiver/Transmitter
UI	User Interface
USB	Universal Serial Bus
(U)SIM	(Universal) Subscriber Identity Module
