

# **EC2x&EG9x&EG25-G Series**

## **QuecOpen Important Data Backup and Update Solution**

**LTE Standard Module Series**

Version: 1.0.0

Date: 2020-11-05

Status: Preliminary



**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, China 200233

Tel: +86 21 5108 6236

Email: [info@quectel.com](mailto: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](mailto: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.

**COPYRIGHT**

THE INFORMATION CONTAINED HERE IS PROPRIETARY TECHNICAL INFORMATION OF QUECTEL WIRELESS SOLUTIONS CO., LTD. TRANSMITTING, REPRODUCTION, DISSEMINATION AND EDITING OF THIS DOCUMENT AS WELL AS UTILIZATION OF 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. 2020. All rights reserved.***

# About the Document

## Revision History

Version	Date	Author	Description
-	2020-10-05	Ramos ZHANG/ Charles Sun	Creation of the document
1.0.0	2020-11-05	Ramos ZHANG/ Charles Sun	Preliminary

## Contents

About the Document .....	2
Contents .....	3
1 Introduction .....	4
2 User Related Partition .....	5
3 User Application Backup and Restore Mechanism .....	6
3.1 Reference process of user application backup and restore mechanism .....	6
3.2 Important Data Backup and Update Scenarios .....	8
3.3 Test Method.....	8
4 Reference Script.....	10

# 1 Introduction

This document mainly introduces the backup and restore mechanism of user application and important parameters, and QuecOpen partition related with user.

The LTE Standard modules currently supporting the function include:

Module Series	Module
EC2x series QuecOpen	EC25 series QuecOpen
	EC21 series QuecOpen
	EC20 R2.1 QuecOpen
EG9x series QuecOpen	EG95 series QuecOpen
	EG91 series QuecOpen
EG2x-G QuecOpen	EG25-G QuecOpen

## 2 User Related Partition

User Related Partition refer to the partitions that can be adjusted for user application and configuration parameter data. Users cannot modify any other unrelated partitions

**Table 1: Partition Related with User**

Partition	Default Size	Available Size	Partition Format	Purpose
usr_data	124.25 M	88.75 M	UBI file system	Readable-writable partition, can store user applications and parameter configuration
sys_back	58 M	Disable	Image	Rootfs volume backup The size can be adjusted according to the actual situation, cannot be used
system	88.75 M	About 19 M	UBI file system	Rootfs volume, readable-writable partition, can store user applications and parameter configuration.

### NOTES

1. If user need to integrate FOTA upgrade function, usr\_data needs to reserve at least 60 M space. sys\_back is used to back up the system partition. If the system partition holds a large amount data, the sys\_back partition should be adjusted accordingly, the minimum space size for sys\_back partition is the user system ubi size + 6M, and in later use, the system partition cannot have large data increased
2. The usr\_data partition is loaded in the `/etc/init.d/find_partitions.sh` script by default, and if the partition fails to mount, it will be formatted. Therefore, there is the risk that the user applications and configuration parameters stored in the partition may be lost (the file is open, and user can also adjust it according to their own programming needs). For the specific implementation method, please refer to the code in `find_partitions.sh`.

## 3 User Application Backup and Restore Mechanism

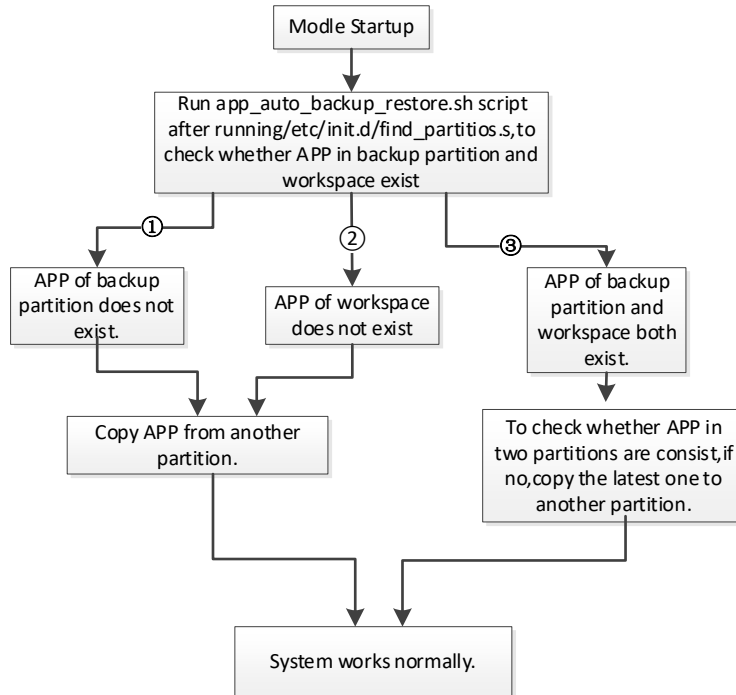
### 3.1 Reference process of user application backup and restore mechanism

User applications (or important configuration parameters) are stored a copy in the system partition and the `usr_data` partition respectively. When the module is powered on, after the script `/etc/init.d/find_partitions.sh` runs, run the automatic backup and restore detection script `app_auto_backup_restore.sh` to detect whether user applications exist in both partitions. If there is no user application in one of the partitions, copy the user application from the other partition to the partition. If user applications exist in both partitions, check the application version number. If the version numbers are different, copy the application with the higher version number to another partition and overwrite the original user application. Since the system partition and the `usr_data` partition will not be damaged at the same time, this backup and restore mechanism can ensure that the user application is not lost, and even if the user application has been upgraded, if the file system upgrade occurs, the user application can be guaranteed to be the latest version.



`app_auto_backup_restore.sh`

(Only for reference)



**Figure 1: Script Processing Flow Chart**

## NOTES

1. Due to the caching principle of Linux file operations, if the power is suddenly turned off during backup, the backup application may be incomplete, but the code is not "perceived", and the system will not run during the next restore. The solution is to copy the user application file and judge the integrity of the file first, then perform the synchronize operation, and finally copy the version tag.
2. The application version tag should be saved separately from the *APPs* directory (storing all applications, even including important parameters), when copying files, first ensure that all files in the *APPs* directory have been copied, and finally synchronize the application version tag, refer to app\_auto\_backup\_restore. If the user's important parameters and applications are stored in other directories, user can refer to the script to add two corresponding directories to achieve the same purpose.
3. The two folders backed up to each other should be placed in different partitions, because the kernel will format the entire partition if detecting the file system corruption



## 3.2 Important Data Backup and Update Scenarios

This chapter analyzes the scenarios and characteristics that may occur in actual operation. The important parameters and applications of the *app\_auto\_backup\_restore* script are stored in the */usrapp/apprun/apps* directory, the backup directory is */usrdata/appbackup/apps*. And below are four scenes

- A. The module run for the first time after downloading, the */usrdata/appbackup/apps* backup partition is empty, the */usrdata/appbackup/app\_ver.txt* tag also does not exist. *app\_auto\_backup\_restore* script will copy contents of the */usrapp/apprun/apps* folder and the tag to the backup partition.
- B. In the subsequent use of the module, if the partition of *usr\_data* is damaged, the *usr\_data* partition will be reformatted when */etc/init.d/find\_partitions.sh* is loaded, at this time, the backup data and backup tag are lost. This script will copy contents under the */usrapp/apprun/apps* folder, and version number in */usrapp/apprun/app\_ver.txt* to the *usr\_data* backup partition.
- C. After the users' application successfully upgraded, the file version number of the */usrapp/apprun/app\_ver.txt* will be incremented simultaneously. Even if the backup partition is not updated, the script will automatically copy the file under *usrapp/apprun/apps* to the */usrdata/appbackup/apps* directory after detecting the change of version number at the next startup.
- D. The users' application has been upgraded and the backup partition has been updated. However, in the later use, the system partition maybe damaged and restored. At this time, the contents of the */usrapp/apprun/apps* folder are restored to the factory state. This script will judge that the version of the backup partition is higher, then synchronize the contents of the */usrdata/appbackup/apps* folder in the backup area to the */usrapp/apprun/apps* folder and the version number will be also synchronized.

## 3.3 Test Method

1. Delete *APPs* and version tag under *usr\_data*, then reboot the module (or run the *app\_auto\_backup\_restore* script directly), check whether the application and version tag in the system partition are synchronized to the *usr\_data* partition.
2. Delete the data in the system partition, then reboot (or run the *app\_auto\_backup\_restore* script directly), and check whether the application and version tag in the *usr\_data* partition are synchronized to the system partition.
3. Add files to *APPs* in the system partition, increase the number of version tag by 1 at the same time, then reboot (or run the *app\_auto\_backup\_restore* script directly), check whether the applications and version tag in the system partition are synchronized to the *usr\_data* partition
4. Similarly, update the *APPs* in the *usr\_data* partition, then reboot (or run the *app\_auto\_backup\_restore* script directly), and check whether the application and version tag in the *usr\_data* partition are synced to the system partition
5. After normal operation, erase the system partition by fastboot method. After rebooting the system, check whether the application exists in the system partition and it is consistent with the *usr\_data* partition.

6. After normal operation, erase the usr\_data partition by fastboot method. After rebooting the system, check whether the application exists in the usr\_data partition and it is consistent with system partition.

## 4 Reference Script

```
#!/bin/sh
# Copyright (c) 2014, The Linux Foundation. All rights reserved.
#
# example app_auto_backup_restore.sh
#
# if app.bin stored in system partition (/usrapp/app.bin, /usrapp/app_Ver)
# app backup partition is usr_data in (/usrdata/appbackup/app.bin, /usrdata/appbackup/app_Ver)
#
# /usrapp/apprun/apps
# /usrapp/apprun/app_ver.txt
# /usrdata/appbackup/apps
# /usrdata/appbackup/app_ver.txt
```

```
Ver1=0
Ver2=0
app1="/usrapp/apprun/apps"
app1_Ver="/usrapp/apprun/app_ver.txt"
#just support 1,2,3,4,5,6,7 .....
app2="/usrdata/appbackup/apps"
app2_Ver="/usrdata/appbackup/app_ver.txt"
# you'd better make sure the usr_data partition is mount ok in here
if [ ! -f $app1_Ver ];then
    Ver1=-1
else
    Ver1=`cat $app1_Ver`
fi
if [ ! -f $app2_Ver ];then
    Ver2=-1
else
    Ver2=`cat $app2_Ver`
fi
```

```
if [ "$Ver1" -eq "$Ver2" ];then
    echo "app verion same , exit !!!"
    exit 0
fi
```

```
echo -n " app version update now!!!!"
if [ "$Ver1" -gt "$Ver2" ];then
    echo -n " app update to usr_data partition /usrdata/appbackup/"
    rm -rf $App2
    rm -rf $App2_Ver
    mkdir -p $App2
    cp -rf $App1 $App2
    sync
    cp -rf $App1_Ver $App2_Ver
    sync
fi
if [ "$Ver1" -lt "$Ver2" ];then
    echo -n " app update to system partition /usrapp/apprun/"
    rm -rf $App1
    rm -rf $App1_Ver
    mkdir -p $App1
    cp -r $App2 $App1
    sync
    cp -rf $App2_Ver $App1_Ver
    sync
fi
```