EC200x&EC600S&EG912Y Series

FILE Application Note

LTE Standard Module Series

Version: 1.0

Date: 2020-08-20

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.

COPYRIGHT

About the Document

Revision History

Version	Date	Author	Description
1.0	2020-08-20	Wayne WEI	Initial

Contents

Table Index

1 Introduction

Quectel EC200x series, EC600S-CN and EG912Y series modules provide AT commands to operate files on different physical storage mediums. This document is a reference guide to these commands.

This document is applicable for the following Quectel modules:

- EC200T Series
- EC200S Series
- EC600S-CN
- EG912Y Series

EC200x series, EC600S-CN and EG912Y series modules support the following storage mediums:

- **UFS**: User File Storage directory. It is a special directory on the flash file system.
- **SD**: SD card directory. (Currently supported by EC200T series only)

NOTES

1. The file name will indicate the storage location. When the file name begins with "UFS:", it means that the file is stored in UFS. When the file name begins with “SD:”, it means that the file is stored in SD card. And if there are no prefix characters in the file name, then the file is also stored in UFS.
2. Currently EC200x series, EC600S-CN and EG912Y series modules do not support the storage medium **RAM**: Random Access Memory.

1.1. The Process of Using FILE AT Commands

The following procedures can be followed to create, read and write a file in the storage:

1. Upload a file to the storage by **AT+QFUPL**, and output/download it through the serial interface by **AT+QFDWL**.
2. Open the file by **AT+QFOPEN**, and then the file can be written or read at any time and any location until the file is closed by **AT+QFCLOSE**.
   - When using **AT+QFOPEN** to open a file, you can set the file to overwrite mode, read-only mode or other modes by the parameter **<mode>** (For more information about **<mode>**, see ***Chapter 2.2.6***). After opening the file, a parameter **<filehandle>** is assigned to it. Then the file can be operated by **<filehandle>**.

- After opening the file, write the data to the file by **AT+QFWRITE** and read the data by **AT+QFREAD** from the current file position.
- Set the file position by **AT+QFSEEK** and query the current file position by **AT+QFPOSITION**.
- Close the file by **AT+QFCLOSE**, after which the **<filehandle>** turns invalid any more.

Use the following commands to manage files in the storage medium:

1. **AT+QFLDS**: get the space information of the storage medium.
2. **AT+QFLST**: list the file information in the storage medium.
3. **AT+QFDEL**: delete the file(s) in the storage medium.

1.2. Description of Data Mode

The COM port of EC200x series, EC600S-CN and EG912Y series has two working modes: AT command mode and data mode. In AT command mode, the inputted data via COM port will be treated as AT command; while in data mode, it will be treated as data.

Inputting "**+++**" or pulling up DTR (**AT&D1** should be set first) can make the COM port exit data mode. To prevent the "**+++**" from being mistaken for data, the following sequence should be followed:

1) Do not input any character within 1 s or longer before inputting "**+++**".
2) Input "**+++**" within 1 s, and no other characters can be inputted during the time.
3) Do not input any character within 1 s after "**+++**" has been inputted.

When **AT+QFUPL**, **AT+QFDWL**, **AT+QFREAD** and **AT+QFWRITE** are executed, the COM port will enter data mode. If you are using "**+++**" or DTR to make the port exit data mode, the executing procedure of these commands will be interrupted before the response is returned. In such case, the COM port cannot reenter data mode by executing **ATO**.

2 Description of FILE AT Commands

2.1. AT Command Syntax

2.1.1. Definitions

- **<CR>** Carriage return character.
- **<LF>** Line feed character.
- **<...>** Parameter name. Angle brackets do not appear on command line.
- **[...]** Optional parameter of a command or an optional part of TA information response. Square brackets do not appear on command line. When an optional parameter is omitted, the new value equals its previous value or its default setting, unless otherwise specified.
- **Underline** Default setting of a parameter.

2.1.2. AT Command Syntax

The **AT** or **at** prefix must be added at the beginning of each command line. Entering **<CR>** will terminate a command line. Commands are usually followed by a response that includes **<CR><LF><response><CR><LF>**. Throughout this document, only the response **<response>** will be presented, **<CR><LF>** are omitted intentionally.

Table 1: Type of AT Commands and Responses

Test Command	**AT+<cmd>=?**	This command returns the list of parameters and value ranges set by the corresponding Write Command or internal processes.
Read Command	**AT+<cmd>?**	This command returns the currently set value of the parameter or parameters.
Write Command	**AT+<cmd>=<p1>[,<p2>[,<p3>[...]]]**	This command sets the user-definable parameter values.
Execution Command	**AT+<cmd>**	This command reads non-variable parameters affected by internal processes in the module.

2.2. AT Command Description

2.2.1. AT+QFLDS Get the Space Information of the Storage Medium

The command gets the space information of the specified storage medium.

AT+QFLDS Get the Space Information of the Storage Medium	
Test Command **AT+QFLDS=?**	Response **OK**
Write Command **AT+QFLDS=<name_pattern>**	Response **+QFLDS: <free_size>,<total_size>** **OK** If there is an error related to ME functionality: **+CME ERROR: <err>**
Execution Command **AT+QFLDS**	Response Return the UFS information **+QFLDS: <UFS_file_size>,<UFS_file_number>** **OK** If there is an error related to ME functionality: **+CME ERROR: <err>**
Maximum Response Time	300 ms
Characteristics	The command takes effect immediately; The configurations will not be saved.

Parameter

<name_pattern>	String type. Storage medium type. "UFS" UFS "SD" SD card
<free_size>	Integer type. The free space size of **<name_pattern>**.
<total_size>	Integer type. The total space size of **<name_pattern>**.
<UFS_file_size>	Integer type. The size of all files in UFS. Unit: byte.
<UFS_file_number>	Integer type. The number of files in UFS.
<err>	Integer type. The code of an error relating to ME. See ***Chapter 4*** for details.

Example

```
AT+QFLDS="UFS"                    //Query the space information of UFS.
+QFLDS: 578847, 917503

OK
AT+QFLDS="SD"                     //Query the space information of SD card.
+QFLDS: 251920384,253132800

OK
```

2.2.2. AT+QFLST List the File Information in the Storage Medium

The command lists the information of a single file or all files in the specified storage medium.

AT+QFLST List the File Information in the Storage Medium	
Test Command **AT+QFLST=?**	Response **OK**
Write Command **AT+QFLST=<name_pattern>**	Response **+QFLST: <filename>,<file_size>** **[+QFLST: <filename>,<file_size>** **[…]]** **OK** If there is an error related to ME functionality: **+CME ERROR: <err>**
Execution Command **AT+QFLST**	Response Return the information of the UFS files: **+QFLST: <filename>,<file_size>** **[+QFLST: <filename>,<file_size>** **[…]]** **OK** If there is an error related to ME functionality: **+CME ERROR: <err>**
Maximum Response Time	300 ms
Characteristics	The command takes effect immediately; The configurations will not be saved.

Parameter

<name_pattern>	String type. The file to be listed.	
	"*"	All the files in UFS
	"**<filename>**"	The specified file **<filename>** in UFS
	"SD:*"	All the files in SD card
	"SD:**<filename>**"	The specified file **<filename>** in SD card
<filename>	String type. File name.	
<file_size>	Integer type. File size. Unit: byte.	
<err>	Integer type. The code of an error relating to ME. See ***Chapter 4*** for details.	

Example

```
AT+QFLST="*"                          //List all the files in UFS.
+QFLST: "UFS:1k.txt",1024
+QFLST: "UFS:2k.txt",2048
+QFLST: "UFS:3k.txt",3072

OK
AT+QFLST="SD:*"                       //List all the files in SD card.
+QFLST: "SD:1k.txt",1024
+QFLST: "SD:10K.txt",10240
+QFLST: "SD:100k.txt",102400

OK
```

NOTE

AT+QFLST queries the actual size of the file currently stored in flash. Use **AT+QFWRITE** to write data. If **AT+QFLST** cannot directly query the file size, you need to execute **AT+QFCLOSE** to close the file and then query the file size.

2.2.3. AT+QFDEL Delete the File(s) in the Storage Medium

The command deletes a single file or all the files in the specified storage medium.

AT+QFDEL Delete the File(s) in the Storage Medium	
Test Command **AT+QFDEL=?**	Response **+QFDEL: <filename>** **OK**
Write Command **AT+QFDEL=<filename>**	Response **OK**

	If there is an error related to ME functionality: **+CME ERROR: <err>**
Maximum Response Time	300 ms
Characteristics	The command takes effect immediately; The configurations will not be saved.

Parameter

<filename>	String type. Name of the file to be deleted. The max length is 80 bytes. "*" Delete all the files in UFS (not delete the directory) "**<filename>**" Delete the specified file **<filename>** in UFS "SD:*" Delete all the files in SD card (not delete the directory) "SD:**<filename>**" Delete the specified file **<filename>** in SD card
<err>	Integer type. The code of an error relating to ME. See ***Chapter 4*** for details.

Example

```
AT+QFDEL="*"                    //Delete all the files in UFS (not delete the directory)
OK
AT+QFDEL="UFS:1.txt"            //Delete the 1.txt file in UFS.
OK
AT+QFDEL="SD:*"                 //Delete all the files in SD card (not delete the directory).
OK
```

2.2.4. AT+QFUPL Upload a File to the Storage Medium

The command uploads a file to storage medium. If there is any file in the storage which has the same name with the file to be uploaded, an error will be reported.

After executing the Write Command and **CONNECT** returns, the module will switch to data mode. When the uploaded data reaches **<file_size>**, or there is no any data inputted when **<timeout>** reaches, then it will exit data mode automatically. During data transmission, you can use "**+++**" or DTR to make the module exit data mode, and more details are provided in ***Chapter 1.2***.

AT+QFUPL Upload a File to the Storage Medium	
Test Command **AT+QFUPL=?**	Response **+QFUPL: <filename>[,(1-<freesize>)[,(**range of supported **<timeout>**s**)[,(**list of supported **<ackmode>**s**)]]]** **OK**

Write Command **AT+QFUPL=<filename>[,<file_size>[,<timeout>[,<ackmode>]]]**	Response **CONNECT** TA switches to the data mode (transparent access mode), and the binary data of file can be inputted. When the total size of the inputted data reaches **<file_size>** (unit: byte), TA will return to command mode and reply the following codes: **+QFUPL: <upload_size>,<checksum>** **OK** If there is an error related to ME functionality: **+CME ERROR: <err>**
Maximum Response Time	5 s
Characteristics	The command takes effect immediately; The configurations will not be saved.

Parameter

<freesize>	Integer type. The free space size of **<name_pattern>**. See **AT+QFLDS** for more details of **<name_pattern>**.
<filename>	String type. Name of the file to be uploaded. The maximum length is 80 bytes. "**<filename>**" Name of the file to be uploaded to UFS "SD:**<filename>**" Name of the file to be uploaded to SD card
<file_size>	Integer type. The file size expected to be uploaded. Default: 10240. Unit: byte.
<upload_size>	Integer type. The actual size of the uploaded data. Unit: byte.
<timeout>	Integer type. The time waiting for data to be inputted to USB/UART. Default: 5. Unit: s. Range: 1–65535.
<ackmode>	Integer type. Whether to use ACK mode. 0 Turn off the ACK mode by default 1 Turn on the ACK mode
<checksum>	Integer type. The checksum of the uploaded data.
<err>	Integer type. The code of an error relating to ME. See ***Chapter 4*** for details.

NOTES

1. It is strongly recommended to use DOS 8.3 file name format for **<filename>**.
2. **<checksum>** is a 16 bit checksum based on bitwise XOR.
   If the number of the characters is odd, set the last character as the high 8 bit, and the low 8 bit as 0, and then use an XOR operator to calculate the checksum. **+++** sequence will cause TA to end the command and switch to command mode. However, the data previously uploaded will be preserved into the file.

3. When executing the command, the data must be entered after **CONNECT** is returned.
4. The ACK mode is provided to avoid the loss of data when uploading large files, in case hardware flow control does not work. The ACK mode works as follows:
   1) Run **AT+QFUPL=<filename>,<file_size>,<timeout>,1** to enable the ACK mode.
   2) The module outputs **CONNECT**.
   3) MCU sends 1 KB bytes data, and then module will respond with an **A**.
   4) MCU receives the **A** and then sends the next 1 KB bytes data.
   5) Repeat step 3) and 4) until the transfer is completed.

2.2.5. AT+QFDWL Download a File from the Storage Medium

The command downloads a specified file from the storage medium.

AT+QFDWL Download a File from the Storage Medium	
Test Command **AT+QFDWL=?**	Response **+QFDWL: <filename>** **OK**
Write Command **AT+QFDWL=<filename>**	Response **CONNECT** TA switches to data mode, and the binary data of the file will be outputted. When the file is read over, TA will return to command mode and reply the following codes: **+QFDWL: <download_size>,<checksum>** **OK** If there is an error related to ME functionality: **+CME ERROR: <err>**
Maximum Response Time	5 s
Characteristics	The command takes effect immediately; The configurations will not be saved.

Parameter

<filename>	String type. Name of the file to be downloaded. The maximum length is 80 bytes. "**<filename>**" Name of the UFS file to be downloaded "SD: **<filename>**" Name of the SD file to be downloaded
<download_size>	Integer type. The size of the downloaded data.
<checksum>	Integer type. The checksum of the downloaded data.
<err>	Integer type. The code of an error relating to ME. See ***Chapter 4*** for details.

NOTES

1. **+++** sequence will cause TA to end the command and switch to command mode.
2. **<checksum>** is a 16 bit checksum based on bitwise XOR.

2.2.6. AT+QFOPEN Open a File

The command opens a file and gets the file handle to be used in commands such as **AT+QFREAD**, **AT+QFWRITE**, **AT+QFSEEK**, **AT+QFPOSITION** and **AT+QFCLOSE**.

AT+QFOPEN Open a File	
Test Command **AT+QFOPEN=?**	Response **+QFOPEN: <filename>[,(**range of supported **<mode>**s**)]** **OK**
Read Command **AT+QFOPEN?**	Response **+QFOPEN: <filename>,<filehandle>,<mode>** **[+QFOPEN: <filename>,<filehandle>,<mode>** **[…]]** **OK**
Write Command **AT+QFOPEN=<filename>[,<mode>]**	Response **+QFOPEN: <filehandle>** **OK** If there is an error related to ME functionality: **+CME ERROR: <err>**
Maximum Response Time	300 ms
Characteristics	The command takes effect immediately; The configurations will not be saved.

Parameter

<filename>	String type. Name of the file to be opened. The maximum length is 80 bytes. "**<filename>**" Name of the UFS file to be opened "SD: **<filename>**" Name of the SD file to be opened
<filehandle>	Integer type. The handle of the file. The data type is 4 bytes.
<mode>	Integer type. The open mode of the file. 0 If the file does not exist, it will be created. If the file exists, it will be directly opened. And both of them can be read and written. 1 If the file does not exist, it will be created. If the file exists, it will be overwritten

	and cleared. And both of them can be read and written. 2 If the file exists, open it and it can be read only. When the file does not exist, it will respond an error.
<err>	Integer type. The code of an error relating to ME. See ***Chapter 4*** for details.

NOTE

If the file is stored in SD card, the obtained **<filehandle>** starts from 20 and increases by 1 by default.

2.2.7. AT+QFREAD Read a File

The command reads the data of a file which is specified by the file handle. The data starts from the current position of the file pointer which belongs to the file handle.

AT+QFREAD Read a File	
Test Command **AT+QFREAD=?**	Response **+QFREAD: <filehandle>[,<length>]** **OK**
Write Command **AT+QFREAD=<filehandle>[,<length>]**	Response **CONNECT <read_length>** TA switches to data mode. When the total size of the data reaches **<length>** (unit: byte), TA will return to command mode, display the result and then reply the following codes: **OK** If there is an error related to ME functionality: **+CME ERROR: <err>**
Maximum Response Time	5 s
Characteristics	The command takes effect immediately; The configurations will not be saved.

Parameter

<filehandle>	Integer type. The handle of the file to be operated.
<length>	Integer type. The length of the file to be read out and the default is the file length. Unit: bytes.
<read_length>	Integer type. The actual read length. Unit: bytes.
<err>	Integer type. The code of an error relating to ME. See ***Chapter 4*** for details.

2.2.8. AT+QFWRITE Write a File

The command writes data into a file. The data starts from the current position of the file pointer which belongs to the file handle.

AT+QFWRITE Write a File	
Test Command **AT+QFWRITE=?**	Response **+QFWRITE: <filehandle>[,<length>[,<timeout>]]** **OK**
Write Command **AT+QFWRITE=<filehandle>[,<length>[,<timeout>]]**	Response **CONNECT** TA switches to data mode. When the total size of the written data reaches **<length>** (unit: byte) or the time reaches **<timeout>**, TA will return to command mode and reply the following codes: **+QFWRITE: <written_length>,<total_length>** **OK** If there is an error related to ME functionality: **+CME ERROR: <err>**
Maximum Response Time	5 s
Characteristics	The command takes effect immediately; The configurations will not be saved.

Parameter

<filehandle>	Integer type. The handle of the file to be operated.
<length>	Integer type. The length of the file to be written, and the default length is 10 KB. The maximum value of this parameter is determined by **<freesize>** of **AT+QFUPL**. Unit: bytes.
<timeout>	Integer type. The time waiting for data to be inputted to USB/UART. Default: 5. Unit: s.
<written_length>	Integer type. The actual written length. Unit: bytes.
<total_length>	Integer type. The total length of the file. Unit: bytes.
<err>	Integer type. The code of an error relating to ME. See ***Chapter 4*** for details.

2.2.9. AT+QFSEEK Set a File Pointer to the Specified Position

The command sets a file pointer to the specified position. This will decide the starting position of commands such as **AT+QFREAD**, **AT+QFWRITE** and **AT+QFPOSITION**

AT+QFSEEK Set File Pointer to a Position	
Test Command **AT+QFSEEK=?**	Response **+QFSEEK: <filehandle>,<offset>[,<position>]** **OK**
Write Command **AT+QFSEEK=<filehandle>,<offset>[,<position>]**	Response **OK** If there is an error related to ME functionality: **+CME ERROR: <err>**
Maximum Response Time	300 ms
Characteristics	The command takes effect immediately; The configurations will not be saved.

Parameter

<filehandle>	Integer type. The handle of the file to be operated.
<offset>	Integer type. The number of bytes of the file pointer movement.
<position>	Integer type. Pointer movement mode. 0 The beginning of the file 1 The current position of the pointer 2 The end of the file
<err>	Integer type. The code of an error relating to ME. See ***Chapter 4*** for details.

NOTES

1. If **<position>** is 0 and **<offset>** exceeds the file size, the command will return **ERROR**.
2. If **<position>** is 1 and the total size of **<offset>** and the current position of the pointer exceed the file size, the command will return **ERROR**.
3. If **<position>** is 2, the handle will move forth.

2.2.10. AT+QFPOSITION Get the Offset of a File Pointer

The command gets the offset of a file pointer from the beginning of the file.

AT+QFPOSITION Get the Offset of a File Pointer	
Test Command **AT+QFPOSITION=?**	Response **+QFPOSITION: <filehandle>** **OK**
Write Command **AT+QFPOSITION=<filehandle>**	Response **+QFPOSITION: <offset>** **OK** If there is an error related to ME functionality: **+CME ERROR: <err>**
Maximum Response Time	300 ms
Characteristics	The command takes effect immediately; The configurations will not be saved.

Parameter

<filehandle>	Integer type. The handle of the file to be operated.
<offset>	Integer type. The offset from the beginning of the file.
<err>	Integer type. The code of an error relating to ME. See ***Chapter 4*** for details.

2.2.11. AT+QFCLOSE Close a File

The command closes a file and ends the operation to the file. After that, the file handle is released and should not be used again, unless the file is opened again by **AT+QFOPEN**.

AT+QFCLOSE Close a File	
Test Command **AT+QFCLOSE=?**	Response **+QFCLOSE: <filehandle>** **OK**
Write Command **AT+QFCLOSE=<filehandle>**	Response **OK** If there is an error related to ME functionality: **+CME ERROR: <err>**
Maximum Response Time	300 ms

Characteristics	The command takes effect immediately; The configurations will not be saved.

Parameter

<filehandle>	Integer type. The handle of the file to be operated.
<err>	Integer type. The code of an error relating to ME. See ***Chapter 4*** for details.

3 Examples

3.1. Upload and Download a File

3.1.1. Upload a File

3.1.1.1. Non ACK Mode

```
AT+QFUPL="test1.txt",10                 //Upload the text file test1.txt to UFS.
CONNECT
<Input file bin data>
+QFUPL: 10,3938

OK
```

3.1.1.2. ACK Mode

The ACK mode can make the data transmission more reliable. When transmitting a large file without hardware flow control, the ACK mode is recommended to be used to prevent the data from being lost. For more details about ACK mode, see **AT+QFUPL**.

```
AT+QFUPL="test.txt",3000,5,1            //Upload the text file test.txt to UFS.
CONNECT
<input file bin data of 1024bytes>
A                                       //After receiving 1024 bytes data, the module will
                                          respond an A. And then the next 1024 bytes data
                                          can be inputted.
<input file bin data of 1024bytes>
A
<input the rest file bin data>
+QFUPL: 3000,B34A

OK
```

3.1.2. Download a File

```
AT+QFDWL="test.txt"                 //Download the text file test.txt from UFS.
CONNECT
<Output Data>
+QFDWL: 10,613e                     //Get the bytes and the checksum value of the uploaded data.

OK
```

3.2. Write and Read a File

3.2.1. Write and Read a UFS File

```
AT+QFOPEN="test.txt",1              //Open the file to get the file handle.
+QFOPEN: 1

OK
AT+QFWRITE=1,10                     //Write 10 bytes to the file.
CONNECT
<Write Data>
+QFWRITE: 10,10                     //The actual bytes written and the size of the file are returned.

OK
AT+QFSEEK=1,0,0                     //Set the file pointer to the beginning of the file.
OK
AT+QFREAD=1,10                      //Read the data.
CONNECT
<Read Data>

OK
AT+QFCLOSE=1                        //Close the file.
OK
```

3.2.2. Write and Read a SD File

```
AT+QFOPEN="SD:1.txt",1              //Open the file to get the file handle.
+QFOPEN: 20

OK
AT+QFWRITE=20,1024                  //Write 10 bytes to the file.
CONNECT
<Write Data>
```

```
+QFWRITE: 1024,1024                 //The actual bytes written and the size of the file are returned.

OK
AT+QFSEEK=20,0,0                    //Set the file pointer to the beginning of the file.
OK
AT+QFREAD=20,1024                   //Read the data.
CONNECT
<Read Data>

OK
AT+QFCLOSE=20                       //Close the file.
OK
```

4 Summary of ERROR Codes

The error code **<err>** indicates an error related to mobile equipment or network. The details about **<err>** are described in the following table.

Table 2: Summary of Error Codes

<err>	Meaning
400	Invalid input value
401	Larger than the size of the file
402	Read zero byte
403	Drive full
405	File not found
406	Invalid file name
407	File already exists
409	Fail to write the file
410	Fail to open the file
411	Fail to read the file
413	Reach the max number of file allowed to be opened
414	The file read-only
416	Invalid file descriptor
417	Fail to list the file
418	Fail to delete the file
419	Fail to get disk info
420	No space

421	Time out
423	File too large
425	Invalid parameter
426	File already opened

5 Appendix A Reference

Table 3: Terms and Abbreviations

Abbreviation	Description
ACK	Acknowledgement
COM	Communication Port
DOS	Disk Operating System
DTR	Data Terminal Ready
ME	Mobile Equipment
RAM	Random Access Memory
SD	Secure Digital
TA	Terminal Adapter
UART	Universal Asynchronous Receiver-Transmitter
UFS	User File Storage
USB	Universal Serial Bus
XOR	Exclusive OR