WCDMA UGxx FILE

AT Commands Manual

UMTS/HSPA Module Series

Rev. WCDMA_UGxx_FILE_AT_Commands_Manual_V1.3

Date: 2016-05-05

	command mode and reply the following codes: **+QFWRITE: <written_length>,<total_length>** **OK** or **+CME ERROR: <err>**

Parameter

<file_handle>	The handle of the file to be operated
<length>	The length of the file to be written, the default length is 10K. The range of this parameter is same with the <free_size> of the "AT+QFUPL"
<timeout>	The time of waiting data to be inputted to USB/UART . Default is 5s
<written_length>	The actual written length
<total_length>	The total length of the file
<err>	The error code from the module (see the ***Appendix A***)

2.9. AT+QFSEEK Seek the File

Set the current position of the file pointer which belongs to the file handle. This will decide the starting position of the "AT+QFREAD", "AT+QFWRITE", "AT+QFPOSITION" and "AT+QFTUCAT".

AT+QFSEEK Seek the File	
Test Command **AT+QFSEEK=?**	Response **+QFSEEK: <file_handle>,<offset>[,<position>]** **OK**
Write Command **AT+QFSEEK=<file_handle>,<offset>[,<position>]**	Response **OK** or **+CME ERROR: <err>**

Parameter

<file_handle>	The handle of the file to be operated	
<offset>	The number of bytes of the file pointer movement	
<position>	Pointer movement mode. Default is 0	
	0	The beginning of the file
	1	The current position of the pointer
	2	The end of the file

<err>	The error code from the module (see the ***Appendix A***)

NOTES

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

2.10. AT+QFPOSITION Get the Offset of the File Pointer

AT+QFPOSITION gets the current position of the file pointer which is relevant to the file handle.

AT+QFPOSITION Get the Offset of the File Pointer	
Test Command **AT+QFPOSITION=?**	Response **+QFPOSITION: <file_handle>** **OK**
Write Command **AT+QFPOSITION=<file_handle>**	Response **+QFPOSITION: <offset>** **OK** or **+CME ERROR: <err>**

Parameter

<file_handle>	The handle of the operated file
<offset>	The offset from the beginning of the file
<err>	The error code from the module (see the ***Appendix A***)

2.11. AT+QFTUCAT Truncate the File from the File Pointer

AT+QFTUCAT will truncate all the data behind the position that the file pointer indicates.

AT+QFTUCAT Truncate the File from the File Pointer	
Test Command **AT+QFTUCAT=?**	Response **+QFTUCAT: <file_handle>**

	OK
Write Command **AT+QFTUCAT=<file_handle>**	Response **OK** or **+CME ERROR: <err>**

Parameter

<file_handle>	The handle of the operated file
<err>	The error code from the module (see the ***Appendix A***)

2.12. AT+QFCLOSE Close the File

AT+QFCLOSE closes the file and ends the operation to the file. The file handle is released and should not be used again, unless open the file again with "AT+QFOPEN".

AT+QFCLOSE Close the File	
Test Command **AT+QFCLOSE=?**	Response **+QFCLOSE: <file_handle>** **OK**
Write Command **AT+QFCLOSE=<file_handle>**	Response **OK** or **+CME ERROR: <err>**

Parameter

<file_handle>	The handle of the operated file
<err>	The error code from the module (see ***Appendix A***)

3 Example

3.1. Upload and Download Files

3.1.1. Upload the File

3.1.1.1. Non ACK Mode

```
AT+QFUPL="RAM:test1.txt",10          //Upload the text file “RAM:test1.txt” to RAM.
CONNECT
<input file bin data>
+QFUPL: 10,613e                      //Get the bytes of the uploaded data and the checksum.

OK
```

3.1.1.2. ACK Mode

The ACK mode can make the data transmission more reliable. When transmitting the large file without hardware flow control, the ACK mode is used to prevent the data from being lost. About the ACK mode, please refer to the details of “AT+QFUPL”.

```
AT+QFUPL="RAM:test.txt",3000,5,1     //Upload the text file “RAM:test.txt” to RAM.
CONNECT
<input file bin data of 1024bytes>
A                                    //After receiving 1024bytes data, the module will respond
                                       an “A”, then 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 the File

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

OK
```

3.2. Write and Read the File

3.2.1. Write and Read RAM File

```
AT+QFLDS="RAM"                          //Query the space information of RAM.
+QFLDS: 524288,524288

OK
AT+QFOPEN="RAM:1",0                     //Open the file in the RAM.
+QFOPEN: 3000

OK
AT+QFWRITE=3000,10                      //Write 10 bytes to the file.
CONNECT
<Write Data>
+QFWRITE: 10,10                         //The actual written bytes and the size of the file are returned.

OK
AT+QFSEEK=3000,0,0                      //Set the file pointer to the beginning of the file.
OK
AT+QFREAD=3000,10                       //Read the data.
CONNECT
<Read Data>

OK
AT+QFCLOSE=3000                         //Close the file.
OK
```

4 Appendix A Summary of <err> Code

The result of the final error code is “+CME ERROR: <err>”. <err> indicates an error relating to the ME or Network. The operation is similar to Error result code. It will be returned when some definition error happens. The <err> codes listed here are just related to UGxx module of the File.

Table 1: Summary of Error Code

<err>	Meaning
400	invalid input value
401	larger than the size of the file
402	read zero bytes
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