GE310-GNSS
AT Command Reference Guide

80598ST10945A Rev. 6 – 2020-09-17

TELIT
TECHNICAL
DOCUMENTATION

[01.2017]

SPECIFICATIONS ARE SUBJECT TO CHANGE WITHOUT NOTICE

NOTICE

While reasonable efforts have been made to assure the accuracy of this document, Telit assumes no liability resulting from any inaccuracies or omissions in this document, or from use of the information obtained herein. The information in this document has been carefully checked and is believed to be reliable. However, no responsibility is assumed for inaccuracies or omissions. Telit reserves the right to make changes to any products described herein and reserves the right to revise this document and to make changes from time to time in content hereof with no obligation to notify any person of revisions or changes. Telit does not assume any liability arising out of the application or use of any product, software, or circuit described herein; neither does it convey license under its patent rights or the rights of others.

It is possible that this publication may contain references to, or information about Telit products (machines and programs), programming, or services that are not announced in your country. Such references or information must not be construed to mean that Telit intends to announce such Telit products, programming, or services in your country.

COPYRIGHTS

This instruction manual and the Telit products described in this instruction manual may be, include or describe copyrighted Telit material, such as computer programs stored in semiconductor memories or other media. Laws in the Italy and other countries preserve for Telit and its licensors certain exclusive rights for copyrighted material, including the exclusive right to copy, reproduce in any form, distribute and make derivative works of the copyrighted material. Accordingly, any copyrighted material of Telit and its licensors contained herein or in the Telit products described in this instruction manual may not be copied, reproduced, distributed, merged or modified in any manner without the express written permission of Telit. Furthermore, the purchase of Telit products shall not be deemed to grant either directly or by implication, estoppel, or otherwise, any license under the copyrights, patents or patent applications of Telit, as arises by operation of law in the sale of a product.

COMPUTER SOFTWARE COPYRIGHTS

The Telit and 3rd Party supplied Software (SW) products described in this instruction manual may include copyrighted Telit and other 3rd Party supplied computer programs stored in semiconductor memories or other media. Laws in the Italy and other countries preserve for Telit and other 3rd Party supplied SW certain exclusive rights for copyrighted computer programs, including the exclusive right to copy or reproduce in any form the copyrighted computer program. Accordingly, any copyrighted Telit or other 3rd Party supplied SW computer programs contained in the Telit products described in this instruction manual may not be copied (reverse engineered) or reproduced in any manner without the express written permission of Telit or the 3rd Party SW supplier. Furthermore, the purchase of Telit products shall not be deemed to grant either directly or by implication, estoppel, or otherwise, any license under the copyrights, patents or patent applications of Telit or other 3rd Party supplied SW, except for the normal non-exclusive, royalty free license to use that arises by operation of law in the sale of a product.

USAGE AND DISCLOSURE RESTRICTIONS

I. License Agreements

The software described in this document is the property of Telit and its licensors. It is furnished by express license agreement only and may be used only in accordance with the terms of such an agreement.

II. Copyrighted Materials

Software and documentation are copyrighted materials. Making unauthorized copies is prohibited by law. No part of the software or documentation may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any language or computer language, in any form or by any means, without prior written permission of Telit

III. High Risk Materials

Components, units, or third-party products used in the product described herein are NOT fault-tolerant and are NOT designed, manufactured, or intended for use as on-line control equipment in the following hazardous environments requiring fail-safe controls: the operation of Nuclear Facilities, Aircraft Navigation or Aircraft Communication Systems, Air Traffic Control, Life Support, or Weapons Systems (High Risk Activities"). Telit and its supplier(s) specifically disclaim any expressed or implied warranty of fitness for such High Risk Activities.

IV. Trademarks

TELIT and the Stylized T Logo are registered in Trademark Office. All other product or service names are the property of their respective owners.

V. Third Party Rights

The software may include Third Party Right software. In this case you agree to comply with all terms and conditions imposed on you in respect of such separate software. In addition to Third Party Terms, the disclaimer of warranty and limitation of liability provisions in this License shall apply to the Third Party Right software.

TELIT HEREBY DISCLAIMS ANY AND ALL WARRANTIES EXPRESS OR IMPLIED FROM ANY THIRD PARTIES REGARDING ANY SEPARATE FILES, ANY THIRD PARTY MATERIALS INCLUDED IN THE SOFTWARE, ANY THIRD PARTY MATERIALS FROM WHICH THE SOFTWARE IS DERIVED (COLLECTIVELY “OTHER CODE”), AND THE USE OF ANY OR ALL THE OTHER CODE IN CONNECTION WITH THE SOFTWARE, INCLUDING (WITHOUT LIMITATION) ANY WARRANTIES OF SATISFACTORY QUALITY OR FITNESS FOR A PARTICULAR PURPOSE.

NO THIRD PARTY LICENSORS OF OTHER CODE SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND WHETHER MADE UNDER CONTRACT, TORT OR OTHER LEGAL THEORY, ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE OTHER CODE OR THE EXERCISE OF ANY RIGHTS GRANTED UNDER EITHER OR BOTH THIS LICENSE AND THE LEGAL TERMS APPLICABLE TO ANY SEPARATE FILES, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

APPLICABILITY TABLE

PRODUCTS

- GE310-GNSS

Contents

1. INTRODUCTION

1.1. Scope

Purpose of this document is providing a detailed specification and a comprehensive listing as a reference for the whole set of AT command for the Telit GE310-GNSS module.

1.2. Audience

Readers of this document should be familiar with Telit modules and their ease of controlling by means of AT Commands.

1.3. Contact Information, Support

For general contact, technical support services, technical questions and report documentation errors contact Telit Technical Support at:

- TS-EMEA@telit.com
- TS-AMERICAS@telit.com
- TS-APAC@telit.com
- TS-SRD@telit.com

Alternatively, use:

http://www.telit.com/support

For detailed information about where you can buy the Telit modules or for recommendations on accessories and components visit:

http://www.telit.com

Our aim is to make this guide as helpful as possible. Keep us informed of your comments and suggestions for improvements.

Telit appreciates feedback from the users of our information.

1.4. Text Conventions

Danger – This information MUST be followed or catastrophic equipment failure or bodily injury may occur.

Caution or Warning – Alerts the user to important points about integrating the module, if these points are not followed, the module and end user equipment may fail or malfunction.

Tip or Information – Provides advice and suggestions that may be useful when integrating the module.

All dates are in ISO 8601 format, i.e. YYYY-MM-DD.

1.5. Related Documents

-

This document is to describe all the AT commands implemented on the Telit wireless module Telit GE310 listed on the Applicability Table.

NOTE:

(EN) The integration of the GE310 cellular module within user application shall be done according to the design rules described in this manual.

(IT) L'integrazione del modulo cellulare GE310 all'interno dell'applicazione dell'utente dovrà rispettare le indicazioni progettuali descritte in questo manuale.

(DE) Die Integration des GE310 Mobilfunk-Moduls in ein Gerät muß gemäß der in diesem Dokument beschriebenen Kunstruktionsregeln erfolgen.

(SL) Integracija GE310 modula v uporabniški aplikaciji bo morala upoštevati projektna navodila, opisana v tem priročniku.

(SP) La utilización del modulo GE310 debe ser conforme a los usos para los cuales ha sido deseñado descritos en este manual del usuario.

(FR) L'intégration du module cellulaire GE310 dans l'application de l'utilisateur sera faite selon les règles de conception décrites dans ce manuel.

(HE) האינטגרטור מתבקש ליישם את ההנחיות המפורטות במסמך זה בתהליך האינטגרציה של המודם הסלולרי GE310 עם המוצר.

The information presented in this document is believed to be accurate and reliable. However, no responsibility is assumed by Telit Communications S.p.A. for its use, nor any infringement of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent rights of Telit Communications S.p.A. other than for circuitry embodied in Telit products. This document is subject to change without notice.

2. V.25TER AT COMMANDS

2.1. ATA

2.1.1. Description

Answers and initiates a connection to an incoming call.

2.1.2. Format

Execution command : ATA

2.1.3. Field

Type	Short name	Parameter/comment
String	text	**28800** Connected with data bit rate of 28800 bits/s (HSCSD) **19200** Connected with data bit rate of 19200 bits/s (HSCSD) **14400** Connected with data bit rate of 14400 bits/s (HSCSD) **9600** Connected with data bit rate of 9600 bits/s **4800** Connected with data bit rate of 4800 bits/s **2400** Connected with data bit rate of 2400 bits/s

2.1.4. Response

Execution command : CONNECT

CONNECT <text>

NO CARRIER

ERROR

NOTE

In UCM project , ATA command will sent to MMI for SYNC

2.2. ATD

2.2.1. Description

Initiates a phone connection, which may be data, facsimile (+FCLASS> 0), or voice (phone number terminated by semicolon). The phone number used to establish the connection will consist of digits and modifiers, or a stored number specification. ATD memory dial can originate call to phone number in entry location <n> (the memory storage of +CPBS setting will be used.). ATDL is used to dial LDN(last dialed number) and it will always dial as voice call.

2.2.2. Format

Execution command : ATD<dial string>

Memory dial command : ATD><n>

2.2.3. Field

Type	Short name	Parameter/comment
String	dial string	**.0 1 2 3 4 5 6 7 8 9 +.** Valid characters for origination **W** The W modifier is ignored but is included for compatibility reasons only **,** The comma modifier is ignored but is included for compatibility reasons only **;** Informs the Infrared Modem that the number is a voice number rather than a fax or data number **T** The T modifier is ignored but is included only for compatibility purposes **P** The P modifier is handled (pulse DTMF dialing functionality)
String	text	**28800** Connected with data bit rate of 28800 bits/s (HSCSD) **19200** Connected with data bit rate of 19200 bits/s (HSCSD) **14400** Connected with data bit rate of 14400 bits/s (HSCSD) **9600** Connected with data bit rate of 9600 bits/s **4800** Connected with data bit rate of 4800 bits/s **2400** Connected with data bit rate of 2400 bits/s

2.2.4. Response

Execution command : CONNECT

CONNECT <text>

NO CARRIER

ERROR

OK

NOTE

The ATD abortability described in V.25 5.6.1 is implemented, except for the ATD memory dial. Aborting of the command is accomplished by the transmission from the DTE to the DCE of any character before the response. In UCM project , ATD command will sent to MMI for SYNC

2.3. ATE

2.3.1. Description

The setting of this parameter determines whether or not the DCE echoes characters received from the DTE during command state and online command state.

2.3.2. Format

Execution command : ATE[<value>]

2.3.3. Field

Type	Short name	Parameter/comment
Integer	value	**0** DCE does not echo characters during command state and online command state. **1** DCE echoes characters during command state and online command state.

2.3.4. Response

Execution command : OK

2.4. ATH

2.4.1. Description

Terminates a connection.

2.4.2. Format

Execution command : ATH

2.4.3. Response

Execution command : NO CARRIER

OK

NOTE

In non-UCM projects (excluding Neptune Gemini with BT supported) projects, ATH can only hang up the call from the same source. In UCM project , ATH command will sent to MMI for SYNC

2.5. ATI

2.5.1. Description

Request Identification Information.

2.5.2. Format

Execution command : ATI[<value>]

2.5.3. Field

Type	Short name	Parameter/comment
Integer	value	used to select from among multiple types of identifying information
String	text	product information

2.5.4. Response

Execution command : <text>

2.6. ATL

2.6.1. Description

Set volume of the monitor speaker.

2.6.2. Format

Execution command : ATL[<value>]

2.6.3. Field

Type	Short name	Parameter/comment
Integer	value	**0** Low speaker volume **1** Low speaker volume **2** Medium speaker volume **3** High speaker volume

2.6.4. Response

Execution command : OK

NOTE

Do not use this command several times in the multiple command in the modem load project because it is not reasonable and might cause unexpected result due to our SW architecture design.

Ex. ATLLLLLLLLLLLLLLLLLL

2.7. ATO

2.7.1. Description

Switch from on-line command mode to on-line data mode during an active call. Returns ERROR when not in on-line command mode.

2.7.2. Format

Execution command : ATO

2.7.3. Field

Type	Short name	Parameter/comment
String	text	**28800** Connected with data bit rate of 28800 bits/s (HSCSD) **19200** Connected with data bit rate of 19200 bits/s (HSCSD) **14400** Connected with data bit rate of 14400 bits/s (HSCSD) **9600** Connected with data bit rate of 9600 bits/s **4800** Connected with data bit rate of 4800 bits/s **2400** Connected with data bit rate of 2400 bits/s

2.7.4. Response

Execution command : CONNECT

CONNECT <text>

NO CARRIER

ERROR

2.8. ATP

2.8.1. Description

Select pulse dialing. (This setting is ignored.)

2.9. ATQ

2.9.1. Description

Set result code suppression mode.

2.9.2. Format

Execution command : ATQ[<value>]

2.9.3. Field

Type	Short name	Parameter/comment
Integer	value	**0** DCE transmits result codes. **1** Result codes are suppressed and not transmitted.

2.9.4. Response

Execution command :

OK If value is 0.

(none) If value is 1 (because result codes are suppressed).

ERROR For unsupported values (if previous value was Q0).

(none) For unsupported values (if previous value was Q1).

NOTE

If use input ATQ, it is equal to ATQ1 by default

2.10. ATS0

2.10.1. Description

Automatic answer.

This S-parameter controls the automatic answering feature of the DCE. If set to 0, automatic answering is disabled. If set to a non-zero value, the DCE shall cause the DCE to answer when the incoming call indication (ring) has occurred the number of times indicated by the value.

2.10.2. Format

Execution command : ATS0=<value>

2.10.3. Field

Type	Short name	Parameter/comment
Integer	value	**0** Automatic answering is disabled..

2.10.4. Response

Execution command : OK

NOTE

In GEMINI architecture, the setting of ATS0 applies both on SIM1 and SIM2.

2.11. ATS3

2.11.1. Description

Command line termination character

This S-parameter represents the decimal IA5 value of the character recognized by the DCE from the DTE to terminate an incoming command line. It is also generated by the DCE as part of the header, trailer, and terminator for result codes and information text, along with the S4 parameter (see the description of the V parameter for usage).

2.11.2. Format

Execution command : ATS3=<value>

2.11.3. Field

Type	Short name	Parameter/comment
Integer	value	**13** Carriage return character (CR, IA5 0/13). **0 to 127** Set command line termination character to this value.

2.11.4. Response

Execution command : OK or ERROR

2.12. ATS4

2.12.1. Description

Response formatting character

This S-parameter represents the decimal IA5 value of the character generated by the DCE as part of the header, trailer, and terminator for result codes and information text, along with the S3 parameter (see the description of the V parameter for usage).

2.12.2. Format

Execution command : ATS4=<value>

2.12.3. Field

Type	Short name	Parameter/comment
Integer	value	**10** Line feed character (LF, IA5 0/10).. **0 to 127** Set response formatting character to this value.

2.12.4. Response

Execution command : OK or ERROR

2.13. ATS5

2.13.1. Description

Command line editing character.

This S-parameter represents the decimal IA5 value of the character recognized by the DCE as a request to delete from the command line the immediately preceding character.

2.13.2. Format

Execution command : ATS5=<value>

2.13.3. Field

Type	Short name	Parameter/comment
Integer	value	**8** Backspace character (BS, IA5 0/8). **0 to 127** Set command line editing character to this value.

2.13.4. Response

Execution command : OK or ERROR

2.14. ATS6

2.14.1. Description

Pause before blind dialing.

The command is ignored.

2.15. ATS7

2.15.1. Description

Connection completion timeout.

This parameter specifies the amount of time, in seconds, that the DCE shall allow between either answering a call (automatically or by the A command) or completion of signaling of call addressing information to network (dialing), and establishment of a connection with the remote DCE. If no connection is established during this time, the DCE disconnects from the line and returns a result code indicating the cause of the disconnection.

2.15.2. Format

Execution command : ATS7=<value>

2.15.3. Field

Type	Short name	Parameter/comment
Integer	value	**1 to 255** Number of seconds in which connection must be established or call will be disconnected.

2.15.4. Response

Execution command : OK or ERROR

2.16. ATS8

2.16.1. Description

Comma dial modifier time.

This parameter specifies the amount of time, in seconds, that the DCE shall pause, during signaling of call addressing information to the network (dialing), when a "," (comma) dial modifier is encountered in a dial string.

2.16.2. Format

Execution command : ATS8=<value>

2.16.3. Field

Type	Short name	Parameter/comment
Integer	value	**0** DCE does not pause when "," encountered in dial string. **1 to 255** Number of seconds to pause. Recommended default setting **2** DCE pauses two seconds when "," is encountered.

2.16.4. Response

Execution command : OK or ERROR

2.17. ATS10

2.17.1. Description

Automatic disconnect delay.

This parameter specifies the amount of time, in tenths of a second, that the DCE will remains connected to the line (off-hook) after the DCE has indicated the absence of received line signal. If the received line signal is once again detected before the time specified in S10 expires, the DCE remains connected to the line and the call continues.

2.17.2. Format

Execution command : ATS10=<value>

2.17.3. Field

Type	Short name	Parameter/comment
Integer	value	**1 to 254** Number of tenths of a second of delay.

2.17.4. Response

Execution command : OK or ERROR

2.18. ATT

2.18.1. Description

We do not support.

This setting is ignored.

2.19. ATV

2.19.1. Description

Set DCE response format.

2.19.2. Format

Execution command : ATV[<value>]

2.19.3. Field

Type	Short name	Parameter/comment
Integer	value	**0** DCE transmits limited headers and trailers and numeric text. **1** DCE transmits full headers and trailers and verbose response text.

2.19.4. Response

Execution command : OK

2.20. ATZ

2.20.1. Description

Reset to default configuration

2.20.2. Format

Execution command : ATZ[<value>]

2.20.3. Field

Type	Short name	Parameter/comment
Integer	value	**0** Set parameters to factory defaults.

2.20.4. Response

Execution command : OK or ERROR

2.21. AT&F

2.21.1. Description

Set to factory-defined configuration

2.21.2. Format

Set command : AT&F[<value>]

2.21.3. Field

Type	Short name	Parameter/comment
Integer	value	**0** Set parameters to factory defaults.

2.21.4. Response

Set command: OK | ERROR | +CME ERROR: <err>

2.22. AT&V

2.22.1. Description

The command displays the current configuration.

2.22.2. Format

Command	Possible response(s)
&V	DEFAULT PROFILE: <user default configuration> USER PROFILE: <user configuration> ACTIVE PROFILE: <current configuration> OK

2.22.1. Example

AT&V

DEFAULT PROFILE

V1 E1 Q0 &C1 &D2 &S0 &K3 X4 W2 +CBST=0,0,1 +CRLP=61,61,48,6 +CIWF=0

S00:000 S02:043 S03:013 S04:010 S05:008 S32:017 S33:019 S95:000

USER PROFILE

V1 E1 Q0 &C1 &D2 &S0 &K3 X4 W2 +CBST=0,0,1 +CRLP=61,61,48,6 +CIWF=0

S00:000 S02:043 S03:013 S04:010 S05:008 S32:017 S33:019 S95:000

ACTIVE PROFILE

V1 E1 Q0 &C1 &D2 &S0 &K3 X4 W2 +CBST=0,0,1 +CRLP=61,61,48,6 +CIWF=0

S00:000 S02:043 S03:013 S04:010 S05:008 S32:017 S33:019 S95:000

OK

2.23. AT+GMI

2.23.1. Description

Same as AT+CGMI

2.24. AT+GMM

2.24.1. Description

Same as AT+CGMM

2.25. AT+GMR

2.25.1. Description

Same as AT+CGMR

2.26. AT+IPR

2.26.1. Description

Specifies the data rate, in addition to 1200 bits/s or 9600 bits/s, at which the DCE will accept commands. May be used to select operation at rates at which the DCE is not capable of automatically detecting the data rate being used by the DTE.

2.26.2. Format

Execution command : AT+IPR=[<rate>]

Read command : AT+IPR? Displays the current <rate> setting.

Test command : AT+IPR=? Shows if the command is supported.

2.26.3. Field

Type	Short name	Parameter/comment
Integer	rate	The rate, in bits per second, at which the DTE-DCE interface should operate. Currently, the following rates are supported: 0, 300, 1200, 2400, 4800, 9600, 14400, 19200, 28800, 38400, 57600, and 115200. If unspecified, or set to zero, automatic detection is selected, and the character format is forced to auto-detect (AT+ICF=0)

2.26.4. Response

Execution command : OK

Read command : +IPR: <rate>

Test command : +IPR: (list of supported <rate>s)

2.27. AT+ICF

2.27.1. Description

Determines the local serial-port asynchronous character framing.

2.27.2. Format

Execution command : AT+ICF=[<format>[,<parity>]]

Read command : AT+ICF? Displays the current <format>, <parity> settings.

Test command : AT+ICF=? Shows if the command is supported.

2.27.3. Field

Type	Short name	Parameter/comment
Integer	parity	**0** Auto-detect **1** 8 Data bits, 2 Stop bits **2** 8 Data bits, 1 Parity bit, 1 Stop bit **3** 8 Data bits, 1 Stop bit **Default setting** **4** 7 Data bits, 2 Stop bits **5** 7 Data bits, 1 Parity bit, 1 Stop bit **6** 7 Data bits, 1 Stop bit
Integer	parity	**0** Odd **Default setting** **1** Even **2** Mark **3** Space

2.27.4. Response

Execution command : OK

Read command : +ICF: <format>,<parity>

Test command : +ICF: (list of supported <format>s), (list of supported<parity>s)

2.28. AT+DS

2.28.1. Description

Controls the V.42 bis data compression function, if provided in the TA.

2.28.2. Format

Execution command : AT+DS=[<direction>[,<compression_negotiation>[,<max_dict>[,<max-string>]]]]

Read command : AT+DS? Displays the current <direction>,<compression_negotiation>,<max_dict>, and <max_string> settings.

Test command : AT+DS=? Shows if the command is supported.

2.28.3. Field

Type	Short name	Parameter/comment
Integer	direction	**0** Disable V.42bis **1** Enable V.42bis in transmit direction only **2** Enable V.42bis in receive direction only **3** Enable V.42bis compression in both directions **Default setting**
Integer	compression_ negotiation	**0** Accept connection if compression is negotiated according to direction **Default setting** **1** Disconnect if compression is not negotiated according to direction
Integer	max_dict	**512 to 4096** Maximum dictionary size **512 is Default setting**
Integer	max_string	**6 to 250** Maximum string length **6 is Default setting**

2.28.4. Response

Execution command : OK

Read command : +DS: <direction>,<compression_negotiation>,<max-dict>,<max_string>

Test command : +DS: (list of supported <direction>s),(list of supported <compression_negotiation>s),(list of supported <max_dict>s), (list of supported <max_string>s)

2.29. AT+GCAP

2.29.1. Description

Request complete capabilities list.

2.29.2. Format

Execution command : AT+GCAP

Test command : AT+GCAP=? Shows if the command is supported.

2.29.3. Response

Execution command : +GCAP: +FCLASS, +CGSM

OK

Test command : OK

3. GENERAL COMMANDS (27.007)

3.1. AT+CGMI – Request manufacturer identification

3.1.1. Description

The command causes the phone to return one or more lines of information text <manufacturer> which is intended to permit the user of the ITAE/ETAE to identify the manufacturer of the phone to which it is connected to.

3.1.2. Format

Command	Possible response(s)
+CGMI	<manufacturer> *+CME ERROR: <err>*
+CGMI=?	

3.2. AT+CGMM – Request model identification

3.2.1. Description

The command causes the phone to return one or more lines of information text <model> which is intended to permit the user of the ITAE/ETAE to identify the specific model of phone to which it is connected to.

3.2.2. Format

Command	Possible response(s)
+CGMM	<model> *+CME ERROR: <err>*
+CGMM=?	

3.3. AT+CGMR – Request revision identification

3.3.1. Description

The command causes the phone to return a string containing information regarding SW version.

3.3.2. Format

Command	Possible response(s)
+CGMR	<revision> *+CME ERROR: <err>*
+CGMR=?	

3.4. AT+CGSN – Request product serial number identification

3.4.1. Description

Returns the IMEI number of the phone.

3.4.2. Format

Command	Possible response(s)
+CGSN	<serial number> <CR><LF> <IMEI> *+CME ERROR: <err>*
+CGSN=?	

3.5. AT+CSCS – Select TE character set

3.5.1. Description

Set command informs TA which character set <chset> is used by the TE. TA is then able to convert character strings correctly between TE and MT character sets.

3.5.2. Format

Command	Possible response(s)
+CSCS=[<chset>]	
+CSCS?	+CSCS: <chset>
+CSCS=?	+CSCS: (list of supported <chset>s)

3.5.3. Field

"GSM" GSM 7 bit default alphabet (3GPP TS 23.038); this setting causes easily software flow control (XON/XOFF) problems

"HEX" character strings consist only of hexadecimal numbers from 00 to FF; e.g. "032FE6" equals three 8-bit characters with decimal values 3, 47 and 230; no conversions to the original MT character set shall be done.

"IRA" international reference alphabet (ITU-T T.50 [13])

"PCCP437" PC character set Code Page 437

"UCS2" 16-bit universal multiple-octet coded character set (ISO/IEC10646 [32]); UCS2 character strings are converted to hexadecimal numbers from 0000 to FFFF; e.g. "004100620063" equals three 16-bit characters with decimal values 65, 98 and 99

"8859-*1*" ISO 8859 Latin character set “UCS2_08X1"

The supported parameters are subject to change according to different compile directives (options).

3.6. AT+CLAC – List all available AT commands

3.6.1. Description

Execution command causes the MT to return one or more lines of AT Commands.

NOTE

This command only returns the AT commands that are available for the user.

3.6.2. Format

Command	Possible response(s)
+CLAC	<AT Command1>[<CR><LF><AT Command2>[…]] *+CME ERROR: <err>*
+CLAC=?	*+CME ERROR: <err>*

3.6.3. Field

<AT Command>:

Defines the AT command including the prefix AT. Text shall not contain the sequence 0<CR> or OK<CR>

3.7. AT+CIMI – Request international mobile subscriber identity

3.7.1. Description

Execution command causes the TA to return <IMSI>, which is intended to permit the TE to identify the individual SIM which is attached to ME. Refer [1] 9.2 for possible <err> values.

3.7.2. Format

Command	Possible response(s)
+CIMI	<IMSI> +CME ERROR: <err>
+CIMI=?	

4. CALL CONTROL COMMANDS (27.007)

4.1. AT+CSTA – Select type of address

4.1.1. Description

Selects the type of number for further dialing commands (D) according to GSM/UMTS specifications.

4.1.2. Format

Command	Possible response(s)
+CSTA=[<type>]	
+CSTA?	+CSTA: <type>
+CSTA=?	+CSTA: (list of supported <type>s)

4.1.3. Field

<type>: type of address octet in integer format (refer 3GPP TS 24.008 subclause 10.5.4.7); default 145 when dialing string includes international access code character "+", otherwise 129

NOTE

If '+' appears at the beginning of <dial string>, the TON to network is set to 145, otherwise the setting of +CSTA will be used.

4.2. AT+CMOD – Call mode

4.2.1. Description

Selects the call mode for future dialing commands or for the next answering command.

4.2.2. Format

Command	Possible response(s)
+CMOD=[<mode>]	
+CMOD?	+CMOD: <mode>
+CMOD=?	+CMOD: (list of supported <mode>s)

4.2.3. Field

<mode>:

0 single mode

1 alternating voice/fax (teleservice 61)

2 alternating voice/data (bearer service 61)

3 voice followed by data (bearer service 81)

4.3. AT+CHUP – Hang up call

4.3.1. Description

Request to hang up the current GSM call.

4.3.2. Format

Command	Possible response(s)
+CHUP	
+CHUP=?	

4.4. AT+CBST – Select bearer service type

4.4.1. Description

Selects the bearer service <name> with the data rate <speed>, and the connection element <ce> to be used when data calls are made. Values may also be used during mobile-terminated data-call setup, especially in the case of single numbering-scheme calls.

4.4.2. Format

Command	Possible response(s)
+CBST=[<speed>[,<name>[,<ce>]]]	
+CBST?	+CBST: <speed>,<name>,<ce>
+CBST=?	+CBST: (list of supported <speed>s),(list of supported <name>s),(list of supported <ce>s)

4.4.3. Field

<speed>:

0 auto bauding (automatic selection of the speed; this setting is possible in case of 3.1 kHz modem and non-transparent service)

4 2400 bps (V.22bis)

5 2400 bps (V.26ter)

6 4800 bps (V.32)

7 9600 bps (V.32)

12 9600 bps (V.34)

14 14400 bps (V.34)

68 2400 bps (V.110 or X.31 flag stuffing)

70 4800 bps (V.110 or X.31 flag stuffing)

71 9600 bps (V.110 or X.31 flag stuffing)

75 14400 bps (V.110 or X.31 flag stuffing)

134 64000 bps(multimedia)

NOTE

when <speed> = 4,5,6,7,12,14 , line type = **Analog**

when <speed> =68,70,71,75 , line type = **ISDN**

<name>:

0	data circuit asynchronous (UDI or 3.1 kHz modem)
1	data circuit synchronous (UDI or 3.1 kHz modem)
2	PAD Access (asynchronous) (UDI)
3	Packet Access (synchronous) (UDI)
4	data circuit asynchronous (RDI)

<ce>:

0	transparent
1	non-transparent
2	both, transparent preferred
3	both, non-transparent preferred

NOTE

the valid parameter might have some differences due to the capability and the configuration of that project.

NOTE

<name> = 2 and 3 are not supported

4.5. AT+CRLP – Radio Link Protocol

4.5.1. Description

Sets the radio link protocol parameters.

4.5.2. Format

Command	Possible response(s)
+CRLP=[<iws>[,<mws>[,<T1>[,<N2>[,<ver>[,<T4>]]]]]]	
+CRLP?	+CRLP: <iws>,<mws>,<T1>,<N2>[,<ver1>[,<T4>]] [<CR><LF>+CRLP: <iws>,<mws>,<T1>,<N2>[,<ver2>[,<T4>]] [...]]
+CRLP=?	+CRLP: (list of supported <iws>s),(list of supported <mws>s), (list of supported <T1>s),(list of supported <N2>s)[,<ver1> [,(list of supported <T4>s)]] [<CR><LF>+CRLP: (list of supported <iws>s),(list of supported <mws>s),(list of supported <T1>s),(list of supported <N2>s) [,<ver1>[,(list of supported <T4>s)]] [...]]

4.5.3. Field

<ver>, <ver*x*>: RLP version number in integer format; only support version 0.

<iws>, <mws>, <T1>, <N2>, <T4>: IWF to MS window size, MS to IWF window size, acknowledgement timer T1, retransmission attempts N2, re-sequencing period T4 in integer format. T1 and T4 are in units of 10 ms.

<ver> and <T4> in set command are ignored.

4.6. AT+CR – Service reporting control

4.6.1. Description

Service reporting control.

Set command controls whether or not intermediate result code +CR: <serv> is returned from the TA to the TE. If enabled, the intermediate result code is transmitted at the point during connect negotiation at which the TA has determined which speed and quality of service will be used, before any error control or data compression reports are transmitted, and before the intermediate result code CONNECT is transmitted.

4.6.2. Format

Command	Possible response(s)
+CR=[<mode>]	
+CR?	+CR: <mode>
+CR=?	+CR: (list of supported <mode>s)

4.6.3. Field

<mode>:

0 disables reporting

1 enables reporting

<serv>:

ASYNC asynchronous transparent

SYNC synchronous transparent

REL ASYNC asynchronous non-transparent

REL SYNC synchronous non-transparent

4.7. AT+CEER – Extended error report

4.7.1. Description

Execution command causes the TA to return one or more lines of information text <report>, which offer the user of the TA an extended report of the reason for

- the failure in the last unsuccessful call setup (originating or answering) or in-call modification;

- the last call release;

4.7.2. Format

Command	Possible response(s)
+CEER	+CEER: <cause>, <report>
+CEER=?	

4.7.3. Field

<cause>: cause value listed in GSM 04.08 annex H.

<report>: string type describes cause value.

NOTE

For error cause other than those listed in GSM 04.08 annex H.

+CEER: 128 ,"ERROR_CAUSE_UNKNOWN" will be given.

If there is no error happened , +CEER: 0 ,"NONE" will be given.

4.8. AT+CRC – Cellular result code

4.8.1. Description

Set command controls whether or not the extended format of incoming call indication or GPRS network request for PDP context activation is used. When enabled, an incoming call is indicated to the TE with unsolicited result code +CRING: <type> instead of the normal RING.

4.8.2. Format

Command	Possible response(s)
+CRC=[<mode>]	
+CRC?	+CRC: <mode>
+CRC=?	+CRC: (list of supported <mode>s)

4.8.3. Field

<mode>:

0 disables extended format

1 enables extended format

<type>:

ASYNC	asynchronous transparent
SYNC	synchronous transparent
REL ASYNC	asynchronous non-transparent
REL SYNC	synchronous non-transparent
FAX	facsimile (TS 62)
VOICE	normal voice (TS 11)
VOICE/XXX	voice followed by data (BS 81) (XXX is ASYNC, SYNC, REL ASYNC or REL SYNC)
ALT VOICE/XXX	alternating voice/data, voice first (BS 61)
ALT XXX/VOICE	alternating voice/data, data first (BS 61)
ALT VOICE/FAX	alternating voice/fax, voice first (TS 61)
ALT FAX/VOICE	alternating voice/fax, fax first (TS 61).
GPRS	GPRS network request for PDP context activation

4.9. AT+CSNS – Single Numbering Scheme

4.9.1. Description

Set command selects the bearer or teleservice to be used when mobile terminated single numbering scheme call is established. Parameter values set with +CBST command shall be used when <mode> equals to a data service.

4.9.2. Format

Command	Possible response(s)
+CSNS=[<mode>]	
+CSNS?	+CSNS: <mode>
+CSNS=?	+CSNS: (list of supported <mode>s)

4.9.3. Field

<mode>:

0	voice
1	alternating voice/fax, voice first (TS 61)
2	fax (TS 62)
3	alternating voice/data, voice first (BS 61)
4	data
5	alternating voice/fax, fax first (TS 61)
6	alternating voice/data, data first (BS 61)
7	voice followed by data (BS 81)

5. NETWORK SERVICE RELATED COMMANDS (27.007)

5.1. AT+CNUM – Subscriber Number

5.1.1. Description

Returns the MSISDNs related to the subscriber (this information can be stored in the SIM/UICC or in the MT).

5.1.2. Format

Command	Possible response(s)
+CNUM	+CNUM: [<alpha1>],<number1>,<type1> [<CR><LF>+CNUM: [<alpha2>],<number2>,<type2>] [...]] *+CME ERROR: <err>*
+CNUM=?	

5.2. AT+CREG – Network Registration

5.2.1. Description

Set command controls the presentation of an unsolicited result code +CREG: <stat> when <n>=1 and there is a change in the MT network registration status, or code +CREG: <stat>[,<lac>,<ci>[,<AcT>]] when <n>=2 and there is a change of the network cell.

Read command returns the status of result code presentation and an integer <stat> which shows whether the network has currently indicated the registration of the MT. Location information elements <lac>,<ci> and <Act> are returned only when <n>=2 and MT is registered in the network.

5.2.2. Format

Command	Possible response(s)
+CREG=[<n>]	
+CREG?	+CREG: <n>,<stat>[,<lac>,<ci>[,<Act>]] *+CME ERROR: <err>*
+CREG=?	+CREG: (list of supported <n>s)

5.2.3. Field

<n>:

0 disable network registration unsolicited result code

1 enable network registration unsolicited result code +CREG: <stat>

2 enable network registration and location information unsolicited result code +CREG: <stat>[,<lac>,<ci>,[<Act>]]

<stat>:

0 not registered, MT is not currently searching a new operator to register to

1 registered, home network

2 not registered, but MT is currently searching a new operator to register to

3 registration denied

4 unknown

5 registered, roaming

<lac>: string type; two byte location area code in hexadecimal format (e.g. "00C3" equals 195 in decimal)

<ci>: string type; four byte cell ID in hexadecimal format

<Act>:

0 GSM

2 UTRAN

3 GSM w/EGPRS

4 UTRAN w/HSDPA

5 UTRAN w/HSUPA

6 UTRAN w/HSDPA and HSUPA

5.3. AT+COPS – Operator Selection

5.3.1. Description

Set command forces an attempt to select and register the GSM/UMTS network operator. If the selected operator is not available, ERROR is returned.

Read command returns the current mode, the currently selected operator.

Test command returns operator list present in the network.

5.3.2. Format

Command	Possible response(s)
+COPS=<mode>[,<format>,<oper>[,<Act>]]	+CME ERROR: <err>
+COPS?	+COPS: <mode>[,<format>,<oper>] +CME ERROR: <err>
+COPS=?	+COPS: [list of supported (<stat>,long alphanumeric <oper> ,short alphanumeric <oper>,numeric <oper>,[,<Act>])s] [,,(list of supported <mode>s),(list of supported <format>s)] +CME ERROR: <err>

5.3.3. Field

<mode>:

0	automatic (<oper> field is ignored)
1	manual (<oper> field shall be present)
2	deregister from network
3	set only <format> (for read command +COPS?), do not attempt registration /deregistration

<format>:

0	long format alphanumeric <oper>
1	short format alphanumeric <oper>
2	numeric <oper>

<oper>: string type

<stat>:

0	unknown
1	available
2	current
3	forbidden

<Act>

0	GSM
2	UTRAN

5.4. AT+CLCK – Facility Lock

5.4.1. Description

Execute command is used to lock, unlock or interrogate a ME or a network facility <fac>.

5.4.2. Format

Command	Possible response(s)
+CLCK=<fac>,<mode>[,<passwd>[,<class>]]	*+CME ERROR: <err>* **when <mode>=2 and command successful:** +CLCK: <status>[,<class1> [<CR><LF>+CLCK: <status>,<class2> [...]]
+CLCK=?	+CLCK: (list of supported <fac>s) *+CME ERROR: <err>*

5.4.3. Field

<fac> : “PF”,"SC","AO","OI","OX","AI","IR","AB","AG","AC","PN","PU","PP","PC”

<mode>:

0 unlock

1 lock

2 query status (only “SC”, “AO”, “OI”, “OX”, “AI”, “IR” support query mode)

<status>:

0 not active

1 active

<passwd>: string type

<classx> is a sum of integers each representing a class of information (default 7):

1 voice (telephony)

2 data (refers to all bearer services)

4 fax (facsimile services)

8 short message service

16 data circuit sync

32 data circuit async

64 dedicated packet access

128 dedicated PAD access

NOTE

The <fac> "AB", "AG" and "AC" are applicable only for <mode>=0

5.5. AT+CPWD – Change Password

5.5.1. Description

Action command sets a new password for the facility lock function defined by command Facility Lock +CLCK..

5.5.2. Format

Command	Possible response(s)
+CPWD=<fac>,<oldpwd>,<newpwd>	*+CME ERROR: <err>*
+CPWD=?	+CPWD: list of supported (<fac>,<pwdlength>)s *+CME ERROR: <err>*

5.5.3. Field

<fac>:

"P2" SIM PIN2

refer Facility Lock +CLCK for other values

<oldpwd>, <newpwd>: string type;

<pwdlength>: integer type maximum length of the password for the facility

5.6. AT+CLIP – Calling line identification presentation

5.6.1. Description

Requests calling line identification. Determines if the +CLIP unsolicited result code is activated. When the presentation of the CLI at the TE is enabled (and calling subscriber allows), +CLIP: <number>,<type>[,<subaddr>,<satype>] response is returned after every RING.

5.6.2. Format

Command	Possible response(s)
+CLIP=[<n>]	
+CLIP?	+CLIP: <n>,<m>
+CLIP=?	+CLIP: (list of supported <n>s)

5.6.3. Field

<n> (parameter sets/shows the result code presentation status to the TE):

0 disable

1 enable

<m> (parameter shows the subscriber CLIP service status in the network):

0 CLIP not provisioned

1 CLIP provisioned

2 unknown (e.g. no network, etc.)

<number>: string type phone number of format specified by <type>

<type>: type of address octet in integer format (refer TS 24.008 subclause 10.5.4.7)

<subaddr>: string type subaddress of format specified by <satype>

<satype>: type of subaddress octet in integer format (refer TS 24.008 subclause 10.5.4.8)

5.7. AT+CLIR – Calling line identification restriction

5.7.1. Description

Requests calling line identification restriction.

5.7.2. Format

Command	Possible response(s)
+CLIR=[<n>]	
+CLIR?	+CLIR: <n>,<m>
+CLIR=?	+CLIR: (list of supported <n>s)

5.7.3. Field

<n> (parameter sets the adjustment for outgoing calls):

0 presentation indicator is used according to the subscription of the CLIR service

1 CLIR invocation

2 CLIR suppression

<m> (parameter shows the subscriber CLIR service status in the network):

0 CLIR not provisioned

1 CLIR provisioned in permanent mode

2 unknown (e.g. no network, etc.)

3 CLIR temporary mode presentation restricted

4 CLIR temporary mode presentation allowed

5.8. AT+COLP – Connected line identification presentation

5.8.1. Description

This command refers to the GSM/UMTS supplementary service COLP (Connected Line Identification Presentation) that enables a calling subscriber to get the connected line identity (COL) of the called party after setting up a mobile originated call. The command enables or disables the presentation of the COL at the TE. It has no effect on the execution of the supplementary service COLR in the network.

When enabled (and called subscriber allows), +COLP: <number>,<type>[,<subaddr>,<satype> [,<alpha>]] intermediate result code is returned from TA to TE before any +CR or V.250 [14] responses.

5.8.2. Format

Command	Possible response(s)
+COLP=[<n>]	
+COLP?	+COLP: <n>,<m>
+COLP=?	+COLP: (list of supported <n>s)

5.8.3. Field

<n> (parameter sets/shows the result code presentation status to the TE):

0 disable

1 enable

<m> (parameter shows the subscriber COLP service status in the network):

0 COLP not provisioned

1 COLP provisioned

2 unknown (e.g. no network, etc.)

<number>, <type>, <subaddr>, <satype>, <alpha>: refer +CLIP

5.9. AT+CCUG - Closed user group

5.9.1. Description

This command allows control of the Closed User Group supplementary service.

Set command enables the served subscriber to select a CUG index, to suppress the Outgoing Access (OA), and to suppress the preferential CUG.

5.9.2. Format

Command	Possible response(s)
+CCUG=[<n>[,<index>[,<info>]]]	
+CCUG?	+CCUG: <n>,<index>,<info>
+CCUG=?	

5.9.3. Field

<n>:

0 disable CUG temporary mode

1 enable CUG temporary mode

<index>:

0...9 CUG index

10 no index (preferred CUG taken from subscriber data)

<info>:

0 no information

1 suppress OA

2 suppress preferential CUG

3 suppress OA and preferential CUG

5.10. AT+CCFC – Call forwarding number and conditions

5.10.1. Description

Sets the call forwarding number and conditions. Registration, erasure, activation, deactivation and status query operations are supported.

5.10.2. Format

Command	Possible response(s)
+CCFC=<reason>,<mode> [,<number>[,<type> [,<class> [,<subaddr>[,<satype> [,<time>]]]]]]	*+CME ERROR: <err>* **when <mode>=2 and command successful:** +CCFC: <status>,<class1>[,<number>,<type> [,<subaddr>,<satype>[,<time>]]][<CR><LF>+CCFC: <status>,<class2>[,<number>,<type> [,<subaddr>,<satype>[,<time>]]] [...]]
+CCFC=?	+CCFC: (list of supported <reason>s)

5.10.3. Field

<reason>:

0 unconditional

1 mobile busy

2 no reply

3 not reachable

4 all call forwarding (refer 3GPP TS 22.030)

5 all conditional call forwarding (refer 3GPP TS 22.030)

<mode>:

0 disable

1 enable

2 query status

3 registration

4 erasure

<number>: string type phone number of forwarding address in format specified by <type>

<type>: type of address

<subaddr>: string type subaddress of format specified by <satype>

<satype>: type of subaddress octet in integer format (refer TS 24.008 subclause 10.5.4.8); default 128

<classx> is a sum of integers each representing a class of information (default 7):

1 voice (telephony)

2 data (refers to all bearer services)

4 fax (facsimile services)

8 short message service

16 data circuit sync

32 data circuit async

64 dedicated packet access

128 dedicated PAD access

<time>:

1...30 when "no reply" is enabled or queried, this gives the time in seconds to wait before call is forwarded

<status>:

0 not active

1 active

5.11. AT+CCWA – Call waiting

5.11.1. Description

This command allows control of the Call Waiting supplementary service. Activation, deactivation and status query are supported. Parameter <n> is used to disable/enable the presentation of an unsolicited result code +CCWA: <number>,<type>,<class> to the TE when call waiting service is enabled.

5.11.2. Format

Command	Possible response(s)
+CCWA=[<n>[,<mode>[,<class>]]]	*+CME ERROR: <err>* **when <mode>=2 and command successful** +CCWA: <status>,<class1> [<CR><LF>+CCWA: <status>,<class2> [...]]
+CCWA?	+CCWA: <n>
+CCWA=?	+CCWA: (list of supported <n>s)

5.11.3. Field

<n> (sets/shows the result code presentation status to the TE):

0 disable

1 enable

<mode> (when <mode> parameter is not given, network is not interrogated):

0 disable

1 enable

2 query status

<classx> is a sum of integers each representing a class of information (default 7):

1 voice (telephony)

2 data (refers to all bearer services)

4 fax (facsimile services)

8 short message service

16 data circuit sync

32 data circuit async

64 dedicated packet access

128 dedicated PAD access

<status>:

0 not active

1 active

<number>: string type phone number of calling address in format specified by <type>

<type>: type of address octet in integer format (refer TS 24.008 subclause 10.5.4.7)

5.12. AT+CHLD – Call related supplementary services

5.12.1. Description

Requests call-related supplementary services. Refers to a service that allows a call to be temporarily disconnected from the ME but the connection to be retained by the network, and to a service that allows multiparty conversation. Calls can be put on hold, recovered, released and added to a conversation.

5.12.2. Format

Command	Possible response(s)
+CHLD=[<n>]	*+CME ERROR: <err>*
+CHLD=?	[+CHLD: (list of supported <n>s)]

5.12.3. Field

<n> (sets/shows the result code presentation status to the TE):

0 Releases all held calls, or sets User-Determined User Busy for a waiting call

1 Releases all active calls and accepts the other (waiting or held) call

1x Releases the specific active call X

2 Places all active calls on hold and accepts the other (held or waiting) call'

2x Places all active calls, except call X, on hold

3 Adds a held call to the conversation

4 Connects two calls and disconnects the subscriber from both calls

5 Activate the Completion of Calls to Busy Subscriber Request. (CCBS)

5.13. AT+CTFR – Call deflection

5.13.1. Description

This refers to a service that causes an incoming alerting call to be forwarded to a specified number.

5.13.2. Format

Command	Possible response(s)
+CTFR=<number>[,<type>[,<subaddr>[,<satype>]]]	*+CME ERROR: <err>*
+CTFR=?	

5.13.3. Field

<number>: string type phone number of format specified by <type>

<type>: type of address

<subaddr>: string type subaddress of format specified by <satype>

<satype>: type of subaddress octet in integer format (refer TS 24.008 subclause 10.5.4.8); default 128

5.14. AT+CUSD – Unstructured supplementary service data (Sec 7.15)

5.14.1. Description

Allows control of the Unstructured Supplementary Service Data (USSD). Both network- and mobile-initiated operations are supported. This command is used to enable the unsolicited result code +CUSD.

5.14.2. Format

Command	Possible response(s)
+CUSD=[<n>[,<str>[,<dcs>]]]	*+CME ERROR: <err>*
+CUSD?	+CUSD: <n>
+CUSD=?	+CUSD: (list of supported <n>s)

5.14.3. Field

<n>:

0 disable the result code presentation to the TE

1 enable the result code presentation to the TE

2 cancel session (not applicable to read command response)

<str>: string type USSD string

<dcs>: 3GPP TS 23.038 Cell Broadcast Data Coding Scheme in integer format (default 15)

<m>:

0 no further user action required

1 further user action required

2 USSD terminated by network

3 other local client has responded

4 operation not supported

5 network time out

5.15. AT+CAOC – Advice of Charge

5.15.1. Description

Sets the current call meter value in hexadecimal format. Must be supported on the SIM card. Enables/Disables the +CCCM unsolicited result code reporting. The unsolicited result code +CCCM: <ccm> is sent when the CCM value changes, but not more that every 10 seconds.

5.15.2. Format

Command	Possible response(s)
+CAOC[=<mode>]	[+CAOC: <ccm>] *+CME ERROR: <err>*
+CAOC?	+CAOC: <mode>
+CAOC=?	[+CAOC: (list of supported <mode>s)

5.15.3. Field

<mode>:

0 query CCM value

1 deactivate the unsolicited reporting of CCM value

2 activate the unsolicited reporting of CCM value

<ccm>: string type; three bytes of the current call meter value in hexadecimal format

(e.g. "00001E" indicates decimal value 30)

5.16. AT+CSSN – Supplementary service notifications

5.16.1. Description

This command refers to supplementary service related network initiated notifications. The set command enables/disables the presentation of notification result codes from TA to TE.

When <n>=1 and a supplementary service notification is received after a mobile originated call setup, intermediate result code +CSSI: <code1>[,<index>] is sent to TE before any other MO call setup result codes presented in the present document or in V.250 [14]. When several different <code1>s are received from the network, each of them shall have its own +CSSI result code.

When <m>=1 and a supplementary service notification is received during a mobile terminated call setup or during a call, or when a forward check supplementary service notification is received, unsolicited result code +CSSU: <code2>[,<index>[,<number>,<type>[,<subaddr>,<satype>]]] is sent to TE. In case of MT call setup, result code is sent after every +CLIP result code (refer command "Calling line identification presentation +CLIP") and when several different <code2>s are received from the network, each of them shall have its own +CSSU result code.

5.16.2. Format

Command	Possible response(s)
+CSSN=[<n>[,<m>]]	
+CSSN?	+CSSN: <n>,<m>
+CSSN=?	+CSSN: (list of supported <n>s),(list of supported <m>s)

5.16.3. Field

<n> (parameter sets/shows the +CSSI result code presentation status to the TE):

0 disable

1 enable

<m> (parameter sets/shows the +CSSU result code presentation status to the TE):

0 disable

1 enable

<code1> (it is manufacturer specific, which of these codes are supported):

0 unconditional call forwarding is active

1 some of the conditional call forwardings are active

2 call has been forwarded

3 call is waiting

4 this is a CUG call (also <index> present)

5 outgoing calls are barred

6 incoming calls are barred

7 CLIR suppression rejected

8 call has been deflected

<index>: refer "Closed user group +CCUG"

<code2> (it is manufacturer specific, which of these codes are supported):

0 this is a forwarded call (MT call setup)

1 this is a CUG call (also <index> present) (MT call setup)

2 call has been put on hold (during a voice call)

3 call has been retrieved (during a voice call)

4 multiparty call entered (during a voice call)

5 call on hold has been released (this is not a SS notification) (during a voice call)

6 forward check SS message received (can be received whenever)

7 call is being connected (alerting) with the remote party in alerting state in explicit call transfer operation (during a voice call)

8 call has been connected with the other remote party in explicit call transfer operation (also number and subaddress parameters may be present) (during a voice call or MT call setup)

9 this is a deflected call (MT call setup)

10 additional incoming call forwarded

<number>: string type phone number of format specified by <type>

<type>: type of address octet in integer format (refer TS 24.008 subclause 10.5.4.7)

<subaddr>: string type subaddress of format specified by <satype>

<satype>: type of subaddress octet in integer format (refer TS 24.008 subclause 10.5.4.8)

5.17. AT+CLCC – List current calls

5.17.1. Description

Returns list of current calls of ME. If command succeeds but no calls are available, no information response is sent to TE.

5.17.2. Format

Command	Possible response(s)
+CLCC	[+CLCC: <id1>,<dir>,<stat>,<mode>,<mpty>[, <number>,<type>] [<CR><LF>+CLCC: <id2>,<dir>,<stat>,<mode>,<mpty>[, <number>,<type>] [...]]] *+CME ERROR: <err>*
+CLCC=?	

5.17.3. Field

<idx>: integer type; call identification number as described in 3GPP TS 22.030 subclause 4.5.5.1;

this number can be used in +CHLD command operations

<dir>:

0 mobile originated (MO) call

1 mobile terminated (MT) call

<stat> (state of the call):

0 active

1 held

2 dialing (MO call)

3 alerting (MO call)

4 incoming (MT call)

5 waiting (MT call)

<mode> (bearer/teleservice):

0 voice

1 data

2 fax

3 voice followed by data, voice mode

4 alternating voice/data, voice mode

5 alternating voice/fax, voice mode

6 voice followed by data, data mode

7 alternating voice/data, data mode

8 alternating voice/fax, fax mode

9 unknown

<mpty>:

0 call is not one of multiparty (conference) call parties

1 call is one of multiparty (conference) call parties

<number>: string type phone number in format specified by <type>

<type>: type of address octet in integer format (refer TS 24.008 subclause 10.5.4.7)

5.18. AT+CPOL – Preferred operator list

5.18.1. Description

This command is used to edit the SIM preferred list of networks. Execute command writes an entry in the SIM list of preferred operators (EFPLMNsel). If <index> is given but <oper> is left out, entry is deleted. If <oper> is given but <index> is left out, <oper> is put in the next free location. If only <format> is given, the format of the <oper> in the read command is changed.

5.18.2. Format

Command	Possible response(s)
+CPOL=[<index>][,<format>[,<oper>[<GSM_AcT>,<GSM_compact_AcT>,<UTRAN_AcT>]]]	*+CME ERROR: <err>*
+CPOL?	+CPOL: <index1>,<format>,<oper1>[,<GSM_AcT1>,<GSM_Com pact_AcT1>,<UTRAN_AcT1>] [<CR><LF>+CPOL: <index2>,<format>,<oper2>[,<GSM_AcT2>,<GSM_Com pact_AcT2>,<UTRAN_AcT2>] [...]] *+CME ERROR: <err>*
+CPOL=?	+CPOL: (list of supported <index>s), (list of supported <format>s) *+CME ERROR: <err>*

5.18.3. Field

<indexn>: integer type; the order number of operator in the SIM/USIM preferred operator list

<format>:

0 long format alphanumeric <oper>

1 short format alphanumeric <oper>

2 numeric <oper>

<opern>: string type; <format> indicates if the format is alphanumeric or numeric (see +COPS)

<GSM_AcT*n*>: GSM access technology:

0 access technology not selected

1 access technology selected

<GSM_Compact_AcT*n*>: GSM access technology:

0 access technology not selected

1 access technology selected

<UTRAN_AcT*n*>: GSM access technology:

0 access technology not selected

1 access technology selected

5.19. AT+CPLS – Selection of preferred PLMN list

5.19.1. Description

This command is used to select one PLMN selector with Access Technology list in the SIM card or active application in the UICC (GSM or USIM), that is used by +CPOL command. Execute command selects a list in the SIM/USIM. Read command returns the selected PLMN selector list from the SIM/USIM. Test command returns the whole index range supported lists by the SIM/USIM

5.19.2. Format

Command	Possible Response(s)
+CPLS=<list>	+CME ERROR: <err>
+CPLS?	+CPLS: <list>
+CPLS=?	+CPLS: <list of supported<lis>s> +CME ERROR: <err>

5.19.3. Field

<list>: integer type

User controlled PLMN selector with Access Technology $EF_{PLMNwAcT}$, if not found in the SIM/UICC then PLMN preferred list $EF_{PLMNsel}$ (this file is only available in SIM card or GSM application selected in UICC)

Operator controlled PLMN selector with Access Technology $EF_{OPLMNwAcT}$

HPLMN selector with Access Technology $EF_{HPLMNwAcT}$

5.20. AT+COPN – Read operator name

5.20.1. Description

Execute command returns the list of operator names from the MT. Each operator code <numericn> that has an alphanumeric equivalent <alphan> in the MT memory shall be returned.

5.20.2. Format

Command	Possible Response(s)
+COPN	+COPN: <numeric1>,<alpha1> [<CR><LF>+COPN: <numeric2>,<alpha2> [...]] +CME ERROR: <err>
+COPN=?	

5.20.3. Field

<numericn>: string type; operator in numeric format (see +COPS)

<alphan>: string type; operator in long alphanumeric format (see +COPS)

5.21. AT+CAEMLPP – eMLPP priority Registration and Interrogation

5.21.1. Description

The execute command is used to change the default priority level of the user in the network. The requested priority level is checked against the eMLPP subscription of the user stored on the SIM card or in the active application in the UICC (GSM or USIM) EF_{eMLPP}. If the user doesn't have subscription for the requested priority level an ERROR or +CMEE ERROR result code is returned.

The read command triggers an interrogation of the provision of the maximum priority level which the service subscriber is allowed to use and default priority level activated by the user.

If the service is not provisioned, a result code including the SS-Status (?) parameter is returned.

5.21.2. Format

Command	Possible Response(s)
+CAEMLPP=<priority>	*+CME ERROR: <err>*
+CAEMLPP?	+CAEMLPP: <default_priority>,<max_priority> *+CME ERROR: <err>*
+CAEMLPP=?	

5.21.3. Field

<priority>: integer type parameter which identifies the default priority level to be activated in the network, values specified in 3GPP TS 22.067

<default_priority>: integer type parameter which identifies the default priority level which is activated in the network, values specified in 3GPP TS 22.067

<max_priority>: integer type parameter which identifies the maximum priority level for which the service subscriber has a subscription in the network, values specified in 3GPP TS 22.067.

5.22. AT+WS46 – Select wireless network

5.22.1. Description

Select the cellular network (Wireless Data Service; WDS) to operate with the TA. This command may be used when TA is asked to indicate the networks in which it can operate.

5.22.2. Format

Command	Possible response(s)
+WS46=[<n>]	
+WS46?	<n>
+WS46=?	(list of supported <n>s)

5.22.3. Field

<n>:

25 3GPP Systems (both GERAN and UTRAN)

6. MT CONTROL AND STATUS COMMAND (27.007)

6.1. AT+CPAS – Phone activity status

6.1.1. Description

Returns the activity status <pas> of the ME. It can be used to interrogate the ME before requesting action from the phone. If the command is executed without the <mode> parameter, only <pas> values from 0 to 128 are returned. If the <mode> parameter is included in the execution command, <pas> values from 129 to 255 may also be returned.

6.1.2. Format

Command	Possible response(s)
+CPAS	+CPAS: <pas> *+CME ERROR: <err>*
+CPAS=?	+CPAS: (list of supported <pas>s) *+CME ERROR: <err>*

6.1.3. Field

<pas>:

0 ready (MT allows commands from TA/TE)

1 unavailable (MT does not allow commands from TA/TE)

2 unknown (MT is not guaranteed to respond to instructions)

3 ringing (MT is ready for commands from TA/TE, but the ringer is active)

4 call in progress (MT is ready for commands from TA/TE, but a call is in progress)

5 asleep (MT is unable to process commands from TA/TE because it is in a low functionality state)

6.2. AT+CFUN – Set Power Status

6.2.1. Description

AT+CFUN = 0 turn off radio and SIM power.

AT+CFUN = 1, 1 or AT+CFUN=4,1 can reset the target.

AT+CFUN = 1 can enter normal mode.

AT+CFUN = 4 can enter flight mode.

6.2.2. Format

Command	Possible response(s)
+CFUN=[<fun>[,<rst>]]	*+CME ERROR: <err>*
+CFUN=?	+CFUN: (list of supported <fun>s), (list of supported <rst>s) *+CME ERROR: <err>*

6.2.3. Field

<fun> : 1 full functionality

4 disable phone both transmit and receive RF circuits (supported only for module solution)

0 minimal functionality, turn off radio and SIM power.

<rst> : 0 do not reset the MT before setting it to <fun> power level reset the MT before setting it to <fun> power level

6.3. AT+CPIN – Enter PIN

6.3.1. Description

Set command sends to the ME a password which is necessary before it can be operated (SIM PIN, SIM PUK, PH-SIM PIN, etc.). If the PIN is to be entered twice, the TA shall automatically repeat the PIN. If no PIN request is pending, no action is taken towards ME and an error message, +CME ERROR, is returned to TE.

If the PIN required is SIM PUK or SIM PUK2, the second pin is required. This second pin, <newpin>, is used to replace the old pin in the SIM.

6.3.2. Format

Command	Possible response(s)
+CPIN=<pin>[,<newpin>]	*+CME ERROR: <err>*
+CPIN?	+CPIN: <code> *+CME ERROR: <err>*
+CPIN=?	

6.3.3. Field

<pin>, <newpin>: string type values

<code> values reserved by the present document:

READY	MT is not pending for any password
SIM PIN	MT is waiting SIM PIN to be given
SIM PUK	MT is waiting SIM PUK to be given
PH-SIM PIN	MT is waiting phone to SIM card password to be given
PH-FSIM PIN	MT is waiting phone-to-very first SIM card password to be given
PH-FSIM PUK	MT is waiting phone-to-very first SIM card unblocking password to be given
SIM PIN2	MT is waiting SIM PIN2 to be given
SIM PUK2	MT is waiting SIM PUK2 to be given
PH-NET PIN	MT is waiting network personalization password to be given
PH-NET PUK	MT is waiting network personalization unblocking password to be given
PH-NETSUB PIN	MT is waiting network subset personalization password to be given
PH-NETSUB PUK	MT is waiting network subset personalization unblocking password to be given
PH-SP PIN	MT is waiting service provider personalization password to be given
PH-SP PUK	MT is waiting service provider personalization unblocking password to be given
PH-CORP PIN	MT is waiting corporate personalization password to be given
PH-CORP PUK	MT is waiting corporate personalization unblocking password to be given

6.4. AT+CBC – Battery Charge

6.4.1. Description

Execution and read command returns battery connection status <bcs> and battery level <bcl> of the ME.

6.4.2. Format

Command	Possible response(s)
+CBC	+CBC: <bcs>,<bcl> *+CME ERROR: <err>*
+CBC=?	+CBC: (list of supported <bcs>s),(list of supported <bcl>s)

6.4.3. Field

<bcs>:

0 MT is powered by the battery

1 MT has a battery connected, but is not powered by it

2 MT does not have a battery connected

3 Recognized power fault, calls inhibited

<bcl>:

0 battery is exhausted, or MT does not have a battery connected

1...100 battery has 1 100 percent of capacity remaining

6.5. AT+CSQ – Signal Quality

6.5.1. Description

The command returns received signal strength indication <rssi> and channel bit error rate <ber> from the ME.

6.5.2. Format

Command	Possible response(s)
+CSQ	+CSQ: <rssi>,<ber> *+CME ERROR: <err>*
+CSQ=?	+CSQ: (list of supported <rssi>s),(list of supported <ber>s)

6.5.3. Field

<rssi>:

0 113 dBm or less

1 111 dBm

2...30 109... 53 dBm

31 51 dBm or greater

99 not known or not detectable

<ber> (in percent):

0...7 as RXQUAL values in the table in TS 45.008 subclause 8.2.4

6.6. AT+CIND – Indicator control

6.6.1. Description

Displays the value of ME indicators.

6.6.2. Format

Command	Possible response(s)
+CIND=[<ind>[,<ind>[,...]]]	*+CME ERROR: <err>*
+CIND?	+CIND: <ind>[,<ind>[,...]] *+CME ERROR: <err>*
+CIND=?	+CIND: (<descr>,(list of supported <ind>s)) [,(<descr>,(list of supported <ind>s))[,...]] *+CME ERROR: <err>*

6.6.3. Field

<ind>: integer type value, which shall be in range of corresponding <descr>

<descr> values reserved by the present document and their <ind> ranges:

"battchg" battery charge level (0 5)

"signal" signal quality (0 5)

"service" service availability (0 1)

"message" message received (0 1)

"call" call in progress (0 1)

"roam" roaming indicator (0 1)

“call setup” call setup indicator(0 3)

“smsfull” a short message memory storage in the MT has become full(1) or memory locations are available (0)

6.7. AT+CIEV - Unsolicited Result Code

6.7.1. Description

This URC is the result code of an indicator event.

6.7.2. Format

Unsolicited result code
+CIEV: <ind>,<value1>[,<value2>,…]

6.7.3. Field

<ind>: integer type value

9: NITZ date/time/timezone information

+CIEV: 9,<UT>,<TZ>[,<DST>]

<UT> , Universal Time , String type

"YY/MM/DD,HH:MM:SS"

<TZ>: Local Time Zone, Integer type

ex: +4 or -4

<DST>: Daylight Saving Time , Integer type

1: Summer time

0: Winter time

ex: +CIEV: 9,"09/05/16,16:56:00",-28,1

6.8. AT+CMER – Mobile Termination event reporting

6.8.1. Description

Set command enables or disables sending of unsolicited result codes from TA to TE in the case of key pressings, display changes, and indicator state changes.

Test command returns the modes supported as compound values.

6.8.2. Format

Command	Possible response(s)
+CMER=[<mode>[,<keyp>[,<disp>[,<ind>[,<bfr>][,<tscrn>]]]]]	*+CME ERROR: <err>*
+CMER?	+CMER: <mode>,<keyp>,<disp>,<ind>,<bfr>
+CMER=?	+CMER: (list of supported <mode>s),(list of supported <key>s),(list of supported <disp>s), (list of supported <ind>s),(list of supported <bfr>s), (list of supported <tscrn>s)

6.8.3. Field

<mode>: integer type

0 buffer unsolicited result codes in the TA; if TA result code buffer is full, codes can be buffered in some other place or the oldest ones can be discarded

1 discard unsolicited result codes when TA-TE link is reserved (e.g. in on-line data mode); otherwise forward them directly to the TE

2 buffer unsolicited result codes in the TA when TA-TE link is reserved (e.g. in on-line data mode) and flush them to the TE after reservation; otherwise forward them directly to the TE

3 forward unsolicited result codes directly to the TE; TA-TE link specific inband technique used to embed result codes and data when TA is in on-line data mode

<keyp>: integer type

0 no keypad event reporting

<disp>: integer type

0 no display event reporting

<ind>: integer type

0 no indicator event reporting

1 indicator event reporting using result code +CIEV: <ind>,<value>. <ind> indicates the indicator order number (as specified for +CIND) and <value> is the new value of indicator. Only those indicator events, which are not caused by +CIND shall be indicated by the TA to TE

2 indicator event reporting using result code +CIEV: <ind>,<value>. All indicator events shall be directed from TA to TE

<bfr>:

0 TA buffer of unsolicited result codes defined within this command is cleared when <mode> 1...3 is entered

1 TA buffer of unsolicited result codes defined within this command is flushed to the TE when <mode> 1...3 is entered (OK response shall be given before flushing the codes)

<tscrn>:

0 no touch screen event reporting

6.9. AT+CPBS – Select Phonebook Memory Storage

6.9.1. Description

Selects the phonebook memory storage <storage> that is used by other phonebook commands.

6.9.2. Format

Command	Possible response(s)
+CPBS=<storage>	*+CME ERROR: <err>*
+CPBS?	+CPBS: <storage>[,<used>,<total>] *+CME ERROR: <err>*
+CPBS=?	+CPBS: (list of supported <storage>s)

6.9.3. Field

"ME" MT phonebook

"SM" SIM/UICC phonebook

“LD” last-dialling phonebook

“MC” MT missed calls list

“RC” MT received calls list.

“DC” MT dialled calls list

“FD” SIM/USIM fixdialling-phonebook

“ON” SIM own numbers (MSISDNs) list

6.10. AT+CPBR – Read phonebook entries

6.10.1. Description

Returns phone book entries in location number range <index1>...<index2> from the current phonebook memory storage selected by AT+CPBS. If <index2> is omitted, only location <index1> is returned. Entry fields returned are location number <indexn>, phone number <number> in <indexn>, and text <text> associated with the number.

6.10.2. Format

Command	Possible response(s)
+CPBR=<index1> [,<index2>]	[+CPBR: <index1>,<number>,<type>,<text>[,<hidden>][[...] <CR><LF>+CPBR: <index2>,<number>,<type>,<text>[,<hidden>]]] *+CME ERROR: <err>*
+CPBR=?	+CPBR: (list of supported <index>s),[<nlength>],[<tlength>] *+CME ERROR: <err>*

6.10.3. Field

<index1>, <index2>, <index>: integer type values in the range of location numbers of phonebook memory

<number>: string type phone number of format <type>

<type>: type of address octet in integer format (refer TS 24.008 subclause 10.5.4.7)

<text>: string type field of maximum length <tlength>; character set as specified by command Select TE Character Set +CSCS

<nlength>: integer type value indicating the maximum length of field <number>

<tlength>: integer type value indicating the maximum length of field <text>

<hidden>: indicates if the entry is hidden or not

0: phonebook entry not hidden

1: phonebook entry hidden

6.11. AT+CPBF – Find Phonebook entries

6.11.1. Description

Execution command returns phonebook entries (from the current phonebook memory storage selected with +CPBS) which alphanumeric field start with string <findtext>(Prefix match). Entry fields returned are location number <indexn>, phone number stored there <number> (of format <type>) and text <text> associated with the number.

6.11.2. Format

Command	Possible response(s)
+CPBF=<findtext>	[+CPBF: <index1>,<number>,<type>,<text> [[...] <CR><LF>+CBPF: <index2>,<number>,<type>,<text>]] *+CME ERROR: <err>*
+CPBF=?	+CPBF: [<nlength>],[<tlength>] *+CME ERROR: <err>*

6.11.3. Field

<index1>, <index2>: integer type values in the range of location numbers of phonebook memory

<number>: string type phone number of format <type>

<type>: type of address octet in integer format (refer TS 24.008 subclause 10.5.4.7)

<findtext>, <text>: string type field of maximum length <tlength>. Only support “IRA”

<nlength>: integer type value indicating the maximum length of field <number>

<tlength>: integer type value indicating the maximum length of field <text>

6.12. AT+CPBW – Write Phonebook entries

6.12.1. Description

Writes phonebook entry in location number <index> in the current phonebook memory storage area, selected with AT+CPBS. If the <number> and <text> parameters are omitted, the entry is deleted. If <index> is omitted but <number> is included, the entry is written to the first free location in the phonebook.

6.12.2. Format

Command	Possible response(s)
+CPBW=[<index>][,<number>[,<type>[,<text>]]]	*+CME ERROR: <err>*
+CPBW=?	+CPBW: (list of supported <index>s),[<nlength>], (list of supported <type>s),[<tlength>] *+CME ERROR: <err>*

6.12.3. Field

<index>: integer type values in the range of location numbers of phonebook memory

<number>: string type phone number of format <type>

<type>: type of address

<text>: string type field of maximum length <tlength>;

character set as specified by command Select TE Character Set +CSCS.

"UCS2", and "IRA" are supported.

<nlength>: integer type value indicating the maximum length of field <number>

<tlength>: integer type value indicating the maximum bytes of field <text> after encoding

6.13. AT+EPBUM – USIM Phonebook Manager

6.13.1. Description

This command is used to query/read/write/delete USIM Phonebook related files:

EF_ANR, EF_SNE, EF_EMAIL, EF_AAS, EF_GAS, EF_GRP

6.13.2. Format

Command	Possible response(s)
+EPBUM=<op>,<type>,<INDEX1>[,<INDEX2>[,<number/email/text/grp_list>]]	<op> = 0 (QUERY) <type>=0 (EF_ANR) +EPBUM: <type>, <INDEX1>, <M_NUM>, <A_NUM>, <L_ANR> <type>=1 (EF_EMAIL) +EPBUM: <type>, <INDEX1>, <M_NUM>, <A_NUM>, <L_EMAIL> <type>=2 (EF_SNE) +EPBUM: <type>, <INDEX1>, <M_NUM>, <A_NUM>, <L_SNE> <op> = 1 (READ) +EPBUM:<type>,<INDEX1>,<INDEX2>, <number/email/text/grp_list> +CME ERROR: <err>
+EPBUM=?	+EPBUM: <N_ANR>,<N_EMAIL>,<N_SNE>,<N_AAS>,<L_AAS>,<N_GAS>,<L_GAS>,<N_GRP> +CME ERROR: <err>

6.13.3. Field

<op>:

0: query EF files information. In this <op>, the valid types are EF_ANR, EF_SNE, and EF_EMAIL

1: read EF files

2: write EF files

3: delete EF files

<type>: the type of USIM phonebook related EF files

0: EF_ANR

1: EF_EMAIL

2: EF_SNE

3: EF_AAS

4: EF_GAS

5: EF_GRP

<M_NUM>: max number of entries in the queried EF files

<A_NUM>: max number of available entries in the queried EF files

<L_ANR>: max supported number length of an entry in the queried EF_ANR file

<L_EMAIL>: max supported email length of an entry in the queried EF_EMAIL file

<L_SNE>: max supported second name length of an entry in the queried EF_SNE file

<INDEX1>, <INDEX2>: has different meaning according to the <op> and <type>

<op>=0 (Query) : only <INDEX1> is needed

<INDEX1> : Assume <index1> is N, N-th EF file associated with an EF_ADN

<op>=1 or 2 or 3: (Read/Write/Delete)

<type>=0 or 1 or 2 **(EF_ANR/EF_EMAIL/EF_SNE)**

<INDEX1>: the index of ADN entry to be accessed

<INDEX2>: Assume < INDEX2> is N, N-th EF entry associated with the ADN entry

<type>=3 or 4 **(EF_AAS or EF_GAS)**

<INDEX1>: the index of EF entry to be accessed

<INDEX2>: ignore

<type>=5 **(EF_GRP)**

<INDEX1>: the index of EF GRP associated with the ADN entry to be accessed

<INDEX2>: ignore

<number/email/text/grp_list>: the format is different according to the <type>

<type>=0 **(EF_ANR)**

<number>, <ton>, <ass_id>

<number>: the telephony number

<ton>: the type of <number>, valid value: 129(normal) or 145(international)

<aas_id>: the associated EF_AAS entry index

<type>=1 **(EF_EMAIL)**

<email> : the email, must be IRA encode

<type>=2 or 3 or 4**(EF_SNE/EF_AAS/EF_GAS)**

<text>, <encode>

<text>: the alpha string, the encoding is according to the <encode>

<encode>:

0: IRA

1: UCS2 0x80

2: UCS2 0x81

<type>=5 **(EF_GRP)**

<GRP1>, <GRP2>, …,<GRP_n>

The valid value of each <GRPx> is 0 ~ 255, n is <N_GRP>

<N_ANR>: maximum number of entries associated with an EF_ADN

<N_EMAIL>: maximum number of entries associated with an EF_EMAIL

<N_SNE>: maximum number of entries associated with an EF_SNE

<N_AAS>: maximum number of entries in the EF_AAS

<L_AAS>: maximum alpha string length of an EF_AAS entry

<N_GAS>: maximum number of entries in the EF_GAS

<L_GAS>: maximum alpha string length of an EF_GAS entry

<N_GRP>: maximum number of groups in an entry of EF_GRP

6.13.4. Example

Assume we insert a SIM card with the following configuration

500 phonebook entries. each one has maximum 3 ANRs, and 1 email with the maximum length 38, 1 SNE with the maximum length 12, and 4 GRP IDs

1 EF_AAS with 5 entries, the maximum length of an AAS entry is 12

1 EF_GAS with 10 entries, the maximum length of a GAS entry is 12

Test Mode

AT+EPBUM=?

+EPBUM: 3,1,1,5,12,10,12,4

OK

Access EF_ANR

QUERY

// query the 2nd EF_ANR where the available number of entries is 95

AT+EPBUM=0,0,2

+EPBUM: 0,2,100,95,40

OK

READ

// read the first ANR of the 500th ADN entry with number(0123456789) and AAS_id(10)

AT+EPBUM=1,0,500,1

+EPBUM: 0,500,1,"0123456789",129,10

OK

// read the 2nd ANR of the 500th ADN entry which is an empty entry

AT+EPBUM=1,0,500,2

OK

WRITE

// Write the 2nd ANR of the 123th ADN entry with number(+0123456789) and AAS_id(10)

AT+EPBUM=2,0,123,2,”0123456789”,145,10

OK

AT+EPBUM=1,0,123,2

+EPBUM: 0,123,2,"0123456789",145,10

OK

DELETE

// Delete the 3nd ANR of the 200th ADN entry

AT+EPBUM=3,0,200,3

OK

// Delete the empty one again

AT+EPBUM=3,0,200,3

OK

Access EF_EMAIL

QUERY

// query the first EF_EMAIL where the available number of entries is 450

AT+EPBUM=0,1,1

+EPBUM: 1,1,500,450,38

OK

READ

// read the first email of the 500th ADN entry with the email “abc@mediatek.com”

AT+EPBUM=1,1,500,1

+EPBUM: 1,500,1,"abc@mediatek.com”

OK

// read the first email of the 300th ADN entry which is an empty entry

AT+EPBUM=1,1,300,1

OK

WRITE

// Write the first ANR of the 123th ADN entry with the email “abcdefghijkl@mediatek.com”

AT+EPBUM=2,1,123,1,”abcdefghijkl@mediatek.com”

OK

AT+EPBUM=1,1,123,1

+EPBUM: 1,123,1,”abcdefghijkl@mediatek.com”

OK

DELETE

// Delete the first email of the 200th ADN entry

AT+EPBUM=3,1,200,1

OK

// Delete the empty one again

AT+EPBUM=3,1,200,1
OK

Access EF_SNE

QUERY

// query the first EF_SNE where the available number of entries is 333
AT+EPBUM=0,2,1
+EPBUM: 2,1,500,333,12
OK

READ

// read the first SNE of the 500th ADN entry with the alpha string “abc”
AT+EPBUM=1,2,500,1
+EPBUM: 2,500,1,"abc”,0
OK
// read the first SNE of the 300th ADN entry which is an empty entry
AT+EPBUM=1,2,300,1
OK

WRITE

// Write the first SNE of the 123th ADN entry with the name “abcdefghijkl”
AT+EPBUM=2,2,123,1,”abcdefghijkl”,0
OK
AT+EPBUM=1,2,123,1
+EPBUM: 2,123,1,”abcdefghijkl”,0
OK

DELETE

// Delete the first SNE of the 200th ADN entry
AT+EPBUM=3,2,200,1
OK
// Delete the empty one again
AT+EPBUM=3,2,200,1
OK

Access EF_AAS

READ

// read the 3rd AAS entry with the alpha string “聯@發”
AT+EPBUM=1,3,3,100
+EPBUM: 3,3,100,"806F0040767C", 1
OK

// read the first AAS entry which is an empty entry

AT+EPBUM=1,3,3,1

OK

WRITE

// Write the first AAS entry with the alpha string “聯@發”

AT+EPBUM=2,3,1,5,”806F0040767C”,1

OK

AT+EPBUM=1,3,1,10

+EPBUM: 3,1,10,”806F0040767C”,1

OK

DELETE

// Delete the 2nd AAS entry

AT+EPBUM=3,3,2,1

OK

// Delete the empty one again

AT+EPBUM=3,3,2,5

OK

Access EF_GAS

READ

// read the 3rd GAS entry with the alpha string “ÇØßæçø”

AT+EPBUM=1,4,3,100

+EPBUM: 4,3,100,"00C700D800DF00E600E700F8", 1

OK

// read the first GAS entry which is an empty entry

AT+EPBUM=1,4,3,1

OK

WRITE

// Write the first GAS entry with the alpha string “ÇØßæçø”

AT+EPBUM=2,4,1,5,”810601090B1E1DE70C”,2

OK

AT+EPBUM=1,4,1,10

+EPBUM: 4,1,10,” 00C700D800DF00E600E700F8”,1

OK

DELETE

// Delete the 2nd GAS entry

AT+EPBUM=3,4,2,1

OK

// Delete the empty one again

AT+EPBUM=3,4,2,5

OK

Access EF_GRP

READ

// read the GRP of the 500th ADN entry with group ID, 1,2,3,4

AT+EPBUM=1,5,500,0

+EPBUM: 5,500,0,1,2,3,4

OK

// read the GRP of the 300th ADN entry which is an empty entry

AT+EPBUM=1,5,300,10

+EPBUM: 5,300,10,255,255,255,255

OK

WRITE

// Write the GRP of the 123th ADN entry with group ID, 10,11,12,13,14

AT+EPBUM=2,5,123,1,10,11,12,13,14

ERROR

AT+EPBUM=2,5,123,1,10,11,12,13

OK

AT+EPBUM=1,5,123,1

+EPBUM: 5,123,1,10,11,12,13

OK

DELETE

// Delete the GRP of the 200th ADN entry

AT+EPBUM=3,5,200,1

OK

// Delete the empty one again

AT+EPBUM=3,5,200,1

OK

6.14. AT+EIND – unsolicited result code- SMS, Phonebook or AT interface

6.14.1. Description

The command sets the +EIND unsolicited result code, to indicate the readiness of SMS, Phonebook or AT interface.

6.14.2. Format

Command	Possible response(s)
+EIND=<flag>	OK ERROR
+EIND?	+EIND: <ind>
+EIND =?	+EIND=(0-4294967295)

6.14.3. Field

<flag>:

Bit 0: Any value (0-4294967295) that bit 0 is 1. E.g. 1, 3, 5, …

Bit 1: Any value (0-4294967295) that bit 1 is 1. E.g. 2, 3, 6, …

Bit 2: Any value (0-4294967295) that bit 2 is 1. E.g. 4, 5, …

Bit 3: Any value (0-4294967295) that bit 3 is 1. E.g. 8, 9, …

Bit 7: Any value (0-4294967295) that bit 7 is 1. E.g. 128, 129, 130, …

<ind>:

1 SMS_READY

2 PHB_READY

3 file change for PLMN files

8 file change for EONS files

16 Invalid SIM

128 AT_READY

Note: the +EIND indications are enabled by default it can be disable by command +EIND=0 it is not saved in NVM

Note: “+EUSIM: 0” indicates a SIM card type, “+EUSIM: 1” indicates a USIM card type

6.15. AT+CCLK – Clock

6.15.1. Description

Set command sets the real-time clock of the MT.

Read command returns the current setting of the clock

6.15.2. Format

Command	Possible response(s)
+CCLK=<time>	*+CME ERROR: <err>*

+CCLK?	+CCLK: <time> *+CME ERROR: <err>*
+CCLK=?	

6.15.3. Field

<time>: string type value; format is "yy/MM/dd,hh:mm:ss",

where characters indicate year (two last digits), month, day, hour, minutes, seconds.

6.16. AT+CALA – Alarm

6.16.1. Description

Sets an alarm time in the ME.

6.16.2. Format

Command	Possible response(s)
+CALA=<time>[,<n>[,<type>[,<text>[,<recurr>]]]]	*+CME ERROR: <err>*
+CALA?	[+CALA: <time>,<n1>,,,<recurr> [<CR><LF>+CALA: <time>,<n2>,,,<recurr> [...]]] *+CME ERROR: <err>*
+CALA=?	OK

6.16.3. Field

<time>: refer +CCLK

<n>: integer type value indicating the index of the alarm.

<type>: integer type. But we don't care about type value.

<text>: string type. But we don't care about text content. MMI doesn't support.

<recurr>: string type value indicating day of weeks for the alarm in one of the following format:

"<1..7>[,<1..7>[…]]" – Sets a recurrent alarm for one or more days in the week. The digits 1 to 7 corresponds to the days in the week, Monday (1), …, Sunday (7).
Example: The string "1,2,3,4,5" may be used to set an alarm for all weekdays.

"0" – Sets a recurrent alarm for all days in the week.

6.17. AT+CSIM – Generic SIM Access

6.17.1. Description

Set command transmits to the MT the <command> it then shall send as it is to the SIM. In the same manner the SIM <response> shall be sent back by the MT to the TA as it is. Refer subclause 9.2 for <err> values.

This command allows a direct control of the SIM by a distant application on the TE. The TE shall then take care of processing SIM information within the frame specified by GSM/UMTS.

6.17.2. Format

Command	Possible response(s)
+CSIM=<length>,<command>	+CSIM: <length>,<response> +CME ERROR: <err>
+CSIM=?	

6.17.3. Field

<length> : integer type; length of the characters that are sent to TE in <command> or <response> (two times the actual length of the command or response)

<command> : command passed on by the MT to the SIM in the format as described in 3GPP TS 51.011 [28] (hexadecimal character format; refer +CSCS)

<response> : response to the command passed on by the SIM to the MT in the format as described in 3GPP TS 51.011 (hexadecimal character format; refer +CSCS)

6.17.4. Example

1 SELECT

(1) (P1 = SELECT MF by file id)

AT+CSIM=14,"00A4000C023F00"

+CSIM: 4, "9000"

OK

2 SELECT

(1) (P1 = SELECT by DF name)

AT+CSIM=42,"00A4040C10A0000000871002FF47700189000001FF"

ERROR

3 READ BINARY

(1) (Pre-condition: SELECT EF_IMSI (P1 = SELECT by path from MF, P2 = return with FCP))

AT+CSIM=20,"00A40804047FFF6F0700"

+CSIM: 64, "621C8202412183026F07A5038001718A01058B036F0605800200098801389000"

OK

(2) READ BINARY

AT+CSIM=10,"00B0000009"

+CSIM: 22, "0849667914305241049000"

OK

4 UPDATE BINARY

(1) (Pre-condition: SELECT EF_PLMNwAcT(P1 = SELECT by path from MF, P2 = return with FCP))

AT+CSIM=20,"00A40804047FFF6F6000"

+CSIM: 64, "621C8202412183026F60A5038001718A01058B036F0606800200878801509000"

OK

(2) READ BINARY

AT+CSIM=10,"00B0000087"

+CSIM: 18,
"888888008854F400808025F510808005F221808015F001808005F520808015F52080

8004F401808004F454808004F429808004F430808004F494808004F404808054F050808025F01080

8054F5108080FFFFFF0000FFFFFF0000FFFFFF0000FFFFFF0000FFFFFF0000FFFFFF0000FFFFFF00

00FFFFFF0000FFFFFF0000FFFFFF0000FFFFFF00009000"

OK

(3) UPDATE BINARY

AT+CSIM=20,"00D600000521F3548080"

+CSIM: 4, "9000"

OK

6.18. AT+CRSM -- Restricted SIM access

6.18.1. Description

Set command transmits to the MT the SIM <command> and its required parameters.

6.18.2. Format

Command	Possible response(s)
+CRSM=<command>[,<fileid> [,<P1>,<P2>,<P3> [,<data>[,<pathid>]]]]	+CRSM: <sw1>,<sw2>[,<response>] *+CME ERROR: <err>*
+CRSM=?	

6.18.3. Field

<command> (command passed on by the MT to the SIM; refer 3GPP TS11.11):

176 READ BINARY

178 READ RECORD

192 GET RESPONSE

214 UPDATE BINARY

220 UPDATE RECORD

242 STATUS

<fileid>: integer type; this is the identifier of a elementary data file on SIM.

<P1>, <P2>, <P3>: integer type; parameters passed on by the MT to the SIM.
(For detailed information , please refer 3GPP TS11.11 Section 9.2)

<data>: information which shall be written to the SIM (hexadecimal character format; refer +CSCS)

<pathid>: string type; contains the path of an elementary file on the SIM/UICC in hexadecimal format as defined in ETSI TS 102 221 (e.g. "7F205F70" in SIM and UICC case). The <pathid> shall only be used in the mode "select by path from MF" as defined in ETSI TS 102 221.

<sw1>, <sw2>: integer type; information from the SIM about the execution of the actual command.

<response>: response of a successful completion of the command previously issued (hexadecimal character format)

NOTE

READ BINARY command is used for transparent EF. READ RECORD is used for linear fixed or cyclic EF

NOTE

Before using READ BINARY, READ RECORD, UPDATE BINARY, UPDATE RECORD, please use command GET RESPONSE to get the exact length information first.

NOTE

- <pathid> + <fileid> can be a unique identifier on the SIM/UICC.
- In USIM, the response of STATUS and GET RESPONSE is TLV format, and length is not fixed. So the P3 should be assigned as “00” as 256 bytes, which is the maximum value of response data.

6.18.4. Example

1. Read EF_{SST} (file_idx= 0x6F38 , structure: transparent)

(1) Get RESPONSE first , 3~4 byte is the file size information.(e.g. 000A=10)

at+crsm=192,28472

+CRSM: 144, 0, "0000000A6F38040015005501010000"

OK

at+crsm=176,28472,0,0,10

+CRSM: 144, 0, "FF3FFFFF00003C03000C"

OK

2. Read a EF_{ADN} (file_idx= 0x6F3A , structure: Linear fixed)

(1)GET RESPONSE first , No.15 byte represents the record length (e.g. 1E =30)

at+crsm=192,28474

+CRSM: 144, 0, "00001D4C6F3A04001100220502011E"

OK

(2) READ RECORD

at+crsm=178,28474,1,4,30

+CRSM: 144, 0, "6F776E6572FFFFFFFFFFFFFFFFFFFFFF06819078303326FFFFFFFFFFFFFF"

OK

3. READ $EF_{ImageInstaceDataFiles}$ (with <pathid>) (fIle_idx = 0x4F20(File id would be different if you use other SIM cards), structure: Transparent)

(1) GET RESPONSE first (without AT command example)

(2) READ BINARY

AT+CRSM=176,20256,0,0,1,,"7F105F50"

+CRSM: 144, 0, "00"

OK

6.19. AT+CRSL – Ringer Sound Level

6.19.1. Description

Set the incoming call ringer sound level.

6.19.2. Format

Command	Possible response(s)
+CRSL=<level>	*+CME ERROR: <err>*
+CRSL?	+CRSL: <level> *+CME ERROR: <err>*
+CRSL=?	+CRSL: (list of supported <level>s) *+CME ERROR: <err>*

6.19.3. Field

<level>: integer type value with manufacturer specific range

6.20. AT+CLVL – Loudspeaker volume level

6.20.1. Description

Sets the volume of the internal speaker in the ME

6.20.2. Format

Command	Possible response(s)
+CLVL=<level>	*+CME ERROR: <err>*
+CLVL?	+CLVL: <level> *+CME ERROR: <err>*
+CLVL=?	+CLVL: (list of supported <level>s) *+CME ERROR: <err>*

6.20.3. Field

<level>: integer type value with manufacturer specific range.

6.21. AT+CMUT – Mute Control

6.21.1. Description

Enable/Disable the uplink voice muting during a voice call.

6.21.2. Format

Command	Possible response(s)
+CMUT=<n>	*+CME ERROR: <err>*
+CMUT?	+CMUT: <n> *+CME ERROR: <err>*
+CMUT=?	+CMUT: (list of supported <n>s)

6.21.3. Field

<n>:

0 mute off

1 mute on

6.22. AT+CACM – Accumulated call meter

6.22.1. Description

Resets the Advice-of-Charge related accumulated call meter value in the SIM file EFACM.

6.22.2. Format

Command	Possible response(s)
+CACM=[<passwd>]	*+CME ERROR: <err>*
+CACM?	+CACM: <acm> *+CME ERROR: <err>*
+CACM=?	

6.22.3. Field

<passwd>: string type; SIM PIN2

<acm>: string type; accumulated call meter value similarly coded as <ccm> under +CAOC

6.23. AT+CAMM – Accumulated call meter maximum

6.23.1. Description

Sets the maximum Advice-of-Charge related accumulated call meter value in the SIM file EFACMmax.

6.23.2. Format

Command	Possible response(s)
+CAMM=[<acmmax>[,<passwd>]]	*+CME ERROR: <err>*
+CAMM?	+CAMM: <acmmax> *+CME ERROR: <err>*
+CAMM=?	

6.23.3. Field

<acmmax>: string type;

accumulated call meter maximum value similarly coded as <ccm> under +CAOC;

value zero disables ACMmax feature

<passwd>: string type; SIM PIN2

6.24. AT+CPUC – Price per unit and currency table

6.24.1. Description

Sets the parameters of Advice-of-Charge related price per unit and currency in SIM file EF_{PUCT}. PUCT information can be used to convert the home units (as used in AT+CAOC, AT+CACM, and AT+CAMM) into currency units.

6.24.2. Format

Command	Possible response(s)
+CPUC=<currency>,<ppu>[,<passwd>]	*+CME ERROR: <err>*
+CPUC?	+CPUC: <currency>,<ppu> *+CME ERROR: <err>*
+CPUC=?	

6.24.3. Field

<currency>: string type; three-character currency code (e.g. "GBP", "DEM");

character set as specified by command Select TE Character Set +CSCS

<ppu>: string type; price per unit; dot is used as a decimal separator (e.g. "2.66")

<passwd>: string type; SIM PIN2

6.25. AT+CCWE – Call Meter maximum event

6.25.1. Description

Shortly before the ACM (Accumulated Call Meter) maximum value is reached, an unsolicited result code +CCWV will be sent, if enabled by this command. The warning is issued approximately when 30 seconds call time remains. It is also issued when starting a call if less than 30 s call time remains.

6.25.2. Format

Command	Possible response(s)
+CCWE=<mode>	*+CME ERROR: <err>*
+CCWE?	+CCWE: <mode> *+CME ERROR: <err>*
+CCWE=?	+CCWE: (list of supported <mode>s) *+CME ERROR: <err>*

6.25.3. Field

<mode>:

0 Disable the call meter warning event

1 Enable the call meter warning event

6.26. AT+CLAN – Set Language

6.26.1. Description

Sets the language in the ME. If the language has been set to .AUTO., the read command returns the current language set from the SIM card. Hence, the .AUTO. code is never returned by the read command.

6.26.2. Format

Command	Possible response(s)
+CLAN=<code>	*+CME ERROR: <err>*
+CLAN?	+CLAN: <code> *+CME ERROR: <err>*
+CLAN=?	+CLAN:(list of supported <code>s) *+CME ERROR: <err>*

6.26.3. Field

<code>:

“AUTO” – Read language from the active application in the SIM card.

“AUTO” is not returned by the read-command.

Note: When the preferred language from SIM card is not recognized or supported by our MMI, AT+CLAN=”AUTO” will remain current ME setting.

"en" -- English.

"zh-TW" – traditional Chinese. (old version: “TW”)

"zh-CN" – simplified Chinese. (old version: “ZH”)

6.27. AT+CLAE – Language Event

6.27.1. Description

To enable/disable unsolicited result code +CLAV: <code>. If <mode>=1, +CLAV: <code > is sent from the ME when the language in the ME is changed.

6.27.2. Format

Command	Possible response(s)
+CLAE=<mode>	*+CME ERROR: <err>*
+CLAE?	+CLAE: <mode> *+CME ERROR: <err>*
+CLAE=?	+CLAE: (list of supported <mode>s) *+CME ERROR: <err>*

6.27.3. Field

<mode>:

0 Disable unsolicited result code +CLAE

1 Enable unsolicited result code +CLAE

<code>: For description see +CLAN.

6.28. AT+CALD –Delete alarm

6.28.1. Description

Action command deletes an alarm in the MT.

6.28.2. Format

Command	Possible response(s)
+CALD=<n>	*+CME ERROR: <err>*
+CALD=?	+CALD: (list of supported <n>s) *+CME ERROR: <err>*

6.28.3. Field

<n>: integer type value indicating the index of the alarm; default is manufacturer specific.

6.29. AT+CTZR – Time Zone Reporting

6.29.1. Description

Enables and disables the time zone change event reporting. If the reporting is enabled the MT returns the unsolicited result code +CTZV: <tz> whenever the time zone is changed.

6.29.2. Format

Command	Possible response(s)
+CTZR=<onoff>	*+CME ERROR: <err>*
+CTZR?	+CTZR: <onoff> *+CME ERROR: <err>*
+CTZR=?	+CTZR: (list of supported <onoff>s) *+CME ERROR: <err>*

6.29.3. Field

<onoff>: integer type value indicating:

0 – Disable automatic time zone update via NITZ (default).

1 – Enable automatic time zone update via NITZ.

7. GPRS COMMANDS (3GPP 27.007)

7.1. AT+CGDCONT – Define PDP Context

7.1.1. Description

Specifies PDP context parameter values for a PDP context identified by the (local) context identification parameter, <cid>.

7.1.2. Format

Command	Possible response(s)
+CGDCONT=[<cid> [,<PDP_type> [,<APN> [,<PDP_addr> [,<d_comp> [,<h_comp> [,<pd1> [,…[,pdN]]]]]]]]	OK ERROR
+CGDCONT?	+CGDCONT: <cid>, <PDP_type>, <APN>, <PDP_addr>, <d_comp>, <h_comp>[,<pd1>[,…[,pdN]]] [<CR><LF>+CGDCONT: <cid>, <PDP_type>, <APN>,<PDP_addr>, <d_comp>, <h_comp>[,<pd1>[,…[,pdN]]] [...]]
+CGDCONT=?	+CGDCONT: (range of supported <cid>s), <PDP_type>,,,(list of supported <d_comp>s), (list of supported <h_comp>s)[,(list of supported <pd1>s)[,…[,(list of supported <pdN>s)]]] [<CR><LF>+CGDCONT: (range of supported <cid>s), <PDP_type>,,,(list of supported <d_comp>s), (list of supported <h_comp>s)[,(list of supported <pd1>s)[,…[,(list of supported <pdN>s)]]] [...]]

7.1.3. Field

<cid>:

(PDP Context Identifier) a numeric parameter which specifies a particular PDP context definition. The parameter is local to the TE-MT interface and is used in other PDP context-related commands. The range of permitted values (minimum value = 1) is returned by the test form of the command.

<PDP_type>: (Packet Data Protocol type) a string parameter.

IP Internet Protocol (IETF STD 5)

<APN>: (Access Point Name) a string parameter which is a logical name that is used to select the GGSN or the external packet data network.

If the value is null or omitted, then the subscription value will be requested.

<PDP_address>: a string parameter that identifies the MT in the address space applicable to the PDP.

If the value is null or omitted, then a value may be provided by the TE during the PDP startup procedure or, failing that, a dynamic address will be requested.

The read form of the command will continue to return the null string even if an address has been allocated during the PDP startup procedure. The allocated address may be read using the +CGPADDR command.

<d_comp>: a numeric parameter that controls PDP data compression (applicable for SNDCP only)

0 - off (default if value is omitted)

<h_comp>: a numeric parameter that controls PDP header compression

0 - off (default if value is omitted)

<pd1>, … <pdN>: zero to N string parameters whose meanings are specific to the <PDP_type>

7.2. AT+CGDSCONT – Define Secondary PDP Context

7.2.1. Description

The set command specifies PDP context parameter values for a Secondary PDP context identified by the (local) context identification parameter, <cid>. The number of PDP contexts that may be in a defined state at the same time is given by the range returned by the test command.

7.2.2. Format

Command	Possible response(s)
+CGDSCONT=[<cid> ,<p_cid> [,<d_comp> [,<h_comp>]]]	OK ERROR
+CGDSCONT?	+CGDSCONT: <cid>, <p_cid>, <d_comp>, <h_comp> [<CR><LF>+CGDSCONT: <cid>, <p_cid>, <d_comp>, <h_comp> [...]]
+CGDSCONT=?	+CGDSCONT: (range of supported <cid>s), (list of <cid>s for active primary contexts),(list of supported <d_comp>s), (list of supported <h_comp>s)

7.2.3. Field

<cid>: (PDP Context Identifier) a numeric parameter which specifies a particular PDP context definition. The parameter is local to the TE-MT interface and is used in other PDP context-related commands. The range of permitted values (minimum value = 1) is returned by the test form of the command.

<p_cid>: (Primary PDP Context Identifier) a numeric parameter which specifies a particular PDP context definition which has been specified by use of the +CGDCONT command. The parameter is local to the TE-MT interface. The list of permitted values is returned by the test form of the command.

<PDP_type>: (Packet Data Protocol type) a string parameter which specifies the type of packet data protocol

IP Internet Protocol (IETF STD 5)

<d_comp>: a numeric parameter that controls PDP data compression

0 - off (default if value is omitted)

<h_comp>: a numeric parameter that controls PDP header compression

0 - off (default if value is omitted)

7.3. AT+CGQREQ – Quality of Service Profile (Requested)

7.3.1. Description

This command allows the TE to specify a Quality of Service Profile that is used when the MT sends an Activate PDP Context Request message to the network.

7.3.2. Format

Command	Possible Response(s)
+CGQREQ=[<cid> [,<precedence > [,<delay> [,<reliability.> [,<peak> [,<mean>]]]]]]	OK ERROR
+CGQREQ?	+CGQREQ: <cid>, <precedence >, <delay>, <reliability>, <peak>, <mean> [<CR><LF>+CGQREQ: <cid>, <precedence >, <delay>, <reliability.>, <peak>, <mean> […]]
+CGQREQ=?	+CGQREQ: <PDP_type>, (list of supported <precedence>s), (list of supported <delay>s), (list of supported <reliability>s) , (list of supported <peak>s), (list of supported <mean>s) [<CR><LF>+CGQREQ: <PDP_type>, (list of supported <precedence>s), (list of supported <delay>s), (list of supported <reliability>s) , (list of supported <peak>s), (list of supported <mean>s) […]]

7.3.3. Field

<cid>: a numeric parameter which specifies a particular PDP context definition

<precedence>: a numeric parameter which specifies the precedence class

<delay>: a numeric parameter which specifies the delay class

<reliability>: a numeric parameter which specifies the reliability class

<peak>: a numeric parameter which specifies the peak throughput class

<mean>: a numeric parameter which specifies the mean throughput class

7.4. AT+CGQMIN – Quality of Service Profile (Minimum acceptable)

7.4.1. Description

This command allows the TE to specify a minimum acceptable profile which is checked by the MT against the negotiated profile returned in the Activate PDP Context Accept message.

7.4.2. Format

Command	Possible Response(s)
+CGQMIN=[<cid> [,<precedence > [,<delay> [,<reliability.> [,<peak> [,<mean>]]]]]]	OK ERROR
+CGQMIN?	+CGQMIN: <cid>, <precedence >, <delay>, <reliability>, <peak>, <mean> [<CR><LF>+CGQMIN: <cid>, <precedence >, <delay>, <reliability.>, <peak>, <mean> […]]
+CGQMIN=?	+CGQMIN: <PDP_type>, (list of supported <precedence>s), (list of supported <delay>s), (list of supported <reliability>s) , (list of supported <peak>s), (list of supported <mean>s) [<CR><LF>+CGQMIN: <PDP_type>, (list of supported <precedence>s), (list of supported <delay>s), (list of supported <reliability>s) , (list of supported <peak>s), (list of supported <mean>s) […]]

7.4.3. Field

<cid>: a numeric parameter which specifies a particular PDP context definition

<precedence>: a numeric parameter which specifies the precedence class

<delay>: a numeric parameter which specifies the delay class

<reliability>: a numeric parameter which specifies the reliability class

<peak>: a numeric parameter which specifies the peak throughput class

<mean>: a numeric parameter which specifies the mean throughput class

7.5. AT+CGATT – PS attach or detach

7.5.1. Description

The execution command is used to attach the MT to, or detach the MT from, the Packet Domain service. After the command has completed, the MT remains in V.250 command state.

7.5.2. Format

Command	Possible Response(s)
+CGATT= [<state>]	OK ERROR
+CGATT?	+CGATT: <state>
+CGATT=?	+CGATT: (list of supported <state>s)

7.5.3. Field

<state>: indicates the state of PS attachment

0 - detached

1 - attached

7.6. AT +CGACT – PDP context activate or deactivate

7.6.1. Description

To activate or deactivate the specified PDP context (s).

7.6.2. Format

Command	Possible Response(s)
+CGACT=[<state> [,<cid>]]	OK ERROR
+CGACT?	+CGACT: <cid>, <state> [<CR><LF>+CGACT: <cid>, <state> [...]]
+CGACT=?	+CGACT: (list of supported <state>s)

7.6.3. Field

<state>: indicates the state of PDP context activation

0 - deactivated

1 - activated

Other values are reserved and will result in an ERROR response to the execution command.

<cid>: a numeric parameter which specifies a particular PDP context definition. If no <cid> is specified, then UE assumes it as 1. The usage of omitted <cid> to activate/deactivate all is not supported.

7.7. AT +CGCMOD –PDP Context Modify

7.7.1. Description

The execution command is used to modify the specified PDP context (s) with respect to QoS profiles and TFTs.

7.7.2. Format

Command	Possible Response(s)
+CGCMOD=<cid>	OK ERROR
+CGCMOD=?	+CGCMOD: (list of <cid>s associated with active contexts)

7.7.3. Field

<cid>: a numeric parameter which specifies a particular PDP context definition (see the +CGDCONT and +CGDSCONT commands).

7.8. AT+CGDATA – Enter data state

7.8.1. Description

The execution command causes the MT to perform whatever actions are necessary to establish communication between the TE and the network using one or more Packet Domain PDP types.

7.8.2. Format

Command	Possible Response(s)
+CGDATA=[<L2P> ,[<cid>]]	CONNECT ERROR
+CGDATA=?	+CGDATA: (list of supported <L2P>s)

7.8.3. Field

<L2P>: a string parameter that indicates the layer 2 protocol to be used between the TE and MT
PPP Point-to-point protocol for a PDP such as IP

Other values will result in an ERROR response.

<cid>: a numeric parameter which specifies a particular PDP context definition (see the +CGDCONT and +CGDSCONT commands).

7.9. AT+CGPADDR – Show PDP address

7.9.1. Description

The execution command returns a list of PDP addresses for the specified context identifiers. The test command returns a list of defined <cid>s.

7.9.2. Format

Command	Possible response(s)
+CGPADDR=<cid>	+CGPADDR: <cid>,<PDP_addr>
+CGPADDR=?	+CGPADDR: (list of defined <cid>s)

7.9.3. Field

<cid>: a numeric parameter which specifies a particular PDP context definition (see the +CGDCONT and +CGDSCONT commands). If no <cid> is specified, an ERROR result code will be returned. Multiple <cid> field is not supported.

<PDP_address>: a string that identifies the MT in the address space applicable to the PDP. The address may be static or dynamic. For a static address, it will be the one set by the +CGDCONT and +CGDSCONT commands when the context was defined. For a dynamic address it will be the one assigned during the last PDP context activation that used the context definition referred to by <cid>.
<PDP_address> is omitted if none is available.

7.10. AT+CGAUTO – Automatic response to network request PDP context activation

7.10.1. Description

The set command disables or enables an automatic positive response (auto-answer) to the receipt of a Request PDP Context Activation message from the network.

When the +CGAUTO=0 command is received, the MT shall not perform a PS detach if it is attached. Subsequently, when the MT announces a network request for PDP context activation by issuing the unsolicited result code RING or +CRING, the TE may manually accept or reject the request by issuing the +CGANS command or may simply ignore the network request.

When the +CGAUTO=1 command is received, the MT shall attempt to perform a PS attach if it is not already attached. Failure will result in ERROR or, if enabled, +CME ERROR being returned to the TE. Subsequently, when the MT announces a network request for PDP context activation by issuing the unsolicited result code RING or +CRING to the TE, this is followed by the intermediate result code CONNECT. The MT then enters V.250 online data state and follows the same procedure as it would after having received a +CGANS=1 with no <L2P> or <cid> values specified.

7.10.2. Format

Command	Possible response(s)
+CGAUTO=<n>	OK ERROR
+CGAUTO?	+CGAUTO: <n>

7.10.3. Field

<n>:

0 turn off automatic response for Packet Domain only
1 turn on automatic response for Packet Domain only

For <n> = 0 Packet Domain network requests are manually accepted or rejected by the +CGANS command.

For <n> = 1 Packet Domain network requests are automatically accepted according to the description above.

7.11. AT+CGANS – Manual response to a network request for PDP context activation

7.11.1. Description

The execution command requests the MT to respond to a network request for Packet Domain PDP context activation which has been signaled to the TE by the RING or +CRING: unsolicited result code. The <response> parameter allows the TE to accept or reject the request.

7.11.2. Format

Command	Possible response(s)
+CGANS=[<response>, [<L2P> ,[<cid>]]]	OK ERROR
+CGANS=?	+CGANS: (list of supported <response>s), (list of supported <L2P>s)

7.11.3. Field

<response>: is a numeric parameter which specifies how the request should be responded to.

0 reject the request

1 accept and request that the PDP context be activated

<L2P>: a string parameter which indicates the layer 2 protocol to be used (see +CGDATA command).

<cid>: a numeric parameter which specifies a particular PDP context definition

7.12. AT+EGPAU – GPRS PPP negotiated authentication protocol

7.12.1. Description

The command is used to set GPRS PPP negotiated authentication protocol.

7.12.2. Format

Command	Possible response(s)
+EGPAU=<op>,<cid>[,<is_chap>]	OK ERROR

+ EGPAU =?	+EGPAU=(0-1),(1-<cid_max>),(0-2) OK

7.12.3. Field

<op>: operation

0 Read

1 Write

<cid>: context ID. Please refer to the value in test command response.

<is_chap>: negotiation protocol

0: PAP

1: CHAP

2: Not documented

7.13. AT+CGCLASS – GPRS mobile station class

7.13.1. Description

The set command is used to set the MT to operate according to the specified GPRS mobile class. If the requested class is not supported, an ERROR or +CME ERROR response is returned. Extended error responses are enabled by the +CMEE command.

The read command returns the current GPRS mobile class.

The test command is used for requesting information on the supported GPRS mobile classes.

7.13.2. Format

Command	Possible response(s)
+CGCLASS=[<class>]	OK ERROR
+CGCLASS?	+CGCLASS:<class>
+CGCLASS=?	+CGCLASS: (list of supported <class>s)

7.13.3. Field

<class>: a string parameter which indicates the GPRS mobile class (in descending order of functionality)

A: class A (highest)

B: class B

CG: class C in GPRS only mode

CC: class C in circuit switched only mode (lowest)

Other values are reserved and will result in an ERROR response to the set command.

If the MT is GPRS attached when the set command is issued with a <class> = CC specified, a detach request shall be sent to the network.

7.14. AT+CGREG – GPRS network registration status

7.14.1. Description

The set command controls the presentation of an unsolicited result code +CGREG: <stat> when <n>=1 and there is a change in the MT's GPRS network registration status, or code +CGREG: <stat>[,<lac>,<ci>[,<Act>]] when <n>=2 and there is a change of the network cell.

The read command returns the status of result code presentation and an integer <stat> which shows whether the network has currently indicated the registration of the MT. Location information elements <lac>,<ci> and <Act> are returned only when <n>=2 and MT is registered in the network.

7.14.2. Format

Command	Possible response(s)
+CGREG=[<n>]	
+CGREG?	+CGREG: <n>,<stat>[,<lac>,<ci>[,<Act>]] *+CME ERROR: <err>*

7.14.3. Field

<n>:

0 disable network registration unsolicited result code

1 enable network registration unsolicited result code +CGREG: <stat>

2 enable network registration and location information unsolicited result code +CGREG:

<stat>:

0 not registered, MT is not currently searching an operator to register to

1 registered, home network

2 not registered, but MT is currently trying to attach or searching an operator to register to

3 registration denied

4 unknown

5 registered, roaming

<lac>: string type; two byte location area code in hexadecimal format (e.g. "00C3" equals 195 in decimal)

<ci>: string type; four byte cell ID in hexadecimal format

<Act>:

0 GSM

2 UTRAN

3 GSM w/EGPRS

4 UTRAN w/HSDPA

5 UTRAN w/HSUPA

6 UTRAN w/HSDPA and HSUPA

7.15. AT+CGSMS – Select service for MO SMS messages

7.15.1. Description

The set command is used to specify the service or service preference that the MT will use to send MO SMS messages.

The read command returns the currently selected service or service preference.

The test command is used for requesting information on the currently available services and service preferences.

7.15.2. Format

Command	Possible Response(s)
+CGSMS= <service>	OK ERROR
+CGSMS?	+CGSMS: <service>

7.15.3. Field

<service>: a numeric parameter which indicates the service or service preference to be used

0 Packet Domain
1 circuit switched
2 Packet Domain preferred (use circuit switched if GPRS not available)
3 circuit switched preferred (use Packet Domain if circuit switched not available)

8. MOBILE TERMINATION ERRORS (27.007)

8.1. AT+CMEE

8.1.1. Description

Set command disables or enables the use of result code +CME ERROR: <err> as an indication of an error relating to the functionality of the MT. When enabled, MT related errors cause +CME ERROR: <err> final result code instead of the regular ERROR final result code. ERROR is returned normally when error is related to syntax, invalid parameters, or TA functionality.

Test command returns values supported as a compound value.

8.1.2. Format

Command	Possible response(s)
+CMEE=[<n>]	
+CMEE?	+CMEE: <n>
+CMEE=?	+CMEE: (list of supported <n>s)

8.1.3. Field

<n>:

0 disable +CME ERROR: <err> result code and use ERROR instead

1 enable +CME ERROR: <err> result code and use numeric <err> values (refer next subclause)

2 enable +CME ERROR: <err> result code and use verbose <err> values (refer next subclause)

<err> values (numeric format followed by verbose format):

9.2.1 General errors

0 phone failure

1 no connection to phone

2 phone adaptor link reserved

3 operation not allowed

4 operation not supported

5 PH SIM PIN required

6 PH-FSIM PIN required

7 PH-FSIM PUK required

10 SIM not inserted

11 SIM PIN required

12 SIM PUK required

13 SIM failure

14 SIM busy

15 SIM wrong
16 incorrect password
17 SIM PIN2 required
18 SIM PUK2 required
20 memory full
21 invalid index
22 not found
23 memory failure
24 text string too long
25 invalid characters in text string
26 dial string too long
27 invalid characters in dial string
30 no network service
31 network timeout
32 network not allowed - emergency calls only
40 network personalization PIN required
41 network personalization PUK required
42 network subset personalization PIN required
43 network subset personalization PUK required
44 service provider personalization PIN required
45 service provider personalization PUK required
46 corporate personalization PIN required
47 corporate personalization PUK required
48 hidden key required (NOTE: This key is required when accessing hidden phonebook entries.)
100 unknown

9.2.2 GPRS-related errors

9.2.2.1 Errors related to a failure to perform an Attach

103 Illegal MS (#3)
106 Illegal ME (#6)
107 GPRS service not allowed (#7)
111 PLMN not allowed (#11)
112 Location area not allowed (#12)
113 Roaming not allowed in this location area (#13)

(Values in parentheses are TS 24.008 cause codes.)

9.2.2.2 Errors related to a failure to Activate a Context

132 service option not supported (#32)

133 requested service option not subscribed (#33)

134 service option temporarily out of order (#34)

149 PDP authentication failure

(Values in parentheses are TS 24.008 cause codes.)

9.2.2.3 Other GPRS errors

150 invalid mobile class

148 unspecified GPRS error

Other values in the range 101-150 are reserved for use by GPRS

9. ANNEX C (27.007)

9.1. AT+FCLASS

9.1.1. Description

Puts the TA in a specific mode of operation. This causes the TA to process information in a manner suitable for that type of information.

9.1.2. Format

Command	Response
+FCLASS=<n>	
+FCLASS?	<n>
+FCLASS=?	(list of supported <n>s)

9.1.3. Field

<n> Mode

0 data

1 fax class 1 (TIA-578-A)

2 fax (manufacturer specific)

2.0 fax class 2 (ITU T T.32 and TIA 592)

9.2. AT+VTS

9.2.1. Description

Allows the transmission of DTMF tones. The command is write-only.

NOTE

The command is used only during voice calls.

9.2.2. Format

Command	Return
+VTS=<dtmf>	
+VTS=?	(list of supported <tone1>s),(list of supported <tone2>s) ,(list of supported <duration>s)

9.2.3. Field

<DTMF>. A single ASCII character in the set .0-9, #, *, A-D.

For example: AT+VTS = 9 or AT+VTS = A

You can use multiple command to achieve continuous DTMF tones.

For example : AT+VTS=6;+VTS=2;+VTS=8;+VTS=2

10. SMS AT COMMANDS (27.005)

Please refer to 27.005 Sec 3.1 Parameter Definition to see more details of the parameter fields in each command.

10.1. AT+CSMS – Select Message Service

10.1.1. Description

Selects the message service and returns the type of messages supported by the ME. If chosen service is not supported by the ME (but supported by the TA), +CME ERROR is returned.

10.1.2. Format

Command	Possible response(s)
+CSMS=<service>	+CSMS: <mt>,<mo>,<bm> *+CMS ERROR: <err>*
+CSMS?	+CSMS: <service>,<mt>,<mo>,<bm>
+CSMS=?	+CSMS: (list of supported <service>s)

10.1.3. Field

<service>:

0 3GPP TS 23.040 and 3GPP TS 23.041

1 3GPP TS 23.040 and 3GPP TS 23.041
the requirement of <service> setting 1 is mentioned under corresponding command descriptions)

<mt>, <mo>, <bm>:

0 type not supported

1 type supported

10.2. AT+CPMS – Preferred Message Storage

10.2.1. Description

Selects memory storage spaces to be used for reading, writing, etc. If chosen storage is not appropriate for the ME (but is supported by the TA), +CME ERROR is returned.

10.2.2. Format

Command	Possible response(s)
+CPMS=<mem1>	+CPMS: <used1>,<total1>,<used2>,<total2>,<used3>,<total3> *+CMS ERROR: <err>*
+CPMS?	+CPMS: <mem1>,<used1>,<total1>,<mem2>,<used2>,<total2>, <mem3>,<used3>,<total3> *+CMS ERROR: <err>*
+CPMS=?	+CPMS: (list of supported <mem1>s),(list of supported <mem2>s), (list of supported <mem3>s)

10.3. AT+CMGF – Message Format

10.3.1. Description

Sets the input and output format to be used by the TA.

10.3.2. Format

Command	Possible response(s)
+CMGF=[<mode>]	
+CMGF?	+CMGF: <mode>
+CMGF=?	+CMGF: (list of supported <mode>s)

10.3.3. Field

<mode>:

0 PDU mode (default)

1 text mode

10.4. AT+CSCA – Service Center Address

10.4.1. Description

Updates the SMCS address, through which mobile-originated SMSs are transmitted. In text mode, the setting is used by send (AT+CMGS) and write (AT+CMGW) commands. In PDU mode, the setting is used by the same commands, but only when the length of the SMCS address (coded into <pdu> parameter) equals zero.

10.4.2. Format

Command	Possible response(s)
+CSCA=<sca>[,<tosca>]	
+CSCA?	+CSCA: <sca>,<tosca>
+CSCA=?	

10.5. AT+CSMP – Set Text Mode Parameters

10.5.1. Description

Setting Text Mode Parameters. Set command is used to select values for additional parameters needed when SM is sent to the network or placed in a storage when text format message mode is selected. It is possible to set the validity period starting from when the SM is received by the SMSC (<vp> is in range 0... 255) or define the absolute time of the validity period termination (<vp> is a string). The format of <vp> is given by <fo>.

10.5.2. Format

Command	Possible response(s)
+CSMP=[<fo>[,<vp>[,<pid>[,<dcs>]]]]	
+CSMP?	+CSMP: <fo>,<vp>,<pid>,<dcs>
+CSMP=?	

10.6. AT+CSDH – Show Text Mode Parameters

10.6.1. Description

Set command controls whether detailed header information is shown in text mode result codes.

Test command returns supported values as a compound value.

10.6.2. Format

Command	Possible response(s)
+CSDH=[<show>]	
+CSDH?	+CSDH: <show>
+CSDH=?	+CSDH: (list of supported <show>s)

10.7. AT+CSCB – Select Cell Broadcast Message Types

10.7.1. Description

Selects which types of CBMs are to be received by the ME.

10.7.2. Format

Command	Possible response(s)
+CSCB=[<mode>[,<mids>]]	
+CSCB?	+CSCB: <mode>,<mids>
+CSCB=?	+CSCB: (list of supported <mode>s)

10.7.3. Field

<mode>:

0 message types specified in <mids> and <dcss> are accepted

1 message types specified in <mids> and <dcss> are not accepted

<mids>: We support **10** message identifiers at most.

string type: all different possible combinations of CBM message identifiers (refer <mid>) (default is empty string);

e.g. "0,1,5,320-478,922"

<dcss>: string type; all different possible combinations of CBM data coding schemes (refer <dcs>) (default is empty string);e.g. “0-3,5”

NOTE 1

For <mids> of <mode>=0, our design is to open the <mids> from user input and close other <mids>.

In the following case, user input <mode>=0 and <mids>=2. So open channel 2 and close other channel (channel 1).

AT+CSCB?

+CSCB: 0,**"1"**,"1"

OK

AT+CSCB=**0**,**"2"**,"2"

OK

AT+CSCB?

+CSCB: 0,**"2"**,"1,2"

OK

In the following case, user input <mode>=0 without <mids>. So don't open any channel and close other channel (channel 1).

AT+CSCB?

+CSCB: 0,**"1"**,"1"

OK

AT+CSCB=**0**

OK

AT+CSCB?

+CSCB: 0,**""**,"1"

OK

For <dcss> of <mode>=0, our design is to **increase** the <dcss> from user input.

In the following case, user input <mode>=0 and <dcss>=2. So **increase** language 2.

AT+CSCB?

+CSCB: 0,"1",**"1"**

OK

AT+CSCB=**0**,"2",**"2"**

OK

AT+CSCB?

+CSCB: 0,"2",**"1,2"**

OK

In the following case, user input <mode>=0 without <dcss>. So don't **increase** any language.

AT+CSCB?

+CSCB: 0,"1",**"1"**

OK

AT+CSCB=**0**

OK

AT+CSCB?

+CSCB: 0,"",**"1"**

OK

NOTE 2

For <mids> of <mode>=1, our design is to close all <mids> no matter with <mids> or not.

In the following case, user input <mode>=1. So close all channel.

AT+CSCB?

+CSCB: 0,**"2"**,"1,2"

OK

AT+CSCB=**1**,"2","2"

OK

AT+CSCB?

+CSCB: 1,**""**,"1"

OK

In the following case, user input <mode>=1 without <mids>. Also close all channel.

AT+CSCB?

+CSCB: 0,**"1"**,"1"

OK

AT+CSCB=**1**

OK

AT+CSCB?

+CSCB: 1,**""**,"1"

OK

For <dcss> of <mode>=1, our design is to **decrease** the <dcss> from user input.

In the following case, user input <mode>=1 and <dcss>=2. So **decrease** language 2.

AT+CSCB?

+CSCB: 0,"2",**"1,2"**

OK

AT+CSCB=**1**,"2",**"2"**

OK

AT+CSCB?

+CSCB: 1,"",**"1"**

OK

In the following case, user input <mode>=1 without <dcss>. So don't **decrease** any language.

AT+CSCB?

+CSCB: 0,"1",**"1"**

OK

AT+CSCB=**1**

OK

AT+CSCB?

+CSCB: 1,"",**"1"**

OK

10.7.4. Usage Note

NOTE

- <mid> 3GPP TS 23.041 CBM Message Identifier in integer format
- <dcs> depending on the command or result code: 3GPP TS 23.038 SM Data Coding Scheme (default 0), or Cell Broadcast Data Coding Scheme in integer format

10.8. AT+CSAS – Save Settings

10.8.1. Description

Execution command saves active message service settings to a non-volatile memory. Settings specified in commands Service Centre Address +CSCA, Set Message Parameters +CSMP and Select Cell Broadcast Message Types +CSCB (if implemented) are saved. Certain settings may not be supported by the storage (e.g. (U)SIM SMS parameters) and therefore cannot be saved.

10.8.2. Format

Command	Possible response(s)
+CSAS[=<profile>]	*+CMS ERROR: <err>*
+CSAS=?	+CSAS: (list of supported <profile>s)

10.8.3. Field

<profile>:

0...255 manufacturer specific profile number where settings are to be stored

10.9. AT+CRES – Restore Settings

10.9.1. Description

Execution command restores message service settings from non-volatile memory to active memory. A TA can contain several profiles of settings. Settings specified in commands Service Centre Address +CSCA, Set Message Parameters +CSMP and Select Cell Broadcast Message Types +CSCB (if implemented) are restored. Certain settings may not be supported by the storage (e.g. (U)SIM SMS parameters) and therefore cannot be restored.

10.9.2. Format

Command	Possible response(s)
+CRES[=<profile>]	*+CMS ERROR: <err>*
+CRES=?	+CRES: (list of supported <profile>s)

10.9.3. Field

<profile>:

0...255 manufacturer specific profile number where settings are to be stored

10.10. AT+CMGL (Text mode) – List Message

10.10.1. Description

Returns messages with status value <stat> from returned message in preferred storage to the TE.

10.10.2. Format

Command	Possible response(s)
+CMGL[=<stat>]	**if text mode (+CMGF=1), command successful and SMS-SUBMITs and/or SMS-DELIVERs:** +CMGL: <index>,<stat>,<oa/da>,[<alpha>],[<scts>]*[,<tooa/toda>,* *<length>]*<CR><LF><data>[<CR><LF> +CMGL: <index>,<stat>,<da/oa>,[<alpha>],[<scts>]*[,<tooa/toda>,* *<length>]*<CR><LF><data>[...]] **if text mode (+CMGF=1), command successful and SMS-STATUS-REPORTs:** +CMGL: <index>,<stat>,<fo>,<mr>,[<ra>],[<tora>],<scts>,<dt>,<st> [<CR><LF> +CMGL: <index>,<stat>,<fo>,<mr>,[<ra>],[<tora>],<scts>,<dt>,<st> [...]] **if text mode (+CMGF=1), command successful and SMS-COMMANDs:** +CMGL: <index>,<stat>,<fo>,<ct>[<CR><LF> +CMGL: <index>,<stat>,<fo>,<ct>[...]] **if text mode (+CMGF=1), command successful and CBM storage:** +CMGL: <index>,<stat>,<sn>,<mid>,<page>,<pages> <CR><LF><data>[<CR><LF> +CMGL: <index>,<stat>,<sn>,<mid>,<page>,<pages> <CR><LF><data>[...]] **otherwise:** *+CMS ERROR: <err>*
+CMGL=?	+CMGL: (list of supported <stat>s)

10.11. AT+CMGL(PDU mode) – List Message

10.11.1. Description

Returns messages with status value <stat> from returned message in preferred storage to the TE.

10.11.2. Format

Command	Possible response(s)
+CMGL[=<stat>]	**if PDU mode (+CMGF=0) and command successful:** +CMGL: <index>,<stat>,[<alpha>],<length><CR><LF><pdu> [<CR><LF>+CMGL:<index>,<stat>,[<alpha>],<length><CR><LF><pdu> [...]] **otherwise:** *+CMS ERROR: <err>*
+CMGL=?	+CMGL: (list of supported <stat>s)

10.12. AT+CMGR (Text mode) – Read Message

10.12.1. Description

Returns messages with location value <index> from preferred message storage <mem1> to the TE. If the status of the message is .received unread., the status in the storage changes to .received read.. If reading fails, +CMS ERROR is returned.

10.12.2. Format

Command	Possible response(s)
+CMGR=<index>	**if text mode (+CMGF=1), command successful and SMS-DELIVER:** +CMGR: <stat>,<oa>,[<alpha>],<scts>*[,<tooa>,<fo>,<pid>,<dcs>, <sca>,<tosca>,<length>]*<CR><LF><data> **if text mode (+CMGF=1), command successful and SMS-SUBMIT:** +CMGR: <stat>,<da>,[<alpha>]*[,<toda>,<fo>,<pid>,<dcs>,[<vp>], <sca>,<tosca>,<length>]*<CR><LF><data> **if text mode (+CMGF=1), command successful and SMS-STATUS-REPORT:** +CMGR: <stat>,<fo>,<mr>,[<ra>],[<tora>],<scts>,<dt>,<st> **if text mode (+CMGF=1), command successful and SMS-COMMAND:** +CMGR: <stat>,<fo>,<ct>*[,<pid>,[<mn>],[<da>],[<toda>],<length>* *<CR><LF><cdata>]* **if text mode (+CMGF=1), command successful and CBM storage:** +CMGR: <stat>,<sn>,<mid>,<dcs>,<page>,<pages><CR><LF><data> **otherwise:** *+CMS ERROR: <err>*
+CMGR=?	

10.13. AT+CMGR (PDU mode) – Read Message

10.13.1. Description

Returns messages with location value <index> from preferred message storage <mem1> to the TE. If the status of the message is .received unread., the status in the storage changes to .received read.. If reading fails, +CMS ERROR is returned.

10.13.2. Format

Command	Possible response(s)
+CMGR=<index>	**if PDU mode (+CMGF=0) and command successful:** +CMGR: <stat>,[<alpha>],<length><CR><LF><pdu> **otherwise:** *+CMS ERROR: <err>*
+CMGR=?	

10.14. AT+CNMA (Text mode) – New Message Acknowledgement to ME/TA

10.14.1. Description

Execution command confirms correct reception of a new message (SMS-DELIVER or SMS-STATUS-REPORT) which is routed directly to the TE. This acknowledgement command (causing ME to send RP-ACK to the network) shall be used when +CSMS parameter <service> equals 1.

10.14.2. Format

Command	Possible response(s)
if text mode (+CMGF=1): +CNMA	*+CMS ERROR: <err>*
+CNMA=?	

10.15. AT+CNMA (PDU mode) – New Message Acknowledgement to ME/TA

10.15.1. Description

Execution command confirms correct reception of a new message (SMS-DELIVER or SMS-STATUS-REPORT) which is routed directly to the TE This acknowledgement command (causing ME to send RP-ACK to the network) shall be used when +CSMS parameter <service> equals 1.

10.15.2. Format

Command	Possible response(s)
if PDU mode (+CMGF=0): +CNMA[=<n>[,<length>[<CR> ***PDU is given***<ctrl-Z/ESC>]]]	*+CMS ERROR: <err>*
+CNMA=?	**if PDU mode (+CMGF=0):** +CNMA: (list of supported <n>s)

10.16. AT+CMGS (Text mode) – Send Message

10.16.1. Description

Execution command sends message from a TE to the network (SMS-SUBMIT). Message reference value <mr> is returned to the TE on successful message delivery.

10.16.2. Format

Command	Possible response(s)
if text mode (+CMGF=1): +CMGS=<da>[,<toda>]<CR> ***text is entered***<ctrl-Z/ESC>	**if text mode (+CMGF=1) and sending successful:** +CMGS: <mr>[,<scts>] **if sending fails:** *+CMS ERROR: <err>*
+CMGS=?	

10.17. AT+CMGS (PDU mode) – Send Message

10.17.1. Description

Execution command sends message from a TE to the network (SMS-SUBMIT). Message reference value <mr> is returned to the TE on successful message delivery.

10.17.2. Format

Command	Possible response(s)
if PDU mode (+CMGF=0): +CMGS=<length><CR> ***PDU is given***<ctrl-Z/ESC>	**if PDU mode (+CMGF=0) and sending successful:** +CMGS: <mr>[,<ackpdu>] **if sending fails:** *+CMS ERROR: <err>*
+CMGS=?	

10.18. AT+CMSS (Text mode) – Send Message from Storage

10.18.1. Description

Execution command sends message with location value <index> from preferred message storage <mem2> to the network (SMS-SUBMIT or SMS-COMMAND). If new recipient address <da> is given for SMS-SUBMIT, it shall be used instead of the one stored with the message. Reference value <mr> is returned to the TE on successful message delivery.

10.18.2. Format

Command	Possible response(s)
+CMSS=<index>[,<da>[,<toda>]]	**if text mode (+CMGF=1) and sending successful:** +CMSS: <mr>[,<scts>] **if sending fails:** *+CMS ERROR: <err>*
+CMSS=?	

10.19. AT+CMSS (PDU mode) – Send Message from Storage

10.19.1. Description

Execution command sends message with location value <index> from message storage <mem2> to the network (SMS-SUBMIT or SMS-COMMAND). If new recipient address <da> is given for SMS-SUBMIT, it shall be used instead of the one stored with the message. Reference value <mr> is returned to the TE on successful message delivery.

10.19.2. Format

Command	Possible response(s)
+CMSS=<index>[,<da>[,<toda>]]	**if PDU mode (+CMGF=0) and sending successful:** +CMSS: <mr>[,<ackpdu>] **if sending fails:** *+CMS ERROR: <err>*
+CMSS=?	

10.20. AT+CMGW (Text mode) – Write Message to Memory

10.20.1. Description

Execution command stores a message to memory storage <mem2>. Memory location <index> of the stored message is returned. By default message status will be set to 'stored unsent', but parameter <stat> allows also other status values to be given, support ‘stored unsent’ and “stored sent”

10.20.2. Format

Command	Possible response(s)
if text mode (+CMGF=1): +CMGW[=<oa/da>[,<tooa/toda>[,<stat>]]]<CR> ***text is entered***<ctrl-Z/ESC>	+CMGW: <index> *+CMS ERROR: <err>*
+CMGW=?	

10.21. AT+CMGW (PDU mode) – Write Message to Memory

10.21.1. Description

Execution command stores a message to memory storage <mem2>. Memory location <index> of the stored message is returned. By default message status will be set to 'stored unsent', but parameter <stat> allows also other status values to be given, support ‘stored unsent’ and “stored sent”

10.21.2. Format

Command	Possible response(s)
if PDU mode (+CMGF=0): +CMGW=<length>[,<stat>]<CR>***PDU is given*** <ctrl-Z/ESC>	+CMGW: <index> *+CMS ERROR: <err>*
+CMGW=?	

10.21.3. Field

<stat> integer type in PDU mode (default 0), or string type in text mode (default "REC UNREAD"); indicates

the status of message in memory; defined values:

0 "REC UNREAD" received unread message (i.e. new message)

1 "REC READ" received read message

2 "STO UNSENT" stored unsent message (only applicable to SMs)

3 "STO SENT" stored sent message (only applicable to SMs)

4 "ALL" all messages (only applicable to +CMGL command)

7 "DRAFT"

10.22. AT+CMGD – Delete Message

10.22.1. Description

Deletes message from preferred message <mem1> (see AT+CPMS) storage location <index>. If deletion fails, +CMS ERROR is returned.

10.22.2. Format

Command	Possible response(s)
+CMGD=<index>[,<delflag>]	*+CMS ERROR: <err>*
+CMGD=?	+CMGD: (list of supported <index>s)[,(list of supported <delflag>s)]

10.22.3. Field

<delflag>: an integer indicating multiple message deletion request as follows:

0 (or omitted) Delete the message specified in <index>

1 Delete all read messages from preferred message storage, leaving unread messages and stored mobile originated messages (whether sent or not) untouched

2 Delete all read messages from preferred message storage and sent mobile originated messages, leaving unread messages and unsent mobile originated messages untouched

3 Delete all read messages from preferred message storage, sent and unsent mobile originated messages leaving unread messages untouched.

4 Delete all messages from preferred message storage including unread messages.

10.23. AT+CMGC (Text mode) – Send Command

10.23.1. Description

Execution command sends a command message from a TE to the network (SMS-COMMAND).

10.23.2. Format

Command	Possible response(s)
if text mode (+CMGF=1): +CMGC=<fo>,<ct>[,<pid>[,<mn>[,<da>[,<toda>]]]]<CR> ***text is entered***<ctrl-Z/ESC>	**if text mode (+CMGF=1) and sending successful:** +CMGC: <mr>[,<scts>] **if sending fails:** *+CMS ERROR: <err>*
+CMGC=?	

10.24. AT+CMGC (PDU mode) – Send Command

10.24.1. Description

Execution command sends a command message from a TE to the network (SMS-COMMAND).

10.24.2. Format

Command	Possible response(s)
if PDU mode (+CMGF=0): +CMGC=<length><CR> ***PDU is given***<ctrl-Z/ESC>	**if PDU mode (+CMGF=0) and sending successful:** +CMGC: <mr>[,<ackpdu>] **if sending fails:** *+CMS ERROR: <err>*
+CMGC=?	

10.25. AT+CMMS – More Message to Send

10.25.1. Description

Set command controls the continuity of SMS relay protocol link. When feature is enabled (and supported by network) multiple messages can be sent much faster as link is kept open.

Test command returns supported values as a compound value.

10.25.2. Format

Command	Possible response(s)
+CMMS=[<n>]	
+CMMS?	+CMMS: <n>
+CMMS=?	+CMMS: (list of supported <n>s)

10.25.3. Field

<n>:

0 disable

2 enable (if the time between the response of the latest message send command and the next send command exceeds 1-5 seconds (the exact value is up to ME implementation), ME shall close the link but TA shall not switch automatically back to <n>=0)

10.26. AT+EQSI – Query storage index

10.26.1. Description

To query storage index.

10.26.2. Format

Command	Possible Response(s)
+EQSI=<storage>	+EQSI: <storage>, <begin>, <end>, <used> OK/ERROR
+EQSI=?	+ESUO: (list of supported <storage>s)

10.26.3. Field

<storage>: string type; SM or ME

<begin>: beginning of index

<end>: ending of index

<used>: number of messages in <storage>

10.27. AT+EMGR (PDU mode) – Read Message

10.27.1. Description

Returns messages with location value <index> from preferred message storage <mem1> to the TE. If the status of the message is .received unread., the status in the storage changes to .received read.. If reading fails, +CMS ERROR is returned. It is similar with AT+CMGR (PDU mode). <stat> is different.

10.27.2. Format

Command	Possible response(s)
+EMGR=<index>	**if PDU mode (+CMGF=0) and command successful:** +EMGR: <stat>,[<alpha>],<length><CR><LF><pdu> **otherwise:** *+CMS ERROR: <err>*
+EMGR=?	

10.27.3. Field

<stat> integer type in PDU mode (default 0), or string type in text mode (default "REC UNREAD"); indicates the status of message in memory; defined values:

0 "REC UNREAD" received unread message (i.e. new message)

1 "REC READ" received read message

2 "STO UNSENT" stored unsent message (only applicable to SMs)

3 "STO SENT" stored sent message (only applicable to SMs)

4 "ALL" all messages (only applicable to +CMGL command)

7 "DRAFT"

11. PROPRIETARY AT COMMANDS

11.1. AT+ESLP – Sleep Mode

11.1.1. Description

This command is used to enable and disable sleep mode in the modem. The module enters in a low power condition with paging activity. During a call or SMS event, the module is operational again: the serial port is fully active for the duration of the call or SMS receiving event. Furthermore, the AT Commands are accepted during the paging events.

11.1.2. Format

Execution command : AT+ ESLP = <op>

Test command : AT+ ESLP =? Show if the command is supported

11.1.3. Field

Type	Short name	Long name	Parameter/comment	
Integer	op	operation	enable	1
			disable	0

11.1.4. Response

Test command : + ESLP: (0, 1)

Execution command : OK

11.2. AT+EPBSE – Band Selection

11.2.1. Description

To set MS preferred band.

11.2.2. Format

Command	Response
+EEBSE=<gsm_band>, <umts_band>	
+EEBSE?	+EPBSE: <gsm_band>, <umts_band>
+EEBSE =?	List of supported bit masks of each band mode +EPBSE: <gsm_band>, <umts_band>

11.2.3. Field

<GSM_band>

bit 1 EGSM900

bit 3 DCS1800

bit 4 PCS1900

bit 7 GSM850

0xff Auto selection → select All supported bands

<UMTS_band>

0xffff Auto selection→ select All supported bands

UMTS not supported on GE310

11.2.4. Example

Set Auto band (select all supported bands)

AT+EPBSE=255, 65535

OK

Set “EURO band” (GSM-900 / DCS-1800 / WCDMA-IMT-2000)

AT+EPBSE=10, 1

OK

NOTE

1 This command is not allowed to set each band mode, GSM or UMTS, as 0, said AT+EPBSE=<gsm_band>,0 or AT+EPBSE=0, <umts_band>.

2 If the band mode is not supported, this command will just ignore the setting

3 After using this command, user should reboot the module to let the setting become effective.

4 If we get 0 in the certain field using AT+EPBSE=? , it means that the field is not supported.

11.3. AT+CGSDATA – Sending uplink data

11.3.1. Description

This command is used to send uplink data to network.

11.3.2. Format

Command	Possible response(s)
+CGSDATA= <byte>	*+CME ERROR: <err>*

11.3.3. Field

<byte> the number of byte sending to network

example:

at+cgsdata = 500 (sending 500 bytes)

11.4. AT+EFTCM – Enable/disable RRM Feature - Jamming Detect

11.4.1. Description

This command enable/disable the power scan result on CCCH Block and SACCH Block status

11.4.2. Format

Command	Possible response(s)
+ETCM=<type>,<on>	*OK* *ERROR*
+ETCM=?	*+ETCM=<type0_on>,<type1_on>,<type2_on>*

11.4.3. Field

<type> : Integer type, Specified feature type

0 Power Scan results send
1 CCCH Block status send
2 SACCH block status send

<on>: a numeric parameter which specifies disable/enable of feature type.

0 off

1 On

<type0_on> : Power Scan results send status disable/enable status

0 off

1 On

<type1_on> : CCCH Block send status disable/enable status

0 off

1 On

<type2_on> : SACCH Block send status disable/enable status

0 off

1 On

NOTE

1 If Power scan results enabled, then Power scan results will be sent by URC +EPWSC

2 If CCCH Block status is enable, the CCCH Block status will be sent by URC +ECCCH

3 If SACCH Block status is enable, the SACCH Block status will be sent by URC +ESACCH

4 Power scan results enabling should be done before PLMN scan

11.5. AT+EPWSC – Unsolicited result

11.5.1. Description

This URC is used to inform about power scan results.

11.5.2. Format

Unsolicited result code
+EPWSC: [list of supported (<arfcn>,<rssi_in_qdbm ,<sch_status>,<bcch_Status>,<end>)s]

11.5.3. Field

<arfcn>: integer type: Indicate the frequency number. range 0~0xFFFFFFFF

Valid arfcn value

<rssi_in_qdbm>: Negative integer.

Received signal strength raw data in quarter dbm. (1 means invalid value)

< sch_status >: integer, used to indicate the Synchronization channel status

0 Not Decoded

1 Decode Failed

2 Decode success

<bcch_status >: integer, used to indicate the Broad cast channel status

0 Not Decoded

1 Decode Failed

2 Decode success

<end> : Integer type, Value 1 indicates End of the list.

Note

- URC is sent only when its activated through AT+ EFTCM = 0,1

11.6. AT+ECCCH– Unsolicited result

11.6.1. Description

This URC is used to inform about CCCH Block status.

11.6.2. Format

Unsolicited result code
+ ECCCH: <status>

11.6.3. Field

<status>: integer type: Indicate the CCCH Block status

1 Bad Block

2 Good Block

Note

- URC is sent only when its activated through AT+ EFTCM = 1,1

11.7. AT+ESACCH– Unsolicited result

11.7.1. Description

This URC is used to inform about SACCH Block status.

11.7.2. Format

Unsolicited result code
+ESACCH: <status>

11.7.3. Field

<status>: integer type: Indicate the SACCH Block status

1 Bad Block

2 Good Block

Note

- URC is sent only when its activated through AT+ EFTCM = 2,1

11.8. AT+EPOF – Power OFF modem

11.8.1. Description

This command is used to power OFF the modem but not cut off the power. This command must be send before cutting off the power of the module in order to permit the detaching from network and to finish the memory saving operations. To completely shot down the module, cut off of the power is needed. To turn ON the module power cut off and power on cycle is necessary.

11.8.2. Format

Command	Possible response(s)
+EPOF	*OK*
+EPOF=?	*OK*

11.9. AT+ESIMS – check SIM Status

11.9.1. Description

The read command is only response the SIM inserted status.

The active command is used to trigger SIM reset procedure and response the SIM inserted status.

The execute command is used to enable/disable +ESIMS URC report

11.9.2. Format

Command	Possible response(s)
+ESIMS?	*+ESIMS: <SIM_INSERTED>* *OK*
+ESIMS	*+ESIMS: <SIM_INSERTED>* *OK*
+ESIMS=<mode>	*OK*

11.9.3. Field

Type	Short Name	Parameter / Comment
integer	SIM_INSERTED	*0= No SIM* *1= Detected*
integer	mode	*0= Disable +ESIMS unsolicited* *1= Enable +ESIMS unsolicited*

11.10. AT+ICCID – Read ICCID of SIM Card

11.10.1. Description

This command is used to read SIM card ICCID if SIM inserted. If SIM not inserted, return +CME

ERROR: 10

11.10.2. Format

Command	Possible response(s)
+ICCID	*<iccid>* *OK* *ERROR / +CME ERROR: 10*

11.10.3. Field

<iccid>: string type

11.11. AT+ETLLISTEN – Server listen service

11.11.1. Description

Server listen service.

11.11.2. Format

Command	Possible response(s)
+ETLLISTEN=<id>,<port>	*+ETLLISTEN:<socket id>* *ERROR*

11.11.3. Field

<id>:

Data account id, total 3 accounts was defined. Value range is 0-2.

A data account id is coupled with a PDP, and the PDP context id will be allocated auto.

<port>:

Listen port

Note:

Server will accept auto when connect coming.

11.12. AT+IFC – DTE-DCE local flow control

11.12.1. Description

The command sets the DTE-DCE local flow control.

11.12.2. Format

Command	Possible response(s)
+IFC=<DCE_by_DTE>,<DTE_by_DCE> +IFC= <DCE_by_DTE>	OK OK
+IFC?	+IFC=<DCE_by_DTE>,<DTE_by_DCE> OK
+IFC=?	+IFC=(0-2),(0-2) OK

11.12.3. Field

<DCE_by_DTE>: specifies the method to be used by DTE to control the flow of received data from DCE

0 None

1 Enable software flow control

2 Enable hardware flow control

<DTE_by_DCE>:>: specifies the method to be used by DCE to control the flow of received data from DTE

0 None

1 Enable software flow control

2 Enable hardware flow control

Note: if only <DCE_by_DTE> is set, without <DTE_by_DCE>, then <DTE_by_DCE> is set as <DCE_by_DTE>

Note: the parameters must be equal

Note: the parameters are not stored in NVM

Note: After set command, the flow control does not permit to see the final characters of the command and the OK return string is missed

12. GPRS TCPIP AT COMMAND

12.1. AT+EGDCONT – Define TCP/IP data account

12.1.1. Description

Define TCP/IP data account

12.1.2. Format

Command	Possible response(s)
+EGDCONT=<id>[,<PDP_type>],<APN>[,<proxy ip>,< proxy port>]	OK +CME ERROR: <err>
+EGDCONT?	+EGDCONT:<id>,<PDP_type>,<APN>,<proxy ip>,< proxy port>

12.1.3. Field

<id>:

Data account id, total 3 accounts was defined.

Value range is 0- (GPRS_MAX_PDP_SUPPORT-1) .

A data account id is coupled with a PDP, and the PDP context id will be allocated auto.

<PDP_type>:

(Packet Data Protocol type) a string parameter.

IP Internet Protocol

<APN>:

(Access Point Name) a string parameter which is a logical name that is used to select the GGSN or the external packet data network

<proxy ip>:

Proxy ip address. Some special APNs may need this content like “CMWAP”. If the value is null or omitted, means do not need this type.

<proxy port>:

Same with proxy ip.

NOTE

Example:

AT+EGDCONT = 0, ”IP”, ”CMNET”

AT+EGDCONT = 0, ””, ”CMNET”

AT+EGDCONT = 1, ”IP”, “CMWAP”, “10.0.0.172”, 80 //CMWAP need proxy ip and port

12.2. AT+ETCPIP – Activate / Deactivate PDP

12.2.1. Description

Activate or deactivate PDP.

12.2.2. Format

Command	Possible response(s)
+ETCPIP=<op>,<id>[,<user name>[,<passwd>[,<auth>]]]	OK +CME ERROR: <err>
+ETCPIP?	+ETCPIP:<id>,<state> +ETCPIP:<id>,<state> ……
+ETCPIP=?	

12.2.3. Field

<op>:

Operation mode:

0: deactivate PDP

1: activate PDP

<id>:

Data account id

<user name>:

string to specify “User Name”

<passwd>:

string to specify “Password”

<auth>:

a numeric parameter used to indicate authentication type. Default is PAP.

0: PAP

1: CHAP

<state>:

0: deactivated

1: activated

NOTE

When deactivate a PDP, only need op and id. Other parameters do not need.

For example, deactivate account 1 PDP: AT+ETCPIP=0, 1

12.3. AT+ETL –Socket operation

12.3.1. Description

Create/close/query socket.

12.3.2. Format

Command	Possible response(s)
+ETL=0,<socket id>	OK +CME ERROR: <err>
+ETL=1,<id>,<type>[,<dest ip>,<dest port>]	+ETL:<socket id> OK
+ETL=2,<id>	+ETL:<socket id>,<type>,<dest ip>,<dest port>
+ETL=?	

12.3.3. Field

<op>:

Operation mode:

0: close a socket with socket id

1: create a socket and return socket id

2: query all the socket info with data account id

<id>:

Data account id

<socket id>:

Socket id. When create a socket, if success will return this value

<type>:

0: TCP

1: UDP

<dest ip>:

string to specify “dest ip address”.

<dest port>:

dest port.

NOTE

1. When use AT+ETL=1,if id is CBM_WIFI_ACCT_ID(0x38),the possible response is +EWFSOCK:<socket id>
2. For UDP type, dest_ip and dest_port can be omitted;
3. For TCP type, dest_ip and dest_port must be specified.
4. When use AT+ETL=2,<id> query the socket info, will return all the create success socket info.

12.4. AT+ETLQ – Query the socket data traffic statistics

12.4.1. Description

Query the socket data transfer statistics.

12.4.2. Format

Command	Possible response(s)
+ETLQ=<socket id>	+ETLQ:<total send>,<total ack>,<total receive> +CME ERROR: <err>
+ETLQ=?	

12.4.3. Field

<socket id>:

Socket id. When create a socket, if success will return this value.

<total send>: Total send data byes in this socket

<total ack>: Total send data byes which get the dest ack in this socket

<total receive>: Total receive data byes in this socket

12.5. AT+EIPSEND – Send data by socket

12.5.1. Description

Send data by socket

12.5.2. Format

Command	Possible response(s)
+EIPSEND=<socket id>,<data>[,<dest ip>,<dest port>]	+EIPSEND: <ret> OK ERROR

12.5.3. Field

<socket id >:

Socket id. When create a socket, if success will return this value.

<data>: Hex format string. For example, if you want send ‘0x01, 0x22, 0xAB, 0xCD’ 4 bytes data, you need send string “0122ABCD”.

<dest ip>: String to specify “dest ip address”. Only UDP socket need.

<dest port>: Dest port. Only UDP socket need.

<ret>: The socket send success data length. If error, <ret> will be a negative number reply the socket error cause

NOTE

For TCP socket, send command should not add ip address and port.

send TCP data to socket 1:

AT+EIPSEND=1,”1122abcd” //data is 0x11, 0x22, 0xab, 0xcd

For UDP socket, if use AT+ETL to create a UDP socket and not with ip and port, send command should add ip address and port, or will return error. If create UDP socket command already with ip and port, the send command format is same with TCP.

send UDP data to socket 2 with ip and port:

AT+EIPSEND=2,”1122abcd”,”10.0.0.1”,80 //send 4 bytes data: 0x11, 0x22, 0xab, 0xcd

The max send data length is 512 bytes raw data.

12.6. AT+EIPRECV– Receive data from socket

12.6.1. Description

Receive data from socket.

12.6.2. Format

Command	Possible response(s)
+EIPRECV=<socket id>	+EIPRECV: <socket id>, <data string> OK ERROR

12.6.3. Field

<socket id >:

Socket id. When create a socket, if success will return this value.

NOTE

The max receive data length is 512 bytes raw data.

12.7. AT+EDNS – DNS service

12.7.1. Description

DNS service to get domain name with ip or get ip address with domain name.

12.7.2. Format

Command	Possible response(s)
+EDNS=<type>,<ip> +EDNS=<type>,<domain name>	+EDNS: <ip>,<domain name> OK ERROR

12.7.3. Field

<type >:

DNS service type.

0: get ip address with domain name

1: get domain name with ip

<ip>:

Ip address string

<domain name>:

Domain name string

Example:

//query IP address

AT+EDNS=0,”www.google.com”

+EDNS: “74.125.128.104”,”www.google.com”

OK

//query domain name

AT+EDNS=1, “74.125.128.104”

+EDNS: “74.125.128.104”,”www.google.com”

OK

12.8. AT+ETLLISTEN – Server listen service

12.8.1. Description

Server listen service.

12.8.2. Format

Command	Possible response(s)
+ETLLISTEN=<id>,<port>	+ETLLISTEN:<socket id> ERROR

12.8.3. Field

<id>:

Data account id, total 3 accounts was defined. Value range is 0-2.

A data account id is coupled with a PDP, and the PDP context id will be allocated auto.

<port>:

Listen port

NOTE

Server will accept auto when connect coming.

12.9. AT+ETLTS – Transparent transmission

12.9.1. Description

Transparent transmission service

12.9.2. Format

Command	Possible response(s)
+ETLTS=<socket id>[,<dest ip>,<dest port>]	If success, no reply ERROR

12.9.3. Field

<socket id >:

Socket id. When create a socket, if success will return this value.

<dest ip>:

String to specify “dest ip address”. Only UDP socket need.

<dest port>:

Dest port. Only UDP socket need.

NOTE

1. When enter transparent transmission mode, UART will be obtained by socket, AT cannot use it until escape from this mode. The escape string is “+++”.
2. For UDP socket, if use AT+ETL to create a UDP socket and not with ip and port, send command should add ip address and port, or will return error. If create UDP socket command already with ip and port, the send command format is same with TCP.

13. SERIAL PORT MULTIPLEXER

13.1. AT+CMUX – Serial Port Multiplexing

13.1.1. Description

The command is used to enable the Serial Port Multiplexing according to 3GPP TS 27.010.

The AT command sets parameters for the control channel.

If the parameters are not set, the system will use the default values.

The product can support 4 virtual ports.

13.1.2. Format

Command	Possible response(s)
+CMUX=?	+CMUX: <mode>, [<subset>], <port_speed>, <N1>, <T1>, <N2>, <T2>, <T3>[,<k>]
+CMUX?	+CMUX: (list of supported <mode>s), (list of supported <subset>s), (list of supported <port_speed>s), (list of supported <N1>s), (list of supported <T1>s), (list of supported <N2>s), (list of supported <T2>s), (list of supported <T3>s), (list of supported <k>s)
+CMUX=<mode>, [<subset>[, <port_speed>[, <N1>[, <T1>[, <N2>[, <T2>[,<T3>[,<k>]]]]]]]]	OK +CME ERROR: <err>

13.1.3. Field

Field

<mode>: integer

0 Basic Option

<subset>: integer

0 UIH frames

1 UI frames

<port_speed> integer (baud rate)

9600	9600 bit/s
19200	19200 bit/s
38400	38400 bit/s
57600	57600 bit/s
115200	115200 bit/s → Default
230400	230400 bit/s

406800 406800 bit/s
921600 921600 bit/s

<N1> integer (maximum frame size)
31 to 1540, where 512 is the default for Basic option

<T1> integer (acknowledgment timer in units of ten milliseconds)
1 to 255, where 10 is default (100ms)

<N2> integer (maximum number of re-transmissions)
0 not supported

<T2> integer (response timer for the multiplexer control channel in units of ten milliseconds)
2 to 255, where 30 is default (300ms)

<T3> integer (wake up response timer in seconds)
1 to 255, where 10 is default (10 s)

<k>: integer (window size, for Advanced operation with Error Recovery options)
0 not supported

13.1.4. Example

AT+CMUX=0,0,115200,512,10,3,30,10

NOTE

- the 3GPP DLC parameter negotiation (PN) command must be supported in the protocol.
- immediately after opening a DLCI with SABM, the tool/library has to send a PN to set CL1 (Convergence Layer Type 1)
- CMUX is not supporting Flow Control

14. AUDIO AT COMMAND

14.1. AT+ESAM – Set Audio Mode

14.1.1. Description

This Command is used to set audio mode.

For GE310-GNSS the required setting to enable the EAR path is 2

14.1.2. Format

Command	Possible response(s)
+ESAM=?	+EECHO: (list of supported <mode>) OK / ERROR
+ESAM =<mode>	OK / *ERROR*

14.1.3. Field

<mode >: integer

2 Enable EAR audio path

Example

AT+ESAM=?

+ESAM: (0,2)

OK

AT+ESAM=2 //Enable EAR audio path

OK

NOTE

At startup the EAR path is disabled

14.2. AT+EECHO – Echo Cancellation

14.2.1. Description

The command is used to activate or deactivate echo cancelation. This command should only sent to target before a call setup

14.2.2. Format

Command	Possible response(s)
+EECHO=?	+EECHO: (list of supported <status>) OK / ERROR
+EECHO?	+EECHO:<status> OK /ERROR
+EECHO =<status>	OK / *ERROR*

14.2.3. Field

<status >: integer

0 Deactivate echo cancellation

1 activate Echo cancellation

Example

AT+EECHO=?

+EECHO: (0,1)

OK

AT+EECHO? //Shows the current configuration

+EECHO: 1

OK

AT+EECHO=0 //Deactivate echo cancellation

OK

AT+EECHO?

+EECHO: 0 //Echo cancellation is deactivated

OK

AT+EECHO=1 //Activate echo cancellation

OK

NOTE

EECHO setting will be saved to NVRAM

14.3. AT+ ENOISE – Noise Cancellation

14.3.1. Description

The command is used to activate or deactivate noise cancelation. This command should only sent to target before a call setup

14.3.2. Format

Command	Possible response(s)
+ENOISE=?	+ENOISE:(list of supported <Receive>s), (list of supported <Transmit>s) OK /ERROR
+ENOISE?	+ENOISE:<Receive>,<Transmit> OK /ERROR
+ENOISE=<Receive>,<Transmit>	OK / *ERROR*

14.3.3. Field

<Receive>: integer

0 OFF

1 ON

<Transmit>: integer

0 OFF

1 ON

Example

AT+ENOISE=?

+ENOISE: (0-1),(0-1)

OK

AT+ENOISE? //Shows the current configuration

+ENOISE:1,1

OK

AT+ENOISE=0,0 // Disable uplink and downlink noise suppression

OK

AT+ENOISE=1,1 //Enable uplink and downlink noise suppression

OK

AT+ENOISE=0,1 //Enable uplink and disable downlink noise suppression

OK

NOTE

ENOISE setting will be saved to NVRAM

14.4. AT+ESST – Set Side Tone

14.4.1. Description

The command is used to set side tone.

14.4.2. Format

Command	Possible response(s)
+ESST=?	+ESST: (list of supported <level>s) OK /ERROR
+ESST?	+ESST: <level> OK /ERROR
+ESST=<level>	OK /ERROR

14.4.3. Field

< level >: integer

0-255: Side tone value (side tone gain from 0 to 255)

0 : disable Side tone

Example

AT+ESST=?

+EST: (0-255)

OK

AT+ESST? //Shows the current value

+EST: 8

OK

AT+ESST=240 //Set side tone gain to 240

OK

AT+ESST=0 //Disable side tone

OK

NOTE

Side tone level will not be saved to NVRAM

14.5. AT+ EARST– Reset audio setting to factory setting

14.5.1. Description

The command is used to recover audio setting to factory setting.

14.5.2. Format

Command	Possible response(s)
+EARST	OK /ERROR

14.5.3. Field

Example

AT+ EARST

OK //Recover audio parameters to factory setting. Note: Setting Recovery won't be valid unless reset modem

14.6. AT+ESLT – Set Audio Gain Value

14.6.1. Description

This Command is used to set audio sound gain value.

14.6.2. Format

Command	Response
AT+ ESLT= <type>,<gain>	*OK*
AT+ ESLT =?	*Show if the command is supported*

14.6.3. Field Format

Type	Short name	Long name	Parameter / Comment	
Integer	Type	Audio type	call tone	0
			keypad tone	1
			microphone	2
			<reserved>	3
			speech sound	4
			side tone	5
			MP3, Wave, melody, Imelody, midi	6
Integer	Gain	Gain value	0-255	

14.6.4. Example

set speech sound gain value 150.

AT+ESLT = 4, 150

OK

14.7. AT+EADP – Set / Get Audio Profile

14.7.1. Description

This Command is used to set and get audio profile command.

14.7.2. Format

Command	Response
AT+ EADP = <op>,<mode>,<audio type>,<level>,[<gain>]	*+EADP: (0,1),(0-2),(0-4),(0-6),(0-255)* *OK*
AT+ EADP =?	*Show if the command is supported*

14.7.3. Field Format

Type	Short name	Long name	Parameter / Comment	
Integer	op	operation	Get	0
			Set	1
Integer	mode	audio mode	Normal mode	0
			Headset mode	1
			Loud speaker mode	2
Integer	type	audio type	Melody	0
			Keytone	1
			Speech	2
			mic	3
			sidetone	4
	level	volume level	0-6 (when type = mic or sidetone, volume level = 0)	
Integer	gain	gain value		

14.7.4. Example

Get Audio mode with Normal Mode , Melody type, volume level is 0. The return value with gain 40 at+EADP=0,0,0,0

OK

14.8. AT+EPLPK – enable/disable PCM loopback

14.8.1. Description

The command is used to enable/disable PCM loopback.

14.8.2. Format

Command	Response
+EPLPK=<LPK>	OK ERROR
+EPLPK?	+IEPLPK=<PCM Loopback state> OK ERROR

+EPLPK=?	List of supported <PCM loopback configuration> OK ERROR

15. TELIT PROPRIETARY AT COMMANDS

15.1. AT#ADC – Read Analog/Digital Converter input

15.1.1. Description

The command is used to read the ADC values.

15.1.2. Format

Command	Possible response(s)
#ADC=?	Test command reports the supported range of values of the command parameters <adc>, <mode> and <dir>. OK / ERROR
#ADC?	Read command reports all pins voltage, converted by ADC, in the format: #ADC: <value>[<CR><LF>#ADC: <value>[…]] OK /ERROR
#ADC = [<adc>,<mode>[,<dir>]]	OK / *ERROR*

15.1.3. Field

<adc> - index of pin

For the number of available ADCs see HW User Guide

<mode> - required action

2 - query ADC value

<dir> - direction; its interpretation is currently not implemented

0 - no effect.

Note: The command returns the last valid measure.

15.2. AT#DAC – Digital/Analog Converter Control

15.2.1. Description

The command is used to enable/disable DAC output.

15.2.2. Format

Command	Possible response(s)
#DAC=?	Test command reports the range for the parameters <enable> and <value>. OK / ERROR
#DAC?	Read command reports whether the DAC_OUT pin is currently enabled or not, along with the integrated output voltage scale factor, in the format: #DAC: <enable>,<value> OK /ERROR
#DAC = [<enable>[,<value>]]	OK / *ERROR*

15.2.3. Field

<enable> - enables/disables DAC output.

0 - disables pin; it is in high impedance status (factory default)

1 - enables pin; the corresponding output is driven

<value> - scale factor of the integrated output voltage; it must be present if <enable>=1

0..1023 (10-bit resolution)

Note: integrated output voltage = MAX_VOLTAGE / value

Example

Enable the DAC out and set its integrated output to the 50% of the max value:

AT#DAC=1,512

OK

Disable the DAC out:

AT#DAC=0

OK

15.3. AT#GPIO – General Purpose Input/Output Pin Control

15.3.1. Description

The command is used to control the module's GPIOs

15.3.2. Format

Command	Possible response(s)
#GPIO=?	Test command reports the supported range of values of the command parameters <pin>, <mode> and <dir> OK / ERROR
#GPIO?	Read command reports the read direction and value of all GPIO pins, in the format: #GPIO: <dir>,<stat>[<CR><LF>#GPIO: <dir>,<stat>[…]] OK /ERROR
#GPIO=[<pin>,<mode>[,<dir>]]	OK / *ERROR*

15.3.3. Field

Execution command sets the value of the general-purpose output pin GPIO<pin> according to <dir> and <mode> parameter.

Not all configurations for the three parameters are valid.

Parameters:

<pin> - GPIO pin number; supported range is from 1 to 6.

<mode> - its meaning depends on <dir> setting:

0 - no meaning if <dir>=0 - INPUT

- output pin cleared to 0 (Low) if <dir>=1 - OUTPUT

1 – Disable Pull-Up or Pull-Down if <dir>=0 - INPUT

- output pin set to 1 (High) if <dir>=1 - OUTPUT

2 - Reports the read value from the input pin if <dir>=0 - INPUT

- Reports the read value from the input pin if <dir>=1 - OUTPUT

3 - Enable Pull-Up if <dir>=0 - INPUT

- no meaning if <dir>=1 - OUTPUT

4 - Enable Pull-Down if <dir>=0 - INPUT

- no meaning if <dir>=1 - OUTPUT

<dir> - GPIO pin direction

0 - pin direction is INPUT

1 - pin direction is OUTPUT

Note: when <mode>=2 (and <dir> is omitted) the command reports the direction and value of pin GPIO<pin> in the format:

#GPIO: <dir>,<stat>

where:

<dir> - current direction setting for the GPIO<pin>

<stat>

- logic value read from pin GPIO<pin> in the case the pin <dir> is set to input;
- logic value present in output of the pin GPIO<pin> in the case the pin <dir> is currently set to output;

15.4. AT#SLED – STAT_LED GPIO Setting

15.4.1. Description

The command is used to control the module's GPIOs

15.4.2. Format

Command	Possible response(s)
AT#SLED=?	Test command returns the range of available values for parameters: <mode>,<on_duration> and <off_duration>. OK / ERROR
AT#SLED?	Read command returns the STAT_LED GPIO current setting, in the format: #SLED: <mode>,<on_duration>,<off_duration> OK /ERROR
AT#SLED=<mode>[,<on_duration>[,<off_duration>]]	OK / *ERROR*

15.4.3. Field

Set command sets the behaviour of the STAT_LED GPIO

Parameters:

<mode> - defines how the STAT_LED GPIO is handled

0 - GPIO tied Low (default

1 - GPIO tied High

2 - GPIO handled by Module Software (factory default) with the following timings:

• not registered : blinking 0.5s on and 0.5s off

• registered in idle: blinking 0.3s on and 2.7s off

3 - GPIO is turned on and off alternatively, with period defined by the sum

<on_duration> + <off_duration>

<on_duration> - duration of period in which STAT_LED GPIO is tied High while <mode>=3

1..100 - in tenth of seconds (default is 10)

<off_duration> - duration of period in which STAT_LED GPIO is tied Low while

<mode>=3

1..100 - in tenth of seconds (default is 10)

15.5. AT#MONI – Cells Information

15.5.1. Description

AT#MONI is both a set and an execution command.

Set command sets one cell out of seven, in the neighbour list of the serving cell including it, from which extract GSM-related information.

Execution command reports GSM-related information for the selected cell.

Test command reports the maximum number of cells, in the neighbour list of the serving cell excluding it, from which we can extract GSM related information, along with the ordinal number of the current selected cell.

15.5.2. Format

Command	Possible response(s)
#MONI=<cell number>	OK ERROR

#MONI	When extracting data for the serving cell: #MONI: <plmn> BSIC:<bsic> RxQual:<qual> LAC:<lac> Id:<cid> ARFCN:<arfcn> PWR:<dbm>dbm TA:<ta> OK When extracting data for an adjacent cell, the format is: #MONI: Adj Cell <n> LAC:<lac> Id:<cid> ARFCN:<arfcn> PWR:<dbm>dbm OK When AT#MONI=7 is the last setting entered, the execution command produces a table-like formatted output, as follows: a. First row reports the identifying name of the 'columns': #MONI: Cell BSIC LAC CellId ARFCN Power C1 C2 TA RxQual PLMN b. Second row reports a complete set of GSM-related information for the serving cell: #MONI: S <bsic> <lac> <cid> <arfcn> <dbm>dbm <c1> <c2> <ta> <qual> <plmn> c. 3rd to 8th rows report a reduced set of GSM-related information for the cells in the neighbours: #MONI: N<cell number> <bsic> <lac> <cid> <arfcn> <dbm>dbm <c1> <c2> OK +CME ERROR: <error>
#MONI=?	#MONI: (<max cell number>,<cell number>) OK

15.5.3. Field

<cell number>:

The number of the cell

0-6: Ordinal number of the cell in the neighbour list (0 – default, serving cell)

7: Special request to obtain GSM-related information from the whole set of seven cells.

<max cell number>

Maximum number of cells, in the neighbour list of the serving cell and excluding it, from which we can extract GSM-related information. This value is always 6.

<plmn>

Name of network operator if it is known, otherwise the globally unique PLMN code which consist of MCC (Mobile Country Code) and MNC (Mobile Network Code

<bsic>

Base station identity code, FF if value is unknown or not detectable.

<lac>

Localization area code in HEX format, FFFF if value is unknown or not detectable.

<cid>

Cell identifier in HEX format, 0000 if value is unknown or not detectable.

<arfcn>

Assigned radio channel.

<dbm>

Received signal strength in dBm.

<c1>

C1 reselection parameter, -1 if value is unknown or not detectable.

<c2>

C2 reselection parameter, -1 if value is unknown or not detectable.

<ta>

Timing advance

0-63: number of steps each representing an advance of one-bit period.

-1: value is unknown or not detectable.

<qual>

Quality of reception

0-7: as RXQUAL values in the table in TS 45.008 subclause 8.2.4

99: value is unknown or not detectable.

NOTE:

1. C1 reselection parameter <c1> and C2 reselection parameter <c2> are always -1 and they are kept only for backwards compatibility.

2. The timing advance <ta> and quality <qual> parameters are meaningful only during voice calls.

16. GNSS AT COMMAND

The following commands can be used on the MAIN UART of GE310 module and when the GNSS receiver is connected in Hosted configuration.

16.1. AT+EGPSC – Power on/off GPS

16.1.1. Description

Power on/off GNSS receiver

16.1.2. Format

Command	Possible response(s)
+EGPSC=<state>	OK +CME ERROR: <err>
+EGPSC?	+EGPSP:<state> +CME ERROR: <err>
+EGPSC=?	+EGPSC:(list of supported <state>s)

16.1.3. Field

<state>:

0 power off GPS

1 power on GPS

When the command is executed the module will start to present the NMEA sentences on the Main UART.

The supported speed is 115200 bps; the standard rate is 1 sec per update.

16.2. AT+EGPSS – Send PMTK Command

16.2.1. Description

Send PMTK command to GPS receiver

16.2.2. Format

Command	Possible response(s)
+EGPSS="<pmtk>"	OK +CME ERROR: <err>
+EGPSS?	Not supported
+EGPSS=?	Not supported

16.2.3. Field

<pmtk>:

PMTK command string without '$' character before PMTK string

16.3. AT+EGPSEPO – Set EPO Parameter

16.3.1. Description

Open/Close EPO downloading and aiding features.

Set EPO data account

16.3.2. Format

Command	Possible response(s)
+EGPSEPO=<status>,<data account>	OK +CME ERROR: <err>
+EGPSEPO?	Not supported
+EGPSEPO=?	Not supported

16.3.3. Field

<status>:

1: Enable EPO

0: Disable EPO

<data account>

0-3 network data account when you send " AT+EGDCONT=0,"IP","cmnet"

NOTE:

When you set EPO enable you must set the correct network data account.

EPO uses TCPIP so you need to set the network data account before activating EPO:

AT+EGDCONT=0,"IP","cmnet"

Enable GPS Time Sync and EPO aiding, set the correct network data account:

AT+EGPSTS=1,0

AT+EGPSEPO=1,0

16.4. AT+EGPSTS – Set GPS Time Sync Parameter

16.4.1. Description

Open/Close GPS time sync and aiding.

Set time sync data account

16.4.2. Format

Command	Possible response(s)
+EGPSTS=<status>,<data account>	OK +CME ERROR: <err>
+EGPSTS?	Not supported
+EGPSTS=?	Not supported

16.4.3. Field

<status>:

1: Enable GPS time sync

0: Disable GPS time sync

<data account>

0-3 network data account when you send " AT+EGDCONT=0,"IP","cmnet"

NOTE:

When you set EPO enable you must set the correct network data account.

EPO uses TCPIP so you need to set the network data account before activating EPO:

AT+EGDCONT=0,"IP","cmnet"

Enable GPS Time Sync and EPO aiding, set the correct network data account:

AT+EGPSTS=1,0

AT+EGPSEPO=1,0

17. PMTK COMMANDS

The PMTK are a set of proprietary commands usable to control the GNSS receiver.

17.1. PMTK Protocol Format

The PMTK is using a proprietary protocol with this structure:

$	PMTK	<command>	<Data Field>	*	<Checksum>	CR	LF

The content is the following:

Item	Type	Length	Description
$	char	1	Preamble Character; Always “$”
PMTK	string	4	String ID; always “PMTK”
<command>	string	3	Numeric string (from 000 to 999) identifying the command
<Data Field>	char	variable	The Data field has a variable length because depending on the specific command. A comma “,” character has to be used between each data field.
*****	char	1	End of Data Field Marker; Always “*”
<Checksum>	char	2	Two Bytes checksum of the data between $ and * The checksum is an XOR of all the bytes between the $ and the * (not including the delimiters themselves) and written in hexadecimal.
CR	Char	1	Carriage Return
LF	Char	1	Line Feed

17.2. PMTK Commands

17.2.1. PMTK000 – Test

17.2.1.1. Description

Test Command

17.2.1.2. Example

$PMTK000*32<CR><LF>

17.2.2. PMTK001 – PMTK Acknowledge

17.2.2.1. Description

Acknowledge to a PMTK command

17.2.2.2. Format

PMTK001,cmd, flag

17.2.2.3. Fields

<cmd>

The command / packet type the acknowledge responds.

<flag>

0 = Invalid command/packet

1 = Unsupported Command / Packet Type

2 = Valid command / packet but action failed

3 = Valid Command/Packet and action correctly completed

17.2.2.4. Example

$PMTK001,604,3*32<CR><LF>

17.2.3. PMTK101 – HOT Start

17.2.3.1. Description

The command forces the GNSS receiver to perform an Hot Start.

The module will use all the data already stored in NVM.

17.2.3.2. Format

PMTK101

17.2.3.3. Fields

None

17.2.3.4. Example

$PMTK101*32<CR><LF>

17.2.4. PMTK102 – WARM Start

17.2.4.1. Description

The command forces the GNSS receiver to perform an Warm Start.

The module will restart without re-using the stored Ephemeris..

17.2.4.2. Format

PMTK102

17.2.4.3. Fields

None

17.2.4.4. Example

$PMTK102*31<CR><LF>

17.2.5. PMTK103 – COLD Start

17.2.5.1. Description

The command forces the GNSS receiver to perform an Cold Start.

The module will restart without using the stored Position, Almanacs and Ephemeris..

17.2.5.2. Format

PMTK103

17.2.5.3. Fields

None

17.2.5.4. Example

$PMTK103*30<CR><LF>

17.2.6. PMTK104 – FULL COLD Start

17.2.6.1. Description

The command forces the GNSS receiver to perform an Cold Start cleaning also the system/user configurations to the factory defaults.

17.2.6.2. Format

PMTK104

17.2.6.3. Fields

None

17.2.6.4. Example

$PMTK104*37<CR><LF>

17.2.7. PMTK161 – StandBy Mode

17.2.7.1. Description

Enter Stand By Mode

17.2.7.2. Format

PMTK161, <mode>

17.2.7.3. Fields

<mode>

0 = Stop Mode

1 = Sleep Mode

17.2.7.4. Example

$PMTK161,0*28<CR><LF>

17.2.8. PMTK223 – Set Period Mode Parameters

17.2.8.1. Description

The command permits to set the Period Mode parameters and need to be used followed by PMTK225.

17.2.8.2. Format

PMTK223,<SV>,<SNR>,<Extension Threshold>,<Extension Gap>

17.2.8.3. Fields

<SV>

number of satellites rise

Range: 1 to 4 (default value is 1)

<SNR>

Minimum SNR level of the rising satellites

25 to 30 (default value is 30)

< Extension Threshold >

Value in msec of Extension Threshold

40000 to 180000 ms (default value is 180000 ms)

< Extension Gap >

Value in msec of Extension Gap.

The Extension Gap is the limitation between neighbor DEE

0 to 3600000 ms (default value is 60000 ms)

17.2.8.4. Example

Example of settings in Period Mode

$PMTK225,0*2B<CR><LF> → set to Normal Mode

$PMTK223,2,25,1,180000,60000*26<CR><LF> → defines the parameters

$PMTK225,2,3000,12000,18000,72000*15<CR><LF> → enable the mode

Description of setting command $PMTK223,2,25,1,180000,60000:

- Period Mode active
- Two more satellites rise to be used
- SNR of new riding satellites is over 25db

If the above three conditions are satisfied the GNSS receiver will extend Run Time to the defined Extension Threshold (180000 ms) for decoding Ephemeris.

The last parameter (Extension Gap defined as 60000 ms) means that the interval of run time extension will be more than 60 seconds.

17.2.9. PMTK225 – Set Periodic Power Saving Mode

17.2.9.1. Description

The command permits to define the periodic Power Saving Mode settings.

In Normal mode the receiver measures and calculates positions.

In Sleep mode the receiver enters in one of the available power saving modes:

- Periodic StandBy Mode
- Periodic Backup Mode

17.2.9.2. Format

PMTK225,<Type>,<Run Time>,<Sleep Time>,<Second Run Time>, <Second Sleep Time>

17.2.9.3. Fields

<Type>

Defines the power saving operation mode.

0 = Normal Mode

1 = Periodic Backup Mode

2 = Periodic StandBy Mode

8 = AlwaysLocate™ StandBy Mode

9 = AlwaysLocate™ Backup Mode

<Run Time>

Duration (in ms) to fix for (or to attempt to fix for) before switching from Normal Mode to Power Saving Mode.

0 = Disabled

>= 1000 Enabled (range 1000 to 518400000)

<Sleep Time>

Interval (in ms) to exit from Power Saving and start Normal Mode to get a new Position Fix.

Accepted Values: from 1000 to 518400000

<Second Run Time>

Duration (in ms) to fix for (or to attempt to fix for) before switching from Normal Mode to Power Saving Mode.

0 = Disabled

>= 1000 Enabled (range 1000 to 518400000)

Note the <Second Run Time> should larger than <Run Time> when non-zero value.

<Second Sleep Time>
Interval (in ms) to exit from Power Saving and start Normal Mode to get a new Position Fix.
Accepted Values: from 1000 to 518400000

17.2.9.4. Example

How to enable a Periodic Backup Mode
$PMTK225,0*2B<CR><LF> → set to Normal Mode
$PMTK223,1,25,180000,60000*38<CR><LF> → defines the parameters
$PMTK225,1,3000,12000,18000,72000*16<CR><LF> → enable the mode

How to enable a Periodic StandBy Mode
$PMTK225,0*2B<CR><LF> → set to Normal Mode
$PMTK223,2,25,1,180000,60000*26<CR><LF> → defines the parameters
$PMTK225,2,3000,12000,18000,72000*15<CR><LF> → enable the mode

How to enable the AlwaysLocate™ StandBy Mode
$PMTK225,0*2B<CR><LF> → set to Normal Mode
$PMTK225,8*23<CR><LF> → enable the mode

How to enable the AlwaysLocate™ BackUp Mode
$PMTK225,0*2B<CR><LF> → set to Normal Mode
$PMTK225,9*22<CR><LF> → enable the mode

17.2.10. PMTK262 – Set GLP Power Saving Mode

17.2.10.1. Description

The command enables the GLP Power saving Mode (1Hz positioning)

17.2.10.2. Format

PMTK262,<enabled>

17.2.10.3. Fields

<enabled>

0 =disabled

3 = enabled

17.2.10.4. Example

$PMTK262,3*2B<CR><LF>

17.2.11. PMTK286 – Interference Cancellation Enable

17.2.11.1. Description

The command enables the Interference Cancellation function.

17.2.11.2. Format

PMTK286,<enabled>

17.2.11.3. Fields

<enabled>

1 = enabled

0 =disabled

17.2.11.4. Example

$PMTK286,1*23<CR><LF>

17.2.12. PMTK301 – Set DGPS Mode

17.2.12.1. Description

The command defines the DGPS data source mode

17.2.12.2. Format

PMTK301,<mode>

17.2.12.3. Fields

<mode>

0 = disabled

2 = SBAS (including WAAS/EGNOSGAGAN/MSAS)

17.2.12.4. Example

$PMTK301,2*2E<CR><LF>

17.2.13. PMTK313 – Set SBAS Enabled

17.2.13.1. Description

The command set the GNSS receiver to include also the SBAS satellites in its position calculation.

17.2.13.2. Format

PMTK313,<enabled>

17.2.13.3. Fields

<enabled>

1 = enabled

0 =disabled

17.2.13.4. Example

$PMTK313,1*2E<CR><LF>

17.2.14. PMTK314 – Set NMEA Output

17.2.14.1. Description

The command permits to define which NMEA sentences are reported by the receiver and their output frequency.

Please refer to the next chapters for the details on the NMEA sentences.

17.2.14.2. Format

PMTK314,<GLL>,<RMC>,<VTG>,<GGA>,<GSA>,<GSV>,0,0,0,0,0,0,0,0,0,0,<ZDA>,0

17.2.14.3. Fields

<GLL>,<RMC>,<VTG>,<GGA>,<GSA>,<GSV> and <ZDA>

Each field represent an NMEA sentence and the possible configurations are the following:

0 = disabled or not supported sentence

1 = Output once every one position Fix

2 = Output once every two position Fixes

3 = Output once every three position Fixes

4 = Output once every four position Fixes

5 = Output once every five position Fixes

17.2.14.4. Example

$PMTK314,0,1,1,1,1,1,0,0,0,0,0,0,0,0,0,0,1,0*28<CR><LF>

This command enables all the sentences with 1 position fix frequency except the GLL that is set as disabled.

The output rates for all the NMEA sentences may be restored to their default values by sending the command: $PMTK314,-1*04<CR><LF>

17.2.15. PMTK352 – Enable QZSS

17.2.15.1. Description

Since QZSS is a regional positioning service, this command allows the user to enable/disable the function.

17.2.15.2. Format

PMTK352,<disable>

17.2.15.3. Fields

<disable>

0 = enabled

1 = disable

17.2.15.4. Example

$PMTK352,0*2A<CR><LF> : enable QZSS function

$PMTK352,1*2B<CR><LF> : disable QZSS function

17.2.16. PMTK353 – Set GNSS Search Mode

17.2.16.1. Description

The command permits to select the preferred satellite system.

17.2.16.2. Format

PMTK353,<GPS>, <GLONASS>,<GALILEO>,<GALILEO FULL>,<BEIDOU>

17.2.16.3. Fields

<GPS>, <GLONASS>,<GALILEO>,<GALILEO FULL>,<BEIDOU>

1 = enabled

0 =disabled

17.2.16.4. Example

$PMTK353,0,1,0,0,0,*2A<CR><LF> : Search GLONASS satellites only

$PMTK353,1,0,0,0,0,*2A<CR><LF> : Search GLONASS satellites only

$PMTK353,1,1,0,0,0,*2B<CR><LF> : Search GPS and GLONASS satellites

$PMTK353,1,1,1,0,0,*2A<CR><LF> : Search GPS, GLONASS and Galileo satellites

$PMTK353,0,0,0,0,1,*2A<CR><LF> : Search BEIDOU satellites only

$PMTK353,1,0,0,0,1,*2B<CR><LF> : Search GPS and BEIDOU satellites

NOTE:

The GLONASS only, BEIDOU only and GALILEO only modes are for testing purpose.

Please use GPS + GLONASS or GPS + BEIDOU in real application configuration.

GLONASS and BEIDOU cannot be enabled at the same time

17.2.17. PMTK386 – Set Static Navigation Threshold

17.2.17.1. Description

Set the speed threshold for static navigation. If the actual speed is below the threshold, output position will keep the same and output speed will be zero. If the threshold value is set to 0 the function is disabled

17.2.17.2. Format

PMTK386,<speed_threshold>

17.2.17.3. Fields

<speed_threshold>

0 = disabled

0.1 to 2.0 = enabled with the set threshold in ms

17.2.17.4. Example

$PMTK386,0.4*39<CR><LF> : enable Static navigation with a threshold of 0.4mS

18. NMEA OUTPUT MESSAGES

- The GE310-GNSS Message protocol is NMEA-0183.
- Default: GPS and GLONASS constellations are enabled
- Default fix rate: 1 Hz. Maximum rate is 10 Hz.
- Multiple GSA and GSV messages may be output on each cycle.

18.1. Available NMEA Messages

The following messages are available by default when the GNSS receiver is activated:

Message ID	Description
RMC	GNSS Recommended minimum navigation data
GGA	GNSS position fix data
GSA	GNSS Dilution of Precision (DOP) and active satellites
GSV	GNSS satellites in view.

The following messages can be enabled by command (ref to PMTK314):

Message ID	Description
GLL	Geographic Position – Latitude & Longitude
VTG	Course Over Ground & Ground Speed
ZDA	Time & Date

The following table shows the Talker IDs used:

Talker ID	Constellation
BD	BeiDou
GA	Galileo
GL	GLONASS
GP	GPS
QZ	QZSS

NOTE:
Some sentences may exceed the NMEA length limitation of 80 characters.

19. BLUETOOTH CONNECTION MANAGER AT COMMAND

19.1. AT+EBTPWR – Power on/off BT

19.1.1. Description

The command is used to power on or off BT. The power on command could only be sent when BT is power off. And the power off command could only be sent when BT is power on.

19.1.2. Format

Command	Possible response(s)
+EBTPWR=?	+EBTPWR: （0-1）/ERROR
+EBTPWR=<op>	OK /*ERROR*

19.1.3. Field

<op>: integer

0 power on

1 power off

Example

AT+EBTPWR=0 //power on BT

OK

19.2. AT+EBTNAME – Read/Write BT device local name

19.2.1. Description

The command is used to read or write BT device local name. The max invalid length of the device name is 54.

19.2.2. Format

Command	Possible response(s)	Description
+EBTNAME=?	OK/ERROR	Show if the command is supported
+EBTNAME?	+EBTNAME:<device name> OK /ERROR	Read BT local name
+EBTNAME=<device name>	OK / *ERROR*	Write BT local name

19.2.4. Field

< device name >: BT name string ,no need use “ ” double quotes.

Example

AT+EBTNAME?//Read BT local name

+EBTNAME：mydevice

OK

AT+EBTNAME=mydevice1 // write BT local name

OK

19.3. AT+EBTADDR – Read/Write BT device local address

19.3.1. Description

The command is used to read or write BT device local address. This CMD should only sent to Target when BT is power off.

19.3.2. Format

Command	Possible response(s)	Description
+EBTADDR=?	OK/ERROR	Show if the command is supported
+EBTADDR?	+EBTADDR:<address > OK /ERROR	Read BT address
+EBTADDR=<address>	OK / *ERROR*	Write BT address

19.3.3. Field

<address>: BT addr string ,no need use “ ” double quotes， length should be 12 characters

Example：

1234565b0102

It means ：LAP is 0x123456, UAP is 0x5b, NAP is 0x0102

Example

AT+EBTADDR?//Read BT local addr

+EBTADDR：1234565b0102

OK

AT+EBTADDR=1234565b0102 // write BT local addr

OK

19.4. AT+EBTINQ – Inquiry BT devices

19.4.1. Description

The command is used to inquiry BT devices.

19.4.2. Format

Command	Possible response(s)	Description
+EBTINQ=?	+EBTINQ：（0-255），（0-255）（0-255）（0-4294967295）OK/ERROR	Show if the command is supported
+EBTINQ?	+EBTADDR: <interval>,<polltime>,<device number>,<Cod> OK /ERROR	Get inquiry parameters
+EBTINQ=<interval>,<polltime>,<device number>,<Cod>	OK / *ERROR*	Set inquiry parameters
+EBTINQ	OK/ERROR	Start to inquiry

19.4.3. Field

<interval >: integer , should greater than poll time. （Note： Current the parameter does not work， because of BT stack already have itself timer ）

<polltime>: integer , the max seconds number for inquiring.

<device number>: integer

<Cod>: integer (ref: https://www.bluetooth.org/en-us/specification/assigned-numbers/baseband）

Example

AT+EBTINQ=60,10,5,16720412 // Cod: 16720412 = 0xFF221C means **Major Service Class ALL & Major Device Class Phone ALL**

OK

AT+EBTINQ?

+EBTINQ:60,10,5, 16720412

OK

AT+EBTINQ

OK

+EBTINQ:1234565b0101 ,devicename1

+EBTINQ:1234565b0102 ,devicename2

+EBTINQ:1234565b0103 ,devicename3

+EBTINQ:1234565b0104 ,devicename4

+EBTINQ:1234565b0105 ,devicename5

19.5. AT+EBTINQC – Cancel inquiry BT devices

19.5.1. Description

The command is used to cancel inquiry BT devices. Should be sent only when it's inquiring.

19.5.2. Format

Command	Possible response(s)	Description
+EBTINQC=?	OK/ERROR	Show if the command is supported
+EBTINQC	OK /ERROR	cancel inquiry BT devices

Example

AT+EBTINQC // cancel inquiry BT devices

OK

19.6. AT+EBTVISB – Set BT visible

19.6.1. Description

The command is used to set BT visible .

19.6.2. Format

Command	Possible response(s)	Description
+EBTVISB=?	+EBTVISB：（0-1）,(0-255) OK/ERROR	Show if the command is supported
+EBTVISB=<n>,[<time>]	OK / *ERROR*	Set BT visible

19.6.3. Field

<n>: integer

0 invisible

1 visible

<times>:integer , visible time ,seconds (0~255)

0 visible forever

1~255 visible time in seconds

Example

AT+EBTVISB=0//invisible

OK

AT+EBTVISB=1,0 // visible forever

OK

AT+EBTVISB=1,60 //visible 60s

OK

AT+EBTNAME=1234565b0102 // write BT local addr

OK

19.7. AT+EBTRNAME – Read remote BT device name

19.7.1. Description

The command is used to read remote BT device name.

19.7.2. Format

Command	Possible response(s)	Description
+EBTRNAME=?	OK/ERROR	Show if the command is supported
+EBTRNAME=<address>	+EBTRNAME:<devicenmae> OK /ERROR	remote BT device name

19.7.3. Field

< device name >: BT name string ,no need use “ ” double quotes.

<address>: BT addr string ,no need use “ ” double quotes， length should be 12 characters

Example：

1234565b0102

It means ：LAP is 0x123456, UAP is 0x5b, NAP is 0x0102

Example

AT+EBTRNAME=1234565B0102 //Read remote BT device name

+EBTRNAME：Yourdevidename1

OK

19.8. AT+EBTPAIR – PAIR BT device

19.8.1. Description

The command is used to pair BT device.

19.8.2. Format

Command	Possible response(s)	Description
+EBTPAIR=?	+EBTPAIR: ,(0-255) OK/ERROR	Show if the command is supported
+EBTPAIR=<address>，<timeout>	+EBTPAIR:<address>,<name>,<enable 16digitspin>[,<password>] OK / *ERROR*	pair BT device

19.8.3. Field

<address>: BT addr string ,no need use “ ” double quotes， length should be 12 characters

Example：

1234565b0102

It means ：LAP is 0x123456, UAP is 0x5b, NAP is 0x0102

<timeout>: integer , time in seconds for pairing timeout.

The value should between 1 and 20. 0 means 20s. Value larger than 20 will be treat as 20.

<name>：BT name string ,no need use “ ” double quotes.

< enable 16digitspin>: integer

0 not enable the 16digits pin

1 enable the 16 digits pin

<password>:String

Example

AT+EBTPAIR=1234565b0102,60 //pair device ,and timeout is 60s

+EBTPAIR：1234565b0102,devicename,0,123456

OK

19.9. AT+EBTPAIRCNF – Allow or disallow BT pair

19.9.1. Description

The command is used to allow or disallow BT pair.

19.9.2. Format

Command	Possible response(s)	Description
+EBTPAIRCNF=?	+EBTPAIRCNF: (0-1), OK/ERROR	Show if the command is supported
+EBTPAIRCNF=<n>[,<password>]	OK / *ERROR*	allow or disallow BT pair

19.9.3. Field

<n>: integer

0 disallow BT pair

1 allow BT pair

<password>: string ， need use “ ” double quotes

Example

AT+EBTPAIRCNF=1,”123456” // allow pair and password is 123456

OK

19.10. AT+EBTRP – Read remote BT device support profiles

19.10.1. Description

The command is used to read remote BT device support profiles which we support also.

19.10.2. Format

Command	Possible response(s)	Description
+EBTRP=?	OK/ERROR	Show if the command is supported
+EBTRP=<address>	+EBTRP:<profile_bitmap> OK / *ERROR*	read remote BT device support profiles

19.10.3. Field

<address>: BT addr string ,no need use “ ” double quotes， length should be 12 characters

Example：

1234565b0102

It means ： LAP is 0x123456, UAP is 0x5b, NAP is 0x0102

<profile_bitmap> integer ,uint64 ,one bit is a profile support or not. 0 :not support ,1 supported

Example : profile_bitmap value is 1 ,

0x0000000000000001 means profile 1 supported ,others not supported.

bit 1 ： SPP Profile, UUID = 0x1101.

bit 2 – bit 64 ： All reserved.

Example

AT+EBTRP=1234565b0102 // read remote BT support profiles

+EBTRP：1

OK

19.11. AT+EBTSENM – Read/Write security mode, encryption mode

19.11.1. Description

The command is used to read or write BT security and encryption mode. Currently we only support security mode setting. Note that the value of queried value of security mode is decide by BT Stack, may not be the same with setting value.)

19.11.2. Format

Command	Possible response(s)	Description
+EBTSENM=?	+EBTSENM：（0-4），（0-2） OK/ERROR	Show if the command is supported
+EBTSENM?	+EBTSENM:<sec-mod>，<enc-mod> OK /ERROR	Read security mode and encryption mode
+EBTSENM=<sec-mod>，<enc-mod>	OK / *ERROR*	Write security mode and encryption mode

19.11.3. Field

< sec-mod >: integer

0- Sec_mode0_off

1- Sec_mode1_non-secure

2- Sec_mode2_service

3- Sec_mode3_link

4- Sec_mod_unknown

< enc-mod >

0- hci_enc_mode_off

1- hci_enc_mode_pt_to_pt

2- hci_enc_mode_pt_to_pt_and_bcast

Example

AT+EBTSENM=0,0 // write mode

OK

AT+EBTSENM?//read mode

+EBTSENM：0,0

OK

19.12. AT+EBTOPAD – Get device list

19.12.1. Description

The command is used to operate device list.

19.12.2. Format

Command	Possible response(s)	Description
+EBTOPAD=?	+EBTOPAD: (0-4), OK/ERROR	Show if the command is supported
+EBTOPAD=<n>,<address>	+EBTOPAD:<index>,<address > OK /ERROR	get device list

19.12.3. Field

<address>: BT addr string ,no need use “ ” double quotes， length should be 12 characters

Example：

1234565b0102

It means ：LAP is 0x123456, UAP is 0x5b, NAP is 0x0102

<n> :integer

0 delete

1 locate

2 delete all

3 most recently used

4 return total list

<index> :integer

Example

AT+EBTOPAD=0,1234565b0102

OK

AT+EBTOPAD=3

+EBTOPAD:1,1234565b0102

OK

AT+EBTOPAD=4

+EBTOPAD:1,1234565b0102

+EBTOPAD:2,1234565b0103

+EBTOPAD:3,1234565b0104

+EBTOPAD:4,1234565b0105

OK

19.13. AT+EBTSTATE – Query connect manger and profile status

19.13.1. Description

The command is used to query BT connect manger and profile status.

19.13.2. Format

Command	Possible response(s)	Description
+EBTSTATE=?	+EBTSTATE： ，（0-4294967295）OK/ERROR	Show if the command is supported
+EBTSTATE	+EBTSTATE: <CM state> OK /ERROR	Query BT connect manger state
+EBTSTATE=<address>,<profile>	+EBTSTATE: <profile state> OK / *ERROR*	Query Profile state

19.13.3. Field

<address>: BT addr string ,no need use “ ” double quotes， length should be 12 characters

Example：

1234565b0102

It means ： LAP is 0x123456, UAP is 0x5b, NAP is 0x0102

<profile>:integer, see BT Profile SPEC for profile UUID.

Example:

4353, it means 0x1101, SPP Profile.

<CM state>:integer

1 START, BT is powering on.

2 READY, BT powered on, stand by for working.

3 PAIRING,

4 INQUIRING,

5 CONNECTING,

6 CONNECTED,

7 NUKNOWN,

<profile state>:integer

1 START, Reserved.

2 IDLE, APP for this profile not activated.

3 ACTIVATE, APP for this profile activated.

4 AUTHORIZING, authorizing for connection.

5 DISCONNECTING, disconnecting for connection.

6 DEACTIVATING, deactivating for profile APP.

Example

AT+EBTSTATE=1234565b0102,1 //query profile state

+EBTSTATE:2

OK

AT+EBTSTATE// query cm state

+EBTSTATE:2

OK

19.14. AT+EBTENSNIFF – Set or get SNIFF mode level

19.14.1. Description

The command is used to read or set BT sniff level.

19.14.2. Format

Command	Possible response(s)	Description
+EBTENSNIFF=?	+EBTENSNIFF：（0-1），（0-4） OK/ERROR	Show if the command is supported
AT+EBTENSNIFF=<op>,[<level>]	+EBTENSNIFF:<level > OK / *ERROR*	Read/Write BT sniff level

19.14.3. Field

<op>: integer

0 write

1 read

<level>:integer

0 reserved.

1 reserved.

2 reserved.

3 reserved.

Example

AT+EBTENSNIFF=0,0// set

OK

AT+EBTENSNIFF=1 // read

+EBTENSNIFF:0

OK

19.15. AT+EBTRSSI – Read BT device signal level

19.15.1. Description

The command is used to read BT device signal level. Only we can get the valid RSSI value when the <address> is connected.

19.15.2. Format

Command	Possible response(s)	Description
+EBTRSSI=?	OK/ERROR	Show if the command is supported
+EBTRSSI=<address>	+EBTRSSI:<signal > OK /ERROR	Read BT signal

19.15.3. Field

<address>: BT addr string ,no need use “ ” double quotes， length should be 12 characters

Example：

1234565b0102

It means ： LAP is 0x123456, UAP is 0x5b, NAP is 0x0102

<signal>:integer

Example

AT+EBTRSSI=1234565b0102

+EBTRSSI:32

OK

19.16. AT+EBTPACT – (De)Active BT profiles

19.16.1. Description

The command is used to (de)active BT profiles.

19.16.2. Format

Command	Possible response(s)	Description
+EBTPACT=?	+EBTPACT：（0-18446744073709551615） OK/ERROR	Show if the command is supported
+EBTPACT=<profile-bitmap>	+EBTPACT:< profile-bitmap > OK / *ERROR*	(de)active BT profiles

19.16.3. Field

<profile_bitmap> integer ,uint64 ,one bit is a profile support or not. 0 :not support ,1 supported

Example : profile_bitmap value is 3 ,

0x0000000000000011 means profile 1 ,2 supported ,others not supported.

bit 1 ： SPP Profile, UUID = 0x1101.

bit 2 – bit 64 ： All reserved.

Example

AT+EBTPACT=1

+EBTPACT：1

OK

19.17. AT+EBTCONN – Connect BT profile

19.17.1. Description

The command is used to connect BT profile.

19.17.2. Format

Command	Possible response(s)	Description
+EBTCONN=?	+EBTCONN：（0-1），，（0-4294967295），（0-1） OK/ERROR	Show if the command is supported
+EBTCONN=<n>,<address>,<profile>,<role>	OK / *ERROR*	Write BT address

19.17.3. Field

<address>: BT addr string ,no need use “ ” double quotes， length should be 12 characters

Example：

1234565b0102

It means ：LAP is 0x123456, UAP is 0x5b, NAP is 0x0102

<n>: integer

0 Disconnect

1 Connect

<profile>:integer, see BT Profile SPEC for profile UUID.

Example:

4353, it means 0x1101, SPP Profile.

<role>:integer, profile role.

1 server

0 client

Example

AT+EBTCONN=1,1234565b0102,32,1

OK

+EBTCONN:1,1234565b0102,32,1

19.18. +EBTPAIR – URC: Pair indication

19.18.1. Description

The command is used to notify that another device wants to pair local BT and may need input password or pin code.

19.18.2. Format

Unsolicited result code
+EBTPAIR:<address>,<name>,<enable 16digitspin>[,<password>]

19.18.3. Field

<address>: BT addr string ,no need use “ ” double quotes， length should be 12 characters

Example：

1234565b0102

It means ：LAP is 0x123456, UAP is 0x5b, NAP is 0x0102

<name>：BT name string ,no need to use “ ” double quotes.

< enable 16digitspin>: integer

0 not enable the 16digits pin

0 enable the 16 digits pin

<password>:String

Example

+EBTPAIR:1234565b0102 ,devicename,0,123456

19.19. +EBTINQ - URC: Pair indication

19.19.1. Description

The command is used to notify other BT device are found.

It should be output after command AT+EBTINQ.

19.19.2. Format

Unsolicited result code
+EBTTINQ:<address>,<name>

19.19.3. Field

<address>: BT addr string ,no need use “ ” double quotes， length should be 12 characters

Example：

1234565b0102

It means ：LAP is 0x123456, UAP is 0x5b, NAP is 0x0102

<name>：BT name string ,no need use “ ” double quotes.

Example

+EBTINQ:1234565b0102 ,devicename

19.20. +EBTIND - URC: Inquiry completed indication

19.20.1. Description

The command is used to notify that the inquiring is completed.

19.20.2. Format

Unsolicited result code
+EBTIND:<result>,<is cancelled>

19.20.3. Field

<result>: integer

0 failed

1 successful

<is cancelled>：integer

0 not be cancelled

1 be cancelled

Example

+EBTIND:1,0

19.21. +EBTPAIRRES - URC: Passive pair response

19.21.1. Description

The command is used to notify the response of passive pairing.

19.21.2. Format

Unsolicited result code
+EBTPAIRRES:<result>,<isfirst>,[address]

19.21.3. Field

<result>: integer

0 failed

1 successful

<is first>：integer

0 not the first

1 be first

<address>: BT addr string ,no need use “ ” double quotes， length should be 12 characters

Example：

1234565b0102

It means ：LAP is 0x123456, UAP is 0x5b, NAP is 0x0102

Example

+EBTPAIRRES:1,0,1234565B0102

19.22. +EBTCONN - URC: Notify profile connected

19.22.1. Description

The command is used to notify BT profile connected.

19.22.2. Format

Unsolicited result code
+EBTCONN:<result>,<address>,<profile id>

19.22.3. Field

<result>: integer

0 failed

1 successful

<address>: BT addr string ,no need use “ ” double quotes， length should be 12 characters

Example：

1234565b0102

It means ： LAP is 0x123456, UAP is 0x5b, NAP is 0x0102

<profile id>:integer, see BT Profile SPEC for profile UUID.

Example:

4353, it means 0x1101, SPP Profile.

Example

+EBTCONN:1,1234565B0102,32

19.23. +EBTPRFAC - URC: Notify all supported profiles are (in)active

19.23.1. Description

The command is used to notify all supported profiles are inactive or active.

19.23.2. Format

Unsolicited result code
+EBTPRFAC:<state>

19.23.3. Field

<state>: integer

0 all inactive

1 all active

19.24. +EBTDISC - URC: Notify profile connected

19.24.1. Description

The command is used to notify BT connections are disconnected.

19.24.2. Format

Unsolicited result code
+EBTDISC:<n>,<address>[,<profileid>,<passive>]

19.24.3. Field

<n>: integer

1 one connection is disconnected

2 all connections are disconnected

<address>: BT addr string ,no need use “ ” double quotes， length should be 12 characters

Example：

1234565b0102

It means ： LAP is 0x123456, UAP is 0x5b, NAP is 0x0102

<profile id>:integer, see BT Profile SPEC for profile UUID.

Example:

4353, it means 0x1101, SPP Profile.

<passive> :integer

0 not passive disconnect

1 passive disconnected

Example

+EBTDISC:1,1234565B0102,32,0

+EBTDISC:2,1234565B0102

19.25. +EBTVISB - URC: Notify visibility is changed

19.25.1. Description

The command is used to notify BT visibility is changed.

19.25.2. Format

Unsolicited result code
+EBTVISB:3,<is because AT>

19.25.3. Field

< is because AT >: integer

0 not because receive AT+EBTVISB

1 because receive AT+EBTVISB

Example

+EBTVISB:3,0

19.26. +EBTRST - URC: Notify BT is Reset

19.26.1. Description

The command is used to notify BT is reset.

19.26.2. Format

Unsolicited result code
+EBTRST:1

19.26.3. Field

Example

+EBTRST:1

19.27. +EBTPRFBND - URC: Notify bond profile fail

19.27.1. Description

The command is used to notify BT bond profile fail.

19.27.2. Format

Unsolicited result code
+EBTPRFBND:<profile-id>,0

19.27.3. Field

<profile id>:integer, see BT Profile SPEC for profile UUID.

Example:

4353, it means 0x1101, SPP Profile.

Example

+EBTPRFBND:10,0

19.28. +EBTPRFAU - URC: Notify BT devices need connect our profile

19.28.1. Description

The command is used to notify other BT devices need connect our profile .

19.28.2. Format

Unsolicited result code
+EBTPRFAU:<Profileid>,<address>,<name>

19.28.3. Field

<address>: BT addr string ,no need use “ ” double quotes， length should be 12 characters

Example：

1234565b0102

It means ：LAP is 0x123456, UAP is 0x5b, NAP is 0x0102

<name>：BT name string ,no need use “ ” double quotes.

<profile id>:integer, see BT Profile SPEC for profile UUID.

Example:

4353, it means 0x1101, SPP Profile.

Example

+EBTPRFAU:3,1234565B0102,devicename

19.29. +EBTPRFCN - URC: Notify profile connected

19.29.1. Description

The command is used to notify BT profile connected.

19.29.2. Format

Unsolicited result code
+EBTPRFCN:<profile>,<ret>

19.29.3. Field

<profile id>:integer, see BT Profile SPEC for profile UUID.

Example:

4353, it means 0x1101, SPP Profile.

<ret>:integer

0 failed

1 successful

Example

+EBTPRFCN:3,1

19.30. +EBTPRFDSCN - URC: Notify profile disconnected

19.30.1. Description

The command is used to notify BT profile disconnected.

19.30.2. Format

Unsolicited result code
+EBTPRFCN:<profile>,<address>

19.30.3. Field

<profile id>:integer, see BT Profile SPEC for profile UUID.

Example:

4353, it means 0x1101, SPP Profile.

<address>: BT addr string ,no need use “ ” double quotes， length should be 12 characters

Example：

1234565b0102

It means ： LAP is 0x123456, UAP is 0x5b, NAP is 0x0102

Example

+EBTPRFDSCN:3,1234565B0102

20. BT SPP PROFILE AT COMMAND

20.1. +ESPPCONN – unsolicited to notify that a new SPP client is connected

20.1.1. Description

This unsolicited is used to notify that a new SPP client is connected in data mode with the device.

20.1.2. Format

Unsolicited result code
+ESPPCONN=<conn id>,<address>

20.1.3. Field

<conn id>:

Connection id for this SPP client.

<address>:

BT address string, no need use “ ” double quotes. Length should be 12 characters.

Example：

1234565b0102

It means : LAP is 0x123456, UAP is 0x5b, NAP is 0x0102

20.2. AT+ESPPREAD – Read a hex format string from a SPP connected client.

20.2.1. Description

Read a hex format string from a SPP connected client.

20.2.2. Format

Command	Possible response(s)
+ESPPREAD=<conn id>	+ESPPREAD: <conn id>,<string> OK +ESPP: <conn id>,<error> ERROR +CME ERROR: <error>
+ESPPREAD=?	

20.2.3. Field

<conn id>:

Connection id of SPP client to read from.

<string>:

Double quoted string read. String is formatted according to ”T-REC-V.25ter 5.4.2.x Numeric and string constants” indications.

<error>
Error description if failed.

20.3. AT+ ESPPREADA – Read ascii data from a SPP connected client

20.3.1. Description

Read ascii data from a SPP connected client.

20.3.2. Format

Command	Possible response(s)
+ESPPREADA=<conn id>	+ESPPREADA: <conn id>,<ret> <data> OK +ESPP: <conn id>,<error> ERROR +CME ERROR: <error>
+ESPPREADA=?	

20.3.3. Field

<conn id>:
Connection id of SPP client to read from.
<ret>:
Number of bytes read.
<data>
Data received in ASCII format.
<error>
Error description if failed

20.4. AT+ ESPPREADS – Read a string from a SPP connected client

20.4.1. Description

Read a string from a SPP connected client.

.

20.4.2. Format

Command	Possible response(s)
+ESPPREADS=<conn id>	+ESPPREADS: <conn id>,<string> OK +ESPP: <conn id>,<error> ERROR ERROR

+ESPPREADS=?	

20.4.3. Field

<conn id>:

Connection id of SPP client to read from.

<string>:

Double quoted string read. String is formatted according to ”T-REC-V.25ter 5.4.2.x Numeric and string constants” indications.

<error>

Error description if failed.

20.5. AT+ ESPPTA – Allows a SPP client to connect

20.5.1. Description

This command allows a SPP client to connect as a terminal adapter in order to send AT commands to module.

20.5.2. Format

Command	Possible response(s)
+ESPPTA=<op>	OK ERROR
+ESPPTA?	+ESSPTA: <state> OK
+ESPPTA	OK ERROR
+ESPPTA=?	

20.5.3. Field

<op>:

Operation mode:

0: Disable SPP terminal adapter.

1: Enable SPP terminal adapter.

<state>:

Current setting for SPP terminal adapter:

0: SPP terminal adapter is disabled (default).

1: SPP terminal adapter is enabled.

NOTE

1. If SPP terminal adapter option is enabled, it is client choice whether to connect as a terminal adapter or in data mode. No data can be written to SPP client until data is at first received by the SPP client itself.

2. To connect as a terminal adapter, SPP client must issue “AT+ESPPTA” followed by command line termination right after the connection is established. Please, note that the command is at first evaluated by module as a string pattern, and for this reason it must be sent in uppercase and without any space character in it.

20.6. AT+ ESPPTS – Transparent data transmission with a SPP

20.6.1. Description

This command starts a transparent data transmission with a SPP connected client.

20.6.2. Format

Command	Possible response(s)
+ESPPTS=<conn id>	If success, no reply +CME ERROR: <error>
+ESPPTS=?	

20.6.3. Field

<conn id>:

Connection id of SPP client to connect in transparent mode.

<error>

Error description if failed.

NOTE

1. When enter transparent transmission mode, no AT command can be issued until escape from this mode. The escape string is “+++”.

2. After transparent mode is ended, a +ESPPTSEND unsolicited is notified.

20.7. AT+ ESPPWRITE – Transparent data transmission with a SPP

20.7.1. Description

Write a hex format string to a SPP connected client.

20.7.2. Format

Command	Possible response(s)
+ESPPWRITE=<conn id>	+ESPPWRITE: <conn id>,<ret> OK +ESPP: <conn id>,<error> ERROR +CME ERROR: <error>
+ESPPWRITE=?	

20.7.3. Field

<conn id>:

Connection id of SPP client to write to.

<ret>

The number of bytes successfully written.

<error>

Error description if failed.

20.8. AT+ ESPPWRITEA – Write ascii data to a SPP connected client

20.8.1. Description

Write ascii data to a SPP connected client. The device responds to the command with the prompt <greater_than><space> and waits for the data to send. To complete the operations send Ctrl-Z char (0x1A hex); to exit without writing the message send ESC char (0x1B hex)..

20.8.2. Format

Command	Possible response(s)
+ESPPWRITEA=<conn id>	+ESPPWRITEA: <conn id>,<ret> OK +ESPP: <conn id>,<error> ERROR +CME ERROR: <error>
+ESPPWRITEA=?	

20.8.3. Field

<conn id>:

Connection id of SPP client to write to.

<ret>

The number of bytes successfully written.

<error>

Error description if failed.

NOTE

The maximum number of bytes to write is 512 bytes; trying to send more data will cause the surplus to be discarded and lost.

20.9. AT+ ESPPWRITES – Write a string to a SPP connected client

20.9.1. Description

Write a string to a SPP connected client.

20.9.2. Format

Command	Possible response(s)
+ESPPWRITES=<conn id>,<string>	+ESPPWRITES: <conn id>,<ret> OK +ESPP: <conn id>,<error> ERROR +CME ERROR: <error>
+ESPPWRITES =?	

20.9.3. Field

<conn id>:

Connection id of SPP client to write to.

<string>:

Double quoted string to write. String must be formatted according to ”T-REC-V.25ter 5.4.2.x Numeric and string constants”.

<ret>

The number of bytes successfully written.

<error>

Socket error description if failed.

21. BT OPP PROFILE AT COMMAND

21.1. AT+EOPPPUSH – OPP client push files

21.1.1. Description

The command is used by OPP client to push files to specified OPP server.

21.1.2. Format

Command	Possible response(s)	Description
+EOPPPUSH=<address>,<file1>[,<file2>[,...,[<file5]]]	OK/ERROR	Max 5 files can be sent in this command

21.1.3. Field

< address >: OPP server BT address string ,no need use “ ” double quotes， length should be 12 characters.

<file1>: First file to send. User must give full file path. “” double quotes are required.

<file2>…<file5>: optional files to send. Must use “” double quotes. If you send several files and some file paths are not valid, we’ll try to send valid file paths. Valid file path count will be reported in URC +EOPPPUSH

file path should use utf-8 encoding

Example：

AT+EOPPPUSH = 1234565b0102, “Z:/AAA/bbb.file”,”Z:/cc.file”

It means ： send to device with BT address “1234565b0102**”. Send 2 local files:** Z:\AAA\bbb.file, Z:\cc.file

If you input 3 file path and only 2 is valid, we’ll try to send two correct files. Total valid files will be returned as URC “EOPPPUSH” filed.

21.2. +EOPPPUSH - URC: Notify when send finish

21.2.1. Description

The command is used to notify when OPP push transaction is finished.

21.2.2. Format

Unsolicited result code
+EOPPPUSH:<result>,<Total_files>,<address>,<file>

21.2.3. Field

<result>: uint 8, 0 succeed, others are error code.

Example: 0b0000000000000101: file 1 and file 3 are successfully sent.

<total_files> : total valid file path. This param is used to notify user how many valid files in AT+EOPPPUSH command. They should expect same amount of +EOPPPUSH URC for this AT command.

<address>: OPP server BT address string ,no “ ” double quotes， length should be 12 characters.

<file>: indicates the file name for this URC

Example

+EOPPPUSH:1,5,1234565B0102,”Z:/aaa.txt”

21.3. AT+EOPPABORT – Abort current OPP transaction with one device

21.3.1. Description

The command is used by OPP client/server to abort OPP transaction with one device. If OPP client is sending multiple objects to server, this command will only abort current sending object.

21.3.2. Format

Command	Possible response(s)	Description
+EOPPABORT=<address>	OK/ERROR	

21.3.3. Field

< address >: address string ,no need use “ ” double quotes， length should be 12 characters.

Example：

AT+EABOART=1234565b0102

It means ：abort OPP transaction with BT address “1234565b0102**”.**

21.4. +EOPPRECV - URC: Notify when OPP server receive OPP client send request

21.4.1. Description

The command is used to notify when OPP server receive OPP client send request.

21.4.2. Format

Unsolicited result code
+EOPPRECV: <address>,<device_name>,<obj_name>,<MIME>

21.4.3. Field

<device_name>: OPP client BT name string, with double quote"". UTF8 encoded

<address>: OPP client BT address

<obj_name>: object name, max 128 chars. with double quotes"". UTF8 encoded.

<MIME>: MIME string, max 80 chars. with double quote""

Example

+EOPPRECV: 1234565B0102, "My BT device", "aa.txt", "text/plain"

It means: device "My BT device" with address "1234565B0102" want to send "aa.txt", which MIME type is "text/plain". Do you want to accept it?

21.5. AT+EOPPACCEPT – OPP server decide whether accept push request by OPP client

21.5.1. Description

The command is used by OPP server to determine whether it accept an object by OPP client.

21.5.2. Format

Command	Possible response(s)	Description
+EOPPACCEPT= <result>, <address>,<path>	OK/ERROR	

21.5.3. Field

< address >: address string ,no need use " " double quotes， length should be 12 characters.

<result>: bool, 1 means accept, 0 means reject;

<path>: full file path to store this object. Double quotes required

Example：

AT+EOPPACCEPT= 1, 1234565b0102,"Z:\aa.txt"

It means ： Accept object send by client with BT address "1234565b0102**" and store it as "Z:\aa.txt".**

21.6. +EOPPACCEPT - URC: Notify when OPP server receive finish

21.6.1. Description

The command is used to notify when OPP server complete receiving one object

21.6.2. Format

Unsolicited result code
+EOPPACCEPT: <result>,<address>,<obj_name>

21.6.3. Field

<address>: OPP client BT address

<obj_name>: object name, max 128 chars. with double quote”” and UTF8 encoded

Example

+EOPPRECV: 1234565B0102, ”aa.txt”

It means: finish receiving “aa.txt” from device “1234565B0102”

22. BT PBAP PROFILE AT COMMAND

22.1. AT+EPBPLIST – PBAP client get PBH list

22.1.1. Description

The command is used by PBAP client to get PBH list

22.1.2. Format

Command	Possible response(s)	Description
+EPBPLIST=<address>, <folder>,[<MaxListCount>, [<ListStartOffset>, [<searchattr>,<searchval>]]]	OK/ERROR	Query PBH list

22.1.3. Field

< address >: PBAP server BT address string ,no need use “ ” double quotes， length should be 12 characters.

<folder>: folder name, 1-10

telecom/pb	1
telecom/ich	2
telecom/och	3
telecom/mch	4
telecom/cch	5
SIM1/telecom/pb	6
SIM1/telecom/ich	7
SIM1/telecom/och	8
SIM1/telecom/mch	9
SIM1/telecom/cch	10

<MaxListCount>: maximum number of entries that PBAP client can handle. <ListStartOffset>: offset to the first entry of PHB. default value is 0.

<searchattr>: 1-3; 1: name; 2:Number; 3:sound;

<searchval>: key word string. Double quotes required. should be UTF8 encoded

Example：

AT+EPBPLIST= 1234565b0102,1,200,0

It means ： query PBH list from “1234565b0102”, max list item is 200, offset is 0.

AT+EPBPLIST= 1234565b0102,1,200,0,1,”a”

It means ： query PBH list from “1234565b0102”, max list item is 200, offset is 0. server only return contact whose name contains “a”.

22.2. +EPBPLIST - URC: Notify when finish pulling list

22.2.1. Description

The command is used to notify when PBAP Client finish pulling list

22.2.2. Format

Unsolicited result code
+EPBPLIST: <result>,<address>,<path>

22.2.3. Field

<result>: bool, 1 succeed, 0 fail

<address>: PBAP server BT address

<path>: full file path of list vCard file. with double quote”” and UTF8 encoded

Example

+EPBPLIST: 1, 1234565B0102,”Z:/@pbap/list.vcd”

It means: get PHB list from device “1234565B0102” succeed

22.3. AT+EPBAPENTRY – PBAP client pull vCard entry

22.3.1. Description

The command is used by PBAP client to get phonebook entry. It will return vCard 2.1 file.

22.3.2. Format

Command	Possible response(s)	Description
+EPBPENTRY = <address>,<folder>, <object>,[<filter>]	OK/ERROR	Download PHB entry

22.3.3. Field

<address>: PBAP server BT address string ,no need use “ ” double quotes， length should be 12 characters.

<folder>: folder name, 1-10

telecom/pb	1
telecom/ich	2
telecom/och	3
telecom/mch	4
telecom/cch	5
SIM1/telecom/pb	6
SIM1/telecom/ich	7
SIM1/telecom/och	8
SIM1/telecom/mch	9
SIM1/telecom/cch	10

<object>: object name, 1-10, max 64 uint16. double quotes required.

<filter>: 32 bit filter. Indicate the attributes contained in the requested vCard objects. Refer to PBAP Spec *Table 5.2 Attribute Mask*. Default value will only filter out PHOTO (0xFFF7). If given filter is 0, we’ll use default filter.

Example：

AT+ EPBPENTRY=1234565b0102,1,”001.vcf”, 0xFFF7

It means ：get phonebook entry “001.vcf” from “1234565b0102”, associated photo not required.

22.4. +EPBPDL - URC: Notify when finish pulling list

22.4.1. Description

The command is used to notify when PBAP Client finish pulling list

22.4.2. Format

Unsolicited result code
+ EPBPDL: <result>,<address>,<path>

22.4.3. Field

<result>: bool, 1 succeed, 0 fail

<address>: PBAP server BT address

<path>: full file path to store returned vcard file.

Example

+ EPBPDL: 1, 1234565B0102,”Z:/@pbapc/temp.vcf”

It means: get PHB list from device “1234565B0102” fail

22.5. AT+EPBPDLCH – PBAP client download call history

22.5.1. Description

The command is used by PBAP client to download one category of call history. It will return vCard 2.1 file.

22.5.2. Format

Command	Possible response(s)	Description
+EPBPDLCH=<address>,<folder>,[<MaxListCount>,<ListStartOffset>,<filter>]	OK/ERROR	Download call history

22.5.3. Field

<path>: full file path to store returned vcard file. Double quotes required

<address>: PBAP server BT address string ,no need use “ ” double quotes， length should be 12 characters.

<folder>: folder name, 2-10. 6 is not valid

telecom/ich	2
telecom/och	3
telecom/mch	4
telecom/cch	5
SIM1/telecom/ich	7
SIM1/telecom/och	8
SIM1/telecom/mch	9
SIM1/telecom/cch	10

<MaxListCount>: maximum number of entries that PBAP client can handle.

<ListStartOffset>: offset to the first entry of CH. Default value is 0.

<filter>: 32 bit filter. Indicate the attributes contained in the requested vCard objects. Refer to PBAP Spec *Table 5.2 Attribute Mask*. Default value will select FN/N/TEL/X-IRMC-CALL-DATETIME (0x10000083). If given filter is 0, we’ll use default filter.

Example：

AT+EPBPLIST= 1234565b0102, 2, 30

It means ： query Incoming call history from “1234565b0102”, max list item is 300, offset is 0.

22.6. +EPBPDLCH - URC: Notify when finish pulling list

22.6.1. Description

The command is used to notify when PBAP Client finish pulling list

22.6.2. Format

Unsolicited result code
+EPBPDLCH: <result>,<address>,<path>

22.6.3. Field

<result>: bool, 1 succeed, 0 fail

<address>: PBAP server BT address

Example

+EOPPRECV: 0, 1234565B0102,”Z:/@pbapc/call_history.vcf”

It means: get CH list from device “1234565B0102” fail

23. BT A2DP AVRCP AT COMMAND

23.1. AT+EBTA2DP

23.1.1. Description

The command is used to play or pause a2dp audio file.

23.1.2. Format

Command	Possible response(s)	Description
+EBTA2DP=<play/pause>, <bt addr>,<"file name">,<sample rate>,<stereo>	+EBTA2DP:<result> result: 0 fail 1 success	Play/pause a2dp audio file

23.1.3. Field

<play/pause> (0,1).play:1,pause:0.

<bt addr>:" 0x0-0xFFFFFFFFFFFF"

<"file name">: string type

<sample rate >: (32000, 44100,48000).

<stereo>:(0,1)

Example：

AT+EBTA2DP=1,
DB4BDD130018," 005A003A005C006300730030003100 2E0061006D0072",32000,0

+EBTA2DP:1

It means ： start paly a2dp audio file, bt addr is DB4BDD130018, file name is cs01.amr, sample rate is 32000, not a stereo.

AT+EBTA2DP=0

+EBTA2DP:1

It means ： pause play a2dp audio file.

23.2. +EBTAVRCP – URC: BT Handset button pressed

23.2.1. Description

The command is used to notify when bluetooth handset button is pressed.

23.2.2. Format

Unsolicited result code
+EBTAVRCP: <command>,<key event>

23.2.3. Field

<command>:

Paly 0x44

Pause 0x46

Forward 0x4B

Backward 0x4C

<key event>：key down: 0, key up: 1

Example

+EBTAVRCP:75,0

+EBTAVRCP:75,1

It means: Bluetooth headset need to play previous audio file.

24. BT HFP PROFILE AT COMMAND

24.1. AT+EHFCLCC

24.1.1. Description

The command is used to query current call info from AG.

24.1.2. Format

Command	Possible response(s)	Description
+EHFCLCC?	OK/ERROR	

24.1.3. Field

None

Example

There should be a BT connection between HF and AG, and please refer to NOTE for details.

AT+EHFCLCC?

[Result]

OK

+EHFCIEV:2,075526630099,1

NOTE

The precondition should be,

3. Power on BT by “at+ebtpwr=0”
   After power on successfully, the HF(M2M) will receive
   +EBTPRFAC:1

   OK

4. Require a BT connection between the HF(M2M) and AG(smart phone):
   Use Smartphone to initial a BT connection, and HF will receive

   +EBTPAIR:65727146FEE1,,0,360695

   And HF needs to send “at+ebtpaircnf=1”to setup the BT connection .

   Then HF will receive

   OK

   It means BT connection full established. And the command for HFP will run properly.

24.2. +EHFCIEV – URC: Notify when call state changed finish

24.2.1. Description

The command is used to notify when call state changed.

24.2.2. Format

Unsolicited result code
+EHFCIEV:<state>, <phb_num>,<call index>

24.2.3. Field

<state>: int

0, call complete

1, outgoing call

2, incoming call

4, call setup

8, current call is hold

<phb_num>: string

Example

+EHFCIEV: 0, 075526630099, 1

24.3. AT+EHFATA – Accept Incoming call from AG/HF

24.3.1. Description

The command is used to accept an incoming call from AG/HF.

24.3.2. Format

Command	Possible response(s)	Description
+EHFATA=<role>	OK/ERROR	Only support HF now.

24.3.3. Field

< role>: bool

0, accept incoming call by AG. Current role is AG, AT+ATA

1, accept incoming call by HF.

Example

[Case 1]
Only one incoming call
AT+EHFATA=1//accept the incoming call
[Result]

+EHFCIEV:2,075526630099,1//incoming call
at+ehfata=1
OK

+EHFCIEV:4,075526630099,1//incoming call active

[Case 2]
One active/held call and incoming call
AT+EHFATA=1//accept the incoming call, put the active call on held(if any)
[Result]
+EHFCIEV:4,13243764932,1//active call

+EHFCIEV:2,075526630099,2//incoming call
at+ehfata=1
OK

+EHFCIEV:4,13243764932,1

+EHFCIEV:2,075526630099,2

+EHFCIEV:8,13243764932,1//held

+EHFCIEV:4,075526630099,2//active

[Case 3]
One active, one hold call, and there is an incoming call
AT+EHFATA=1// Terminate all active calls (if any), and accept the incoming call
[Result]
+EHFCIEV:8,13243764932,1//hold

+EHFCIEV:4,10010,2//active call

+EHFCIEV:2,075526630099,3
at+ehfata=1
OK

+EHFCIEV:0,10010,2

+EHFCIEV:8,13243764932,1

+EHFCIEV:2,075526630099,3

+EHFCIEV:8,13243764932,1

+EHFCIEV:4,075526630099,3

24.4. AT+EHFATD - start phone connection by AG/HF

24.4.1. Description

The command is used to start phone connection by AG/HF.

24.4.2. Format

Command	Possible response(s)	Description
+EHFATD= <role>,<type>[,<number>]	OK/ERROR	Only support HF and dial with phone number now.

24.4.3. Field

<role>:

0, Initial an outgoing call by AG

1, Initial an outgoing call by HF

<type>:

0, dial with string

1, redial last number

2, dial with memory(dial with phone number stored in AG phonebook)

<number>:

String with “0~9” or phone number index in AG phonebook.

When type is 0, use phone number string, and when type is 2, use phone number index.

Example

[Case 1]

AT+EHFATD=1,0,”075526630099”//initial an outgoing call

[Result]

at+ehfatd=1,0, 075526630099

OK

+EHFCIEV:1, 075526630099,1

[Case 2]

AT+EHFATD=1,2,1//initial an outgoing call with the first phone number in AG phonebook

[Result]

at+ehfatd=1,2,1

OK

+EHFCIEV:1,13243764932,1

24.5. AT+EHFCHUP - AG/HF to terminate the current call

24.5.1. Description

The command is used by AG/HF to terminate the current call.

24.5.2. Format

Command	Possible response(s)	Description
+EHFCHUP=<role>	OK/ERROR	Only support HF now.

24.5.3. Field

< role>:

0, terminate call connection by AG

1, terminate call connection by HF.

Example

[Case]

AT+EHFCHUP=1//hangup current call

[Result]

+EHFCIEV:2,075526630099,2

at+ehfchup=1

OK

+EHFCIEV:0,075526630099,2

24.6. AT+EHFCHLD - AG/HF to release/hold/retrieve/swap call

24.6.1. Description

The command is used by AG/HF to release/hold/retrieve/swap call.

24.6.2. Format

Command	Possible response(s)	Description
+EHFCHLD=<role>,<act>[,<index>]	OK/ERROR	Only support HF now.

24.6.3. Field

<role>:

0, act from AG

1, act from HF

<act>[,<index>]

0, releases all held calls

1, release all active calls and accepts the other (waiting or held)call

1,x, release the specific active call x

2, place all active calls on hold and accept the (waiting or held) call

2,x, place all active calls, except call x, on hold

3, add a held call to conference call

4, swap/retrieve

<index>: int

1~127

Example

[Case 1]

AT+EHFCHLD=1, 0// release all held calls

[Result]

+EHFCIEV:2,075526630099,1

at+ehfchld=1,0

OK

+EHFCIEV:0,075526630099,1

[Case 2]

AT+EHFCHLD=1, 1// release all active calls and accepts the other (waiting or //held)call, supported in multiple call.

[Result]

+EHFCIEV:2,075526630099,1

at+ehfchld=1,1

OK

+EHFCME:100

+EHFCIEV:2,075526630099,1

[Case 3]

AT+EHFCHLD=1, 1,x// release the specific active call x, only supported in //conference call

[Result]

+EHFCIEV:2,075526630099,1

at+ehfchld=1,1,1

OK

+EHFCME:100

+EHFCIEV:2,075526630099,1

[Case 4]

AT+EHFCHLD=1, 2// place all active calls on hold and accept the (waiting or held) call, //only supported in multiple call

[Result]

+EHFCIEV:2,075526630099,1

at+ehfchld=1,2

OK

+EHFCIEV:2,075526630099,1

+EHFCIEV:4,075526630099,1

[Case 5]

AT+EHFCHLD=1, 2,x// place all active calls, except call x, on hold, only //supported in conference call

[Result]
+EHFCIEV:2,075526630099,1
at+ehfchld=1,2,1
OK

+EHFCME:100

+EHFCIEV:2,075526630099,1

[Case 6]
AT+EHFCHLD=1, 3// add a held call to conference call

[Result]
+EHFCIEV:2,075526630099,1
at+ehfchld=1,3
OK

+EHFCIEV:2,075526630099,1

[Case 7]
AT+EHFCHLD=1, 4// swap the active/held call

[Result]
+EHFCIEV:2,075526630099,1
at+ehfchld=1,4
OK

+EHFCIEV:2,075526630099,1

+EHFCIEV:4,075526630099,1

24.7. AT+EHFVTS - transmit DTMF codes by HF

24.7.1. Description

The command is used to transmit DTMF codes by HF.

24.7.2. Format

Command	Possible response(s)	Description
+EHFVTS=<digit>	OK/ERROR	

24.7.3. Field

<digit>: U8, digit should one of “*# 0-9”

Example

[precondition]

[Case]

AT+EHFVTS=*//play tone

[Result]

OK

24.8. AT+EHFRVC - set or sync volume among HF/AG

24.8.1. Description

The command is used to (Remote audio volume control) set or sync volume among HF/AG.

24.8.2. Format

Command	Possible response(s)	Description
+EHFRVC=<role>,<act>,<vol>	OK/ERROR	Only supported HF

24.8.3. Field

<role>: bool

0, action from AG

1, action from HF

<act>: int

0, set the volume of AG speaker

1, set the volume of AG mic

<vol>: int 0-15

Example

[precondition]

[Case]

AT+EHFRVC=1,0,15//set the volume of AG speaker

[Result]

OK

24.9. +EHFVGS - URC: Notify when call state changed finish

24.9.1. Description

The command is used to notify when the volume of speaker in AG changed, and the volume of HF speaker will sync with this.

24.9.2. Format

Unsolicited result code
+EHFVGS:<value>

24.9.3. Field

<state>: int

0-15(0, means mute. And 15 means the maximum volume)

Example

[precondition]

+EHFVGS: 12

24.10. +EHFCME - URC: Notify when call state changed finish

24.10.1. Description

The command is used to notify when HFP AT CMD execute failed.

24.10.2. Format

Unsolicited result code
+EHFCME:<result>

24.10.3. Field

<state>: U8

0-255(255 means ok, other means fail)

Example

+EHFCME: 100

25. BLE GATT AT COMMAND

25.1. AT+GATSREG – (De)Register GATT Server

25.1.1. Description

(De)Register GATT Server with user_id..

25.1.2. Format

Command	Possible response(s)
+GATSREG=<op>,<user_id>	+GATSREG:<op>,<result>,<user_id> OK / ERROR

25.1.3. Field

< op >:

0 deregister

1 register

<user_id>:

User id of gatt server, or the name of the gatt server.

A Hex value string, each char of it should in set { ‘0’~’9’,’a’~’f’,’A’~’F’ }

Max length of it is 32.

<result>:

0 SUCCESS

Other un-success

OK / ERROR: at argument error will return ERROR, other will return OK.

25.1.4. Note

Each GATT Server AT can only send after the prior AT return OK/ERROR.

25.1.5. Example:

AT+GATSREG=1,"AFAF123BC" //register a gatt server with user id “AFAF123BC”

+GATSREG=1,0, AFAF123BC

OK

25.2. AT+GATSCON: (Dis)Connect GATT Server to another device

25.2.1. Description

(Dis)Connect GATT Server.

25.2.2. Format

Command	Possible response(s)
+GATSCON=<op>,<user_id>,<addr>,<direct>	+GATSCON: <op>,<result>,<user_id>,<addr>,<conn_id> OK / ERROR

25.2.3. Field

< op >:

0 disconnect

1 connect

<user_id>:

User id of gatt server, or the name of the gatt server.

A Hex value string, each char of it should in set { ‘0’~’9’,’a’~’f’,’A’~’F’ }

Max length of it is 32.

<addr>:

BT Address string, needn’t use “” double quotes, length should be 12.

<direct>:

0: non-direct

1: direct

<result>:

0 SUCCESS

Other un-success

<conn_id>:

The id of this connection.

OK / ERROR: at argument error will return ERROR, other will return OK.

NOTE

Each GATT Server AT can only send after the prior AT return OK/ERROR.

25.2.4. Example:

AT+GATSCON=1,"AFAF123BC",1234565b0102,1 // connect server to another device whose LAP is 0x123456, UAP is 0x5b, NAP is 0x0102.

+GATSCON: 1,0, AFAF123BC,1234565b0102,2

OK

25.3. URC +GATSCON: indicate server a connection's status

25.3.1. Description

Notify server a connection's status.

25.3.2. Format

URC
+GATSCON: <op>,<result>,<user_id>,<addr>,<conn_id>

25.3.3. Field

<op>:

0: disconnect

1: connect

<result>:

0: Success

Other: un-success

<user_id>:

User id of gatt server, or the name of the gatt server.

A Hex value string, each char of it should in set { '0'~'9','a'~'f','A'~'F' }

Max length of it is 32.

<addr>:

Address of peer. HEX value.

<conn_id>:

The id of this connection.

25.3.4. Example:

+GATSCON: 1,0,1809, 1234565b0102,33

25.4. AT+GATSS: add/remove a service

25.4.1. Description

Add or remove a service to or from a GATT Server.

25.4.2. Format

Command	Possible response(s)
+GATSS=<op>,<user_id>,<uuid>,<num_handles>,<is_primary>,<inst>	+GATSS: <op>,<result>,<user_id>,<uuid>,<inst>,<is_primary>,<service_handle> OK / ERROR

25.4.3. Field

< op >:

0 remove

1 add

<user_id>:

User id of gatt server, or the name of the gatt server.

A Hex value string, each char of it should in set { ‘0’~’9’,’a’~’f’,’A’~’F’ }

Max length of it is 32.

<uuid>:

The uuid of service, a string with hex value, max len is 32, min len is 4.

<num_handles>:

Number of handles of this service. Dec format.

<is_primary>

0: not primary

1: primary

<inst>

Instance id of this uuid. Dec format.

<result>:

0: Success

Other: un-success

<service_handle>:

Service handle of this service. Dec format.

OK / ERROR: at argument error will return ERROR, other will return OK.

NOTE

Each GATT Server AT can only send after the prior AT return OK/ERROR.

25.4.4. Example:

AT+GATSS=1,"aabb230","0123456789abcdefa",4,1,254

+GATSS: 1,0, aabb230, 0123456789abcdefa,254,1,32765

OK

25.5. AT+GATSI: add/remove a include service

25.5.1. Description

Add or remove an include service to or from service.

25.5.2. Format

Command	Possible response(s)
+GATSI=<op>,<user_id>,<service_handle>,<inc_service_handle>	+GATSI: <op>,<result>,<user_id>,<service_handle>,<inc_service_handle> OK/ERROR

25.5.3. Field

<op>:

0 remove

1 add

<user_id>:

User id of gatt server, or the name of the gatt server.

A Hex value string, each char of it should in set { ‘0’~’9’,’a’~’f’,’A’~’F’ }

Max length of it is 32.

<service_handle>:

Service handle of this service. Dec format.

< inc_service_handle >:

Service handle of this include service. Dec format.

<result>:

0: Success

Other: un-success

OK / ERROR: at argument error will return ERROR, other will return OK.

NOTE

Each GATT Server AT can only send after the prior AT return OK/ERROR.

25.5.4. Example:

```
AT+GATSI=1,"aabb230",1199,19900
+GATSI: 1,0, aabb230,1199,19900
OK
```

25.6. AT+GATSC: add/remove a characteristic to/from an existed service

25.6.1. Description

Add or remove an include service to or from service.

25.6.2. Format

Command	Possible response(s)
+GATSC=<op>,<user_id>,<service_handle>,<char_uuid>,<inst>,<prop>,<permission>	+GATSC: <op>,<result>,<user_id>,<service_handle>,<char_uuid>,<inst>,<char_handle> OK/ERROR

25.6.3. Field

<op>:

0 remove

1 add

<user_id>:

User id of gatt server, or the name of the gatt server.

A Hex value string, each char of it should in set { ‘0’~’9’,’a’~’f’,’A’~’F’ }

Max length of it is 32.

<service_handle>:

Service handle of this service. Dec format.

<char_uuid>:

The uuid of characteristic, a string with hex value, max len is 32, min len is 4.

<inst>

Instance id of this uuid. Dec format.

<prop>

prop of this characteristic. Dec format. (0 - 4294967295)

<permission>

permission of this characteristic. Dec format. (0 - 4294967295)

<result>:

0: Success

Other: un-success

< char_handle>:

Characteristic handle of this Characteristic . Dec format.

OK / ERROR: at argument error will return ERROR, other will return OK.

NOTE

Each GATT Server AT can only send after the prior AT return OK/ERROR.

25.6.4. Example:

AT+GATSC=1," aabb230",0099,"0123456789abcdef",2,12313,4444

+GATSC: 1,0, aabb230,99, "0123456789abcdef",2,9980

OK

25.7. AT+GATSD: add/remove a descriptor to/from an existed service

25.7.1. Description

Add or remove an descriptor to or from an existed characteristic.

25.7.2. Format

Command	Possible response(s)
+GATSD=<op>,<user_id>,<service_handle>,<desc_uuid>,<inst>,<permission>	+GATSD: <op>,<result>,<user_id>,<service_handle>,<desc_uuid>,<inst>,<desc_handle> OK/ERROR

25.7.3. Field

<op>:

0 remove

1 add

<user_id>:

User id of gatt server, or the name of the gatt server.

A Hex value string, each char of it should in set { ‘0’~’9’,’a’~’f’,’A’~’F’ }

Max length of it is 32.

<service_handle>:

Handle of characteristic . Dec format.

<desc_uuid>:

The uuid of descriptor, a string with hex value, max len is 32, min len is 4.

<inst>

Instance id of the descriptor uuid. Dec format.

<permission>

permission of this descriptor. Dec format. (0 - 4294967295)

<result>:

0: Success

Other: un-success

<desc_handle>:

Handle of this descriptor. Dec format.

OK / ERROR: at argument error will return ERROR, other will return OK.

NOTE

Each GATT Server AT can only send after the prior AT return OK/ERROR.

25.7.4. Example:

AT+GATSD=1," aabb230",99,"0123456789abcdef",2,12313

+GATSD: 1,0, aabb230,99, 0123456789abcdef,2,9999

OK

25.8. AT+GATSST: start/stop a service

25.8.1. Description

Start or stop a service.

25.8.2. Format

Command	Possible response(s)
+GATSST=<op>,<user_id>,<service_handle>,<transport>	+GATSST: <op>,<result>,<user_id>,<service_handle> OK/ERROR

25.8.3. Field

<op>:

0 stop

1 start

<user_id>:

User id of gatt server, or the name of the gatt server.

A Hex value string, each char of it should in set { ‘0’~’9’,’a’~’f’,’A’~’F’ }

Max length of it is 32.

<service_handle>:

Handle of service . Dec format.

<transport>

transport to start service.

0 : LE,

1 : BREDR,

2 : Dual

<result>:

0: Success

Other: un-success

OK / ERROR: at argument error will return ERROR, other will return OK.

NOTE

Each GATT Server AT can only send after the prior AT return OK/ERROR.

25.8.4. Example:

AT+GATSST=1," aabb230",99,1

+GATSST: 1,0, aabb230,9987

OK

25.9. AT+GATSIND: send an indication to a client

25.9.1. Description

Send an indication to a client.

25.9.2. Format

Command	Possible response(s)
+GATSIND=<user_id>,<conn_id>,<attr_handle>,<need_confirm>,<value>	+GATSIND: <result>,<user_id>,<conn_id>,<attr_handle>,<need_confirm> OK/ERROR

25.9.3. Field

<user_id>:

User id of gatt server, or the name of the gatt server.

A Hex value string, each char of it should in set { ‘0’~’9’,’a’~’f’,’A’~’F’ }

Max length of it is 32.

<conn_id>:

The id of this connection.

<attr_handle>:

Handle of attrbuite. Dec format.

<need_confirm>:

Need client confirm or not. 0: no, 1: yes.

<value>

The value need to be notify . Hex format.

<result>:

0: Success

Other: un-success

OK / ERROR: at argument error will return ERROR, other will return OK.

NOTE

Each GATT Server AT can only send after the prior AT return OK/ERROR.

25.9.4. Example:

```
AT+GATSIND="1806",1,1321,0,"abcdefghigk"
+GATSIND:0,1806,1,1321,1
OK
```

25.10. AT+GATSRSP: send response to a client

25.10.1. Description

send response to a client.

25.10.2. Format

Command	Possible response(s)
+GATSRSP=<user_id>,<result>,<conn_id>,<trans_id>,<attr_handle>,<value>	+GATSRSP: <result>,<user_id>,<conn_id>,<attr_handle>,<need_confirm> OK/ERROR

25.10.3. Field

<user_id>:

User id of gatt server, or the name of the gatt server.

A Hex value string, each char of it should in set { ‘0’~’9’,’a’~’f’,’A’~’F’ }

Max length of it is 32.

<result>:

Result of response for client request. 0-255

<conn_id>:

The id of this connection.

<trans_id>:

id of the current transaction. 0 - 255.

<attr_handle>:

Handle of attrbuite. Dec format.

<value>

The value need to be notified . Hex format.

<result>:

0: Success

Other: un-success

<need_confirm>:

Need client confirm or not. 0: no, 1: yes.

OK / ERROR: at argument error will return ERROR, other will return OK.

NOTE

Each GATT Server AT can only send after the prior AT return OK/ERROR.

25.10.4. Example:

```
AT+GATSRSP="1806",1,0,2,9999,"aaaaaaaaaaaaaaaaaaaa:"
+GATSRSP: 0,1806,1, 9999,0
OK/ERROR
```

25.11. +GATRREQ - URC: indicate server that a client has a read request

25.11.1. Description

Notify server a client's read request.

25.11.2. Format

URC
+GATRREQ: <user_id>,<conn_id>,<trans_id>,<addr>,<attr_handle>,<is_long>,<offset>

25.11.3. Field

<user_id>:

User id of gatt server, or the name of the gatt server.

A Hex value string, each char of it should in set { '0'~'9','a'~'f','A'~'F' }

Max length of it is 32.

<conn_id>:

The id of this connection.

<trans_id>:

id of the current transaction. 0 - 255.

<addr>:

Address of peer. HEX value.

<attr_handle>:

Handle of attrbuite. Dec format.

<is_long>:

Tell server that the request is one of several requests.

<offset>:

Offset of the request. 0 – 65535

25.11.4. Example:

+GATRREQ: 2202,3,4, 1234565b0102,10010,1,9980

25.12. +GATWREQ - URC: indicate server that a client has a write request

25.12.1. Description

Notify server a client's write request.

25.12.2. Format

URC
+GATWREQ: <user_id>,<conn_id>,<trans_id>,<addr>,<attr_handle>,<value>,<need_rsp>,<is_prepare>,<offset>

25.12.3. Field

<user_id>:

User id of gatt server, or the name of the gatt server.

A Hex value string, each char of it should in set { '0'~'9','a'~'f','A'~'F' }

Max length of it is 32.

<conn_id>:

The id of this connection.

<trans_id>:

id of the current transaction. 0 - 255.

<addr>:

Address of peer. HEX value.

<attr_handle>:

Handle of attribute. DEC format.

<value>

The value need to be write. Hex format.

<need_rsp>

Whether client need server's response. 1 yes, 0 no.

<is_prepare>

Whether or not server execute request immediately. 1 no, 2 yes..

<offset>:

Offset of the request. 0 – 65535

25.12.4. Example:

+GATWREQ: 2202,3,4, 1234565b0102,10010,ABABABABABABABABABA,1,1,0

25.13. +GATEWREQ - URC: indicate server that a client ask server to exec or cancel the write request indicated (trans_id) before

25.13.1. Description

Notify server to execute the request.

25.13.2. Format

URC
+GATEWREQ: <user_id>,<conn_id>,<trans_id>,<addr>,<cancel>

25.13.3. Field

<user_id>:

User id of gatt server, or the name of the gatt server.

A Hex value string, each char of it should in set { ‘0’~’9’,’a’~’f’,’A’~’F’ }

Max length of it is 32.

<conn_id>:

The id of this connection.

<trans_id>:

id of the current transaction. 0 - 255.

<addr>:

Address of peer. HEX value.

<cancel>:

To execute the transaction or cancel it. 1 cancel, 0 exec.

25.13.4. Example:

+GATEWREQ: 2202,3,4, 1234565b0102,1

26. BLE GATT CLIENT AT COMMAND

26.1. AT+GATCREG: (De)Register GATT Client

26.1.1. Description

(De)Register GATT Client with user_id..

26.1.2. Format

Command	Possible response(s)
+GATCREG=<op>,<user_id>	+GATCREG: 1, <result>,<user_id> OK/ERROR

26.1.3. Field

< op >:

0 deregister

1 register

<user_id>:

User id of gatt client, or the name of the gatt client.

A Hex value string, each char of it should in set { ‘0’~’9’,’a’~’f’,’A’~’F’ }

Max length of it is 32.

<result>:

0 SUCCESS

Other un-success

OK / ERROR: at argument error will return ERROR, other will return OK.

NOTE

Each GATT Server AT can only send after the prior AT return OK/ERROR.

26.1.4. Example:

AT+GATCREG=1,"AFAF123BC" //register a gatt client with user id “AFAF123BC”

+GATCREG=1,0, AFAF123BC

OK

26.2. AT+GATCSCAN: Start or Stop Scan

26.2.1. Description

Start or stop Scan.

26.2.2. Format

Command	Possible response(s)
+GATCSCAN=<op>,<user_id>	+GATCSCAN: 1,<result>,<user_id> OK/ERROR

26.2.3. Field

< op >:

0 deregister

1 register

<user_id>:

User id of gatt client, or the name of the gatt client.

A Hex value string, each char of it should in set { ‘0’~’9’,’a’~’f’,’A’~’F’ }

Max length of it is 32.

<result>:

0 SUCCESS

Other un-success

OK / ERROR: at argument error will return ERROR, other will return OK.

NOTE

Each GATT Server AT can only send after the prior AT return OK/ERROR.

26.2.4. Example:

AT+GATCSCAN=1,"AFAF123BC" //register a gatt client with user id “AFAF123BC”

+GATCSCAN=1,0, AFAF123BC

OK

26.3. +GATCSCANRESULT - URC: report scan result

26.3.1. Description

Notify server to execute the request.

26.3.2. Format

URC
+GATCSCANRESULT: <user_id>,<addr>,<RSSI>,<eir>

26.3.3. Field

<user_id>:

User id of gatt client, or the name of the gatt client.

A Hex value string, each char of it should in set { ‘0’~’9’,’a’~’f’,’A’~’F’ }

Max length of it is 32.

<addr>:

Address of peer. HEX value.

<RSSI>:

Signal strength. 0~255 means -127 ~ 127

<eir>:

raw eir data string. HEX format.

26.3.4. Example:

+GATCSCANRESULT: 2202, 1234565b0102,186,890FAB10887376257

26.4. AT+GATCCON: (Dis)Connect GATT Client to another device

26.4.1. Description

(Dis)Connect GATT Client.

26.4.2. Format

Command	Possible response(s)
+GATCCON=<op>,<user_id>,<addr>,<direct>	+GATCCON: <op>,<result>,<user_id>,<addr>,<conn_id> OK/ERROR

26.4.3. Field

< op >:

0 disconnect

1 connect

<user_id>:

User id of gatt client, or the name of the gatt client.

A Hex value string, each char of it should in set { ‘0’~’9’,’a’~’f’,’A’~’F’ }

Max length of it is 32.

<addr>:

BT Address string, needn’t use “” double quotes, length should be 12.

<direct>:

0: non-direct

1: direct

<result>:

0 SUCCESS

Other un-success

<conn_id>:

The id of this connection.

OK / ERROR: at argument error will return ERROR, other will return OK.

NOTE

Each GATT Server AT can only send after the prior AT return OK/ERROR.

26.4.4. Example:

AT+GATCCON=1,"AFAF123BC",1234565b0102,1 // connect server to another device whose LAP is 0x123456, UAP is 0x5b, NAP is 0x0102.

+GATCCON: 1,0, AFAF123BC,1234565b0102,2

OK

26.5. +GATCCON - URC: indicate client a connection's status

26.5.1. Description

Notify client a connection's status.

26.5.2. Format

URC
+GATCCON: <op>,<result>,<user_id>,<addr>,<conn_id>

26.5.3. Field

<op>:

0: disconnect

1: connect

<result>:

0: Success

Other: un-success

<user_id>:

User id of gatt client, or the name of the gatt client.

A Hex value string, each char of it should in set { '0'~'9','a'~'f','A'~'F' }

Max length of it is 32.

<addr>:

Address of peer. HEX value.

<conn_id>:

The id of this connection.

26.5.4. Example:

+GATSCON: 1,0,1809, 1234565b0102,33

26.6. AT+GATCRSH: Refresh peer's service

26.6.1. Description

(Dis)Connect GATT Client.

26.6.2. Format

Command	Possible response(s)
AT+GATCRSH=<user_id>,<btaddr>	+GATCRSH: <result>,<user_id> OK/ERROR

26.6.3. Field

<user_id>:

User id of gatt client, or the name of the gatt client.

A Hex value string, each char of it should in set { '0'~'9','a'~'f','A'~'F' }

Max length of it is 32.

<addr>:

BT Address string, needn't use "" double quotes, length should be 12.

<result>:

0 SUCCESS

Other un-success

OK / ERROR: at argument error will return ERROR, other will return OK.

26.6.4. Note

Each GATT AT can only send after the prior AT return OK/ERROR.

This CMD is only an pseudo interface, nothing will be done in BT Stack.

26.6.5. Example:

AT+GATCRSH="AFAF123BC",1234565b0102

+GATCRSH: 0, "AFAF123BC"

OK

"AT+GATCSS=<user_id>,<conn_id>,<service_uuid>,<inst>

+GATCSS: <result>,<user_id>,<conn_id>

OK/ERROR"

+GATCSS: <result>,<user_id>,<conn_id>,<service_uuid>,<inst>,<is_primary>

"AT+GATCGIS=<user_id>,<conn_id>,<service_uuid>,<service_inst>,<service_is_primary>[,<inc_service_uuid>,<inc_service_inst>,<inc_service_is_primary>]

+GATCGIS:
<result>,<user_id>,<conn_id>,<service_uuid>,<service_inst>,<service_is_primary>,<inc_service_uuid>,<inc_service_inst>,<inc_service_is_primary>

OK/ERROR"

"AT+GATCGC=<user_id>,<conn_id>,<service_uuid>,<service_inst>,<service_is_primary>[,<char_uuid>,<char_inst>]

+GATCGC:
<result>,<user_id>,<conn_id>,<service_uuid>,<service_inst>,<service_is_primary>,<char_uuid>,<char_inst>,<prop>

OK/ERROR"

"AT+GATCGD=<user_id>,<conn_id>,<service_uuid>,<service_inst>,<service_is_primary>,<char_uuid>,<char_inst>[,<desc_uuid>,<desc_inst>]

+GATCGD:
<result>,<user_id>,<conn_id>,<service_uuid>,<service_inst>,<service_is_primary>,<char_uuid>,<char_inst>,<desc_uuid>,<desc_inst>

OK/ERROR"

"AT+GATCRC=<user_id>,<conn_id>,<service_uuid>,<service_inst>,<service_is_primary>,<char_uuid>,<char_inst>,<auth_req>

+GATCRC:
<result>,<user_id>,<conn_id>,<service_uuid>,<service_inst>,<service_is_primary>,<char_uuid>,<char_inst>,<value>

OK/ERROR"

"AT+GATCWC=<user_id>,<conn_id>,<write_type>,<service_uuid>,<service_inst>,<service_is_primary>,<char_uuid>,<char_inst>,<value>,<auth_req>

+GATCWC:
<result>,<user_id>,<conn_id>,<service_uuid>,<service_inst>,<service_is_primary>,<char_uuid>,<char_inst>

OK/ERROR"

"AT+GATCRD=<user_id>,<conn_id>,<service_uuid>,<service_inst>,<service_is_primary>,<char_uuid>,<char_inst>,<desc_uuid>,<desc_inst>,<auth_req>

+GATCRD:
<result>,<user_id>,<conn_id>,<service_uuid>,<service_inst>,<service_is_primary>,<char_uuid>,<char_inst>,<desc_uuid>,<desc_inst>,<value>

OK/ERROR"

"AT+GATCWD=<user_id>,<conn_id>,<write_type>,<service_uuid>,<service_inst>,<service_is_primary>,<char_uuid>,<char_inst>,<desc_uuid>,<desc_inst>,<value>,<auth_req>

+GATCWD:
<result>,<user_id>,<conn_id>,<service_uuid>,<service_inst>,<service_is_primary>,<char_uuid>,<char_inst>,<desc_uuid>,<desc_inst>

OK/ERROR"

"AT+GATCEW=<user_id>,<conn_id>,<execute>

+GATCEW: <result>,<user_id>,<conn_id>

OK/ERROR"

"AT+GATCRN=1,<user_id>,<btaddr>,<service_uuid>,<service_inst>,<service_is_primary>,<char_uuid>,<char_inst>

+GATCRN: 1,
<result>,<user_id>,<btaddr>,<service_uuid>,<service_inst>,<service_is_primary>,<char_uuid>,<char_inst>

OK/ERROR"

"AT+GATCRN=0,<user_id>,<btaddr>,<service_uuid>,<service_inst>,<service_is_primary>,<char_uuid>,<char_inst>

+GATCRN: 0,
<result>,<user_id>,<btaddr>,<service_uuid>,<service_inst>,<service_is_primary>,<char_uuid>,<char_inst>

OK/ERROR"

"AT+GATCGDT=<user_id>,<btaddr>

+GATCGDT: <result>,<user_id>,<btaddr>,<dev_type>

OK/ERROR"

+GATCN:
<user_id>,<conn_id>,<btaddr>,<service_uuid>,<service_inst>,<service_is_primary>,<char_uuid>,<char_inst>,<is_notify>,<value>

"AT+GATCL=<user_id>,<start>

+GATCL=<user_id>,<result>

OK/ERROR

27. ASCII MODE AT COMMAND

27.1. AT+EIPRECVA: Receive ASCII data from socket

27.1.1. Description

Receive ASCII data from socket

27.1.2. Format

Command	Possible response(s)
+EIPRECVA=<socket id>	+EIPRECVA: <socket id>,<ret> <data> OK +ESOCK: <socket id>,<error> ERROR
+EISPRECVA=?	

27.1.3. Field

<socket id>:

Socket id returned by a successful ETL=1,… command

<ret>:

Number of bytes received from socket

<data>

Data received from socket.

<error>

Socket error description if failed.

27.2. AT+EIPRECVS: Receive a string from socket

27.2.1. Description

Receive a string from socket

27.2.2. Format

Command	Possible response(s)
+EIPRECVS=<socket id>	+EIPRECVS: <socket id>,<string> OK +ESOCK: <socket id>,<error> ERROR
+EISPRECVS=?	

27.2.3. Field

<socket id>:

Socket id returned by a successful ETL=1,… command

<string>:

Double quoted string received. String is formatted according to ”T-REC-V.25ter 5.4.2.x Numeric and string constants” indications.

<error>

Socket error description if failed.

27.3. AT+EIPSENDA: Send ASCII data by socket

27.3.1. Description

Send ASCII data by socket. The device responds to the command with the prompt <greater_than><space> and waits for the data to send. To complete the operations send Ctrl-Z char (0x1A hex); to exit without writing the message send ESC char (0x1B hex).

27.3.2. Format

Command	Possible response(s)
+EIPSENDA=<socket id>>[,<dest ip>,<dest port>]	+EIPSENDA: <socket id>,<ret> OK +ESOCK: <socket id>,<error> ERROR
+EISPSENDS=?	

27.3.3. Field

<socket id>:

Socket id returned by a successful ETL=1,… command

<dest ip>:

String to specify destination IP address (Only UDP socket).

<port>:

Destination port (Only UDP socket).

<ret>

The number of bytes successfully sent.

<error>

Socket error description if failed.

NOTE

The maximum number of bytes to send is 512 bytes; trying to send more data will cause the surplus to be discarded and lost.

27.4. AT+EIPSENDS: Send a string by socket

27.4.1. Description

Send a string by socket

27.4.2. Format

Command	Possible response(s)
+EIPSENDS=<socket id>,<string>[,<dest ip>,<dest port>]	+EIPSENDS: <socket id>,<ret> OK +ESOCK: <socket id>,<error> ERROR
+EISPSENDS=?	

27.4.3. Field

<socket id>:

Socket id returned by a successful ETL=1,… command

<string>:

Double quoted string to send. String must be formatted according to ”T-REC-V.25ter 5.4.2.x Numeric and string constants”.

<dest ip>:

String to specify destination IP address (Only UDP socket).

<port>:

Destination port (Only UDP socket).

<ret>

The number of bytes successfully sent.

<error>

Socket error description if failed.

28. FTP AT COMMAND

28.1. AT+EFTPAPP: Open an FTP data connection and starts sending a file to the server

28.1.1. Description

Receive Open an FTP data connection and starts sending a file to the server. If file doesn't exist, file is created, otherwise data is appended at the end of the existing file. Currently only online mode transfers are supported.

In online mode:

if the data connection succeeds, a CONNECT indication is sent. Afterward a NO CARRIER indication is sent when the socket is closed. Transfer is terminated when an escape sequence is detected, or when all <file_size> bytes are sent to server socket.

28.1.2. Format

Command	Possible response(s)
+EFTPAPP=<ftp_id>,<filename>[,<conn_mode>[,file_size]]	OK ERROR
+ EFTPAPP=?	

28.1.3. Field

<ftp_id>:

Id of the FTP client

<filename>:

Name of the file to transfer

<conn_mode>:

0 - online mode

<file_size>:

Size of the file to transfer.

28.2. AT+EFTPCLOSE: Close an opened FTP client

28.2.1. Description

Close an opened FTP client

28.2.2. Format

Command	Possible response(s)
+EFTPCLOSE=<ftp_id>	OK ERROR

+EFTPCLOSE=?	

28.2.3. Field

<ftp_id>:

Id of the FTP client to close

28.3. AT+EFTPCWD: Change current working directory

28.3.1. Description

Change current working directory

28.3.2. Format

Command	Possible response(s)
+EFTPCWD=<ftp_id>[,<dirname>]	OK ERROR
+EFTPCWD=?	

28.3.3. Field

<ftp_id>:

Id of the FTP client

<dirname>:

Name of the directory.

28.4. AT+EFTPDEL: Delete a file

28.4.1. Description

Delete a file

28.4.2. Format

Command	Possible response(s)
+EFTPDEL=<ftp_id>[,<dirname>]	OK ERROR
+EFTPDEL=?	

28.4.3. Field

<ftp_id>:

Id of the FTP client

<dirname>:

Name of the directory.

28.5. AT+EFTPGET: Open an FTP data connection and starts downloading a file from the server

28.5.1. Description

Open an FTP data connection and starts downloading a file from the server. Online mode transfer are supported.

In online mode:

if the data connection succeeds, a CONNECT indication is sent. afterward a NO CARRIER indication is sent when the socket is closed. Transfer is terminated when an escape sequence is detected, or when all bytes are received form the server.

28.5.2. Format

Command	Possible response(s)
+EFTPGET=<ftp_id>,<filename>[,<conn_mode>]	OK ERROR
+ EFTPGET=?	

28.5.3. Field

<ftp_id>:

Id of the FTP client

<filename>:

Name of the file to transfer

<conn_mode>

0 - online mode

28.6. AT+EFTPLIST: List the content of a specified directory or the properties of a specified file

28.6.1. Description

List the content of a specified directory or the properties of a specified file.

28.6.2. Format

Command	Possible response(s)
+EFTPLIST=<ftp_id>[,<name>]	OK ERROR
+EFTPCLOSE=?	

28.6.3. Field

<ftp_id>:

Id of the FTP

<name>

The name of a directory or a file. If not specified, commands list current working directory

28.7. AT+EFTPMKD: Create a new directory

28.7.1. Description

Create a new directory

28.7.2. Format

Command	Possible response(s)
+EFTPMKD=<ftp_id>[,<dirname>]	OK ERROR
+EFTPMKD=?	

28.7.3. Field

<ftp_id>:

Id of the FTP client

<dirname>:

Name of the directory.

28.8. AT+EFTPOPEN: Create a new directory

28.8.1. Description

Open a FTP client

28.8.2. Format

Command	Possible response(s)
+EFTPOPEN=<id>,<server:port>[,<username>,<password>]	+ EFTPOPEN=<ftp id> OK ERROR
+EFTPOPEN=?	

28.8.3. Field

<id>:

Data account id

<server:port>:

Host name or ip address and port of FTP server (factory default port 21)

<username>:

Username to use in FTP authentication

<password>:

Password to use in FTP authentication

28.9. AT+EFTPPUT: Open an FTP data connection and starts sending a file to the server

28.9.1. Description

Open an FTP data connection and starts sending a file to the server. If file doesn't exist, file is created, otherwise existing file is overwritten. Online mode transfers are supported.

In online mode:

if the data connection succeeds, a CONNECT indication is sent. Afterward a NO CARRIER indication is sent when the socket is closed. Transfer is terminated when an escape sequence is detected, or when all <file_size> bytes are sent to server.

28.9.2. Format

Command	Possible response(s)
+EFTPPUT=<ftp_id>,<filename>[,<conn_mode>[,file_size]]	OK ERROR
+ EFTPPUT=?	

28.9.3. Field

<ftp_id>:
Id of the FTP client

<filename>:
Name of the file to transfer

<conn_mode>:
0 - online mode

<file_size>:
Size of the file to transfer

28.10. AT+EFTPPWD: Print current working directory

28.10.1. Description

Print current working directory

28.10.2. Format

Command	Possible response(s)
+EFTPPWD=<ftp_id>	+ EFTPOPEN=<current directory> OK ERROR
+EFTPCWD=?	

28.10.3. Field

<ftp_id>:
Id of the FTP client

28.11. AT+EFTPRMD: Removing directory

28.11.1. Description

Remove a directory

28.11.2. Format

Command	Possible response(s)
+EFTPRMD=<ftp_id>[,<dirname>]	OK ERROR
+EFTPRMD=?	

28.11.3. Field

<ftp_id>:
Id of the FTP client

<dirname>:
Name of the directory.

29. TLS AT COMMAND

29.1. AT+ECERT: Manage certificates

29.1.1. Description

Manage certificates. Provides services for users in importing, exporting, retrieval and deletion of certificates on device

29.1.2. Format

Command	Possible response(s)
+ECERT=<op>…	
+ECERT=0,<cert id/label >	OK ERROR
+ECERT=1,<label>,<type>,<format>, <size>[,password]	+ECERT=<cert id>,<label> OK
+ECERT=2,<cert id/label >,<format>	+ECERT=< cert id>,<label><size> <data> OK
+ECERT=3[,<type>]	+ECERT=<cert id>,<label> OK
+ECERT=?	

29.1.3. Field

<op>:

0: delete certificate

1: store certificate

2: retrieve certificate

3: list certificates

<cert id/label>:

Certificate id -or- label

<label>:

A unique string assigned to certificate

<type>:

0: Personal Certificate (with Private Key)

1: CA Certificate

<format>

0: Auto detect format from data (Not applicable for personal certificate, personal certificates support only PKCS#12 encoding format)

1: DER format

2: PEM format

3: PKCS#12 format

NOTE

a. personal certificates support only PKCS#12 encoding format.
b. command AT+ECERT=1,… set the module in online mode waiting for data from serial port. If user specify the <size>, the module waits until all data is transmitted from host. <Size> parameter can be omitted only for PEM files, in this case the modem waits till a <CTRL-Z> or <ESC> character is received
c. Number of certificates: Only one server certificate can be stored and used at a time. To use another certificate, host MUST delete the first one and install new one.
d. Certificate size: The maximum certificate size supported is 8KB.

29.2. AT+ETLSCFG: Open a TLS connection over TCP connection identified by a socket id.

29.2.1. Description

Open a TLS connection over TCP connection identified by a socket id

29.2.2. Format

Command	Possible response(s)
ETLSCFG=<socket id>[,<cipher suite>[,<auth mode>[,<server cert id/label>[,<client cert id/label>]]]]	OK ERROR
+ETLSCFG=?	

29.2.3. Field

<socket id>:

Socket id of an already connected TCP client

<cipher suite>:

0: Default TLS v1.2 cipher suites

<auth mode>:

0: no verification [default].

1: server authentication

2: server authentication and client authentication (if requested by remote server)

<server cert id/label>:

Certificate Id or label of CA certificate to use in server authentication.

<client cert id/label>

Certificate Id or label of personal certificate to use in client authentication. personal certificate is also called as client certificate.

NOTE

a. Server cert id/label option: The server cert id/label option is not supported, so do not use it. SW searches certificates and uses the right one for the server being connected to.
b. ECDHE cipher suites: ECDHE cipher suites work only with servers that don't expect EC extensions in client hello.
c. Server Name Validation: Server name validation is not supported.
d. Server Name Indication: Server Name Indication (SNI) extension is not supported.
e. Data Transfer: The first unsolicited data transfer from server immediately after TLS handshake fails. The subsequent ones and other data transfers from client and server work.
f. Data transfer: Once *+ESOCK: <TCP socket id> READY RECV* URC indication is received, receive data from TLS socket using *at+eiprecv(a)(s)* sometimes returns WOULD BLOCK error. But the device sends another *READY RECV* indication in such cases and host should issue at+eiprecv command again to read data.

29.3. AT+ETLSCLOSE: Close a TLS connection created using ETLSCFG over a TCP connection identified by a socket id.

29.3.1. Description

Close a TLS connection created using ETLSCFG over a TCP connection identified by a socket id.

29.3.2. Format

Command	Possible response(s)
ETLSCLOSE=<socket id>	OK ERROR
+ETSCLOSE=?	

29.3.3. Field

<socket id>:

Socket id of TLS connection created using ETLSCFG over a TCP connection.

Note: Once TLS connection is closed, TCP connection also MUST be closed using AT+ETL=0,<socket id> command before starting a new TLS connection again.

30. DOCUMENT HISTORY

Revision	Date	Changes
0	2018-10-01	First issue
1	2018-10-10	SSL Chapter removed AT+CGEQREQ removed because not applicable to 2G product
2	2018-10-30	Updated GNSS commands section
3	2018-12-17	Added +CMUX command Updated GNSS section
4	2019-04-03	Added +ESAM command
5	2020-07-24	Added AT+ESLT; AT+EFTCM; AT+EFTCM; AT+EPWSC; AT+ECCCH; AT+ESACCH; AT+EPOF; AT+EADP; AT+ESIMS; AT+ICCID; AT+ETLLISTEN; AT#SLED; AT&V; AT+EGPAU; AT+EIND; AT+EPLPK; AT+IFC commands Added ASCII; added FTP; added TLS functions Added GPS GNSS command Added AT#MONI command
6	2020-09-17	Updated Chapter 11.1. AT+ESLP – Sleep Mode Removed commands AT+CVHU and AT+CNMI

SUPPORT INQUIRIES

Link to **www.telit.com** and contact our technical support team for any questions related to technical issues.

www.telit.com

Telit Communications S.p.A.
Via Stazione di Prosecco, 5/B
I-34010 Sgonico (Trieste), Italy

Telit IoT Platforms LLC
5300 Broken Sound Blvd, Suite 150
Boca Raton, FL 33487, USA

Telit Wireless Solutions Inc.
3131 RDU Center Drive, Suite 135
Morrisville, NC 27560, USA

Telit Wireless Solutions Co., Ltd.
8th Fl., Shinyoung Securities Bld.
6, Gukjegeumyung-ro8-gil, Yeongdeungpo-gu
Seoul, 150-884, Korea

Telit Wireless Solutions Ltd.
10 Habarzel St.
Tel Aviv 69710, Israel

Telit Wireless Solutions
Technologia e Servicos Ltda
Avenida Paulista, 1776, Room 10.C
01310-921 São Paulo, Brazil

[01.2017]

Mod. 0805 2017-01 Rev.6