灵龙芯
smart core

SCP5
AT Commands Manual

Rev. SCP5_AT_Commands_Manual_V1.3

Date: 2020-04-21

Status: Released

About the Document

History

Revision	Date	Author	Description
1.0	2019-10-10	Wang	Initial.
1.1	2019-11-15	Wang	1. Updated document format. 2. Added the following AT commands: AT+CGMI/AT+CGMM/AT+ECBCINFO/ AT+ECDNS/AT+ECDNSCFG. 3. Updated the parameter description for the following AT commands: AT+MQTTCREATE/AT+MQTTPUB/
1.2	2020-03-20	Wang	1. Add the following AT Commands: AT+ECPCFG/AT+ECSLEEP/ AT+ECDMCONFIG/AT+ECURC/ AT+ECPTWEDRXS/AT+ECADC 2. Add Socket solution B Command; 3. Update MQTT Command format; 4. Update COAP Command format;
1.3	2020-04-21	Wang	1. Add the following AT Commands: AT+ECIPR/AT+ECSNTP 2. Modify section 3.1.5AT+ECFREQ

Contents

About the Document **2**
Contents **3**
1 Introduction **1**
1.1 Definitions **1**
1.2 AT command syntax **1**
1.3 3GPP compliance **2**
2 General Control Commands **3**
2.1 Basic Commands（ITU-T V.250） **3**
2.1.1 ATE Command Echo 3
2.1.2 ATQ Result Code Suppression 3
2.1.3 ATI Display Product identification Information 4
2.2 3GPP Commands (27.007) **5**
2.2.1 AT+CFUN Set Phone Functionality 5
2.2.2 AT+CGSN Request Product Serial Number 6
2.2.3 AT+CGMR Request Manufacturer Revision 8
2.2.4 AT+CGMI Request Manufacturer Identification 8
2.2.5 AT+CGMM Request Manufacturer Model 9
2.2.6 AT+CMEE Report Mobile Termination Error 10
2.2.7 AT+COPS PLMN Selection 11
2.2.8 AT+CREG Network Registration 13
2.2.9 AT+CEREG EPS Network Registration Status 15
2.2.10 AT+CSQ Get Signal Quality 18
2.2.11 AT+CESQ Get Extended Signal Quality 19
2.2.12 AT+CPSMS Power Saving Mode Setting 20
2.2.13 AT+CEDRXS eDRX Setting 22
2.2.14 AT+CEDRXRDP eDRX Read Dynamic Parameters 24
2.2.15 AT+CCIOTOPT CIot Optimization Configuration 25
2.2.16 AT+CGCMOD PDP Context Modify 26
2.2.17 AT+CGATT PS Attach or Detach 27
2.2.18 AT+CGACT PDP Context Activate or Deactivate 28
2.2.19 AT+CGDATA Enter Data State 29
2.2.20 AT+CGDCONT Define a PDP Context 30
2.2.21 AT+CGCONTRDP PDP Context Read Dynamic Parameters 35
2.2.22 AT+CGEQOS Define EPS Quality of Service 37
2.2.23 AT+CGEQOSRDP EPS Quality of Service Read Dynamic Parameters 38
2.2.24 AT+CGTFT Traffic Flow Template 40
2.2.25 AT+CSODCP Sending of Originating Data Via The Control Plane 42
2.2.26 AT+CRTDCP Reporting of Terminating Data Via The Control Plane 44
2.2.27 AT+CGAPNRC APN Rate Control 45
2.2.28 AT+CGEREP Packet Domain Event Reporting 47

2.2.29 +CGEV Used to Indicate EPS PDN Connection and Bearer Resources Operations Status 48
2.2.30 AT+CGPADDR Show PDP Address(es) 49
2.2.31 AT+CSCON Signalling Connection Status 50
2.2.32 AT+CCLK Return Current Date and Time 51
2.2.33 AT+CIMI Request International Mobile Subscriber Identity 52
2.2.34 AT+CPIN Enter PIN 53
.2.35 AT+CLCK Facility Lock 54
2.2.36 AT+CPWD Change Password 55
2.2.37 AT+CSIM Generic SIM Access 56
2.2.38 AT+CRSM Restricted SIM 57
2.2.39 AT+CTZU Automatic Time Zone Update 59
2.2.40 AT+CTZR Time Zone Reporting 60
2.3 3GPP Commands (27.005) 62
2.3.1 AT+CMGS Send Message 62
2.3.2 AT+CSCA Service Center Address 64
2.3.3 AT+CMGF Message Format 65
2.3.4 AT+CSMP Set Text Mode Parameters 66
2.3.5 +CMT New Message Received 67
3 Extended Commands 68
3.1 EC General Commands 68
3.1.1 AT+ECBAND 68
3.1.2 AT+ECCFG 69
3.1.3 AT+ECPING 72
3.1.4 AT+ECIPERF 73
3.1.5 AT+ECFREQ 76
3.1.6 AT+ECRMFPLMN 77
3.1.7 AT+ECATTBEARER 78
3.1.8 AT+ECSENDDATA 79
3.1.9 +RECVNONIP 81
3.1.10 AT+ECPMUCFG 81
3.1.11 AT+ECSMSSEND 82
3.1.12 AT+ECRFSTAT 83
3.1.13 AT+ECRST 83
3.1.14 AT+ECPSMR 84
3.1.15 AT+ECPLMNS 85
3.1.16 AT+ECCESQS 86
3.1.17 AT+ECSTATUS 87
3.1.18 AT+ECICCID 91
3.1.19 AT+ECBCINFO 91
3.1.20 AT+ECDNS 92
3.1.21 AT+ECDNSCFG 93

3.1.22 AT+ECPCFG....94
3.1.23 AT+ECSLEEP....95
3.1.24 AT+DMCONFIG....96
3.1.25 AT+ECURC....97
3.1.26 AT+ECPTWEDRXS....98
3.1.27 AT+ECADC....101
3.1.28 AT+ECIPR....102
3.1.29 AT+ECSNTP....103
3.2 Socket Commands(Solution A)....105
3.2.1 AT+SKTCREATE....105
3.2.2 AT+SKTCONNECT....106
3.2.3 AT+SKTBIND....106
3.2.4 AT+SKTSEND....107
3.2.5 +SKTRECV....108
3.2.6 +SKTERR....109
3.2.7 AT+SKTSTATUS....109
S3.2.8 AT+SKTDELETE....110
3.3 LwM2M Commands....111
3.3.1 AT+LWM2MCREATE....111
3.3.2 AT+LWM2MDELETE....112
3.3.3 AT+LWM2MADDOBJ....112
3.3.4 AT+LWM2MDELOBJ....113
3.3.5 +LWM2MREAD....114
3.3.6 +LWM2MWRITE....114
3.3.7 +LWM2MEXECUTE....115
3.3.8 +LWM2MOBSERVE....116
3.3.9 AT+LWM2MREADCONF....117
3.3.10 AT+LWM2MWRITECONF....118
3.3.11 AT+LWM2MEXECUTECONF....119
3.3.12 AT+LWM2MNOTIFY....120
3.3.13 AT+LWM2MUPDATE....121
3.3.14 Summary of <err> Codes....122
3.4 CoAP Commands....123
3.4.1 AT+COAPCREATE....123
3.4.2 AT+COAPDEL....123
3.4.3 AT+COAPADDRES....124
3.4.4 AT+COAPHEAD....125
3.4.5 AT+COAPOPTION....126
3.4.6 AT+COAPSEND....127
3.4.7 AT+COAPDATASTATUS....128
3.4.8 AT+COAPCFG....129
3.4.9 AT+COAPALISIGN....130

3.4.10 +COAPURC 131
3.4.10 +COAPURC 132
3.5 MQTT Commands 134
3.5.1 AT+ECMTCFG 134
3.5.2 AT+ECMTOPEN 138
3.5.3 AT+ECMTCLOSE 139
3.5.4 AT+ECMTCONN 140
3.5.5 AT+ECMTDISC 141
3.5.6 AT+ECMTSUB 141
3.5.7 AT+ECMTUNS 142
3.5.8 AT+ECMTPUB 143
3.5.9 +ECMTSTAT 144
3.5.10 +ECMTRECV 145
3.6 HTTP Commands 146
3.6.1 AT+HTTPCREATE 146
3.6.2 AT+HTTPCON 147
3.6.3 AT+HTTPDESTROY 148
3.6.4 AT+HTTPSEND 149
3.6.5 +HTTPRESPH 150
3.6.6 +HTTPRESPC 151
3.6.7 +HTTPERR 151
3.7 OneNET Extension Commands 153
3.7.1 AT+MIPLCONFIG 153
3.7.2 AT+MIPLCREATE 153
3.7.3 AT+MIPLDELETE 154
3.7.4 AT+MIPLOPEN 155
3.7.5 AT+MIPLCLOSE 155
3.7.6 AT+MIPLADDOBJ 156
3.7.7 AT+MIPLDELOBJ 157
3.7.8 AT+MIPLNOTIFY 158
3.7.9 AT+MIPLREADRSP 160
3.7.10 AT+MIPLWRITERSP 161
3.7.11 AT+MIPLEXECUTERSP 162
3.7.12 AT+MIPLOBSERVERSP 164
3.7.13 AT+MIPLDISCOVERRSP 165
3.7.14 AT+MIPLPARAMETERRSP 166
3.7.15 AT+MIPLUPDATE 167
3.7.16 AT+MIPLVER 168
3.7.17 +MIPLREAD 168
3.7.18 +MIPLWRITE 169
3.7.19 +MIPLEXECUTE 170
3.7.20 +MIPLOBSERVE 171

3.7.20 +MIPLDISCOVER 172
3.7.21 +MIPLPARAMETER 172
3.7.22 +MIPLEVENT 173
3.7.23 Summary of <err> Codes 174
3.8 OceanConnect Extension Commands 175
3.8.1 AT+CTM2MVER 175
3.8.2 AT+CTM2MSETMOD 175
3.8.3 AT+CTM2MSETPM 177
3.8.4 AT+CTM2MREG 178
3.8.4 AT+CTM2MUPDATE 179
3.8.5 AT+CTM2MDEREG 180
3.8.6 AT+CTM2MSEND 180
3.8.7 AT+CTM2MCMDRSP 181
3.8.8 +CTM2MRECV 182
3.8.9 +CTM2M 183
3.8.10 +CTM2MCMD 184
3.9 Socket Commands(Solution B) 186
3.9.1 AT+ECSOCR 186
3.9.2 AT+ECSOST 187
3.9.3 AT+ECSOSTF 188
3.9.4 AT+ECQSOS 190
3.9.5 AT+ECSORF 191
3.9.6 AT+ECSOCO 192
3.9.7 AT+ECSOSD 193
3.9.8 AT+ECSOCL 194
3.9.9 AT+ECSONMI 195
3.9.10 AT+ECSONMIE 197
3.9.11 +ECSOCLI 199
3.9.12 +ECSOSTR 199
3.9.13 Summary of <err> Codes(Socket solution B) 200
4 Error Values 201

1 Introduction

1.1 Definitions

For the purposes of the present document, the following syntactical definitions apply

<CR> Carriage return character.

<LF> Linefeed character.

<...> Name enclosed in angle brackets is a syntactical element. Brackets themselves do not appear in the command line.

[...] Optional subparameter of a command or an optional part of TA information response is enclosed in square brackets. Brackets themselves do not appear in the command line. When subparameter is not given in parameter type commands, new value equals to its previous value. In action type commands, action should be done on the basis of the recommended default setting of the subparameter.

NO_SAVE The parameter of the current AT command will be lost, if module is rebooted or current AT command doesn't have parameter

AUTO_SAVE The parameter of the current AT command will be kept in NVRAM automatically and take in effect immediately, and it won't be lost if module is reboot

AUTO_SAVE_REBOOT The parameter of the current AT command will be kept in NVRAM automatically and take in effect after reboot, and it won't be lost if module is reboot

- '-' means this AT command doesn't care the parameter saving mode

1.2 AT command syntax

1.2.1 AT command type

Type	Format	Description
Test Command	AT+<cmd>=?	Check possible sub-parameter values
Read Command	AT+<cmd>?	Check current sub-parameter values
Set Command	AT+<cmd>=p1[,p2[,p3[…]]]	Set command
Execution Command	AT+<cmd>	Execution command

SmartCore use AT+EC<cmd> to implement self-extended command.

1.2.2 Command line

See figure 1 for general structure of a command line. Standardized basic commands are found only in ITU-T Recommendation V.250 [14]. The commands in this specification use syntax rules of extended commands. Every extended command has a test command (trailing =?) to test the existence of the command and to give information about the type of its subparameters. Parameter type commands also have a read command (trailing ?) to check the current values of subparameters. Action type commands do not store the values of any of their possible subparameters, and therefore do not have a read command.

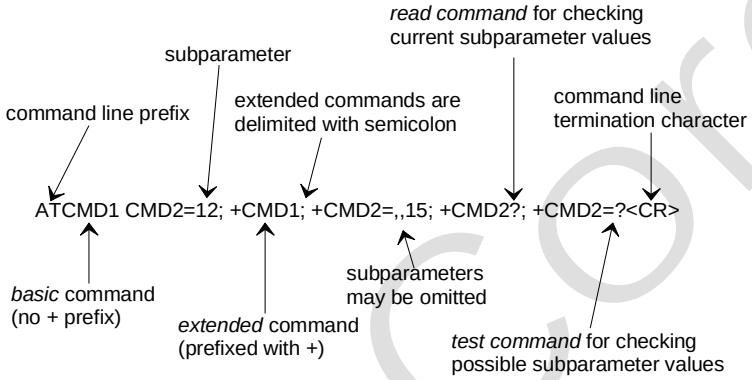

Figure 1: Basic structure of a command line

If all commands in a command line has been performed successfully, result code `<CR><LF>OK<CR><LF>` is sent from the TA to the TE.

If subparameter values of a command are not accepted by the TA (or command itself is invalid, or command cannot be performed for some reason), result code `<CR><LF>ERROR<CR><LF>` is sent to the TE and no subsequent commands in the command line are processed. ERROR response may be replaced by `+CME ERROR: <err>` (refer clause 4) when command was not processed due to an error related to MT operation.

1.3 3GPP compliance

Basic commands are compiled with ITU-T V.250(07/2003)

3GPP commands are complied with the 3GPP TS 27.007 V16.0.0 (2019-03) and 3GPP TS 27.005 V15.0.0 (2018-06).

2 General Control Commands

2.1 Basic Commands（ITU-T V.250）

2.1.1 ATE Command Echo

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

ATE	
Set Command ATE<value>	Response OK If there is any error, response: +CME ERROR: <err>
Maximum Response Time	5s
Parameter Saving Mode	AUTO_SAVE

Parameter

<value>	Integer type	
	0	DCE does not echo characters during command state and online command state
	1	DCE echoes characters during command state and online command state

Example

```
ATE0
OK
```

2.1.2 ATQ Result Code Suppression

The setting of this parameter determines whether or not the DCE transmits unsolicited result codes to the DTE. When result codes are being suppressed, unsolicited result is not transmitted.

NOTE: currently, this command is not fully implemented as defined in ITU-T V.250.

ATQ	
Set Command ATQ<value>	Response OK If there is any error, response:

	+CME ERROR: <err>
Maximum Response Time	5s
Parameter Saving Mode	AUTO_SAVE

Parameter

<value>	Integer type	
	0	DCE transmits unsolicited result codes
	1	Unsolicited result codes are suppressed and not transmitted. Note: a) If set to "1", all unsolicited result codes are all suppressed, including: PING/IPERF/LWM2M unsolicited result codes; b) If set to "1", only suppress the unsolicited result codes; And AT response/result codes are not suppressed;

Example

```
ATQ0
OK
```

2.1.3 ATI Display Product identification Information

The execution command returns product identification information. Please refer to Chapter 6 for possible <err> values.

ATI	
Execution Command ATI	Response SmartCore <Object Id> Revision:<revision> OK If there is any error, response: +CME ERROR: <err>
Maximum Response Time	5s

Parameter

<Object Id>	identifier of device type
<revision>	Revision of software release

Example

```
ATI
SmartCore
```

```
SCP5
Revision:SCP5NBR01A01

OK
```

2.2 3GPP Commands (27.007)

2.2.1 AT+CFUN Set Phone Functionality

Set command selects the level of functionality in the MT. Level "full functionality" is where the highest level of power is drawn. "Minimum functionality" is where minimum power is drawn.

Read command returns the current setting of <fun>.

Test command returns values supported by the MT as compound values.

AT+CFUN	
Set Command AT+CFUN=<fun>[,<rst>]	Response OK If there is any error, response: +CME ERROR: <err>
Read Command AT+CFUN?	Response +CFUN: <fun> OK
Test Command AT+CFUN=?	Response +CFUN: (list of supported <fun>s),(list of supported <rst>s) OK
Maximum Response Time	25s
Parameter Saving Mode	NO_SAVE

Parameter

<fun>	Integer type	
	0	Minimum functionality
	1	Full functionality
	4	Turn off RF
<rst>	Integer type	
	0	Do not reset the MT before setting it to <fun> power level. This shall always be defaulted when <rst> is not given.
	1	Reset the MT before setting it to <fun> power level. (not supported and

	will be ignored)

Example

```
AT+CFUN=?
+CFUN:(0,1,4),(0)
OK

AT+CFUN?
+CFUN:1
OK

AT+CFUN=1
OK
```

2.2.2 AT+CGSN Request Product Serial Number

The execution command returns the IMEI (International Mobile Station Equipment Identity) number and related information.

Test command returns values supported as a compound value.

AT+CGSN	
Set Command AT+CGSN=<snt>	Response When <snt>=0 and command successful: +CGSN: <sn> OK When <snt>=1 and command successful: +CGSN: <imei> OK When <snt>=2 and command successful: +CGSN: <imeisv> OK When <snt>=3 and command successful: +CGSN: <svn> OK If there is any error, response: +CME ERROR: <err>
Execution Command AT+CGSN	Response <sn> OK

	If there is any error, response: ERROR or +CME ERROR: <err>
Test Command AT+CGSN=?	Response +CGSN: (list of supported <snt>s) OK
Maximum Response Time	5s
Parameter Saving Mode	NO_SAVE

Parameter

<snt>	Integer type; indicating the serial number type that has been requested	
	0	Return <sn>
	1	Returns the IMEI(International Mobile station Equipment Identity)
	2	Returns the IMEISV(International Mobile station Equipment Identity and Software Version number)
	3	Returns the SVN(Software Version Number)
<sn>	One or more lines of information text determined by the MT manufacturer (not support now)	
<imei>	String type; in decimal format indicating the IMEI	
<imeisv>	String type; in decimal format indicating the IMEISV	
<svn>	String type; in decimal format indicating the current SVN which is a part of IMEISV	

Example

```
AT+CGSN=1
+CGSN: "863806040000440"

OK
AT+CGSN=2
+CGSN: "863806040000440"

OK

AT+CGSN=3
+CGSN: “01”

OK

AT+CGSN=?
+CGSN: (0,1,2,3)
OK
```

2.2.3 AT+CGMR Request Manufacturer Revision

The execution command returns the manufacturer revision. Now it returns the firmware revision and build time.

AT+CGMR	
Execution Command AT+CGMR	Response +CGMR: <"Board Version&&SDK Version&&EVB Version&&Compiled Time"> OK
Test Command AT+CGMR=?	Response OK
Maximum Response Time	5s
Parameter Saving Mode	NO_SAVE

Example

```
AT+CGMR
+CGMR:
-- Board: EC616_EVK --
-- SDK Version: EC616_SW_V001.000.xxx --
-- EVB Version: EC616_HW_V1.0 --
-- Compiled: Jul 23 2019 20:50:16 --

OK

AT+CGMR=?
OK
```

2.2.4 AT+CGMI Request Manufacturer Identification

The execution command returns manufacturer information. By default, it will return "SmartCore" on the standard platform.

AT+CGMI	
Execution Command AT+CGMI	Response <manufacturer>

	OK
Test Command AT+CGMR=?	Response OK
Maximum Response Time	5s
Parameter Saving Mode	NO_SAVE

Example

```
AT+CGMI
SmartCore

OK

AT+CGMI=?
OK
```

2.2.5 AT+CGMM Request Manufacturer Model

The execution command returns manufacturer model information.

AT+CGMI	
Execution Command AT+CGMM	Response <model> OK
Test Command AT+CGMM=?	Response OK
Maximum Response Time	5s
Parameter Saving Mode	NO_SAVE

Example

```
AT+CGMM
SCP5NB01-STD

OK

AT+CGMM=?
OK
```

2.2.6 AT+CMEE Report Mobile Termination Error

The write command disables or enables the use of final 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.

The read command returns the current setting of <n>.

The test command returns values supported as a compound value.

AT+CMEE	
Set Command AT+CMEE=<n>	Response OK
Read Command AT+CMEE?	Response +CMEE: <n> OK
Test Command AT+CMEE=?	Response +CMEE: (list of supported <n>s) OK
Maximum Response Time	5s
Parameter Saving Mode	AUTO_SAVE

Parameter

<mode>	Integer type	
	0	Disable +CME ERROR: <err> result code and use ERROR instead
	1	Enable +CME ERROR: <err> result code and use numeric <err> values
	2	Enable +CME ERROR: <err> result code and use verbose <err> values

Example

```
AT+CMEE=?
+CMEE: (0-2)

OK

AT+CMEE?
+CMEE: 1

OK

AT+CMEE=2
```

```
OK
```

2.2.7 AT+COPS PLMN Selection

The set command forces an attempt to select and register the network operator using the USIM card installed in the currently selected card slot. <mode> is used to select whether the selection is done automatically by the MT or is forced by this command to operator <oper> (it shall be given in format <format>) to a certain access technology, indicated in <AcT>. If the selected operator is not available, no other operator shall be selected (except <mode>=4). If the selected access technology is not available, then the same operator shall be selected in other access technology. The selected operator name format shall also apply to further read commands (AT+COPS?). <mode>=2 forces an attempt to deregister from the network. The selected mode affects all further network registration (e.g. after <mode>=2, MT shall be unregistered until <mode>=0 or 1 is selected). This command should be abortable when registration/deregistration attempt is made.

The read command returns the current mode, the currently selected operator and the current access technology. If no operator is selected, <format>, <oper> and <AcT> are omitted.

The test command returns a set of five parameters, each representing an operator present in the network. A set consists of an integer indicating the availability of the operator <stat>, long and short alphanumeric format of the operator's name, numeric format representation of the operator and access technology. Any of the formats may be unavailable and should then be an empty field. The list of operators shall be in order: home network, networks referenced in USIM or active application in the UICC (GSM or USIM) in the following order: HPLMN selector, user controlled PLMN selector, operator controlled PLMN selector and PLMN selector (in the USIM or GSM application), and other networks. After the operator list MT returns lists of supported <mode>s and <format>s. These lists shall be delimited from the operator list by two commas.

AT+COPS	

Set Command
`AT+COPS=<mode>[,<format>[,<oper>[,<AcT>]]]`
	Response
	`OK`
	If there is any error, response:
	`+CME ERROR: <err>`

Read Command
`AT+COPS?`
	Response
	`+COPS: <mode>[,<format>,<oper>][,<AcT>]`
	`OK`
	If there is any error, response:

	+CME ERROR: <err>
Test Command AT+COPS=?	Response +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) OK If there is any error, response: +CME ERROR: <err>
Maximum Response Time	305s
Parameter Saving Mode	AUTO_SAVE Note: Set <mode> to 0,3,4 will save to NVM

Parameter

<mode>	Integer type	
	0	Automatic(<oper> field is ignored)
	1	Manual(<oper> field shall be present, and <AcT> is optional) Note: <format> set to 2 is only supported in this case;
	2	Deregister from network
	3	Set only <format> (for read command AT+COPS?), do not attempt to register/deregister (<oper> and <AcT> fields are ignored);
	4	Manual/automatic (<oper> field shall be present); if manual selection fails, automatic mode (<mode>=0) is entered; Note: <format> set to 2 is only supported in this case;
<format>	Integer type	
	0	Long format alphanumeric <oper> (not support now)
	1	Short format alphanumeric <oper> (not support now)
	2	Numeric <oper>
<oper>	String type	
	<format> indicates if the format is alphanumeric or numeric; long alphanumeric format can be up to 16 characters long and short format up to 8 characters; numeric format is the GSM location area identification number which consists of a three BCD digit ITU-T country code coded, plus a two or three BCD digit network code, which is administration specific.	
<stat>	Integer type	
	0	Unknown
	1	Available
	2	Current
	3	Forbidden

<AcT>	Integer type; access technology selected	
	9	NB-IoT

Example

```
AT+COPS=1,2,"46000"
OK
```

2.2.8 AT+CREG Network Registration

Set command controls the presentation of an unsolicited result code +CREG: <stat> when <n>=1 and there is a change in the MT's circuit mode network registration status in GERAN/UTRAN/E-UTRAN, or unsolicited result code +CREG: <stat>[,[<lac>],[<ci>],[<AcT>]] when <n>=2 and there is a change of the network cell in GERAN/UTRAN/E-UTRAN. The parameters <AcT>, <lac> and <ci> are sent only if available. The value <n>=3 further extends the unsolicited result code with [,<cause_type>,<reject_cause>], when available, when the value of <stat> changes.

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>, if available, are returned only when <n>=2 and MT is registered in the network. The parameters [,<cause_type>,<reject_cause>], if available, are returned when <n>=3.

Test command returns values supported as a compound value.

AT+CREG	
Set Command AT+CREG=[<n>]	Response OK If there is any error, response: +CME ERROR: <err>
Read Command AT+CREG?	Response +CREG: <n>,<stat> [,[<tac>],[<ci>],[<AcT>][,<cause_type>,<reject_cause>]] OK
Test Command AT+CREG=?	Response +CREG:(list of supported <n>s) OK
Maximum Response Time	5s
Parameter Saving Mode	AUTO_SAVE

	Note: Set of <n> will save to NVM, and the default value is 0.

Parameter

<n>	Integer type	
	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>]
	3	Enable network registration, location information and cause value information unsolicited result code +CREG: <stat>[,[<lac>],[<ci>],[<AcT>] [,<cause_type>,<reject_cause>]]
<stat>	Integer type	
	0	not registered, MT is not currently searching a new operator to register to
	1	registered, home network (not applicable) Note: As no CS service for NB, this is not applicable for NB.
	2	not registered, but MT is currently searching a new operator to register to
	3	registration denied
	4	unknown (e.g. out of GERAN/UTRAN/E-UTRAN coverage)
	5	registered, roaming (not applicable) Note: As no CS service for NB, this is not applicable for NB.
	6	registered for "SMS only", home network (applicable only when <AcT> indicates E-UTRAN)
	7	registered for "SMS only", roaming (applicable only when <AcT> indicates E-UTRAN)
	8	attached for emergency bearer services only (not applicable) Note: As no emergency bearer for NB, this is not applicable for NB.
	9	registered for "CSFB not preferred", home network (not applicable) Note: As no CS service for NB, this is not applicable for NB.
	10	registered for "CSFB not preferred", roaming (not applicable) Note: As no CS service for NB, this is not applicable for NB.
<tac>	String type	
	two byte tracking area code	
<ci>	String type	
	Four byte cell ID in hexadecimal format	
<AcT>	Integer type; access technology of the serving cell	
	9	E-UTRAN(NB-S1 mode)

<cause_type>	Integer type; indicates the type of <reject_cause>	
	0	Indicates that <reject_cause> contains an MM cause value, see 3GPP TS 24.008 [8] Annex G.
	1	Indicates that <reject_cause> contains a manufacturer specific cause.
<reject_cause>	Integer type; contains the cause of the failed registration. The value is of type as defined by<cause_type>.	

Example

```
AT+CREG?
+CREG: 3,0


OK
```

2.2.9 AT+CEREG EPS Network Registration Status

The set command controls the presentation of an unsolicited result code +CEREG: <n>,<stat> when <n>=1 and there is a change in the MT's EPS network registration status in E-UTRAN, or unsolicited result code +CEREG: <n>,<stat>[,[<tac>],[<ci>],[<AcT>]] when <n>=2 and there is a change of the network cell in EUTRAN. The parameters <AcT>,<tac> and <ci> are provided only if available. The value <n>=3 further extends the unsolicited result code with [,<cause_type>,<reject_cause>], when available, when the value of <stat> changes.
If the UE requests PSM for reducing its power consumption, the set command controls the presentation of an unsolicited result code: +CEREG: <n>,<stat>[,[<tac>],[<ci>],[<AcT>][,[<cause_type>],[<reject_cause>][,[<Active-Time>],[<Periodic-TAU>]]]]. When <n>=4, the unsolicited result code will provide the UE with additional information for the active time value and the extended periodic TAU value if there is a change of the network cell in E-UTRAN. The value <n>=5 further enhances the unsolicited result code with <cause_type> and <reject_cause> when the value of <stat> changes. The parameters <AcT>, <tac>,<ci>,<cause_type>,<reject_cause>,<Active-Time> and <Periodic-TAU> are provided only if available.

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 <tac>, <ci> and <AcT>, if available, are returned only when <n>=2 and MT is registered in the network. The parameters [,<cause_type>,<reject_cause>], if available, are returned when <n>=3.

Test command returns values supported as a compound value.

AT+CEREG	
Set Command AT+CEREG=<n>	Response OK If there is any error, response: +CME ERROR: <err>
Read Command AT+CEREG?	When <n>=0,1,2 or 3 and command successful: +CEREG: <n>,<stat>[,[<tac>],[<ci>],[<AcT>[,<cause_type>,<reject_cause>]]] OK When <n>=4 or 5 and command successful: +CEREG: <n>,<stat>[,[<tac>],[<ci>],[<AcT>],[<cause_type>,<reject_cause>[,[<Active_Time>],[<Periodic_TAU>]]]] OK If there is any error, response: +CME ERROR: <err>
Test Command AT+CEREG=?	Response +CEREG: (list of supported <n>s) OK
Maximum Response Time	5s
Parameter Saving Mode	AUTO_SAVE Note: Set of <n> will save to NVM, and the default value is 0.

Parameter

<n>	Integer type;
0	disable network registration unsolicited result code
1	enable network registration unsolicited result code +CEREG: <stat>
2	enable network registration and location information unsolicited result code +CEREG: <stat>[,[<tac>],[<ci>],[<AcT>]]
3	enable network registration, location information and EMM cause value information unsolicited result code +CEREG: <stat>[,[<tac>],[<ci>],[<AcT>][,<cause_type>,<reject_cause>]]
4	For a UE that wants to apply PSM, enable network registration and location information unsolicited result code +CEREG: <stat>[,[<tac>],[<ci>],[<AcT>][,,[,[<Active-Time>],[<Periodic-TAU>]]]]
5	For a UE that wants to apply PSM, enable network registration, location information and EMM cause value information unsolicited result code +CEREG:

	<stat>[,[<tac>],[<ci>],[<AcT>][,[<cause_type>],[<reject_cause>][,[<Active-Time>],[<Periodic-TAU>]]]]
<stat>	Integer type; indicates the EPS registration status
	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 (e.g. out of E-UTRAN coverage)
	5 Registered, roaming
	6 Registered for “SMS only”, home network (not applicable)
	7 Registered for “SMS only”, roaming (not applicable)
	8 Attached for emergency bearer services only (not applicable)
	9 Registered for “CSFB not preferred”, home network (not applicable)
	1 Registered for “CSFB not preferred”, roaming (not applicable)
	0
<tac>	String type; two bytes tracking area code in hexadecimal format (e.g. “00C3” equals 195 in decimal)
<ci>	String type; four bytes E-UTRAN cell ID in hexadecimal format
<AcT>	Integer type; indicates the access technology of the serving cell
	0 GSM (not applicable)
	1 GSM Compact (not applicable)
	2 UTRAN (not applicable)
	3 GSM w/EGPRS (not applicable)
	4 UTRAN w/HSDPA (not applicable)
	5 UTRAN w/HSUPA (not applicable)
	6 UTRAN w/HSDPA and HSUPA (not applicable)
	7 E-UTRAN (not applicable)
	8 EC-GSM-IoT (A/Gb mode) (not applicable)
	9 E-UTRAN (NB-S1 mode)
<cause_type>	Integer type; indicates the type of <reject_cause>
	0 Indicates that <reject_cause> contains an EMM cause value
	1 Indicates that <reject_cause> contains a manufacturer-specific cause value
<reject_cause>	Integer type; contains the cause of the failed registration. The value is of type as defined by <cause_type>.
<Active_Time>	String type; one byte in an 8-bit format. Requested Active Time value (T3324) to be allocated to the UE. (e.g. "00100100" equals 4 minutes). Bits 5 to 1 represent the binary coded timer value. Bits 6 to 8 defines the timer value unit for the GPRS timer
<Perio	String type; one byte in an 8-bit format. Requested extended periodic TAU value (T3412) to

dic_TA	be allocated to the UE in E-UTRAN. (e.g. "01000111" equals 70 hours).
U>	Bits 5 to 1 represent the binary coded timer value
	Bits 6 to 8 define the timer value unit

Example

```
AT+CEREG?
+CEREG: 5,1,”5b49”,”0190271a”,9
OK
```

2.2.10 AT+CSQ Get Signal Quality

The execution command returns received signal quality <rssi> and channel bit error rate <ber> from the MT. Please refer to Chapter 4 for possible <err> values.

The test command returns values supported as compound values.

AT+CSQ	
Execution Command AT+CSQ	Response +CSQ: <rssi>,<ber> OK If there is any error, response: +CME ERROR: <err>
Test Command AT+CSQ=?	Response +CSQ: (list of supported <rssi>s),(list of supported <ber>s) OK
Maximum Response Time	5s
Parameter Saving Mode	NO_SAVE

Parameter

<rssi>	Integer type	
	0	-113dBm or less
	1	-111dBm
	2…30	-109…-53 dBm
	31	-51 dBm or greater
	99	not known or not detectable
<ber>	Integer type	

	0…7	as RXQUAL values in the table in 3GPP TS 45.008 [20] subclause 8.2.4
	99	not known or not detectable

Example

```
AT+CSQ
+CSQ: 27,0


OK
```

2.2.11 AT+CESQ Get Extended Signal Quality

The execution command returns received signal quality parameters. Since it only supports NB-IoT <rxlev> and <ber> are set to value 99, <rscp> and <ecno> is set to 255.

The test command returns values supported as compound values.

AT+CESQ	
Execution Command AT+CESQ	Response +CESQ: <rxlev>,<ber>,<rscp>,<ecno>,<rsrq>,<rsrp> OK If there is any error, response: +CME ERROR: <err>
Test Command AT+CESQ=?	Response +CESQ: (list of supported <rxlev>s),(list of supported <ber>s, list of supported <rscp>s),(list of supported <ecno>s, list of supported <rsrq>s),(list of supported <rsrp>s) OK
Maximum Response Time	5s
Parameter Saving Mode	NO_SAVE

Parameter

<rxlev>	Integer type; not supported by NB-IoT	
	99	not known or not detectable
<ber>	Integer type; not supported by NB-IoT	
	99	not known or not detectable
<rscp>	Integer type; not supported by NB-IoT	
	255	not known or not detectable
<ecno>	Integer type; not supported by NB-IoT	

	255	not known or not detectable
<rsrq>	Integer type	
	0	rsrq<-19.5dB
	1	-19.5dB<=rsrq<-19dB
	2	-19dB<=rsrq<-18.5dB
	:	: : :
	32	-4dB<=rsrq<-3.5dB
	33	-3.5dB<=-3 dB
	34	-3 dB <=rsrq
	255	not known or not detectable
<rsrp>	Integer type	
	0	rsrp<-149dBm
	1	-140dBm<=rsrp<-139dBm
	2	-139dBm<=rsrp<-138dBm
	:	: : :
	95	-46dBm<=rsrp<-45dBm
	96	-45dBm<=rsrp<-44dBm
	97	-44dBm<=rsrp
	255	not known or not detectable

Example

```
AT+CESQ
+CESQ: 99,99,255,255,26,56
OK
AT+CESQ=?
+CESQ: (99),(99),(255),(255),(0-34,255),(0-97,255)
OK
```

2.2.12 AT+CPSMS Power Saving Mode Setting

The set command controls the setting of the UE's power saving mode (PSM) parameters. The command controls whether the UE wants to apply PSM or not. Please refer to the unsolicited result codes provided by AT+CEREG for the active time value and the extended periodic TAU value that are allocated to the UE by the network in E-UTRAN.
A special form of the command can be given as AT+CPSMS=2. In this form, the use of PSM will be disabled and data for all parameters in the command +CPSMS will be removed.

The read command returns the current parameter values.

The test command returns the supported <mode>s and the value ranges for the requested extended

periodic TAU value in E-UTRAN and the requested Active Time value as compound values.

AT+CPSMS	
Set Command AT+CPSMS=<mode>[,<Requested_Periodic-RAU>[,<Requested_GPRS-READY-timer>[,<Requested_Periodic-TAU>[,<Requested_Active-Time>]]]]	Response OK If there is any error, response: +CME ERROR: <err>
Read Command AT+CPSMS?	Response +CPSMS: <mode>,[<Requested_Periodic-RAU>],[<Requested_GPRS-READY-timer>],[<Requested_Periodic-TAU>],[<Requested_Active-Time>]
Test Command AT+CPSMS=?	Response +CPSMS: (list of supported <mode>s),(list of supported <Requested_Periodic-RAU>s, list of supported <Requested_GPRS-READY-timer>s),(list of supported <Requested_Periodic-TAU>s, list of supported <Requested_Active-Time>s) OK
Maximum Response Time	5s
Parameter Saving Mode	AUTO_SAVE

Parameter

<mode>	Integer type; indicates to disable or enable the use of PSM in the UE	
	0	Disable the use of PSM
	1	Enable the use of PSM
	2	Disable the use of PSM and discard all parameters for PSM
<Requested_Periodic-RAU>	String type; not supported by NB-IoT	
<Requested_GPRS-READY-timer>	String type; not supported by NB-IoT	
<Requested_Periodic-TAU>	String type; one byte in an 8-bit format. Requested extended periodic TAU value (T3412) to be allocated to the UE in E-UTRAN. (e.g. "01000111" equals 70 hours). Bits 5 to 1 represent the binary coded timer value Bits 6 to 8 define the timer value unit as follows: Bits 8 7 6 0 0 0 Value is incremented in multiples of 10 minutes	

	0 0 1 Value is incremented in multiples of 1 hour 0 1 0 Value is incremented in multiples of 10 hours 0 1 1 Value is incremented in multiples of 2 seconds 1 0 0 Value is incremented in multiples of 30 seconds 1 0 1 Value is incremented in multiples of 1 minute 1 1 0 Value is incremented in multiples of 320 hours 1 1 1 Value indicates that the timer is deactivated
<Requested_Active-Time>	String type; one byte in an 8-bit format. Requested Active Time value (T3324) to be allocated to the UE. (e.g. "00100100" equals 4 minutes). Bits 5 to 1 represent the binary coded timer value. Bits 6 to 8 defines the timer value unit for the GPRS timer as follows: Bits 8 7 6 0 0 0 Value is incremented in multiples of 2 seconds 0 0 1 Value is incremented in multiples of 1 minute 0 1 0 Value is incremented in multiples of decihours 1 1 1 Value indicates that the timer is deactivated

Example

```
AT+CPSMS=1,,,,”00100010”
OK
AT+CPSMS?
+CPSMS: 1,,,,”00100010”
OK
AT+CPSMS=?
+CPSMS: (0,1,2),,,(“00000000”-“11111111”),(“00000000”-“11111111”)
OK
```

2.2.13 AT+CEDRXS eDRX Setting

The set command controls the setting of the UE’s eDRX parameters. It can be used to control whether the UE wants to apply eDRX or not, as well as the requested eDRX value for NB-IoT.
A special form of the command can be given as AT+CEDRXS=3. In this form, eDRX will be disabled and data for all parameters in AT+CEDRXS command will be removed.

The read command returns the current settings for each defined value of <AcT-type>.

The test command returns the supported <mode>s and the value ranges for the access technology and the requested eDRX value as compound values.

AT+CEDRXS	
Set Command AT+CEDRXS=<mode>,<AcT-type>[,<Requested_eDRX_value>]	Response OK If there is any error, response: +CME ERROR: <err>
Read Command AT+CEDRXS?	Response +CEDRXS: <AcT-type>,<Requested_eDRX_value> OK
Test Command AT+CEDRXS=?	Response +CEDRXS: (list of supported <mode>s),(list of supported <AcT-type>s),(list of supported <Requested_eDRX_value>s)
Maximum Response Time	5s
Parameter Saving Mode	AUTO_SAVE

Parameter

<mode>	Integer type; indicates to disable or enable the use of eDRX in the UE. This parameter is applicable to all specified types of access technology, i.e. the most recent setting of <mode> will take effect for all specified values of <AcT-type>.	
	0	Disable the use of eDRX
	1	Enable the use of eDRX
	2	Enable the use of eDRX and enable the unsolicited result code: +CEDRXP: <AcT-type>[,<Requested_eDRX_value>[,<NW-provided_eDRX_value>[,<Paging_time_window>]]]
	3	Disable the use of eDRX and discard all parameters for eDRX.
<AcT-type>	Integer type; indicates the type of access technology. AT+CEDRXS? is used to specify the relationship between the type of access technology and the requested eDRX value.	
	5	E-UTRAN(NB-S1 mode)
<Requested_eDRX_value>	String type; half a byte in a 4-bit format. (e.g."0010" equals 20.48 seconds)	

Example

```
AT+CEDRXS=1,5,"0010"
OK
AT+CEDRXS?
+CEDRXS: 5,"0010"
```

```
OK
AT+CEDRXS=?
+CEDRXS: (0,1,2,3), (5) ,(“0000”-“1111”)
OK
```

2.2.14 AT+CEDRXRDP eDRX Read Dynamic Parameters

The execution command returns <AcT-type>, <Requested_eDRX_value>, <NW-provided_eDRX_value> and <Paging_time_window>. If eDRX is used for the cell that the UE is currently registered to. If the cell that the UE is currently registered to is not using eDRX, <AcT-type>=0 is returned.

AT+CEDRXRDP	
Execution Command AT+CEDRXRDP	Response +CEDRXRDP: <AcT-type>[,<Requested_eDRX_value>[,<NW-provided_eDRX_value>[,<Paging_time_window>]]] OK
Test Command AT+CEDRXRDP=?	Response OK
Maximum Response Time	5s
Parameter Saving Mode	NO_SAVE

Parameter

<AcT-type>	Integer type; indicates the type of access technology. 5 NB IOT 0 No act or not using eDRX
<Requested_eDRX_value>	String type; half a byte in a 4-bit format. (e.g.”0010” equals 20.48 seconds)
<NW-provided_eDRX_value>	String type; half a byte in a 4-bit format. (e.g.”0010” equals 20.48 seconds)
<Paging_time_window>	String type; half a byte in a 4-bit format. (e.g.”0000” equals 2.56 seconds)

Example

```
AT+CEDRXRDP
+CEDRXRDP: 5,”0010”,”1101”,”0100”
OK
AT+CEDRXRDP=?
OK
```

2.2.15 AT+CCIOTOPT CIot Optimization Configuration

The set command controls which CIoT EPS optimizations the UE indicates as supported and preferred in the ATTACH REQUEST and TRACKING AREA UPDATE REQUEST messages. The command also allows reporting of the CIoT EPS optimizations that are supported by the network.
The set command is used also to control the unsolicited result code `+CCIOTOPTI`. An unsolicited result code `+CCIOTOPTI: <supported_Network_opt>` is used to indicate the supported CIoT EPS optimizations by the network.

The read command returns the current settings for supported and preferred CIoT EPS optimizations and the current status of unsolicited result code `+CCIOTOPTI`.

The test command returns values supported as compound values.

AT+CCIOTOPT	

Set Command
`AT+CCIOTOPT=<n>[,<support_UE_opt>[,<preferred_ue_opt>]]`
	Response
	`OK`
	If there is any error, response:
	`+CME ERROR: <err>`

Read Command
`AT+CCIOTOPT?`
	Response
	`+CCIOTOPT: <n>,<support_UE_opt>,<preferred_UE_opt>`
	`OK`

Test Command `AT+CCIOTOPT=?`	Response `+CCIOTOPT:` (list of supported <n>s),(list of supported <support_UE_opt>s),(list of supported <preferred_UE_opt>s)
Maximum Response Time	5s
Parameter Saving Mode	AUTO_SAVE

Parameter

`<n>`	Integer type; indicates the type of access technology.	
	`0`	Disable reporting
	`1`	Enable reporting
	`3`	Disable reporting and reset the parameters for CIoT EPS optimization to the default values.
`<support_UE_opt>`	Integer type; indicates the UE's support for CIoT EPS optimizations	

	1	Support for control plane CIoT EPS optimization
	2	Support for user plane CIoT EPS optimization (not support now)
	3	Support for both control plane CIoT EPS optimization and user plane CIoT EPS optimizations
<preferred_UE_opt>	Integer type; indicates the UE's preference for CIoT EPS optimizations	
	0	No preference
	1	Preference for control plane CIoT EPS optimization
	2	Preference for user plane CIoT EPS optimization

Example

```
AT+CCIOTOPT=?
+CCIOTOPT: (0,1,3),(1,3),(0,1,2)

OK

AT+CCIOTOPT?
+CCIOTOPT: 0,3,1

OK
```

2.2.16 AT+CGCMOD PDP Context Modify

The execution command is used to modify the specified PDP context with request to QoS profiles and TFTs. If the requested modification for any specified context cannot be achieved, an ERROR or +CME ERROR response is returned..

The test command returns a list of <cid>s associated with active contexts.

AT+CGCMOD	
Set Command AT+CGCMOD=<cid> (Note1)	Response OK If there is any error, response: +CME ERROR: <err>
Test Command AT+CGCMOD=?	Response +CGCMOD: (list of <cid>s associated with active contexts) OK
Maximum Response Time	70s
Parameter Saving Mode	NO_SAVE

Parameter

<cid>	Integer type; specifies a particular PDP context definition. <cid> values of 1-11 are supported.

Note1:

1) Not support to specify several <cid>s, just not support: AT+CGCMOD=<cid>,<cid>[,..]

Example

```
AT+CGCMOD=?
+CGCMOD: (5)

OK
```

2.2.17 AT+CGATT PS Attach or Detach

The set command is used to attach the MT to, or detach the MT from, the Packet Domain service. If the MT is already in the requested state, the command is ignored and the OK response is returned. If the requested state cannot be achieved, +CME ERROR response is returned. Any active PDP contexts will be automatically deactivated when the attachment state changes to detached.

The read command returns the current Packet Domain service state.

The test command is used for requesting information on the supported Packet Domain service states.

AT+CGATT	
Set Command AT+CGATT=<state>	Response OK If there is any error, response: +CME ERROR: <err>
Read Command AT+CGATT?	Response +CGATT: <state> OK
Test Command AT+CGATT=?	Response +CGATT: (list of supported <state>s) OK
Maximum Response Time	70s
Parameter Saving Mode	NO_SAVE

Parameter

<state>	Integer type; indicates the state of PS attachment.	
	0	Detached
	1	Attached

Example

```
AT+CGATT=?
+CGATT: (0,1)

OK

AT+CGATT?
+CGATT: 1

OK
```

2.2.18 AT+CGACT PDP Context Activate or Deactivate

The set command is used to activate or deactivate the specified PDP context. If any PDP context is already in the requested state, the state for that context remains unchanged. If the requested state for any specified context cannot be achieved, an +CME ERROR response is returned. If the MT is not PS attached when the activation form of the command is executed, the MT first performs a PS attach and then attempts to activate the specified contexts. If the attach fail, then the MT responds with +CME ERROR.

For EPS, if an attempt is made to disconnect the last PDN connection, then the MT responds with a +CME ERROR.

For EPS, the activation request for an EPS bearer resource will be answered by the network by either an EPS dedicated bearer activation or EPS bearer modification request. The request must be accepted by the MT before the PDP context can be set in to established state.

The read command returns the current activation states for all the defined PDP contexts.

The test command is used for requesting information on the supported PDP context activation states.

AT+CGACT	
Set Command AT+CGACT=<state>,<cid> (Note1)	Response OK If there is any error, response: +CME ERROR: <err>
Read Command	Response

AT+CGACT?	[+CGACT: <cid>,<state>] [<CR><LF>+CGACT:<cid>,<state>. […]] OK
Test Command AT+CGACT=?	Response +CGACT: (list of supported <state>s) OK
Maximum Response Time	70s
Parameter Saving Mode	NO_SAVE

Parameter

<state>	Integer type; indicates the activation state of PDP context activation.	
	0	Deactivated
	1	Activated
<cid>	Integer type; specifies a particular PDP context definition. Only one <cid> can be activated or deactivated at the same time. <cid> values of 1-11 are supported.	

Note1

1) <cid> must be specified, just not support to activate/deactivate all defined/activated bearers.
2) Not support to specify several <cid>s, just not support:
   AT+CGACT=<state>,<cid>,<cid>[,..].

Example

```
AT+CGACT=?
+CGACT: (0,1)

OK

AT+CGACT?
+CGACT: 5,1

OK
```

2.2.19 AT+CGDATA Enter Data State

The execution command causes the MT to perform whatever actions are necessary to establish communication between the TE and the network using one Packet Domain PDP types. This may include performing a PS attach and one PDP context activations. <cid> should be specified (see the

+CGDCONT) in order to provide the information needed for the context activation request.

The test command is used for requesting information on the supported <L2P> protocols.

AT+CGDATA	
Set Command AT+CGDATA=[<L2P>],<cid>	Response OK If there is any error, response: +CME ERROR: <err>
Test Command AT+CGDATA=?	Response +CGDATA: (list of supported <L2P>s) OK
Maximum Response Time	70s
Parameter Saving Mode	NO_SAVE

Parameter

<L2P>	String type; indicates the layer 2 protocol to be used between the TE and MT.	
	M-PT	SmartCore specified protocol – PDP Type, such as IP/IPV6/IPV4V6/Non-IP
<cid>	Integer type; specifies a particular PDP context definition. <cid> values of 1-11 are supported.	

Note:

1) This AT command is not fully followed the 3GPP 27.007, execution command just trigger MT to activate a PDP context, just same as: +CGACT=1,<cid>.
2) If PDP activation success, MT issues the result code: OK, not: CONNECT, as not support V.250 online data state now.

Example

```
AT+CGDATA=?
+CGDATA: "M-PT"

OK
```

2.2.20 AT+CGDCONT Define a PDP Context

The set command specifies PDP context parameter values for a PDP context identified by the (local) context identification parameter, <cid> It also allows the TE to specify whether security protected transmission of ESM information is requested, because the PCO can include information that requires

ciphering. There can be other reasons for the UE to use security protected transmission of ESM information, e.g. if the UE needs to transfer an APN. 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.
For EPS the PDN connection and its associated EPS default bearer is identified herewith.
A special form of the set command, `+CGDCONT=<cid>` causes the values for context number `<cid>` to become undefined.

The read command returns the current settings for each defined context.

The test command returns values supported as compound values

AT+CGDCONT	

Set Command
`AT+CGDCONT=<cid>[,<PDP_type>[,APN>[,<PDP_addr>[,<d_comp>[,<h_comp>[,<IPv4AddrAlloc>[,<request_type>[,<P-CSCF_discovery>[,<IM_CN_Signalling_Flag_Ind>[,<NSLPI>[,<securePCO>[,<IPv4_MTU_discovery>[,<Local_Addr_Ind>[,<Non-IP_MTU_discovery]]]]]]]]]]]]`
	Response
	`OK`
	If there is any error, response:
	`+CME ERROR: <err>`

Read Command
`AT+CGDCONT?`
	Response
	`+CGDCONT: <cid>,<PDP_type>,<APN>,<PDP_addr>,<d_comp>,<h_comp>[,<IPv4AddrAlloc>[,<request_type>[,<P-CSCF_discovery>[,<IM_CN_Signalling_Flag_Ind>[<NSLPI>[,<securePCO>[,<IPv4_MTU_discovery>[,<Local_Addr_Ind>[,<Non-IP_MTU_discovery>]]]]]]]]]`
	`[<CR><LF>+CGDCONT:<cid>,<PDP_type>,<APN>,<PDP_addr>,<d_comp>,<h_comp>[,<IPv4AddrAlloc>[,<request_type>[,<P-CSCF_discovery>[,<IM_CN_Signalling_Flag_Ind>[<NSLPI>[,<securePCO>[,<IPv4_MTU_discovery>[,<Local_Addr_Ind>[,<Non-IP_MTU_discovery>]]]]]]]]]]`
	`[…]]`
	`OK`

Test Command `AT+CGDCONT=?`	Response `+CGDCONT:` (range of supported

	<cid>s),<PDP_type>,,,(list of supported <d_comp>s),(list of supported <h_comp>s),(list of supported <IPv4AddrAlloc>s),(list of supported <request_type>s),(list of supported <PCSCF_discovery>s),(list of supported <IM_CN_Signalling_Flag_Ind>s),(list of supported <NSLPI>s),(list of supported <securePCO>s),(list of supported <IPv4_MTU_discovery>s),(list of supported <Local_Addr_Ind>s),(list of supported <NonIP_MTU_discovery>s),(list of supported <Reliable_Data_Service>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 <IPv4AddrAlloc>s),(list of supported <request_type>s),(list of supported <PCSCF_discovery>s),(list of supported <IM_CN_Signalling_Flag_Ind>s),(list of supported <NSLPI>s),(list of supported <securePCO>s),(list of supported <IPv4_MTU_discovery>s),(list of supported <Local_Addr_Ind>s),(list of supported <NonIP_MTU_discovery>s),(list of supported <Reliable_Data_Service>s) [...]]
Maximum Response Time	5s
Parameter Saving Mode	NO_SAVE

Parameter

<cid>	Integer type; 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 is returned by the test form of the command. <cid> values of 1-11 are supported.	
<PDP_type>	String type; specifies the type of packet data protocol. The default value is manufacturer specific.	
	IP	Internet Protocol
	IPV6	Internet Protocol, version 6
	IPV4V6	Virtual <PDP_type> introduced to handle dual IP stack UE capability
	Non-IP	Transfer of Non-IP data to external packet data

	network
<APN>	String type; a logical name that is used to select the GGSN or the external packet data network.
<PDP_addr>	String type; identifies the MT in the address space applicable to the PDP. (ignored with the set command)
<d_comp>	Integer type; don't need for NB-IOT
<h_comp>	Integer type; don't need for NB-IOT
<IPv4AddrAlloc>	Integer type; control how the MT/TA requests to get the IPv4 address information
	0 IPv4 address allocation through NAS signalling
	1 IPv4 address allocated through DHCP (Not support)
<request_type>	Integer type; indicates the type of PDP context activation request for the PDP context
	0 PDP context is for new PDP context establishment or for handover from a non-3GPP access network
	1 PDP context is for emergency bearer services (Not support)
	2 PDP context is for new PDP context establishment (Not support)
	3 PDP context is for handover from a non-3GPP access network (Not support)
<P-CSCF_discovery>	Integer type; influences how the MT/TA requests to get the P-CSCF address
	0 Preference of P-CSCF address discovery not influenced by +CGDCONT
	1 Preference of P-CSCF address discovery through NAS signaling (Not support)
	2 Preference of P-CSCF address discovery through DHCP (Not support)
<IM_CN_Signalling_Flag_Ind>	Integer type; indicates to the network whether the PDP context is for IM CN subsystem-related signaling only or not
	0 UE indicates that the PDP context is not for IM CN subsystem-related signaling only
	1 UE indicates that the PDP context is for IM CN subsystem-related signaling only (Not support)
<NSLPI>	Integer type; indicates the NAS signaling priority requested for this PDP context
	0 Indicates that this PDP context is to be activated with the value for the low priority indicator configured in the MT.
	1 Indicates that this PDP context is to be activated with the value for the low priority indicator set to "MS is not configured for NAS signaling low priority"

<securePCO>	Integer type; specifies if security protected transmission of PCO is requested or not	
	0	Security protected transmission of PCO is not requested
	1	Security protected transmission of PCO is requested (Not support)
<IPv4_MTU_discovery>	Integer type; influences how the MT/TA requests to get the IPv4 MTU size	
	0	Preference of IPv4 MTU size discovery not influenced by +CGDCONT
	1	Preference of IPv4 MTU size discovery through NAS signalling
<Local_Addr_Ind>	Integer type; indicates to the network whether or not the MS supports local IP address in TFTs	
	0	Indicates that the MS does not support local IP address in TFTs
	1	Indicates that the MS supports local IP address in TFTs (Not support)
<Non-IP_MTU_discovery>	Integer type; influences how the MT/TA requests to get the Non-IP MTU size	
	0	Preference of Non-IP MTU size discovery not influenced by +CGDCONT
	1	Preference of Non-IP MTU size discovery through NAS signalling

Example

```
AT+CGDCONT=5,"IP","CMNbiot.mnc004.mcc460.gprs",,1,1,0,0,0,0,0,0,1,0,1
OK

AT+CGDCONT?
+CGDCONT: 5,"IP","snbiot.mnc006.mcc460.gprs","10.212.162.96",0,0

OK

AT+CGDCONT=1,"ipv4v6"
OK

AT+CGDCONT?
+CGDCONT: 5,"IP","snbiot.mnc006.mcc460.gprs","10.212.154.7",0,0
+CGDCONT: 1,"IPV4V6",,,0,0

OK
```

```
AT+CGDCONT=?
+CGDCONT: (1-11),"IP",,,(0),(0),(0),(0),(0),(0),(0),(0),(0,1),(0),(0),(0)
+CGDCONT: (1-11),"IPV6",,,(0),(0),(0),(0),(0),(0),(0),(0),(0),(0),(0),(0)
+CGDCONT: (1-11),"IPV4V6",,,(0),(0),(0),(0),(0),(0),(0),(0),(0,1),(0),(0),(0)
+CGDCONT: (1-11),"Non-IP",,,(0),(0),(0),(0),(0),(0),(0),(0),(0),(0),(0,1),(0)

OK
```

2.2.21 AT+CGCONTRDP PDP Context Read Dynamic Parameters

The execution command returns the relevant information for an active non-secondary PDP context with the context identifier <cid>. If the MT has dual stack capabilities, at least one pair of lines with information is returned per <cid>. First one line with the IPv4 parameters followed by one line with the IPv6 parameters. If this MT with dual stack capabilities indicates more than more than two IP addresses of DNS servers, multiple of such pairs of lines are returned.
If the parameter <cid> is omitted, the relevant information for all active non secondary PDP contexts is returned.

The test command returns a list of <cid>s associated with active non secondary contexts.

AT+CGCONTRDP	
Set Command AT+CGCONTRDP[=<cid>]	Response [+CGCONTRDP: <cid>,<bearer_id>,<apn>[,<local_addr and subnet_mask>[,<gw_addr>[,<DNS_prim_addr>[,<DNS_sec_addr>[,<PCSCF_prim_addr>[,<PCSCF_sec_addr>[,<IM_CN_Signalling_Flag>[,<LIPA_indication>[,<IPv4_MTU>[,<WLAN_Offload>[,<Local_Addr_Ind>[,<NonIP_MTU>[,<Serving_PLMN_rate_control_value>]]]]]]]]]]]]] [<CR><LF>+CGCONTRDP: <cid>,<bearer_id>,<apn>[,<local_addr and subnet_mask>[,<gw_addr>[,<DNS_prim_addr>[,<DNS_sec_addr>[,<PCSCF_prim_addr>[,<PCSCF_sec_addr>[,<IM_CN_Signalling_Flag>[,<LIPA_indication>[,<IPv4_MTU>[,<WLAN_Offload>[,<Local_Addr_Ind>[,<NonIP_MTU>[,<Serving_PLMN_rate_control_value>]]]]]]]]]]]]] [...]] OK If there is any error, response:

	+CME ERROR: <err>
Test Command AT+CGCONTRDP=?	Response +CGCONTRDP: (list of <cid>s associated with active contexts) OK
Maximum Response Time	5s
Parameter Saving Mode	NO_SAVE

Parameter

<cid>	Integer type; specifies a particular non secondary PDP context definition. <cid> values of 1-11 are supported.
<bearer_id>	Integer type; identifies the bearer.
<apn>	String type; a logical name that was used to select the GGSN or the external packet data network.
<local_addr_and_subnet_mask>	String type; shows the IP address and subnet mask of the MT. The string is given as dot-separated numeric (0-255)
<DNS_prim_addr>	String type; the IP address of the primary DNS server.
<DNS_sec_addr>	String type; the IP address of the secondary DNS server.
<P_CSCF_prim_addr>	String type; shows the IP address of the primary P-CSCF server. (Not displayed)
<P_CSCF_sec_addr>	String type; shows the IP address of the secondary P-CSCF server. (Not displayed)
<IM_CN_Signalling_Flag>	Integer type; shows whether the PDP context is for IM CN subsystem-related signalling only or not. (Not displayed)
<LIPA_indication>	Integer type; indicates that the PDP context provides connectivity using a LIPA PDN connection. (Not displayed)
<IPv4_MTU>	Integer type; shows the IPv4 MTU size in octets
<WLAN_Offload>	Integer type; indicates whether traffic can be offloaded using the specified PDN connection via a WLAN or not. (Not displayed)
<Local_Addr_Ind>	integer type; indicates whether or not the MS and the network support local IP address in TFTs. (Not displayed)
<NonIP_MTU>	Integer type; shows the Non-IP MTU size in octets
<Serving_PLMN_rate_control_value>	Integer type; indicates the maximum number of uplink messages the UE is allowed to send in a 6 minute interval.

Example

```
AT+CGCONTRDP=5
+CGCONTRDP:
```

```
5,5,"CMNbiot.mnc004.mcc460.gprs","100.115.240.198.255.255.255.0","211.136.20.203","211.136
.17.107"
OK
```

2.2.22 AT+CGEQOS Define EPS Quality of Service

The set command allows the TE to specify the EPS Quality of Service parameters <cid>, <QCI>, [<DL_GBR> and <UL_GBR>] and [<DL_MBR> and <UL_MBR>] for a PDP context or Traffic Flows (see 3GPP TS 24.301 [83] and 3GPP TS 23.203 [85]).
A special form of the set command, +CGEQOS=<cid> causes the values for context number <cid> to become undefined.

The read command returns the current settings for each defined QoS.

The test command returns the ranges of the supported parameters as compound values.

AT+CGEQOS	
Set Command AT+CGEQOS=<cid>[,<QCI>[,<DL_GBR>,<UL_GBR>[,<DL_MBR>,<UL_MBR>]]]	Response OK If there is any error, response: +CME ERROR: <err>
Read Command AT+CGEQOS?	Response [+CGEQOS: <cid>,<QCI>,[<DL_GBR>,<UL_GBR>],[<DL_MBR>,<UL_MBR>]] [<CR><LF> +CGEQOS:<cid>,<QCI>,[<DL_GBR>,<UL_GBR>],[<DL_MBR>,<UL_MBR>][…]] OK
Test Command AT+CGEQOS=?	Response +CGEQOS: (range of supported <cid>s),(list of supported <QCI>s)
Maximum Response Time	5s
Parameter Saving Mode	NO_SAVE

Parameter

<cid>	Integer type; specifies a particular EPS Traffic Flows definition in EPS and PDP context definition. <cid> values of 1-11 are supported.

<QCI>	Integer type;specifies a class of EPS QoS	
	0	QCI is selected by network
	[1-4]	Value range for guaranteed bit rate Traffic Flows
	75	Value for guaranteed bit rate Traffic Flows
	[5-9]	Value range for non-guaranteed bit rate Traffic Flows
	79	Value for non-guaranteed bit rate Traffic Flows
<DL_GBR>	Integer type; indicates DL GBR in case of GBR QCI. The value is in kbit/s. This parameter is omitted for a non-GBR QCI	
<UL_GBR>	Integer type; indicates UL GBR in case of GBR QCI. The value is in kbit/s. This parameter is omitted for a non-GBR QCI	
<DL_MBR>	Integer type; indicates DL MBR in case of GBR QCI. The value is in kbit/s. This parameter is omitted for a non-GBR QCI	
<UL_MBR>	Integer type; indicates UL MBR in case of GBR QCI. The value is in kbit/s. This parameter is omitted for a non-GBR QCI	

Example

```
AT+CGEQOS=5,9,64,64,64,64
OK
```

2.2.23 AT+CGEQOSRDP EPS Quality of Service Read Dynamic Parameters

The execution command returns the quality of service parameters <QCI>,[<DL_GBR> and <UL_GBR>] and [<DL_MBR> and <UL_MBR>] of the active secondary or non-secondary PDP context associated to the provided context identifier <cid>.

If the parameter <cid> is omitted, the quality of service parameters for all secondary and non-secondary active PDP contexts are returned.

The test command returns a list of <cid>s associated with secondary or non-secondary active PDP contexts.

AT+CGEQOSRDP	
Set Command AT+CGEQOSRDP [=<cid>]	Response +CGEQOSRDP: <cid>,<QCI>,[<DL_GBR>,<UL_GBR>],[<DL_MBR>,<UL_MBR>][<DL_AMBR>,<UL_AMBR>] [<CR><LF>+CGEQOSRDP: <cid>,<QCI>,[<DL_GBR>,<UL_GBR>],[<DL_MBR>,<UL_MBR>][<D

	L_AMBR>,<UL_AMBR>] [...]] OK If there is any error, response: +CME ERROR: <err>
Test Command AT+CGEQOSRDP=?	Response +CGEQOSRDP: (list of <cid>s associated with active contexts) OK
Maximum Response Time	5s
Parameter Saving Mode	NO_SAVE

Parameter

<cid>	Integer type; specifies a particular PDP context definition (see the +CGDCONT commands). <cid> values of 1-11 are supported.	
<QCI>	Integer type;specifies a class of EPS QoS	
	0	QCI is selected by network
	[1-4]	Value range for guaranteed bit rate Traffic Flows
	75	Value for guaranteed bit rate Traffic Flows
	[5-9]	Value range for non-guaranteed bit rate Traffic Flows
	79	Value for non-guaranteed bit rate Traffic Flows
	[128-254]	Value range for Operator-specific QCIs
<DL_GBR>	Integer type; indicates DL GBR in case of GBR QCI. The value is in kbit/s. This parameter is omitted for a non-GBR QCI	
<UL_GBR>	Integer type; indicates UL GBR in case of GBR QCI. The value is in kbit/s. This parameter is omitted for a non-GBR QCI	
<DL_MBR>	Integer type; indicates DL MBR in case of GBR QCI. The value is in kbit/s. This parameter is omitted for a non-GBR QCI	
<UL_MBR>	Integer type; indicates UL MBR in case of GBR QCI. The value is in kbit/s. This parameter is omitted for a non-GBR QCI	
<DL_AMBR>	Integer type; indicates DL APN aggregate MBR. The value is in kbit/s.	
<UL_AMBR>	Integer type; indicates UL APN aggregate MBR. The value is in kbit/s.	

Example

```
AT+CGEQOSRDP
+CGEQOSRDP: 5,9


OK
```

2.2.24 AT+CGTFT Traffic Flow Template

This command allows the TE to specify a Packet Filter - PF for a Traffic Flow Template - TFT that is used in the GGSN in UMTS/GPRS and Packet GW in EPS for routing of packets onto different QoS flows towards the TE. A TFT consists of from one and up to 16 Packet Filters, each identified by a unique <packet filter identifier>. A Packet Filter also has an <evaluation precedence index> that is unique within all TFTs associated with all PDP contexts that are associated with the same PDP address.

The set command specifies a Packet Filter that is to be added to the TFT stored in the MT and used for the context identified by the (local) context identification parameter, <cid>.
A special form of the set command, +CGTFT=<cid> causes all of the Packet Filters in the TFT for context number <cid> to become undefined. At any time there may exist only one PDP context with no associated TFT amongst all PDP contexts associated to one PDP address. At an attempt to delete a TFT, which would violate this rule, an +CME ERROR response is returned.

The read command returns the current settings for all Packet Filters for each defined context.
The test command returns values supported as compound values. If the MT supports several PDP types, the parameter value ranges for each PDP type are returned on a separate line. TFTs shall be used for PDP-type IP only.

AT+CGTFT	
Set Command AT+CGTFT=<cid>,[<packet filter identifier>,<evaluation procedure index>[,remote address and subnet mask>[,<protocol number (ipv4)/next header (ipv6)>[,<local port range>[,<remote port range>[,<ipsec security parameter index (spi)>[,<type of service (tos) (ipv4) and mask>[,<flow label (ipv6)>[,<direction>]]]]]]]]	Response OK If there is any error, response: +CME ERROR: <err>
Read Command AT+CGTFT?	Response [+CGTFT: <cid>,<packet filter identifier>,<evaluation precedence index>,<remote address and subnet mask>,<protocol number (ipv4) / next

	header (ipv6)>,<local port range>,<remote port range>,<ipsec security parameter index (spi)>,<type of service (tos) (ipv4) and mask /traffic class (ipv6) and mask>,<flow label (ipv6)>,<direction>] [<CR><LF>+CGTFT: <cid>,<packet filter identifier>,<evaluation precedence index>,<remote address and subnet mask>,<protocol number (ipv4) / next header (ipv6)>,<local port range>,<remote port range>,<ipsec security parameter index (spi)>,<type of service (tos) (ipv4) and mask /traffic class (ipv6) and mask>,<flow label (ipv6)>,<direction> […]] OK If there is any error, response: +CME ERROR: <err>
Test Command AT+CGTFT=?	Response +CGTFT: (list of supported<cid>s),(list of supported <packet filter identifier>s),(list of supported <evaluation precedence index>s),(list of supported <remote address and subnet mask>s),(list of supported <protocol number (ipv4) / next header (ipv6)>s),(list of supported <local port range>s),(list of supported <remote port range>s),(list of supported <ipsec security parameter index (spi)>s),(list of supported <type of service (tos) (ipv4) and mask / traffic class (ipv6) and mask>s),(list of supported <flow label (ipv6)>s),(list of supported <direction>s) OK
Maximum Response Time	5s
Parameter Saving Mode	NO_SAVE

Parameter

<cid>	Integer type; Specifies a particular PDP context definition <cid> values of 1-11 are supported.
<packet filter identifier>	Integer type; Value range is from 1 to 16.

<evaluation precedence index>	Integer type; The value range is from 0 to 255.
<remote address and subnet mask>	string type; The string is given as dot-separated numeric (0-255)
<protocol number (ipv4) / next header (ipv6)>	Integer type; Value range is from 0 to 255.
<local port range>	string type; The string is given as dot-separated numeric (0-65535)
<remote port range>	string type; The string is given as dot-separated numeric (0-65535)
<ipsec security parameter index>	Integer type; numeric value in hexadecimal format
<type of service (tos) (ipv4) and mask / traffic class (ipv6) and mask>	string type; The string is given as dot-separated numeric (0-255)
<flow label (ipv6)>	Integer type ;numeric value in hexadecimal format
<direction>	Integer type. Specifies the transmission direction in which the packet filter shall be applied.
	0 Pre-Release 7 TFT filter
	1 Uplink
	2 Downlink
	3 Birectional (Up & Downlink)

Example

```
AT+CGTFT=5,2,6,"32.1.11.160.0.0.0.0.0.0.0.0.0.0.0.0.255.255.255.255.0.0.0.0.0.0.0.0.0.0.0.0",17,"60001.60001","60350.60450",,168.252,,1
OK
```

2.2.25 AT+CSODCP Sending of Originating Data Via The Control Plane

The set command is used by the TE to transmit data over control plane to network via MT. Context identifier <cid> is used to link the data to particular context.
This command optionally indicates that the application on the MT expects that the exchange of data:
- will be completed with this uplink data transfer; or
- will be completed with the next received downlink data.
This command also optionally indicates whether or not the data to be transmitted is an exception data.
This command causes transmission of an ESM DATA TRANSPORT message, as defined in 3GPP TS 24.301.

AT+CSODCP	
Set Command AT+CSODCP=<cid>,<cpdata_length>,<cpdata>[,<RAI>[,<type_of_user_data>]]	Response OK If there is any error, response: +CME ERROR: <err>
Test Command AT+CSODCP=?	Response +CSODCP: (range of supported <cid>s),(maximum number of octets of user data indicated by <cpdata_length>),(list of supported <RAI>s),(list of supported <type_of_user_data>s)
Maximum Response Time	5s
Parameter Saving Mode	NO_SAVE

Parameter

<cid>	Integer type; specifies a particular PDP context definition. <cid> values of 0-11 are supported. Note: If <cid> set to 0, just use current default bearer to send this originating data.	
<cpdata_length>	Integer type. Indicates the number of octets of the <cpdata> information element. The max length is 950.	
<cpdata>	string of octets.	
<RAI>	Integer type. Indicates the value of the release assistance indication	
	0	No information available
	1	The MT expects that exchange of datawill be completed with the transmission of the ESM DATA TRANSPORT message.
	2	The MT expects that exchange of data will be completed with the receipt of an ESM DATA TRANSPORT message.
<type_of_user_data>	Integer type. Indicates whether the user data that is transmitted is regular or exceptional.	
	0	Regular data
	1	Exception data

Example

```
AT+CSODCP=0,20,"A1B2C3E4F50011223344A1B2C3E4F50011223344”,0,0
OK
```

2.2.26 AT+CRTDCP Reporting of Terminating Data Via The Control Plane

The set command is used to enable and disable reporting of data from the network to the MT that is transmitted via the control plane in downlink direction. If reporting is enabled, the MT returns the unsolicited result code `+CRTDCP: <cid>,<cpdata_length>,<cpdata>` when data is received from the network.

AT+CRTDCP	

Set Command
`AT+CRTDCP=[<reporting>]`
	Response
	`OK`
	If there is any error, response:
	`+CME ERROR: <err>`

Read Command
`AT+CRTDCP?`
	Response
	`+CRTDCP: <reporting>`
	`OK`

Test Command `AT+CRTDCP=?`	Response `+CRTDCP:` (list of supported `<reporting>`s),(range of supported `<cid>`s),(maximum number of octets of user dataindicated by `<cpdata_length>`) OK
Maximum Response Time	5s
Parameter Saving Mode	NO_SAVE

Parameter

`<reporting>`	Integer type, controlling reporting of mobile terminated control plane data events	
	`0`	Disable reporting of MT control plane data.
	`1`	Enable reporting of MT control plane data by the unsolicited result code +CRTDCP.
`<cid>`	Integer type. A numeric parameter which specifies a particular PDP context or EPS bearer context definition. The `<cid>` parameter is local to the TE-MT interface and identifies the PDP or EPS bearer contexts which have been setup via AT command (see the `+CGDCONT` commands). `<cid>` values of 1-11 are supported.	

<cpdata_length>	Integer type. Indicates the number of octets of the <cpdata> information element. When there is no data to transmit, the value shall be set to zero.
<cpdata>	string of octets. Contains the user data container contents (refer 3GPP TS 24.301 [83] subclause 9.9.4.24). When there is no data to transmit, the <cpdata> shall be an empty string (""). And support “HEX” character format type.

Example

```
AT+CRTDCP=0
OK

AT+CRTDCP?
+CRTDCP: 0

OK
AT+CRTDCP=1
OK

AT+CRTDCP?
+CRTDCP: 1

OK
```

2.2.27 AT+CGAPNRC APN Rate Control

This execution command returns the APN rate control parameters (see 3GPP TS 24.008 [8]) associated to the provided context identifier <cid>. If the parameter <cid> is omitted, the APN rate control parameters for all active PDP contexts are returned.
The test command returns a list of <cid>s associated with secondary and non secondary active PDP contexts.

AT+CGAPNRC	
Set Command AT+CGAPNRC[=<cid>]	Response +CGAPNRC: <cid>[,<Additional_exception_reports>[,<Uplink_time_unit>[,<Maximum_uplink_rate>]]] [<CR><LF>+CGAPNRC: <cid>[,<Additional_exception_reports>[,<Uplink_time_unit>[,<Maximum_uplink_rate>]]] [...]]

	OK If there is any error, response: +CME ERROR: <err>
Test Command AT+CGAPNRC=?	Response +CGAPNRC: (list of <cid>s associated with active contexts) OK
Maximum Response Time	5s
Parameter Saving Mode	NO_SAVE

Parameter

<cid>	Integer type; specifies a particular PDP context definition (see the +CGDCONT commands). <cid> values of 1-11 are supported.	
<Additional_exception_reports>	Integer type; indicates whether or not additional exception reports are allowed to be sent when the maximum uplink rate is reached. This refers to bit 4 of octet 1 of the APN rate control parameters IE as specified in 3GPP TS 24.008 [8] subclause 10.5.6.3.2.	
	0	Additional_exception_reports at maximum rate reached are not allowed to be sent.
	1	Additional_exception_reports at maximum rate reached are allowed to be sent.
<Uplink_time_unit>	Integer typ; specifies the time unit to be used for the maximum uplink rate. This refers to bits 1 to 3 of octet 1 of the APN rate control parameters IE as specified in 3GPP TS 24.008 [8] subclause 10.5.6.3.2.	
	0	unrestricted
	1	minute
	2	hour
	3	day
	4	week
<Maximum_uplink_rate>	Integer type; specifies the maximum number of messages the UE is restricted to send per uplink time unit. This refers to octet 2 to 4 of the APN rate control parameters IE as specified in 3GPP TS 24.008 [8] subclause 10.5.6.3.2	

Example

```
AT+CGAPNRC=?
+CGAPNRC: (5)

OK
```

2.2.28 AT+CGEREP Packet Domain Event Reporting

Set command enables or disables sending of unsolicited result codes, +CGEV: XXX from MT to TE in the case of certain events occurring in the Packet Domain MT or the network. <mode> controls the processing of unsolicited result codes specified within this command. <bfr> controls the effect on buffered codes when <mode> 1 or 2 is entered. If a setting is not supported by the MT, ERROR or +CME ERROR: is returned.

Read command returns the current mode and buffer settings.

Test command returns the modes and buffer settings supported by the MT as compound values.

AT+CGEREP	
Set Command AT+CGEREP=<mode>[,<bfr>]	Response OK If there is any error, response: +CME ERROR: <err>
Read Command AT+CGEREP?	Response +CGEREP: <mode>,<bfr> OK
Test Command AT+CGEREP=?	Response +CGEREP: (list of supported <mode>s),(list of supported <bfr>s) OK
Maximum Response Time	5s
Parameter Saving Mode	AUTO_SAVE

Parameter

<mode>	integer type;	
	0	buffer unsolicited result codes in the MT; if MT result code buffer is full, the oldest ones can be discarded.No codes are forwarded to the TE. Note: default value
	1	discard unsolicited result codes when MT-TE link is reserved (e.g. in on-line data mode); otherwise forward them directly to the TE
<bfr>	integer type;	
	0	MT buffer of unsolicited result codes defined within this command is cleared when <mode> 1 ; Only it now

Example

```
AT+CGEREP=1,0

OK

AT+CGEREP?
+CGEREP: 1,0

OK

AT+CGEREP=?
+CGEREP: (0,1),(0)

OK
```

2.2.29 +CGEV Used to Indicate EPS PDN Connection and Bearer Resources Operations Status

This is an unsolicited message to indicate EPS PDN connection and bearer resources operations status

+CGEV
+CGEV: <xxx>

Parameter

+CGEV: NW PDN DEACT <cid>	The network has forced a context deactivation.
+CGEV: ME PDN DEACT <cid>	The mobile termination has forced a context deactivation.
+CGEV: ME PDN ACT <cid>[,<reason>]	The ME has activated a context.
+CGEV: NW MODIFY <cid>,<change_reason>,<event_type>	The network has modified a context.
+CGEV: ME MODIFY <cid>,<change_reason>,<event_type>	The mobile termination has modified a context.

Parameter

<cid>	The format is found in command +CGDCONT. <cid> values of 1-11 are supported.	
<pdnReason>	0	IPV4 only allowed

	1	IPV6 only allowed
	2	Single address bearer only allowed
	3	Single address bearer only allowed and active second bearer failed
	4	No reason
<bearerType>	0	NULL
	1	default
	2	Dedicated (Not application)
<change_reason>	Integer type; a bit map that indicates what kind of change occurred.the value is determined by summing all the applicable bits.	
	Bit 1	TFT changed
	Bit 2	Qos changed
	Bit 3	WLAN Offload changed

Example

```
+CGEV: ME PDN ACT 5,0
```

2.2.30 AT+CGPADDR Show PDP Address(es)

The execution command returns a list of PDP addresses for the specified context identifiers. If no <cid> is specified, the addresses for all defined contexts are returned.

The test command returns a list of defined <cid>s

AT+CGPADDR	
Set Command AT+CGPADDR [=<cid>]	Response +CGPADDR: <cid>[,<PDP_addr_1>[,<PDP_addr_2>]] [<CR><LF>+CGPADDR:<cid>,[<PDP_addr_1>[,<PDP_addr_2>]]] [...] OK If there is any error, response: +CME ERROR: <err>
Test Command AT+CGPADDR=?	Response +CGPADDR: (list of defined <cid>s) OK
Maximum Response Time	5s
Parameter Saving Mode	NO_SAVE

Parameter

<cid>	Integer type; specifies a particular PDP context definition (see the

	+CGDCONT commands). <cid> values of 1-11 are supported.
<PDP_addr_1> <PDP_addr_2>	<PDP_addr_1> and <PDP_addr_2>: each is a string type that identifies the MT in the address space applicable to the PDP. Both <PDP_addr_1> and <PDP_addr_2> are omitted if none is available. Both <PDP_addr_1> and <PDP_addr_2> are included when both IPv4 and IPv6 addresses are assigned, with <PDP_addr_1> containing the IPv4 address and <PDP_addr_2> containing the IPv6 address. The string is given as dot-separated numeric (0-255) parameter of the form: a1.a2.a3.a4 for IPv4 and a1.a2.a3.a4.a5.a6.a7.a8.a9.a10.a11.a12.a13.a14.a15.a16 for IPv6.

Example

```
AT+CGPADDR
+CGPADDR: 5,"100.120.44.90"

OK
```

2.2.31 AT+CSCON Signalling Connection Status

The set command controls the presentation of an unsolicited result code +CSCON. If <n>=1, +CSCON: <mode> is sent from the MT when the connection mode of the MT is changed.

The read command returns the status of result code presentation and an integer <mode> which shows whether the MT is currently in idle mode or connected mode.

Test command returns supported values as a compound value.

AT+CSCON	
Set Command AT+CSCON=<n>	Response OK If there is any error, response: +CME ERROR: <err>
Read Command AT+CSCON?	Response +CSCON: <n>,<mode> OK
Test Command AT+CSCON=?	Response +CSCON: (list of supported <n>s)

	OK
Maximum Response Time	5s
Parameter Saving Mode	AUTO_SAVE

Parameter

<n>	Integer type	
	0	Disable unsolicited result code Note: default value
	1	Enable unsolicited result code +CSCON:<mode>
<mode>	Integer type; indicates the signalling connection status	
	0	idle
	1	connected

Example

```
AT+CSCON=?
+CSCON: (0,1)

OK

AT+CSCON=1
OK

AT+CSCON?
+CSCON: 1,0

OK

AT+CSCON=0
OK

AT+CSCON?
+CSCON: 0,0

OK
```

2.2.32 AT+CCLK Return Current Date and Time

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

The read command returns the current setting of the clock.

AT+CCLK	
Set Command AT+CCLK=<time>	Response OK If there is any error, response: +CME ERROR: <err>
Read Command AT+CCLK?	Response +CCLK: <time> OK
Test Command AT+CCLK=?	Response OK
Maximum Response Time	5s
Parameter Saving Mode	AUTO_SAVE

Parameter

<time>	String type
	String type. The format is "yy/MM/dd,hh:mm:ss±zz", where characters indicate year (two last digits), month, day, hour, minute, second and time zone (indicates the difference, expressed in quarters of an hour, between the local time and GMT; and range is -47 ~ +48). For instance, 6th of May 2014, 22:10:00 GMT+2 hours equals "2014/05/06,22:10:00+08" Note: the year should be after 2000 years, otherwise there will be asserted

Example

```
AT+CCLK="2018/07/25,02:22:22+00"

OK

AT+CCLK?
+CCLK: "2018/07/25,02:22:30+00"

OK

AT+CCLK=?
OK
```

2.2.33 AT+CIMI Request International Mobile Subscriber Identity

Execution command causes the TA to return <IMSI>, which is intended to permit the TE to identify the individual SIM card or active application in the UICC which is attached to MT.

AT+CIMI	
Set Command AT+CIMI	Response <IMSI> OK If there is any error, response: +CME ERROR: <err>
Test Command AT+CIMI=?	Response OK
Maximum Response Time	5s
Parameter Saving Mode	NO_SAVE

Parameter

<IMSI>	String type
	International Mobile Subscriber Identity (string without double quotes)

Example

```
AT+CIMI=?
OK

AT+CIMI
460043263600043

OK
```

2.2.34 AT+CPIN Enter PIN

Set command sends to the MT 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 MT 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.

Read command returns an alphanumeric string indicating whether some password is required or not.

AT+CPIN	
Set Command AT+CPIN=<pin>[,<newpin>]	Response OK If there is any error, response: +CME ERROR: <err>

Read Command AT+CPIN?	Response +CPIN: <code> OK
Test Command AT+CPIN=?	Response OK
Maximum Response Time	5s
Parameter Saving Mode	NO_SAVE

Parameter

<pin>,<newpin>	String type	
<code>	String type	
	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
	SIM PIN2	MT is waiting SIM PIN2 to be given (this <code> is recommended to be returned only when the last executed command resulted in PIN2 authentication failure (i.e. +CME ERROR: 17); if PIN2 is not entered right after the failure, it is recommended that MT does not block its operation)
	SIM PUK2	MT is waiting SIM PUK2 to be given (this <code> is recommended to be returned only when the last executed command resulted in PUK2 authentication failure (i.e. +CME ERROR: 18); if PUK2 and new PIN2 are not entered right after the failure, it is recommended that MT does not block its operation)

Example

```
AT+CPIN?
+CPIN: READY


OK
```

.2.35 AT+CLCK Facility Lock

Execute command is used to lock, unlock or interrogate a MT or a network facility <fac>. Password is normally needed to do such actions. When querying the status of a network service (<mode>=2) the response line for 'not active' case (<status>=0) should be returned only if service is not active for any <class>. This command should be abortable when network facilities are set or interrogated.
Test command returns facility values supported as a compound value.

AT+CLCK	
Set Command AT+CLCK=<fac>,<mode>[,<passwd>]	Response OK If there is any error, response: +CME ERROR: <err> When <mode>=2 and command successful: +CLCK: <status>
Test Command AT+CLCK=?	Response +CLCK: (list of supported <fac>s) OK If there is any error, response: +CME ERROR: <err>
Maximum Response Time	5s
Parameter Saving Mode	NO_SAVE

Parameter

<fac>	String type	
	"SC"	SIM (lock SIM/UICC card installed in the currently selected card slot) (SIM/UICC asks password in MT power-up and when this lock command issued)
<mode>	Integer type	
	0	Unlock
	1	Lock
	2	Query status
<status>	Integer type	
	0	Not active
	1	active
<passwd>	String type; shall be the same as password specified for the facility from the MT user interface or with command Change Password +CPWD	

Example

```
AT+CLCK=?
+CLCK:("SC")

OK
```

2.2.36 AT+CPWD Change Password

Command sets a new password for the facility lock function defined by command Facility Lock +CLCK. Test command returns a list of pairs which present the available facilities and the maximum length of

their password.

AT+CPWD	
Set Command AT+CPWD=<fac>,<oldpwd>,<newpwd>	Response OK If there is any error, response: +CME ERROR: <err>
Test Command AT+CPWD=?	Response +CPWD: list of supported (<fac>,<pwdlength>)s OK If there is any error, response: +CME ERROR: <err>
Maximum Response Time	5s
Parameter Saving Mode	NO_SAVE

Parameter

<fac>	String type	
	"SC"	SIM (lock SIM/UICC card installed in the currently selected card slot) (SIM/UICC asks password in MT power-up and when this lock command issued)
<oldpwd>,<newpwd>	String type	
		<oldpwd> shall be the same as password specified for the facility from the MT user interface or with command Change Password +CPWD and <newpwd> is the new password, maximum length of password can be determined with <pwdlength>
<pwdlength>	Integer type; maximum length of the password for the facility	

Example

```
AT+CPWD=?
+CPWD: ("SC",8)

OK
```

2.2.37 AT+CSIM Generic SIM Access

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.
This command allows a direct control of the SIM that is installed in the currently selected card slot, by an distant application on the TE. The TE shall then take care of processing SIM information within the frame

specified by GSM/UMTS.

AT+CSIM	
Set Command AT+CSIM=<length>,<command>	Response +CSIM: <length>,<response> OK If there is any error, response: +CME ERROR: <err>
Test Command AT+CSIM=?	Response OK
Maximum Response Time	5s
Parameter Saving Mode	NO_SAVE

Parameter

<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>	String type
	command passed on by the MT to the SIM in the format as described in 3GPP TS 51.011 [28] (hexadecimal character format)
<response>	String type
	response to the command passed on by the SIM to the MT in the format as described in 3GPP TS 51.011 [28] (hexadecimal character format)

Example

```
AT+CSIM=?
OK

AT+CSIM=14,"00A4000C023F00"
+CSIM: 4, "9000"

OK
```

2.2.38 AT+CRSM Restricted SIM

By using this command instead of Generic SIM Access +CSIM TE application has easier but more limited access to the SIM database. Set command transmits to the MT the SIM <command> and its required parameters. If a SIM installed in the currently selected card slot, the MT handles internally all SIM-MT interface locking and file selection routines. As response to the command, MT sends the actual SIM

information parameters and response data. MT error result code +CME ERROR may be returned when the command cannot be passed to the SIM, but failure in the execution of the command in the SIM is reported in <sw1> and <sw2> parameters.
Coordination of command requests to SIM and the ones issued by GSM/UMTS application inside the MT is implementation dependent. However, the TE should be aware of the precedence of the GSM/UMTS application commands to the TE commands.

AT+CRSM	
Set Command AT+CRSM=<command>[,<fileid> [,<P1>,<P2>,<P3>[,<data>[,<pathid>]]]]	Response +CRSM: <sw1>,<sw2>[,<response>] OK If there is any error, response: +CME ERROR: <err>
Test Command AT+CRSM=?	Response OK
Maximum Response Time	5s
Parameter Saving Mode	NO_SAVE

Parameter

<command>	Integer type; command passed on by the MT to the SIM; refer 3GPP TS 51.011 [28]	
	176	READ BINARY
	178	READ RECORD
	192	GET RESPONSE
	214	UPDATE BINARY
	220	UPDATE RECORD
	242	STATUS
	All other values are reserved	
<fileid>	Integer type; this is the identifier of a elementary datafile on SIM. Mandatory for every command except STATUS	
	The range of valid file identifiers depends on the actual SIM and is defined in 3GPP TS 51.011 [28]. Optional files may not be present at all.	
<P1>,<P2>,<P3>	Integer type; parameters passed on by the MT to the SIM. These parameters are mandatory for every command, except GET RESPONSE and STATUS. The values are described in 3GPP TS 51.011 [28]	
<data>	String type; information which shall be written to the SIM (hexadecimal character format)	
<pathid>	String type; contains the path of an elementary file on the SIM/UICC in hexadecimal format as defined in ETSI TS 102 221 [60] (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 [60]	
<sw1>,<sw2>	Integer type; information from the SIM about the execution of the actual	

	command. These parameters are delivered to the TE in both cases, on successful or failed execution of the command
<response>	String type; response of a successful completion of the command previously issued (hexadecimal character format). STATUS and GET RESPONSE return data, which gives information about the current elementary datafield. This information includes the type of file and its size (refer 3GPP TS 51.011 [28]). After READ BINARY, READ RECORD or RETRIEVE DATA command the requested data will be returned. <response> is not returned after a successful UPDATE BINARY, UPDATE RECORD or SET DATA command.

Example

```
AT+CRSM=176,28423,0,0,18
+CRSM: 144, 0, "08490660"

OK
```

2.2.39 AT+CTZU Automatic Time Zone Update

Set command enables and disables automatic time zone update via NITZ. If setting fails in an MT error, +CME ERROR: <err> is returned.

Read command returns the current settings in the MT.

Test command returns supported on- and off-values as a compound value.

AT+CTZU	
Set Command AT+CTZU=<onoff>	Response OK If there is any error, response: +CME ERROR: <err>
Read Command AT+CTZU?	Response +CTZU: <onoff> OK
Test Command AT+CTZU=?	Response +CTZU: (lists of supported <onoff>s) OK
Maximum Response Time	5s
Parameter Saving Mode	AUTO_SAVE

Parameter

<onoff>	Integer type	
	0	Disable automatic time zone update via NITZ.
	1	Enable automatic time zone update via NITZ. Note: default value

Example

```
AT+CTZU=1

OK

AT+CTZU?
+CTZU: 1

OK

AT+CTZU=?
+CTZU: (0-1)

OK
```

2.2.40 AT+CTZR Time Zone Reporting

This set command controls the time zone change event reporting. If reporting is enabled the MT returns the unsolicited result code +CTZV: <tz>, +CTZE: <tz>,<dst>,[<time>], or +CTZEU: <tz>,<dst>,[<utime>]whenever the time zone is changed. The MT also provides the time zone upon network registration if provided by the network. If setting fails in an MT error, +CME ERROR: <err> is returned.

Read command returns the current reporting settings in the MT.

Test command returns supported <reporting>-values as a compound value.

AT+CTZR	
Set Command AT+CTZR=<reporting>	Response OK If there is any error, response: +CME ERROR: <err>
Read Command AT+CTZR?	Response +CTZR: <reporting> OK

Test Command AT+CTZR=?	Response +CTZR: (lists of supported <reporting>s) OK
Maximum Response Time	5s
Parameter Saving Mode	AUTO_SAVE

Parameter

<reporting>	Integer type	
	0	disable time zone change event reporting. Note: default value.
	1	Enable time zone change event reporting by unsolicited result code +CTZV: <tz>.
	2	Enable extended time zone and local time reporting by unsolicited result code +CTZE: <tz>,<dst>,[<time>]. Note: not support.
	3	Enable extended time zone and universal time reporting by unsolicited result code +CTZEU: <tz>,<dst>,[<utime>]
<tz>	String type	
	representing the sum of the local time zone (difference between the local time and GMT expressed in quarters of an hour) plus daylight saving time. The format is "±zz", expressed as a fixed width, two digit integer with the range -48 ... +56. To maintain a fixed width, numbers in the range -9 ... +9 are expressed with a leading zero, e.g. "-09", "+00" and "+09"	
<dst>	Interger type, indicating whether <tz> includes daylight savings adjustment.	
	0	<tz> includes no adjustment for Daylight Saving Time
	1	<tz> includes +1 hour (equals 4 quarters in <tz>) adjustment for daylight saving time
	2	<tz> includes +2 hours (equals 8 quarters in <tz>) adjustment for daylight saving time
<time>	String type	
	Value representing the local time. The format is "YYYY/MM/DD,hh:mm:ss", expressed as integers representing year (YYYY), month (MM), date (DD), hour (hh), minute (mm) and second (ss). The local time can be derived by the MT from information provided by the network at the time of delivering time zone information and will be present in the unsolicited result code for extended time zone and local time reporting if the universal time is provided by the network.	
<utime>	String type	
	Value representing the universal time. The format is "YYYY/MM/DD,hh:mm:ss", expressed as integers representing year (YYYY),	

	month (MM), date (DD), hour (hh), minute (mm) and second (ss). The universal time can be provided by the network at the time of delivering time zone information and will be present in the unsolicited result code for extended time zone and universal time reporting if provided by the network.

Example

```
AT+CTZR=3

OK

AT+CTZR?
+CTZR: 3

OK

AT+CTZR=?
+CTZR: (0,1,3)

OK
```

2.3 3GPP Commands (27.005)

2.3.1 AT+CMGS Send Message

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. Value can be used to identify message upon unsolicited delivery status report result code. If sending fails in a network or an ME error, final result code +CMS ERROR: <err> is returned. This command should be abortable.

For text mode:

- entered text (3GPP TS 23.040 [3] TP-Data-Unit) is sent to address <da> and all current settings (refer Set Text Mode Parameters +CSMP and Service Centre Address +CSCA) are used to construct the actual PDU in ME/TA.
- the TA shall send a four-character sequence <CR><LF><greater_than><space> (IRA 13, 10, 62, 32) after command line is terminated with <CR>; after that text can be entered from TE to ME/TA.
- the DCD signal shall be in ON state while text is entered.
- the echoing of entered characters back from the TA is controlled by V.25ter echo command E.
- the entered text should be formatted as follows:
- if <dcs> (set with +CSMP) indicates that 3GPP TS 23.038 [2] GSM 7 bit default alphabet is used and <fo> indicates that 3GPP TS 23.040 [3] TP-User-Data-Header-Indication is not set:
- if TE character set other than "HEX" (refer command Select TE Character Set +CSCS in 3GPP TS

27.007 [9]): ME/TA converts the entered text into the GSM 7 bit default alphabet according to rules of Annex A; backspace can be used to delete last character and carriage returns can be used (previously mentioned four character sequence shall be sent to the TE after every carriage return entered by the user);

- if TE character set is "HEX": the entered text should consist of two IRA character long hexadecimal numbers which ME/TA converts into the GSM 7 bit default alphabet characters. (e.g. 17 (IRA 49 and 55) will be converted to character Π (GSM 7 bit default alphabet 23)).
- if <dcs> indicates that 8-bit or UCS2 data coding scheme is used or <fo> indicates that 3GPP TS 23.040 [3] TP-User-Data-Header-Indication is set: the entered text should consist of two IRA character long hexadecimal numbers which ME/TA converts into 8-bit octet (e.g. two characters 2A (IRA 50 and 65) will be converted to an octet with integer value 42).
- sending can be cancelled by giving <ESC> character (IRA 27).
- <ctrl-Z> (IRA 26) must be used to indicate the ending of the message body.

For PDU mode:

- <length> must indicate the number of octets coded in the TP layer data unit to be given (i.e. SMSC address octets are excluded).
- the TA shall send a four character sequence <CR><LF><greater_than><space> (IRA 13, 10, 62, 32) after command line is terminated with <CR>; after that PDU can be given from TE to ME/TA.
- the DCD signal shall be in ON state while PDU is given.
- the echoing of given characters back from the TA is controlled by V.25ter echo command E.
- the PDU shall be hexadecimal format (similarly as specified for <pdu>) and given in one line; ME/TA converts this coding into the actual octets of PDU.
- when the length octet of the SMSC address (given in the PDU) equals zero, the SMSC address set with command Service Centre Address +CSCA is used; in this case the SMSC Type-of-Address octet shall not be present in the PDU, i.e. TPDU starts right after SMSC length octet.
- sending can be cancelled by giving <ESC> character (IRA 27).
- <ctrl-Z> (IRA 26) must be used to indicate the ending of PDU.

AT+CMGS	
Set Command If text mode(AT+CMGF=1): AT+CMGS=<da>[,<toda>]<CR> Text is entered<ctrl-Z/ESC> If PDU mode(AT+CMGF=0): AT+CMGS=<length><CR> PDU is given<ctrl-z/ESC>	If sending successful: +CMGS: <mr> If there is any error, response: +CMS ERROR: <err>
Maximum Response Time	60s
Parameter Saving Mode	NO_SAVE

Parameter

<da>	String type; in text mode (AT+CMGF=1)

	Destination address
<toda>	integer type; Type of destination address
<length>	Integer type; in PDU mode (AT+CMGF=0), the length of the actual TP data unit in octets (i.e. the RP layer SMSC address octets are not counted in the length). The range is 7-220.
<mr>	3GPP TS 23.040 [3] TP-Message-Reference in integer format.

Example

```
AT+CMGF=1
OK
AT+CMGS=”1064899990000”
>TEST
CTRL+Z(1a(hex))
+CMGS: 1
OK
```

2.3.2 AT+CSCA Service Center Address

Set command updates the SMSC address, through which mobile originated SMs are transmitted. In text mode, setting is used by send and write commands. In PDU mode, setting is used by the same commands, but only when the length of the SMSC address coded into <pdu> parameter equals zero.

AT+CSCA	
Set Command AT+CSCA=<sca>[,<tosca>]	Response OK If there is any error, response: +CMS ERROR: <err>
Read Command AT+CSCA?	Response +CSCA: <sca>,<tosca>
Maximum Response Time	5s
Parameter Saving Mode	NO_SAVE

Parameter

<sca>	String type; 3GPP TS 24.011 [6] RP SC address Address-Value field in string format; BCD numbers (or GSM 7-bit default alphabet characters) are converted to characters
<tosca>	Integer type. 3GPP TS 24.011 [6] RP SC address Type-of-Address octet in integer format (when first character of <da> is + (IRA 43), default value is 145, otherwise default value is 129).

Example

```
AT+CSCA="8613800200569"
OK

AT+CSCA?
+CSCA: "8613800200569",129

OK
```

2.3.3 AT+CMGF Message Format

Set command tells the TA, which input and output format of messages to use. <mode> indicates the format of messages used with send, list, read and write commands and unsolicited result codes resulting from received messages. Mode can be either PDU mode (entire TP data units used) or text mode (headers and body of the messages given as separate parameters).

Test command returns supported modes as a compound value.

AT+CMGF	
Set Command AT+CMGF=<mode>	Response OK If there is any error, response: +CMS ERROR: <err>
Read Command AT+CMGF?	Response +CMGF: <mode>
Test Command AT+CMGF=?	Response +CMGF: (list of supported <mode>s)
Maximum Response Time	5s
Parameter Saving Mode	AUTO_SAVE

Parameter

<mode>	integer type;	
	0	PDU mode
	1	Text mode

Example

```
AT+CMGF=1
OK

AT+CMGF?
```

```
+CMGF: 1

OK
```

2.3.4 AT+CSMP Set 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>. If TA supports the EVPF, see 3GPP TS 23.040 [3], it shall be given as a hexadecimal coded string (refer e.g. <pdu>) with double quotes.

AT+CSMP	
Set Command AT+CSMP=<fo>[,<vp>[,<pid>[,<dcs>]]]	Response OK If there is any error, response: +CMS ERROR: <err>
Read Command AT+CSMP?	Response +CSMP: <fo>,<vp>,<pid>,<dcs>
Maximum Response Time	5s
Parameter Saving Mode	AUTO_SAVE

Parameter

<fo>	integer type;
	First octet for sms submit pdu,status report required
<vp>	integer type;
	Validity period
<pid>	integer type;
	Protocol identifier
<dcs>	integer type;
	Data coding scheme

Example

```
AT+CSMP=33,167,0,0
OK
AT+CSMP?
+CSMP: 33,167,0,0

OK
```

2.3.5 +CMT New Message Received

SMS-DELIVERs are routed directly to the TE using unsolicited result code.

+CMT
+CMT: <length><CR><LF><pdu> (PDU mode enabled)
+CMT: <oa>, <scts><CR><LF><data> (text mode enabled)

Parameter

<length>	Integer type; Length of PDU(PDU mode enabled) Deliver Message's source address(text mode enabled)
<oa>	String type; Deliver Message's source address(text mode enabled)
<scts>	String type; TP-Service-Centre-Time-Stamp in time-string format 3GPP TS 23.040 [3] TP-Discharge-Time in time-string format: "yy/MM/dd,hh:mm:ss±zz", where characters indicate year (two last digits), month, day, hour, minutes, seconds and time zone. E.g. 6th of May 1994, 22:10:00 GMT+2 hours equals to "94/05/06,22:10:00+08"
<data>	String type; The content of deliver message in HEX string format

Example

```
+CMT: ”106499990000”,”19.05.16 16:27:55 GMT:+8” hell
```

3 Extended Commands

3.1 EC General Commands

3.1.1 AT+ECBAND

The command sets the network mode and bands to be used.

Read command returns the current network mode and band list.

Test command returns network mode and bands supported by the UE.

AT+ECBAND	
Set Command AT+ECBAND=<mode>[,<band1>[,<band2>…]]	Response OK If there is any error, response: +CME ERROR: <err>
Read Command AT+ECBAND?	Response +ECBAND: <mode>,<band1>,<band2>,… OK
Test Command AT+ECBAND=?	Response +ECBAND: (list of supported <mode>s),(list of supported <band>s) OK
Maximum Response Time	25s
Parameter Saving Mode	AUTO_SAVE

Parameter

<mode>	Integer type
	0 NB-IOT mode(current support NB-IoT only)
<band>	Integer type
	Band list in decimal number. <band> values of 1, 3, 5, 8 are supported. The default value is decided by RF Calibration table

Example

```
AT+ECBAND?
+ECBAND: 0,5,8,1,3
```

```
OK

AT+ECBAND=?
+ECBAND: (0),(1,3,5,8)

OK

AT+ECBAND=0,5,8
OK
```

3.1.2 AT+ECCFG

The command set UE extended configuration.

The read command return current setting of each parameters.

The test command returns values supported as a compound value.

AT+ECCFG	
Set Command AT+ECCFG=<param1>,<value1>[,<param2>,<value2>[,< param3>,<value3>,[…]]]	Response OK If there is any error, response: +CME ERROR: <err>
Read Command AT+ECCFG?	Response +ECCFG: <param1>,<value1>,<param2>,<value2>…<paramN>,<valueN> OK
Test Command AT+ECCFG=?	Response +ECCFG: (list of supported <param>s) OK
Maximum Response Time	5s
Parameter Saving Mode	AUTO_SAVE_REBOOT

Parameter

<param>	String type, name of configuration parameter.	
	"AutoApn"	Whether UE auto set the attached APN according to the inserted SIM card. Note: a) Supported values: (0,1)

		b) Default value: 0, just use the APN which is set by AT+ECATTBEARER
	"Rohc"	Whether UE support ROHC. Note: a) Supported values: (0,1) b) Default value: 1.
	"PowerCfun"	Default CFUN state after UE power-on or reboot; Note: a) Support values: (0,1,4) b) Default values: 1 c) if set to 0,UE remain CFUN0 state(neither turn on protocol/Rf nor SIM)after power-on or reboot;And could turn on protocol/RF and SIM via AT+CFUN=1 d) if set to 1;UE auto turn on protocol,and connect the network after power-on or reboot; e) if set to 4,UE only turn on SIM ,disable (turn off)protocol/RF,after power-on or reboot.
	"Ipv6RsForTestSim"	Whether UE trigger IPv6 NDP (RS) procedure to get IPv6 prefix address, when the SIM card inserted is a TEST SIM. Note: a) Supported values: (0,1) b) Default value 0. c) IPv6 NDP (RS) procedure is triggered by default if the inserted SIM card is not for testing.
	"SupportSms"	Whether UE support SMS. Note: a) Supported values: (0,1) b) Default value: 1
	"TauForSms"	Whether need to trigger TAU procedure, if UE support SMS capability, while NW not support. Note: a) Supported values: (0,1) b) Default value: 0
	"PlmnSearchPowerLevel"	Set the PLMN search level when UE OOS; Note: a) Supported values: (0,1,2,3) 0 - OOS PLMN search interval: 30 sec, 1 min, 2 min 1 - OOS PLMN search interval: 5 min, 10 min,

	15 min 2 - OOS PLMN search interval: 10 min, 30 min, 1 hour 3 - OOS PLMN search interval: 30 sec, then stop PLMN search, and let AT: AT+ECPLMNS to start PLMN search b) Default value: 1
"Epco"	Whether UE need to use "EPCO" in "PDN CONNECTION REQUEST" carried in "ATTACH REQUEST", and "ESM INFORMATION RESPONSE"; If set to 0, just use "PCO". Note: a) Supported values: (0,1) b) Default value: 1
"MultiCarrier"	Whether UE support multi-carrier feature. Note: a) Support values: (0,1) b) Default value: 1
"MultiTone"	Whether UE support multi-tone feature. Note: a) Supported values: (0,1) b) Default value: 1
"SupportUpRai"	Whether UE support L2 (MAC layer) RAI feature, which only valid whether set to R14 version. Note: a) Supported values: (0,1) b) Default value: 0
"DataInactTimer"	Set the value of "data inactivity timer" in seconds, if this timer is not configured by NW (in MAC-MainConfig-NB), just use this setting value. Note: a) Supported value: (0,40-254) b) Default value: 60 c) If set to 0, just means this timer is invalid, don't need to start.
"RelaxMonitorDeltaP"	Set the value of "SearchDeltaP" in DB for Relex-Monitor feature. If this value is not configured by NW (in SIB-NB), just use this setting value. Note: a) Supported values: (0-15) b) Default value: 0
"RelVersion"	Set the NB release version.

	Note: a) Supported values: (13,14) b) Default value: 13
<value>	Integer type
	value of configuration

Example

```
AT+ECCFG="Rohc",0
OK
```

3.1.3 AT+ECPING

The command sends an ICMP packet to the specified host address. AT+ECPING initiates the sending of a PING packet with payload size: <size> to the specified address. This will either cause a packet to be returned if the remote system is connected and responding to PING packets or no response will be received. If none of the response packet received within the timeout period <timeout>. It will continue to send PING packet until the <count> number of times.

The test command returns values supported as a compound value.

AT+ECPING	
Set Command AT+ECPING=[<ipaddr/Url>/<mode>[,<count>[,<size>[,<timeout>]]]]	Response OK If there is any error, response: +SOCKET ERROR: <err>
Test Command AT+ECPING=?	Response +ECPING: (list of supported <ipaddr/Url/mode>s),(list of supported <count> s),(list of supported <size>s),(list of supported <timeout>s) OK
Maximum Response Time	5s
Parameter Saving Mode	NO_SAVE

Parameter

<mode>	Integer type
	0 Stop ping
<ipaddr/Url>	String type, IP address or URL
<count>	Integer type, ping times Note: default value: 4

<size>	Integer type, payload size Note: default value: 32
<timeout>	Integer type, UE ping reply timeout after ping request.(ms) Note: <timeout> values between 1 to 600000 are supported, and default value: 20000

Note:

a) When one PING reply received in <timeout>, an unsolicited result code: +ECPING: SUCC, dest: <dest_ip_addr>, RTT: <rtt_time>ms will sent to TE.

b) If no PING reply received in <timeout>, an unsolicited result code: +ECPING: FAIL, dest: <dest_ip_addr>, time out: <timeout>ms will sent to TE.

c) If this is an ERROR meet during PING procedure, an unsolicited result code: +ECPING: ERROR, cause: <cause> will sent to TE.

d) When PING procedure is done, an unsolicited result code: +ECPING: DONE<CR><LF>+ECPING: dest: <dest_ip_addr>, <count> packets transmittted, <reply_count> received, <lost_percent>% packet loss<CR> rtt min/avg/max = <rtt_min> / <rtt_avg> / <rtt_max> ms will sent to TE.

Example

Ping 180.97.33.107 10 times with 32 bytes payload, timeout is 60 seconds:

```
AT+ECPING=”180.97.33.107”,10,32,60000
OK
+ECPING: SUCC, dest: 180.97.33.107, RTT: 334 ms

+ECPING: SUCC, dest: 180.97.33.107, RTT: 179 ms

…

Stop ping:
AT+ECPING=0
OK
```

3.1.4 AT+ECIPERF

The command tests the TCP/IP’s uplink and downlink IPERF performance.

The test command returns values supported as a compound value.

AT+ECIPERF	
Set Command AT+ECIPERF=<action>[,<protocol>[,<port>[,<ipaddr>[,<tpt>[,payload_size[,<packet_number>[,<duration>[,<report_interval>]]]]]]]	Response OK If there is any error, response: +SOCKET ERROR: <err>
Test Command	Response

AT+ECIPERF=?	+ECIPERF: (list of suppprted <action>s),(list of supported <protcol>s),(list of supported <port>s), (list of supported <tpt>s),(list of supported <payload_size>s),(list of supported <pkg_num>s), (list of supported <duration>s),(list of supported <report_interval>s) OK
Maximum Response Time	5s
Parameter Saving Mode	NO_SAVE

Parameter

<action>	Integer type, IPERF command	
	0	Terminate all IPERF services
	1	Start IPERF client
	2	Stop IPERF client
	3	Start IPERF server
	4	Start IPERF UDP NAT server Note: One type of IPERF UDP server, in this mode, UE will send one UDP packet to remote server to setup the UDP connection, then UE wait to receive the DL UDP packets, and start the DL UDP IPERF server.
	5	Stop IPERF server
<protocol>	Integer type	
	0	UDP
	1	TCP
<port>	Integer type, UDP/TCP port number. Note: a) if <action> is 1 or 4, this <port> is the destination server port number. b) if <action> is 3, this <port> is the local IPERF server port number. c) Default value: 5001	
<ipaddr>	String type, destination server IP address. Note: if <action> is 1 or 4, this <ipaddr> is mandatory.	
<tpt>	Integer type, throughput in bps. Note: default value: 20000	
<payload_size>	Integer type, payload size of UL UDP/TCP IPERF packet. Used for client mode	

<packet_number>	Integer type, packet number of UE send, when acted as a client mode.
< report_interval >	Integer type, report internal of IPERF service result. UE send the following unsolicited result codes periodically in this interval (in seconds). a) If <action> is 1, the unsolicited result codes: +ECIPERF: Client SUCC, pkg sent bytes: <bytes>, UL through put: <tpt> bps b) If <action> is 3 or 4, the unsolicited result codes: +ECIPERF: Server SUCC, pkg recv bytes: <bytes>, DL through put: <tpt> bps Note: default value: 10
<duration>	Integer type, IPERF service duration in seconds Note: if not specified, IPERF will not stop, before meet an error, or received a terminate command.

Note:

a) When IPERF client service is finished (terminated/timeout), UE send the unsolicited result codes: +ECIPERF: Client END, pkg sent total bytes: <bytes>, average UL through put: <tpt> bps

b) When IPERF server service is finished (terminated/timeout), UE send the unsolicited result codes: +ECIPERF: Server END, pkg recv total bytes: <bytes>, average DL through put: <tpt> bps

c) If happens an error which caused the IPERF service can't go on, UE send the unsolicited result codes: +ECIPERF: Client FAIL, <err>; or +ECIPERF: Server FAIL, <err>

Example

```
AT+ECIPERF=1,0,5001,"180.167.122.150",10000

OK

+ECIPERF: Client SUCC, pkg sent bytes: 13720, UL through put: 10976 bps

+ECIPERF: Client SUCC, pkg sent bytes: 9604, UL through put: 7683 bps

+ECIPERF: Client SUCC, pkg sent bytes: 12348, UL through put: 9878 bps

+ECIPERF: Client SUCC, pkg sent bytes: 12348, UL through put: 9878 bps
AT+ECIPERF=0

OK

+ECIPERF: Client END, pkg sent total bytes: 52136, average UL through put: 9268 bps
```

3.1.5 AT+ECFREQ

The command set prefer EARFCN list, lock or unlock cell.

Read command returns the current EARFCN setting.

The test command returns values supported as a compound value.

AT+ECFREQ	
Set Command If cell unlock or remove prefer EARFCN (mode = 0): AT+ECFREQ=<mode> If set prefer EARFCN list (mode = 1): AT+ECFREQ=<mode>[,<earfcn1>[,<earfcn2>…]] If cell lock (mode = 2): AT+ECFREQ=<mode>,<earfcn>[,<phyCellId>]	Response OK If there is any error, response: +CME ERROR: <err>
Read Command AT+ECFREQ?	Response If neither set prefer EARFCN list nor lock EARFCN/cell : OK If set prefer EARFCN list: +ECFREQ: <1>,<arfcn1>,<arfcn2>,… OK if lock EARFCN or lock cell: +ECFREQ:<mode>,<arfcn>,<phyCellId> OK if both set prefer EARFCN list and lock EARFCN/cell +ECFREQ:<1>,<arfcn1>,<arfcn2>,… +ECFREQ:<mode>,<arfcn>,<phyCellId> OK
Test Command AT+ECFREQ=?	Response +ECFREQ: (list of supported <mode>s) OK
Maximum Response Time	5s
Parameter Saving Mode	AUTO_SAVE_REBOOT

Note: AT+ECFREQ must be restricted to execute in power off or air plane state.

Parameter

<mode>	Integer type	
	0	Cancel cell lock
	1	Set prefer EARFCN list

	2	EARFCN lock, or cell lock
	3	Clear prefer EARFCN
<earfcn>	Integer type	
	E-UTRA Absolute Radio Frequency Channel Number	
<phyCellId>	Integer type	
	PhysicalCell ID	

Example

```
AT+ECFREQ=2,3734,145
OK
AT+ECREQ?
+ECFREQ: 2,3734,145

OK
AT+ECFREQ=3
OK
```

3.1.6 AT+ECRMFPLMN

Set command remove FPLMN in NVM or SIM.

The test command returns values supported as a compound value.

AT+ECRMFPLMN	
Set Command AT+ECRMFPLMN=<mode>	Response OK If there is any error, response: +CME ERROR: <err>
Test Command AT+ECRMFPLMN=?	Response +ECRMFPLMN: (list of supported <mode>s) OK
Maximum Response Time	5s
Parameter Saving Mode	AUTO_SAVE

Parameter

<mode>	Integer type	
	0	Remove FPLMN in NVM file and in SIM card
	1	Remove FPLMN in NVM file
	2	Remove FPLMN in SIM card

Example

```
AT+ECRMFPLMN=0
OK
```

3.1.7 AT+ECATTBEARER

The set command is used to configure the PDN info request to establish during the attach process, if attach with PDN required.

The read command is used to obtain the configuration of the PDN info request to establish during the attach process.

The test command returns values supported as a compound value.

AT+ECATTBEARER	
Set Command AT+ECATTBEARER=<PDP_type>[,<eitf>[,<apn>[,<IPv4 AddrAlloc>[,<NSLPI>[,<IPv4_MTU_discovery>[,<Non IP_MTU_discovery>]]]]]]	Response OK If there is any error, response: +CME ERROR: <err>
Read Command AT+ECATTBEARER?	Response +ECATTBEARER: <pdnType>,<eitf>,<apnStr>, <ipv4allocType>,<NSLPI>,<ipv4Mtu>,<nonIpMtu> OK
Test Command AT+ECATTBEARER=?	Response +ECATTBEARER: (list of supported <PDP_type>s),(list of supported <eitf>s),(list of supported <IPv4AddrAlloc>s),(list of supported <NSLPI>s),(list of supported <IPv4_MTU_discovery>s),(list of supported <NonIP_MTU_discovery>s)
Maximum Response Time	5s
Parameter Saving Mode	AUTO_SAVE_REBOOT

Parameter

<PDP_type>	Integer type, PDP type

	1	IPv4
	2	IPv6
	3	IPv4v6
	5	NON IP
	The default value is 3	
<eitf>	Integer type	
	0	Security protected ESM information transfer not required
	1	Security protected ESM information transfer required
	The default value is 1	
<apn>	string type	
	Apn string	
	The default value is “”(NULL)	
<IPv4AddrAlloc>	Integer type	
	0	IPv4 address allocate through NAS signaling
	1	IPv4 address allocate through DHCP (not applicable)
	The default value is 1	
<NSLPI>	Integer type	
	0	indicates that this PDP context is to be activated with the value for the low priority indicator configured in the MT.
	1	indicates that this PDP context is is to be activated with the value for the low priority indicator set to "MS is not configured for NAS signaling low priority".
	The default value is 0	
<IPv4_MTU_discovery>	Integer type	
	0	IPv4 MTU size discovery not influenced by +ECATTBEARER
	1	IPv4 MTU size discovery through NAS signaling
	The default value is 1	
<NonIP_MTU_discovery>	Integer type	
	0	Non-IP MTU size discovery not influenced by +ECATTBEARER
	1	Non-IP MTU size discovery through NAS signaling
	The default value is 0	

Example

3.1.8 AT+ECSENDDATA

The set command could send data via control plane or user plane

AT+ECSENDDATA	
Set Command	Response

AT+ECSENDDATA=<cid>,<data_length>,<data>[,<RAI>[,<type_of_user_data>]]	OK If there is any error, response: +CME ERROR: <err>
Test Command AT+ECSENDDATA=?	Response +ECSENDDATA: (range of supported <cid>s),(maximum number of octets of user data indicated by <data_length>),(list of supported <RAI>s),(list of supported <type_of_user_data>s)
Maximum Response Time	5s
Parameter Saving Mode	NO_SAVE

Parameter

<cid>	Integer type; specifies a particular PDP context definition. Note: If <cid> set to 0, just use current default bearer to send this originating data.	
<data_length>	Integer type. Indicates the number of octets of the <data> information element. The max length is 950.	
<data>	string of octets.	
<RAI>	Integer type. Indicates the value of the release assistance indication	
	0	No information available
	1	The MT expects that exchange of data will be completed with the transmission of this UL packet.
	2	The MT expects that exchange of data will be completed with the receipt of a DL packet.
<type_of_user_data>	Integer type. Indicates whether the user data that is transmitted is regular or exceptional.	
	0	Regular data
	1	Exception data

Note:

Difference with AT+CSODCP, AT+CSODCP limit to transmit data over control plane to network, but this AT don't have such limitation.

Example

```
AT+ECSENDDATA=5,2,"ABCD"
OK
```

3.1.9 +RECVNONIP

This is an unsolicited code message used to indicate downlink NON-IP data.

+RECVNONIP
+RECVNONIP: <cid>,<data_length>,<data>

Parameter

<cid>	Integer type; specifies a particular PDP context definition. <cid> values of 1-11 are supported.
<data_length>	Integer type. Indicates the number of octets of the <data> information element.
<data>	string of octets.

Example

3.1.10 AT+ECPMUCFG

The command set PMU mode.
Read command returns the current setup.
Test command returns values supported as a compound value.

AT+ECPMUCFG	
Set Command AT+ECPMUCFG=<enable>[,<mode>]	Response OK If there is any error, response: +CME ERROR: <err>
Read Command AT+ECPMUCFG?	Response +ECPMUCFG: <enable>[,<mode>] OK Note: If PMU is disabled, <mode> will not return.
Test Command AT+ECPMUCFG?	Response +ECPMUCFG: (range of supported <enable>s) , (list of supported <mode>s) OK
Maximum Response Time	5s
Parameter Saving Mode	AUTO_SAVE

Parameter

<ENABLE>	Integer type; specifies to enable PMU or not	
	0	Disable the PMU
	1	Enable the PMU
	The default value is 0	
<MODE>	Integer type; specifies to depth of sleep mode	
	0	Active
	1	Idle
	2	Sleep1
	3	Sleep2
	4	Hibernate
	The default value is 0	

Example

```
AT+ECPMUCFG=1,4
OK

AT+ECPMUCFG=0
OK
```

3.1.11 AT+ECSMSSEND

The command is used to send one SMS.

AT+ECSMSSEND	
Set Command AT+ECSMSSEND=<mode>,<phoneNum>,<payload>	Response OK If there is any error, response: +CME ERROR: <err>
Maximum Response Time	60s
Parameter Saving Mode	NO_SAVE

Parameter

<mode>	Integer type	
	0	PDU mode
	1	TXT mode
<phoneNum>	String type	
	Destination address	
<payload>	String type	
	PDU for PDU mode Message's content for TXT mode	

Example

```
AT+ECSMSSEND=1,”1064899990000”,”hello”
OK
```

3.1.12 AT+ECRFSTAT

The command shows the status of RF calibration.

AT+ECRFSTAT	
Test Command AT+ECRFSTAT=?	Response +ECRFSTAT: calibrate done OK If RF is not calibrated, response: +ECRFSTAT: not calibrate
Read Command AT+ECRFSTAT?	Response +ECRFSTAT: calibrate done OK If RF is not calibrated, response: +ECRFSTAT: not calibrate
Maximum Response Time	5s
Parameter Saving Mode	NO_SAVE

Example

```
AT+ECRFSTAT?
+ECRFSTAT: calibrate done
OK
```

3.1.13 AT+ECRST

The command restart the chip.

AT+ECRST	
Execution Command AT+ECRST	Response OK
Maximum Response Time	5s
Parameter Saving Mode	NO_SAVE

Example

```
AT+ECRST
OK
```

3.1.14 AT+ECPSMR

The command report the power saving mode status.

AT+ECPSMR	
Test Command AT+ECPSMR=?	Response +ECPSMR: (range of supported <n>s) OK
Set Command AT+ECPSMR=<n>	Response OK If there is any error, response: +CME ERROR: <err>
Read Command AT+ECPSMR?	Response +ECPSMR: <n>,<mode> OK If there is any error, response: +CME ERROR: <err>
Indicate	Response +ECPSMR: <mode>
Maximum Response Time	5s
Parameter Saving Mode	AUTO_SAVE

Parameter

<n>	Integer type
	0: disable unsolicited result code
	1: enable unsolicited result code +ECPSMR: <mode>
	The default value is 0
<mode>	Integer type
	0: normal mode
	1: power saving mode

Example

```
AT+ECPSMR=1
OK

AT+ECPSMR?
+ECPSMR: 1,0
OK
```

```
+ECPSMR: 1
```

3.1.15 AT+ECPLMNS

Set command is used to trigger a PLMN search while UE is out of service, if UE is not out of service, `+CME ERROR: <err>` is returned.

Read command returns the current PLMN search state, and the reset of time of PLMN search timer.

AT+ECPLMNS	

Set Command
`AT+ECPLMNS`
	Response
	`OK`
	If there is any error, response:
	`+CME ERROR: <err>`

Read Command
`AT+ECPLMNS?`
	Response
	`+ECPLMNS: <state>[,<oosTimeStep>]`
	`OK`

Test Command `AT+ECPLMNS=?`	Response `OK`
Maximum Response Time	5s
Parameter Saving Mode	NO_SAVE

Parameter

`<state>`	Integer type	
	`0`	Deactivated, no PLMN search is ongoing
	`1`	Searching, PLMN search is ongoing
	`2`	Selected, already selected a PLMN
	`3`	OOS, UE is out of service and has started a PLMN search timer
`<oosTimeStep>`	Integer type. The rest of time (in seconds) of OOS PLMN search timer, only present when `<state>` is 3.	

Example

```
AT+ECPLMNS
OK

AT+ECPLMNS?
+ECPLMNS: 3, 108
OK
```

3.1.16 AT+ECCESQS

The set command controls the extended signal quality change event reporting. If reporting is enabled the MT returns the unsolicited result codes: +CESQ: <rxlev>,<ber>,<rscp>,<ecno>,<rsrq>,<rsrp>, or +ECCESQ: RSRP,<rsrp>,RSRQ,<rsrq>,SNR,<snr> whenever the extended signal quality is changed. If setting fails in an MT error, +CME ERROR: <err> is returned.

The read command returns the current reporting settings in the MT.

The test command returns values supported as compound values.

AT+ECCESQS	
Execution Command AT+ECCESQS=<report level>	Response OK If there is any error, response: +CME ERROR: <err>
Read Command AT+ECCESQS?	Response +ECCESQS: <report level> OK
Test Command AT+ECCESQS=?	Response +ECCESQS: (list of supported <report level>s) OK
Maximum Response Time	5s
Parameter Saving Mode	AUTO_SAVE

Parameter

<report level>	Integer type
	0 disable unsolicited report.
	1 report +CESQ: <rxlev>,<ber>,<rscp>,<ecno>,<rsrq>,<rsrp>
	2 report +ECCESQ: RSRP,<rsrp>,RSRQ,<rsrq>,SNR,<snr>
	The default value is 0
<rxlev>	Integer type; not supported by NB-IoT
	99 not known or not detectable
<ber>	Integer type; not supported by NB-IoT
	99 not known or not detectable
<rscp>	Integer type; not supported by NB-IoT
	255 not known or not detectable
<ecno>	Integer type; not supported by NB-IoT

	255 not known or not detectable
<rsrp>	Integer type
	For +CESQ reporting, refer to sub clause 2.2.9 AT+CESQ
	For +ECCESQ reporting, the range is -156 dBm to -44 dBm
<rsrq>	Integer type
	For +CESQ reporting, refer to sub clause 2.2.9 AT+CESQ
	For +ECCESQ reporting, the range is -34 dB to 2.5 dB
<snr>	Integer type
	The range is -30 dB to 30 dB

Example

```
AT+ECCESQS=2
OK

AT+ECCESQS?
+ECCESQ: 2

OK

AT+ECCESQS=?
+ECCESQS: (0-2)

OK
```

3.1.17 AT+ECSTATUS

This read command returns some key parameter in UE side.

AT+ECSTATUS	
Read Command	Response
AT+ECSTATUS	+ECSTATUS: PHY, DlEarfcn:<dlEarfcn>, UlEarfcn:<ulEarfcn>, PCI:<pci>, Band:<band>, RSRP:<rsrp>, RSRQ:<rsrq>, SNR:<snr>, CeLevel:<ceLevel>, DlBler:<dlBler>, UlBler:<ulBler>, DataInactTimerS:<dataInactTimers>, RetxBSRTimerP:<retxBSRTimer0>, NBMode:<nbMode> +ECSTATUS: L2, SrbNum:<srbNum>, DrbNum:<drbNum> +ECSTATUS: RRC, State:<rrcState>, TAC:<tac>,

	CellId:<cellId> +ECSTATUS: EMM, EmmState:<emmState>, EmmMode:<emmMode>, PTWMs:<ptwMs>, EDRXPeriodMs:<eDRXPeriodMs>, PsmExT3412TimerS:<psmExT3412TimerS>, T3324TimerS:<T3324TimerS>, T3346RemainTimeS:<T3346RemainTimeS> +ECSTATUS: PLMN, PlmnState:<plmnState>, PlmnType:<plmnType>, SelectPlmn:<selectPlmn> +ECSTATUS: ESM, ActBearerNum:<actBearerNum>, APN:<apn>, IPv4:<ipaddr> +ECSTATUS: CCM, Cfun:<cfun>, IMSI:<imsi> OK
Test Command AT+ECSTATUS=?	Response OK
Maximum Response Time	5s
Parameter Saving Mode	NO_SAVE

Parameter

<dlEarfcn>	Integer type	
	Downlink earfcn, value range is 0~262143	
<ulEarfcn>	Integer type	
	Uplink earfcn, value range is 0~262143	
<pci>	Integer type	
	Physical cell ID, value range is 0~503,255	
<band>	Integer type	
	Band, value range is 0~70	
<rsrp>	Integer type	
	Value in dBm, range is -156dBm ~ -44dBm	
<rsrq>	Integer type	
	Value in dB, range is -34dB ~ -2.5dB	
<snr>	Integer type	
	Value in dB, range is -30dB ~ 30dB	
<ceLevel>	Integer type	
	0	CE level 0
	1	CE level 1
	2	CE level 2
<dlBler>	Integer type	

	Downlink block error, value range is 0~10000
<ulBler>	Integer type
	Uplink block error, value range is 0~10000
<dataInactTimerS>	Integer type
	Data inactive timer in seconds, value range is 0~180
<retxBSRTimerP>	Integer type
	Timer for BSR reporting, value in number of PDCCH periods. Value pp4 corresponds to 4 PDCCH periods, pp16 corresponds to 16 PDCCH periods and so on.
<NBMode>	String type
	Value range is "InBand Same PCI," InBand Diff PCI," Guard Band," Stand alone"
<srbNum>	Integer type
	Value range is 0~2
<drbNum>	Integer type
	Value range is 0~2
<rrcState>	String type
	Value range is "DEACT," OOS," IDLE," SUSPEND IDLE," CONNECTED," UNKONWN"
<tac>	Integer type
	Value range is 0~65534
<cellId>	Integer type
	Value range is 0~268435455
<emmState>	String type
	Value range is "NULL," DEREG," REG INIT," REG," DEREG INIT," TAU INIT," SR INIT," UNKNOWN"
<emmMode>	String type
	Value range is "IDLE," PSM," CONNECTED," UNKNOWN"
<ptwMs>	Integer type
	eDRX Paging Time Window in milliseconds
<edrxPeriodMs>	Integer type
	eDRX period in milliseconds
<psmExT3412TimerS>	Integer type
	Extended T3412 timer value in seconds
<T3324TimerS>	Integer type
	T3324 timer value in seconds
<T3346RemainTimeS >	Integer type
	If T3346 is running, set to the remaining time, else set to 0
<plmnState>	String type
	Value range is "NO PLMN", "SEARCHING", "SELECTED", "UNKNOWN"

<plmnType>	String type	
	Value range is "HPLMN", "EHPLMN", "VPLMN", "UPLMN", "OPLMN", "UNKNOWN"	
<selectPlmn>	String type	
	Selected PLMN	
<actBearerNum>	Integer type	
	activated bearer number	
<apn>	String type	
	access point name	
<Ipv4Addr ipv6Addr>	String type	
	Ipv4/Ipv6 address	
<fun>	Integer type	
	0	Minimum functionality
	1	Full functionality
	4	Turn off RF
<IMSI>	String type	
	International Mobile Subscriber Identity (string with double quotes)	

Example

```
AT+ECSTATUS

+ECSTATUS: PHY, DlEarfcn:3738, UlEarfcn:21738, PCI:11, Band:8, RSRP:-91, RSRQ:-8, SNR:8, CeLevel:0, DlBler:0/100, UlBler:0/100, DataInactTimerS:0, RetxBSRTimerP:0, NBMode:"Stand alone"

+ECSTATUS: L2, SrbNum:0, DrbNum:0

+ECSTATUS: RRC, State:"IDLE", TAC:23369, CellId:26224411

+ECSTATUS: EMM, EmmState:"REG", EmmMode:"IDLE", PTWMs:5120, EDRXPeriodMs:40960, PsmExT3412TimerS:0, T3324TimerS:300, T3346RemainTimeS:0

+ECSTATUS: PLMN, PlmnState:"SELECTED", PlmnType:"EHPLMN", SelectPlmn:"0x460,0xf000"

+ECSTATUS: ESM, ActBearerNum:1, APN:"cmnbiot.MNC004.MCC460.GPRS", IPv4:"100.83.34.10"

+ECSTATUS: CCM, Cfun:1, IMSI:"460043263600041"

OK
```

3.1.18 AT+ECICCID

Execution command causes the TA to return the ICCID of the UICC.

AT+ECICCID	
Set Command AT+ECICCID	Response +ECICCID: <ICCID> OK If there is any error, response: +CME ERROR: <err>
Maximum Response Time	5s
Parameter Saving Mode	NO_SAVE

Parameter

<ICCID>	String type
	Integrated circuit card identification

Example

```
AT+ECICCID
+ECICCID: 89861119220009636664

OK
```

3.1.19 AT+ECBCINFO

Execution command to return the basic serving cell information and neighbor cells information,mainly used for location service.

AT+ECBCINFO	
Set Command AT+ECBCINFO	Response +ECBCINFOSC: <earfcn>,<pci>,<rsrp>,<rsrq>,<mcc>,<mnc>,<cellid> [<CR><LF> +ECBCINFONC:<earfcn>,<pci>,<rsrp>,<rsrq> […]] OK If there is any error, response: +CME ERROR: <err>
Test Command	Response

AT+ECBCINFO=?	OK
Maximum Response Time	8s
Parameter Saving Mode	NO_SAVE

Parameter

<earfcn>	Integer type
	Indicate the EARFCN of the cell.Range 0~262143
<pci>	Integer type
	Indicate the physical cell ID. Range 0~503
<rsrp>	Integer type
	Indicate the measurement of RSRP value,in uint of dBm.Range -156~-44
<rsrq>	Integer type
	Indicate the measurement of RSRQ value,in uint of dBm.Range -34~25
<mcc>	String type
	Indicate the mobile country code
<mnc>	String type
	Indicate the mobile network code
<cellID>	String type
	Four byte E-UTRAN cell ID in hexadecimal format

Example

3.1.20 AT+ECDNS

This command to get the IP address for a specific URL .As a limitation now,only one IP address is return for a URL.

AT+ECDNS	
Set Command AT+ECDNS=<URL>	Response +ECDNS: <ipaddr> OK If there is any error, response: +SOCKET ERROR: <err>
Test Command AT+ECDNS=?	Response OK
Maximum Response Time	30s
Parameter Saving Mode	NO_SAVE

Parameter

<url>	String type
	Domain name
<ipaddr>	String type
	If IPV4 type ,output is dot-notation format,such as :”32.1.13.184”
	If IPV6 type output is colon-notation format,such as:”2001:0DB8:0000:CD30:0000:0000:0000:0002”

Example

```
AT+ECDNS=”www.baidu.com”
+ECDNS: “39.156.66.14”

OK
```

3.1.21 AT+ECDNSCFG

Set command set the default DNS addresses configuration. If DNS address not configured by network when activate a default bearer, just using these DNS addresses.

The read command return current setting of default DNS address.

AT+ECDNSCFG	
Set Command AT+ECDNSCFG=<ipaddr1>[,<ipaddr2>[,<ipaddr3>[,<ipaddr4>]]]	Response OK If there is any error, response: +SOCKET ERROR: <err>
Read Command AT+ECDNSCFG?	Response +ECDNSCFG: <ipaddr1>[,<ipaddr2>[,<ipaddr3>[,<ipaddr4>]]] OK
Test Command AT+ECDNSCFG=?	Response OK
Maximum Response Time	5s
Parameter Saving Mode	AUTO_SAVE

Parameter

<ipaddr>	String type
	If IPV4 type ,output is dot-notation format,such as :”32.1.13.184”
	If IPV6 type output is colon-notation format,such

	as:”2001:0DB8:0000:CD30:0000:0000:0000:0002”

Example

3.1.22 AT+ECPCFG

Set command is used to set plat config, if UE is not out of service, `+CME ERROR: <err>`is returned.

Read command returns the current plat config setting.

AT+ECPCFG	

Set Command
`AT+ECPCFG=<mode>,<value>`
	Response
	`OK`
	If there is any error, response:
	`+CME ERROR: <err>`

Read Command
`AT+ECPCFG?`
	Response
	`+ECPCFG: "faultAction":<value>,"dumpToATPort":<value>,"startWDT":<value>,"logCtrl":<value>,"logLevel":<value>,"logBaudrate":<value>,"slpWaitTime":<value>`
	
	`OK`

Test Command
`AT+ECPCFG=?`
	Response
	`+ECPCFG: "faultAction":<value>,"dumpToATPort":<value>,"startWDT":<value>,"logCtrl":<value>,"logLevel":<value>,"logBaudrate":<value>,"slpWaitTime":<value>`
	
	`OK`

Maximum Response Time	5s
Parameter Saving Mode	SAVE

Parameter

`<mode>`	String type	
	faultAction	Set the hardfault action mode
	startWDT	Set watch dog mode
	logCtrl	Set log control mode
	logLevel	Set log print level
	logBaudrate	Set log print baud rate

	slpWaitTime Set sleep wait time
<value>	Integer type
	For faultAction, the values range is from 0 to 3 0: dump full exception info to flash and EPAT tool then trapped in endless loop 1: print necessary exception info then reset 2: dump full exception info to flash then reset 3: dump full exception info to flash and EPAT tool then reset
	For startWDT, the values range is from 0 to 1 0: stop WDT 1: start WDT
	For logCtrl, the values range is from 0 to 2 0: unilog is disabled 1: only sw log is enabled 2: All log is enabled
	For logLevel, the values range is from 0 to 5 0: debug log level 1: info log level 2: value log level 3: signal log level 4: warning log level 5: error log level
	For logBaudrate, the values range is from 921600 to 6000000
	For slpWaitTime, the values range is from 0 to 0xffff

3.1.23 AT+ECSLEEP

This command is used for power consumption test. After executing this command, UE will enter related low power state. And UE could be wake up by wakeup PAD, after wake up, UE will reboot.

AT+ECSLEEP	
Set Command AT+ECSLEEP=<state>	Response +ECSLEEP: <mode> OK If there is any error, response: +CME ERROR: <err>
Test Command AT+ECSLEEP?	Response +ECPMUCFG: <state> OK

Maximum Response Time	5s
Parameter Saving Mode	SAVE

Parameter

<state>	Integer type	
	0	HIB2
	1	HIB1
	2	SLEEP2
	3	SLEEP1
<mode>	string type	
	HIB2	Hibenrnate2 status
	HIB1	Hibenrnate1 status
	SLEEP2	Sleep2 status
	SLEEP1	Sleep1 status

3.1.24 AT+DMCONFIG

The command set the parameters need by registering to the Device Manager Platform of China Mobile.

The read command returns the relevant parameters that have been set.

AT+DMCONFIG	
Set Command AT+DMCONFIG=<mode>,<lifetime>,<appkey>,<secret>,<paltform>	Response OK If there is any error, response: +CME ERROR: <err>
Test Command AT+DMCONFIG?	Response +DMCONFIG:<mode>,<lifetime>,<appkey>,<secret>,<paltform> OK
Maximum Response Time	5s
Parameter Saving Mode	SAVE

Parameter

<mode>	Integer type	
	0	Not register to DM paltform
	1	Register to DM paltform
<lifetime>	Integer type	

	Lifetime,uint: minute	
<appkey>	String type	
	A string of characters gived by China Mobile	
<secret>	String type	
	A string of characters gived by China Mobile	
<platform>	Integer type	
	0	Register to commercial paltform
	0	Register to test platform

3.1.25 AT+ECURC

The command close/open URC (unsolicited result code) report

AT+ECURC	
Set Command AT+ECURC=<urcStr>,<value>	**Response** OK **If there is any error, response:** +CME ERROR: <err>
Read Command AT+ECURC?	**Response** +ECURC: "CREG":<value>,"CEREG":<value>,"CEDRXP":<value>, "CCIOTOPTI":<value>,"CSCON":<value>,"CTZEU":<value>,"ECCESQ":<value>,"CGEV":<value>,"ECPSMR":<value>,"ECPTWEDRXP":<value>,"ECPIN":<value>,"ECPADDR":<value>,"ECPCFUN":<value> OK **If there is any error, response:** +CME ERROR: <err>
Test Command AT+ECURC=?	**Response** +ECPCFG: "ALL":(0-1),"CREG":(0-1),"CEREG":(0-1),"CEDRXP":(0-1),"CCIOTOPTI":(0-1),"CSCON":(0-1),"CTZEU":(0-1),"ECCESQ":(0-1),"CGEV":(0-1),"ECPSMR":(0-1),"ECPTWEDRXP":(0-1),"ECPIN":(0-1),"ECPADDR":(0-1),"ECPCFUN":(0-1) OK

Maximum Response Time	**5s**
Parameter Saving Mode	**AUTO_SAVE** Note: Set of <value value value> will save to NVM, and the default value is 0.

Parameter

<urcStr>	String type	
	ALL	All unsolicited result codes included as below
	CREG	unsolicited result code +CREG
	CEREG	unsolicited result code +CEREG
	CEDRXP	unsolicited result code +CEDRXP
	CCIOTOPT	unsolicited result code +CCIOTOPT
	CSCON	unsolicited result code +CSCON
	ECCESQ	unsolicited result code +ECCESQ
	CGEV	unsolicited result code +CEGV
	ECPSMR	unsolicited result code +ECPSMR
	PTWEDRX	unsolicited result code +ECPTWEDRX
	ECPTWEDRXP	unsolicited result code +ECPTWEDRXP
	ECPIN	unsolicited result code +ECPIN
	ECADDR	unsolicited result code +ECCADDR
	ECPCFUN	unsolicited result code +ECPCFUN
<value>	Integer type	
	0	disable unsolicited result code report
	1	enable unsolicited result code report

3.1.26 AT+ECPTWEDRXS

The set command controls the setting of the UE's paging time window and eDRX parameters. It can be used to control whether the UE wants to apply paging time window and eDRX or not, as well as the requested eDRX value for NB-IoT.

The set command also contols the presentation of the URC when <n>=2 and there is a change of the paging time window and eDRX parameters provided by network: +ECPTWEDRXP: <AcT-type> [,<Requested_Paging_time_window>[,<Requested_eDRX_value>[,<NW_provided_eDRX_value> [,<Paging_time_window>]]]]

A special form of the command can be given as AT+ECPTWEDRXS =3 . In this form, paging time window and eDRX will be disabled and data for all parameters in AT+ECPTWEDRXS command will be removed.

The read command returns the current settings for each defined value of <AcT -type>.

The test command returns the supported <mode>s and the value ranges for the access technology and the requested paging time window and requested eDRX value as compound values.

AT+ECPTWEDRXS	
Set Command AT+ECPTWEDRXS=<mode>,<AcT-type> [,<Requested_Paging_time_windows> [,<Requested_eDRX_value>]	**Response:** OK **If there is any error, response::** +CME ERROR: <err>
Read Command AT+ECPTWEDRXS?	**Response:** +ECPTWEDRXS:<Act-type>, <Requested_Paging_time_window>, <Requested_eDRX_value> [,<NW_provided_eDRX_value> [,<Paging_time_window>]] OK
Test Command AT+ECPTWEDRXS=?	**Response:** +ECPTWEDRXS: (list of supported <mode>),(list of supported <AcT-type>),(list of supported Requested_Paging_time_windows,)(list of supported <Requested_eDRX_value>)
Maximum Response Time	5s
Parameter Saving Mode	**AUTO_SAVE**

Parameter

<mode>	Integer type; indicates to disable or enable the use of requested paging time window and eDRX in the UE. This parameter is applicable to all specified types of access technology, i.e. the most recent setting of <mode> will take effect for all specified values of <AcT-type>	
	0	Disable the use of requested paging time window and eDRX
	1	Enable the use of requested paging time window and eDRX
	2	Enable the use of requested paging time window and eDRX and enable the unsolicited result code: +ECPTWEDRXP:<AcTtype>[,<Requested_Paging_time_window>[,<Requested_eDRX_value>[,<NW_provided_eDRX_value>[,<Paging_time_window>]]]]
	3	Disable the use of paging time window and eDRX and

	discard all parameters for pagint time window and eDRX.
<AcT-type>	Integer type; indicates the type of access technology. 5 E-UTRAN(NB-S1 mode)
<Requested_Paging_time_window>	String type; half a byte in a 4 bit format. The paging time window referes to bit 8 to 5 of octet 3 of the Extended DRX parameters information element. bits 4 3 2 1 Paging time window 0 0 0 0 2.56s 0 0 0 1 5.12s 0 0 1 0 7.68s 0 0 1 1 10.24s 0 1 0 0 12.8s 0 1 0 1 15.36s 0 1 1 0 17.92s 0 1 1 1 20.48s 1 0 0 0 23.04s 1 0 0 1 25.6s 1 0 1 0 28.16s 1 0 1 1 30.72s 1 1 0 0 33.28s 1 1 0 1 35.84s 1 1 1 0 38.4s 1 1 1 1 40.96s
<Requested_eDRX_value>	String type; half a byte in a 4-bit format. The eDRX value refers to bit 4 to 1 of octet 3 of the Extended DRX parameters information element. bits 4 3 2 1 E-UTRAN eDRX value 0 0 1 0 20.48s 0 0 1 1 40.96s 0 1 0 1 81.92s 1 0 0 1 163.84s 1 0 1 0 327.68s 1 0 1 1 655.36s 1 1 0 0 1310.72s 1 1 0 1 2621.44s 1 1 1 0 5242.88s 1 1 1 1 10485.76s
<NW-provided_eDRX_value>	String type; half a byte in a 4-bit format. The eDRX value refers to bit 4 to 1 of octet 3 of the Extended DRX parameters information element.

	bits 4 3 2 1 E-UTRAN eDRX value 0 0 1 0 20.48s 0 0 1 1 40.96s 0 1 0 1 81.92s 1 0 0 1 163.84s 1 0 1 0 327.68s 1 0 1 1 655.36s 1 1 0 0 1310.72s 1 1 0 1 2621.44s 1 1 1 0 5242.88s 1 1 1 1 10485.76s
<Paging_time_window>	String type; half a byte in a 4 bit format. The paging time window referes to bit 8 to 5 of octet 3 of the Extended DRX parameters information element. bits 4 3 2 1 Paging time window 0 0 0 0 2.56s 0 0 0 1 5.12s 0 0 1 0 7.68s 0 0 1 1 10.24s 0 1 0 0 12.8s 0 1 0 1 15.36s 0 1 1 0 17.92s 0 1 1 1 20.48s 1 0 0 0 23.04s 1 0 0 1 25.6s 1 0 1 0 28.16s 1 0 1 1 30.72s 1 1 0 0 33.28s 1 1 0 1 35.84s 1 1 1 0 38.4s 1 1 1 1 40.96s

3.1.27 AT+ECADC

This command is used to get thermal temperature and VBAT values sampled by ADC.

AT+ECADC	
Set Command AT+ECADC=<option>	Response +ECADC: <option>,<value>[,<option>,<value>] OK
Test Command AT+ECADC=?	Response +ECADC: <option> OK
Maximum Response Time	5s
Parameter Saving Mode	NO_SAVE

Parameter

<option>	String type	
	temp	Get current thermal temperature in unit of degree centigrade with 1 degree resolution.
	vbat	Get current VBAT value in unit of mV.
	all	Get current thermal temperature and VBAT value.
<lifetime>	Integer type; Corresponding value of option	

Example

```
AT+ECADC=all
+ECADC: TEMP,26,VBAT,3604

OK
```

3.1.28 AT+ECIPR

Set command sets the UE baud rate to be used.

Read command returns the current baud rate.

Test command returns baud rates supported by the UE.

AT+ECIPR	
Set Command AT+ECIPR=<rate>	Response OK

	If there is any error, response:： +CME ERROR: <err>
Read Command AT+ECIPR?	Response +ECIPR: <rate> OK
Test Command AT+ECIPR=?	Response +ECIPR: (list of fixed-only <rate> value) OK
Maximum Response Time	5s
Parameter Saving Mode	NO_SAVE

Parameter

<rate>	Baud rate at which the UE will accept commands.

Example

```
AT+ECIPR=9600
OK

AT+ECIPR?
+ECIPR: 115200

OK
AT+ECIPR=?
+ECIPR:(300,600,1200,2400,4800,9600,19200,38400,57600,115200,230400,460800)

OK
```

3.1.29 AT+ECSNTP

This command is used to synchronize the local time with the Universal Time Coordinated (UTC) via the SNTP server.

The test command returns the supported parameters

The set command sets server name and start to synchronize the local time with the Universal Time Coordinated (UTC). It will return immediately, the UTC content will be returned via URC.

AT+ECSNTP	
Test Command AT+ECSNTP=?	Response +ECSNTP: ("IP ADDR\URL"),(0-65535),(0,1) OK
Set Command AT+ECSNTP=<server>[,<port>,<autosync>]	Response OK +ECSNTP: <time> If there is any error, response:： +CME ERROR: <err>
Maximum Response Time	5s
Parameter Saving Mode	NO_SAVE

Parameter

<server>	String type	
	Address of the NTP server. The format is a domain name or a dotted decimal IP address.	
<port>	Integer type	
	The NTP server port number, default is 123	
<autosync>	Integer type; Whether to automatically set synchronized time to local UTC. Default is 1	
	0	not set
	1	set
<time>	String type	
	Time synchronized from NTP server The format is: “yy/mm/dd: hh/mm/ss”	

Example

3.2 Socket Commands(Solution A)

3.2.1 AT+SKTCREATE

The command creates a socket on the UE and associates with specified protocol. UE supports up to five sockets(TCP or UDP) at the same time. And will return error if it is exceeded.

The test command returns values supported as a compound value.

AT+SKTCREATE	
Set Command AT+SKTCREATE=<domain>,<type>,<protocol>	**Response** +SKTCREATE: <fd> OK **If there is any error, response:** +SOCKET ERROR: <err>
Test Command AT+SKTCREATE=?	**Response** +SKTCREATE: (list of supported <domain>s), (list of supported <type>s), (list of supported <protocol>s) OK
Maximum Response Time	5s
Parameter Saving Mode	NO_SAVE

Parameter

<fd>	Integer type	
	1-7	Socket file description
<domain>	Integer type	
	1	IPV4
	2	IPV6
<type>	Integer type	
	1	TCP
	2	UDP
<protocol>	Integer type; standard internet protocol definition	
	6	IPPROTO_TCP
	17	IPPROTO_UDP

Example

```
AT+SKTCREATE=1,1,17
+SKTCREATE: 1
```

```
OK
```

3.2.2 AT+SKTCONNECT

For TCP, the command connect socket with remote address and port.
For UDP, the command save remote address and port for send

AT+SKTCONNECT	
Set Command AT+SKTCONNECT=<fd>,<addr>,<port>	**Response** OK **If there is any error, response:** +SOCKET ERROR: <err>
Test Command AT+SKTCONNECT=?	**Response** +SKTCONNECT: (list of supported <fd>s), (<addr>), (list of supported <port>s) OK
Maximum Response Time	10s
Parameter Saving Mode	NO_SAVE

Parameter

<fd>	Integer type	
	1-7	Socket file description returned by +SKTCREATE
<addr>	string type	
	Remote address to connect or send to	
<port>	Integer type	
	Remote port to connect or send to	

Example

```
```

3.2.3 AT+SKTBIND

The command bind socket with local address and port. If the address is default, it means any address.

AT+SKTBIND	
Set Command AT+SKTBIND=<fd>,<addr>,<port>	**Response** OK **If there is any error, response:** +SOCKET ERROR: <err>
Test Command	**Response**

AT+SKTBIND=?	+SKTBIND: (list of supported <fd>s), (<addr>), (list of supported <port>s) OK
Maximum Response Time	5s
Parameter Saving Mode	NO_SAVE

Parameter

<fd>	Integer type	
	1-7	Socket file description returned by +SKTCREATE
<addr>	string type	
	address to bind. If address is defaults means any address.	
<port>	Integer type	
	port to bind	

Example

3.2.4 AT+SKTSEND

Send a containing length bytes of data to remote port on remote addr.

AT+SKTSEND	
Set Command AT+SKTSEND=<fd>,<datalen>,<data>[,<rai info>[,<except info>]]	**Response** OK **If there is any error, response:** +SOCKET ERROR: <err>
Test Command AT+SKTSEND=?	**Response** +SKTSEND: (list of supported <fd>s), (list of supported <data len>s), (<data>), (list of supported <rai info>s), (list of supported <expect info>s) OK
Maximum Response Time	5s
Parameter Saving Mode	NO_SAVE

Parameter

<fd>	Integer type	
	1-7	Socket file description returned by +SKTCREATE
<data len>	Integer type	
	length of data in hex string format, the max length is 512	

<data>	Integer type	
	Data in hex string	
<rai info>	Integer type (option)	
	0-2	release assistance indication
	0	no rai info
	1	no further uplink or downlink data transmission subsequent to the uplink data transmission subsequent to the uplink data transmission is expected
	2	only a single downlink data transmission and no further uplink data transmission subsequent to the uplink data transmission is expected
	Note: default value: 0	
<except info>	Integer type (option)	
	0-1	expect data indication
	0	disable expect data indication
	1	enable expect data indication
	Note: default value: 0	

Example

Send data:23456

```
AT+SKTSEND=0,5,3233343536
OK
```

3.2.5 +SKTRECV

This is an unsolicited message to show data has been received on a socket.

+SKTRECV

```
+SKTRECV: <fd>,<len>,<data>
```

Parameter

<fd>	Integer type	
	1-7	Socket file description that data from
<len>	String type	
	Received data length(bytes)	
<data>	String type	
	Received data in hex string format	

Example

3.2.6 +SKTERR

This is an unsolicited message to show the error number when error occur.

+SKTERR
+SKTERR: <fd>,<errno>

Parameter

<fd>	Integer type	
	1-7	Socket file description that data from
<errno>	Integer type	(Posix Errno defines)
	12	Out of memory error
	105	No buffer space available
	62	Timer expired
	113	No route to host
	115	Operation now in progress
	22	Invalid argument
	11	Operation would block
	107	Transport endpoint is not connected
	103	Software caused connection abort
	104	Connection reset by peer

Note: if there is some error with the socket which is connected, the socket will be closed automatic

Example

3.2.7 AT+SKTSTATUS

Get the status of a socket by file description.

AT+SKTSTATUS	
Set Command AT+SKTSTATUS=<fd>	**Response** +SKTSTATUS: <status> OK **If there is any error, response:** +SOCKET ERROR: <err>
Test Command AT+SKTSTATUS=?	**Response** +SKTSTATUS: (list of supported <fd>s) OK

Maximum Response Time	5s
Parameter Saving Mode	NO_SAVE

Parameter

<fd>	Integer type	
	1-7	Socket file description returned by +SKTCREATE
<status>	Integer type	
	1	Not connected
	2	Connecting
	3	Connected

Example

S3.2.8 AT+SKTDELETE

Delete a socket by file description.

AT+SKTDELETE	
Set Command AT+SKTDELETE=<fd>	**Response** OK **If there is any error, response:** +SOCKET ERROR: <err>
Test Command AT+SKTDELETE=?	**Response** +SKTDELETE: (list of supported <fd>s) OK
Maximum Response Time	5s
Parameter Saving Mode	NO_SAVE

Parameter

<fd>	Integer type	
	1-7	Socket file description returned by +SKTCREATE

Example

3.3 LwM2M Commands

3.3.1 AT+LWM2MCREATE

This command creates an instance of lwM2M client and register with lwM2M server. It need specify sever, port, ender point name, lifetime. Also can set psk id and psk if need DTLS.

AT+LWM2MCREATE	
Set Command AT+LWM2MCREATE=<server>,<port>,<local port>,<enderpoint>,<life time>[,<psk id>,<psk>]	Response +LWM2MCREATE: <clientId> OK If there is any error, response: +LWM2M ERROR: <err>
Test Command AT+LWM2MCREATE=?	Response +LWM2MCREATE: "<server>",(range of supported<port>),(range of supported<local port>),"<enderpoint>",(range of supported<lifetime>),"<psk_id>","<psk>" OK
Maximum Response Time	5s
Parameter Saving Mode	NO_SAVE

Parameter

<clientId>	Integer type
	LwM2M client's id
<server>	String type
	LwM2M server's URL or IP address
<port>	Integer type
	LwM2M server's port number
<local port>	Integer type
	LwM2M client's local port
<enderpoint>	String type
	LwM2M client's enderpoint name
<life time>	Integer type
	LwM2M client's life time
<psk id>	String type
	LwM2M client's public identity
<psk>	String type

	LwM2M client's pre shared key

Example

```
AT+LWM2MCREATE="180.167.122.150",5683,56830,"client0",60
+LWM2MCREATE: 0
OK
```

3.3.2 AT+LWM2MDELETE

This command deletes a specified lwM2M client instance.

AT+LWM2MDELETE	
Set Command AT+LWM2MDELETE=<clientId>	Response OK If there is any error, response: +LWM2M ERROR: <err>
Test Command AT+LWM2MDELETE=?	Response +LWM2MDELETE: (list of supported<clientId>) OK
Maximum Response Time	5s
Parameter Saving Mode	NO_SAVE

Parameter

<clientId>	Integer type
	LwM2M client's id returned by +LWM2MCREATE

Example

3.3.3 AT+LWM2MADDOBJ

This command adds a lwM2M object to a specified lwM2M client instance. Definitions of object, instance and resource. please refer to Lightweight Machine to Machine Technical Specification, ext-label Objects Produced by IPSO Alliance and oma-label Objects Produced by OMA.
http://www.openmobilealliance.org/wp/OMNA/LwM2M/LwM2MRegistry.html

AT+LWM2MADDOBJ	
Set Command	Response

AT+LWM2MADDOBJ=<clientId>,<objectId>,<instanceId>,<resourceCount>,<resourceIds>	OK If there is any error, response: +LWM2M ERROR: <err>
Test Command AT+LWM2MADDOBJ=?	Response +LWM2MADDOBJ: (list of supported<clientId>),(range of supported<objectId>),(range of supported<instanceId>),(range of supported<resourceCount>),"<resourceIds>" OK
Maximum Response Time	5s
Parameter Saving Mode	NO_SAVE

Parameter

<clientId>	Integer type
	LwM2M client's id returned by +LWM2MCREATE
<objectId>	Integer type
	Object id number
<instanceId>	Integer type
	Instance id number
<resourceCount>	Integer type
	Number of resources
<resourceIds>	String type
	Resources numbers separated by semicolons

Example

```
AT+LWM2MADDOBJ=0,3306,111,3,"5750;5850;5851"
OK
```

3.3.4 AT+LWM2MDELOBJ

This command deletes an object from a specified lwM2M client instance.

AT+LWM2MDELOBJ	
Set Command AT+LWM2MDELOBJ=<clientId>,<objectId>	Response OK If there is any error, response: +LWM2M ERROR:<err>
Test Command	Response

AT+LWM2MDELOBJ=?	+LWM2MDELOBJ: (list of supported<clientId>),(range of supported<objectId>) OK
Maximum Response Time	5s
Parameter Saving Mode	NO_SAVE

Parameter

<clientId>	Integer type
	LwM2M client's id returned by +LWM2MCREATE
<objectId>	Integer type
	Object id number

Example

3.3.5 +LWM2MREAD

This is an unsolicited message to represent lwM2M client receive the LwM2M server's reading command.

+LWM2MREAD
+LWM2MREAD: <clientId>,<objectId>,<instanceId>,<resId>

Parameter

<clientId>	Integer type
	LwM2M client's id returned by +LWM2MCREATE
<objectId>	Integer type
	Object id number that lwM2M server want to read
<instanceId>	Integer type
	Instance id number that lwM2M server want to read
<resId>	Integer type
	Ressource id number that lwM2M server want to read

Example

```
+LWM2MREAD: 0,3306,111,5750
```

3.3.6 +LWM2MWRITE

This is an unsolicited message to represent lwM2M client receive the LwM2M server's writing command.

+LWM2MWRITE

```
+LWM2MWRITE:
<clientId>,<objectId>,<instanceId>,<num>[,<resId>,<type>,<length>,<valueStr
>]
```

Parameter

<clientId>	Integer type	
	LwM2M client's id returned by +LWM2MCREATE	
<objectId>	Integer type	
	Object id number that lwM2M server want to write	
<instanceId>	Integer type	
	Instance id number that lwM2M server want to write	
<num>	Integer type	
	The number of resources need to be write	
<resId>	Integer type	
	Resource id number that lwM2M server want to write	
<type>	String type	
	"S"	String type
	"O"	Opaque type
	"I"	Integer type
	"F"	Float type
<length>	Integer type	
	value length in bytes	
<valueStr>	String type	
	value need to write to resource	

Example

```
+LWM2MWRITE: 0,3306,111,5750,O,4,"54595045"
```

3.3.7 +LWM2MEXECUTE

This is an unsolicited message to represent lwM2M client receive the LwM2M server's execution command.

+LWM2MEXECUTE

```
+LWM2MEXECUTE:
<clientId>,<objectId>,<instanceId>,<resId>,<length>,<valueStr>
```

Parameter

<clientId>	Integer type
	LwM2M client's id returned by +LWM2MCREATE
<objectId>	Integer type
	Object id number that lwM2M server want to execute
<instanceId>	Integer type
	Instance id number that lwM2M server want to execute
<resId>	Integer type
	Resource id number that lwM2M server want to execute
<length>	Integer type
	value length
<valueStr>	String type
	value of execute command

Example

```
+LWM2MEXECUTE: 0,3303,0,5605,2,"ok"
```

3.3.8 +LWM2MOBSERVE

This is an unsolicited message to represent lwM2M client receive the LwM2M server's observation command.

+LWM2MOBSERVE

```
+LWM2MOBSERVE: <clientId>,<oper>,<objectId>,<instanceId>,<resId>
```

Parameter

<clientId>	Integer type	
	LwM2M client's id returned by +LWM2MCREATE	
<oper>	Integer type	
	0	Observe object instance
	1	Cancel observe
<objectId>	Integer type	
	Object id number that lwM2M server want to observe	
<instanceId>	Integer type	
	Instance id number that lwM2M server want to observe	
<resId>	Integer type	
	Resource id number that lwM2M server want to observe	

Example

To observe 3306/111/5750:

```
+LWM2MOBSERVE: 0,0,3306,111,5750
```

Cancel observe 3306/111/5750:

```
+LWM2MOBSERVE: 0,1,3306,111,5750
```

3.3.9 AT+LWM2MREADCONF

This command response lwM2M server's read command

AT+LWM2MREADCONF	
Set Command AT+LWM2MREADCONF=<clientId>,<objectId>,<instanceId>,<resId>,<valuetype>,<valuelen>,<value>	Response OK If there is any error, response: +LWM2M ERROR: <err>
Test Command AT+LWM2MREADCONF=?	Response +LWM2MREADCONF:(list of supported<clientId>),(range of supported<objectId>),(range of supported<instanceId>),(range of supported<resId>),(range of supported<valuetype>),(range of supported<valuelen>),"<value>" OK
Maximum Response Time	5s
Parameter Saving Mode	NO_SAVE

Parameter

<clientId>	Integer type
	LwM2M client's id returned by +LWM2MCREATE
<objectId>	Integer type
	Object id number
<instanceId>	Integer type
	Instance id number
<resId>	Integer type

	Resource id number	
<valuetype>	Integer type	
	0	string
	1	opaque
	2	Integer
	3	float
	4	bool
	Other value	undefined
<valuelen>	Integer type	
	Value length	
<value>	String type	
	Value, if type is opaque, value in hex string format	

Example

Value type is string:

```
AT+LWM2MREADCONF=0,3306,0,5750,0,5,”hello”
```

Value type is opaque:

```
AT+LWM2MREADCONF=0,12001,0,4,1,5,”3432383330”
```

Value type is Integer:

```
AT+LWM2MREADCONF=0,3306,0,5851,2,3,”206”
```

Value type is float:

```
AT+LWM2MREADCONF=0,3303,0,5601,3,4,”3.14”
```

Value type is bool:

```
AT+LWM2MREADCONF=0,3306,0,5850,4,1,”1”
OK
```

3.3.10 AT+LWM2MWRITECONF

This command response lwM2M server’s write command

AT+LWM2MWRITECONF	
Set Command AT+LWM2MWRITECONF=<clientId>,<result>	Response OK If there is any error, response: +LWM2M ERROR: <err>
Test Command AT+LWM2MWRITECONF=?	Response +LWM2MWRITECONF: (list of supported<clientId>),(range of supported<result>)

	OK
Maximum Response Time	5s
Parameter Saving Mode	NO_SAVE

Parameter

<clientId>	Integer type	
	LwM2M client's id returned by +LWM2MCREATE	
<result>	Integer type; result of write command	
	0x44	Write success
	0x8c	Time out
	0x84	Object not found
	Refer to rfc 7252	

Example

```
AT+LWM2MWRITECONF=0,68
OK
```

3.3.11 AT+LWM2MEXECUTECONF

This command response lwM2M server's execute command

AT+LWM2MEXECUTECONF	
Set Command AT+LWM2MEXECUTECONF=<clientId>,<result>	Response OK If there is any error, response: +LWM2M ERROR: <err>
Test Command AT+LWM2MEXECUTECONF=?	Response +LWM2MEXECUTECONF: (list of supported<clientId>),(range of supported<result>)
Maximum Response Time	5s
Parameter Saving Mode	NO_SAVE

Parameter

<clientId>	Integer type	
	LwM2M client's id returned by +LWM2MCREATE	
<result>	Integer type; result of write command	
	0x44	execute success
	0x8c	Time out

	0x84	Object not found
	Refer to rfc 7252	

Example

```
AT+LWM2MEXECUTECONF=0,68
OK
```

3.3.12 AT+LWM2MNOTIFY

Notify lwM2M server a specified resource changed

AT+LWM2MNOTIFY	
Set Command AT+LWM2MNOTIFY=<clientId>,<objectId>,<instanceId>,<resourceId>,<valuetype>,<valuelen>,<value>	Response OK If there is any error, response: +LWM2M ERROR: <err>
Test Command AT+LWM2MNOTIFY=?	Response +LWM2MNOTIFY:(list of supported<clientId>),(range of supported<objectId>),(range of supported<instanceId>),(range of supported<resourceId>),(range of supported<valuetype>),(range of supported<valuelen>),"<value>" OK
Maximum Response Time	5s
Parameter Saving Mode	NO_SAVE

Parameter

<clientId>	Integer type
	LwM2M client's id returned by +LWM2MCREATE
<objectId>	Integer type
	Object id number
<instanceId>	Integer type
	Instance id number
<resource>	Integer type
	Resource id number
<valuetype>	Integer type
	0 string

	1	opaque
	2	Integer
	3	float
	4	bool
	Other value	undefined
<valuelen>	Integer type	
	Value length	
<value>	String type	
	Value, if type is opaque, value in hex string format	

Example

```
AT+LWM2MNOTIFY=0,3303,0,5601,3,4,”3.14”
OK
```

3.3.13 AT+LWM2MUPDATE

This command updates the register information, with or not with object id’s update

AT+LWM2MNOTIFY	
Set Command AT+LWM2MUPDATE=<clientId>[,<withobj>]	Response OK If there is any error, response: +LWM2M ERROR: <err>
Test Command AT+LWM2MUPDATE=?	Response +LWM2MUPDATE: (list of supported<clientId>),(list of supported<withobj>) OK
Maximum Response Time	5s
Parameter Saving Mode	NO_SAVE

Parameter

<clientId>	Integer type	
	LwM2M client’s id returned by +LWM2MCREATE	
<withobj>	Integer type	
	0	not update the object information
	1	Update the object information

Example

```
AT+LWM2MUPDATE=0
OK
```

3.3.14 Summary of <err> Codes

<err> Codes	Description
ERROR	Input wrong AT command, for example: contains chinese characters
PARAMETER ERROR	Input wrong parameter, for example: parameter out of range
CANNOT CREATE SEMPH	Can't create semaphore
CONFIG ERROR	Configuration lwm2m client error
NO FREE CLIENT	No free client left, now only support one client at one time
OPERATION NO SUPPORT	Not support operation, such as GET command
NO FIND CLIENT	Not find this client
ADD OBJECT FAILED	Fail to add object
NO FIND OBJECT	Not find this object id
DELETE OBJECT FAILED	Fail to delete the object
NETWORK NOT READY	Network not ready, can't use data service

3.4 CoAP Commands

3.4.1 AT+COAPCREATE

The command creates a CoAP client.

AT+COAPCREATE	
Set Command AT+COAPCREATE=<port>	**Response** OK **If there is any error, response:** +CME ERROR: <err>
Test Command AT+COAPCREATE=?	**Response** +COAPCREATE: (range of supported<port>) OK **If there is any error, response:** +CME ERROR: <err>
Maximum Response Time	10s
Parameter Saving Mode	NO_SAVE

Parameter

<port>	Integer type; server's port, values of 0-65535 are supported

Example

3.4.2 AT+COAPDEL

The command deletes the CoAP client

AT+COAPDEL	
Set Command AT+COAPDEL	**Response** OK **If there is any error, response:** +CME ERROR: <err>

Maximum Response Time	5s
Parameter Saving Mode	NO_SAVE

Example

3.4.3 AT+COAPADDRES

The command adds the CoAP resource

AT+COAPCREATE	
Set Command AT+COAPADDRES=<length>,<resource>	**Response** OK **If there is any error, response:** +CME ERROR: <err>
Test Command AT+COAPADDRES=?	**Response** +COAPADDRES: (range of supported<length>),"<resource>" OK **If there is any error, response:** +CME ERROR: <err>
Maximum Response Time	10s
Parameter Saving Mode	NO_SAVE

Parameter

<length>	Integer type
	The CoAP client resources, range: 1-50
<resource>	String type
	The resource name

Note: This command is not supported now.

Example

3.4.4 AT+COAPHEAD

The command adds the CoAP head.

AT+COAPHEAD	
Set Command AT+COAPHEAD=<mode>[,[<msgid>][,<tkl>,<token>]]	**Response** OK **If there is any error, response:** +CME ERROR: <err>
Test Command AT+COAPHEAD=?	**Response** +COAPHEAD: <mode>[,[<msgid>][,<tkl>,<token>]] OK **If there is any error, response:** +CME ERROR: <err>
Maximum Response Time	10s
Parameter Saving Mode	NO_SAVE

Parameter

<mode>	Integer type, The CoAP Head and Token parameter	
	1	Generate message id and token values randomly.
	2	Generate message id, and configure the token values.
	3	Only configure message id, not needed token values.
	4	Configure message id, and generate the token values randomly.
	5	Configure message and token values.
<msgid>	Integer type	
	The message id, only needed configure when the <mode> value is 3, 4, 5. Range value: 0-65535	
<tkl>	Integer type	
	The token values length, only needed configure when the <mode> value is 1, 2. Range value: 1-8.	
<token>	String type	
	The token values, hexadecimal format string, only need configure when the <mode> value is 1, 2.	

Example

3.4.5 AT+COAPOPTION

The command adds the CoAP option.

AT+COAPOPTION	
Set Command AT+COAPOPTION=<opt_cnt>,<opt_name>,<opt_value>[,…]	**Response** OK **If there is any error, response:** +CME ERROR: <err>
Test Command AT+COAPOPTION=?	**Response** +COAPOPTION: <opt_cnt>,<opt_name>,<opt_value>[,…] OK **If there is any error, response:** +CME ERROR: <err>
Maximum Response Time	10s
Parameter Saving Mode	NO_SAVE

Parameter

<opt_cnt>	Integer type	
	The option parameter count, range value: 1-12	
<opt_name>	String type, The option name, refer the RFC 7252	
	1	If-Match
	3	Uri-Host
	4	ETag
	5	If-None-Match
	6	Observe
	7	Uri-Port
	8	Location-Path
	11	Uri-Path
	12	Content-Format
	14	Max-Age
	15	Uri-Query
	17	Accept
	20	Location-Query
	23	Block2
	27	Block1
	28	SIZE
	35	Proxy-Uri

	39	Proxy-Scheme
	60	Size1
<opt_value>	String type, The length of value string: 1-180. If the <opt_name> is 12 or 17, the <opt_value> should be the below value	
	0	text-plain
	40	application/link-format
	41	application/xml
	42	application/octet-stream
	47	application/exi
	50	application/json

Example

```
```

3.4.6 AT+COAPSEND

The command send data to CoAP server.

AT+COAPSEND	
Set Command AT+COAPSEND=<msgType>,<method>,<ipAddr>,<port>[,<length>,<data>] **Note: After ">" is responded,input the data to be sent.Tab "CTRL + Z" to send,and tab "ESC" to cancel the opration.**	**Response** OK **If there is any error, response:** +COAP ERROR: <err>
Test Command AT+COAPSEND=?	**Response** +COAPSEND: <msgType>,<method>,<ipAddr>,<port>[,<length>,<data>] OK **If there is any error, response:** +CME ERROR: <err>
Maximum Response Time	10s
Parameter Saving Mode	NO_SAVE

Parameter

<msgType>	Integer type	
	0	CON, confirmable message(requires ACK/RST)
	1	NON, non-confirmable message(one-shot message)
	2	ACK, used to acknowledge confirmable message
	3	RST, reset, indicates error in received message
<methon>	String type	
	1	GET
	2	POST
	3	PUT
	4	DELETE
<ipAddr>	String type	
	The CoAP Server ip address	
<method>	Integer type	
	1	GET
	2	POST
	3	PUT
	4	DELETE
<port>	Integer type	
	The CoAP Server Port	
<length>	Integer type	
	The length of data to be sent, the max length is 512 Bytes	
<data>	string type	
	The length of data to be sent, hex string	

Example

3.4.7 AT+COAPDATASTATUS

The command gets the CoAP data status.

AT+COAPDATASTATUS	
Test Command	**Response**
AT+COAPDATASTATUS=?	+COAPDATASTATUS: <status> OK **If there is any error, response:**

	+CME ERROR: <err>
Maximum Response Time	10s
Parameter Saving Mode	NO_SAVE

Parameter

<status>	Integer type	
	0	Have not sent
	1	Sent, waiting response of IoT platform(not supported)
	2	Sent failed(not supported)
	3	Timeout(not supported)
	4	Success
	5	Got reset message(not supported)

Example

3.4.8 AT+COAPCFG

The command configs the CoAP client

AT+COAPCFG	
Set Command AT+COAPCFG="Showra"[,<Showra>]	**Response** OK **If there is any error, response:** +CME ERROR: <err>
Set Command AT+COAPCFG="Showrspopt"[,<Showrspopt>]	**Response** OK **If there is any error, response:** +CME ERROR: <err>
Test Command AT+COAPCFG=?	**Response** OK
Maximum Response Time	10s
Parameter Saving Mode	NO_SAVE

Parameter

<Showra>	Integer type; Set whether or not to display the address of sender	
	1	Do not display ip address and port
	2	The IP address and port are displayed in URC in the following format: +COAPURC: "rsp",[<ip_addr>,<port>],<type>,<rspcode>,<msgid>[,<length>,<data>]
<Showrspopt>	Integer type, Set whether or not to display the coap option of sender	
	0	Do not show CoAP options
	1	The CoAP options are displayed in URC in the following format: +COAPURC:"rsp",<type>,<rspcode>,<msgid>[,<opt_count>,<opt_name>,"<opt_value>"[,...]][,<length>,<data>]

3.4.9 AT+COAPALISIGN

The command gets the ali cloud sign

AT+COAPALISIGN	
Set Command AT+COAPALISIGN=”<devID>”,”<devName>”,”<devSecret>”,”<productKey>”	**Response** +COAPALISIGN: ”<sign>” OK **If there is any error, response:** +CME ERROR: <err>
Test Command AT+COAPALISIGN=?	**Response** +COAPALISIGN: " <devID>”,”<devName>”,”<devSecret>”,”<productKey>” OK **If there is any error, response:** +CME ERROR: <err>
Maximum Response Time	10s
Parameter Saving Mode	NO_SAVE

Parameter

<devID>	String type
	Device ID issued by AliCloud.
<devName>	String type
	Device name issused by AliCloud.
<devSecret>	String type

	Device secret key issused by AliCloud
<productKey>	String type
	Product key issued by AliClode
<sign>	String type
	The calculated sign value

3.4.10 +COAPURC

This is an unsolicited message to indicate CoAP client receive response from CoAP server.

+COAPURC
+COAPURC: "rsp",[<ip_addr>,<port>,]<type>,<rspcode>,<msgid>[,<opt_count>,<opt_name>,"<opt_value>"[,...]][,<length>,<data>]

Parameter

<ip_addr>	String type
	The CoAP server ip address, it will show when set AT+COAPCFG=”Showra”,1
<port>	String type
	The CoAP server port, it will show when set AT+COAPCFG=”Showra”,1
<type>	Integer type; The CoAP Protocol of message type, range: 0-3, refer the RFC 7252
	0 CON
	1 NON
	2 ACK
	3 RST
<rspcode>	String type; The response code of CoAP Protocol. Refer to the RFC 7252
	0
	2.01
	2.02
	2.03
	2.04
	2.05
	4.00
	4.01
	4.02
	4.03
	4.04
	4.05
	4.06
	4.12
	4.13

	4.15
	5.00
	5.01
	5.02
	5.03
	5.04
	5.05
<msgid>	Integer type
	The CoAP message id
<opt_cnt>	Integer type
	The count of option, it will show when set AT+COAPCFG=”Showrspopt”,1
<opt_name>	Integer type
	The option name, it will show when set AT+COAPCFG=”Showrspopt”,1
<opt_value>	Integer type
	The option value, it will show when set AT+COAPCFG=”Showrspopt”,1
<length>	Integer type
	The data length. The max length is 512 bytes
<data>	String type
	Receive data from server

Example

```
```

3.4.10 +COAPURC

This is an unsolicited message to indicate CoAP client receive data from CoAP server.

+COAPURC
+COAPURC: "req",[<ip_addr>,<port>,]<type>,<method>,<msgid>,<mode>[,<tkl>,<token>][,<opt_name>,"<opt_value>"[,...]][,<length>,<data>]

Parameter

<ip_addr>	String type
	The CoAP server ip address, it will show when set AT+COAPCFG=”Showra”,1
<port>	String type
	The CoAP server port, it will show when set AT+COAPCFG=”Showra”,1
<type>	Integer type; The CoAP Protocol of message type, range: 0-3, refer the RFC 7252

	0	CON
	1	NON
	2	ACK
	3	RST
<method>	Integer type	
	1	GET
	2	POST
	3	PUT
	4	DELETE
<msgid>	Integer type	
	The CoAP message id	
<mode>	Integer type; Indicates the existence of token, option, and data. Hexadecimal format	
	Bit 0 : The existence of token	
	Bit 1-6: The count of option	
	Bit 7 : The exitence of data	
<tkl>	Integer type	
	The token value length	
<token>	String type	
	The token value. Hexadecimal format	
<opt_cnt>	Integer type	
	The count of option, it will show when set AT+COAPCFG=”Showrspopt”,1	
<opt_name>	Integer type	
	The option name, it will show when set AT+COAPCFG=”Showrspopt”,1	
<opt_value>	Integer type	
	The option value, it will show when set AT+COAPCFG=”Showrspopt”,1	
<length>	Integer type	
	The data length. The max length is 512 bytes	
<data>	String type	
	Receive data from server	

Example

3.5 MQTT Commands

3.5.1 AT+ECMTCFG

The command config a MQTT client.

AT+ECMTCFG	
Set Command AT+ECMTCFG="echomode",<tcpconnectID>[,<echomode>]	**Response** OK if **<echomode>** is omitted,query the data echomode: +ECMTCFG: "echomode",<echomode> OK **If there is any error, response:** ERROR
Set Command AT+ECMTCFG="dataformat",<tcpconnectID>[,<send_format>[,<recv_format>]]	**Response** OK if <**send_format**> and <**recv_format**> are both omitted, query the format of sent/received data: +ECMTCFG: "dataformat",<send_format>,<recv_format> OK **If there is any error, response:** ERROR
Set Command AT+ECMTCFG="version",<tcpconnectID>[,<version>]	**Response** OK if <**version**> and <**recv_format**> is omitted,query the MQTT Protoco version: +ECMTCFG: "dataformat",<send_format>,<recv_format> OK **If there is any error, response:** ERROR
Set Command AT+ECMTCFG="keepalive",<tcpconnectID>[,<keep-alive time>]	**Response** OK

	if <**keep-alive time**> is omitted,query the keep-alive time: +ECMTCFG: "keepalive",<keep-alive time> OK **If there is any error, response:** ERROR
Set Command AT+ECMTCFG="session",<tcpconnectID>[,<clean_session>]	**Response** OK if <**clean_session**> is omitted,query the session type: +ECMTCFG: "session",<clean_session> OK **If there is any error, response:** ERROR
Set Command AT+ECMTCFG="timeout",<tcpconnectID>[,<pkt_timeout>[,<retry_times>][,<timeout_notice>]]	**Response** OK if <**pkt_timeout**>,<**retry_times**>,<**timeout_notice**> are ommitted,query the timeout value of message delivery: +ECMTCFG: "timeout",<pkt_timeout>,<retry_times>,<timeout_notice> OK **If there is any error, response:** ERROR
Set Command AT+ECMTCFG="will",<tcpconnectID>[,<will_fg>[,<will_qos>,<will_retain>,"<will_topic>","<will_msg>"]]	**Response** OK if <**will_fg**>,<**will_qos**>,<**will_retain**>,<**will_topic**>and <**will_msg**>are ommitted,query the will information: +ECMTCFG: "will",<will_fg>,<will_qos>,<will_retain>,<will_topic>,<will_msg> OK **If there is any error, response:** ERROR
Set Command AT+ECMTCFG="aliauth",<tcpconnectI	**Response** OK

D>[,"<product_key>","<device_name>","<device_secret>"]	if <**product_key**>,<**device_name**> and <**device_secret**> are ommitted,query the device information: +ECMTCFG:"aliauth",<product_key>,<device_name>,<device_secret> OK **If there is any error, response:** ERROR
Test Command AT+ECMTCFG=?	**Response** +ECMTCFG: "echomode",(0),(0,1) "dataformat",(0),(0,1),(0,1) "keepalive",(0),(0-3600), "session",(0),(0,1) "timeout",(0),(1-60),(1-10),(0,1) "will",(0),(0,1),(0-2),(0,1),"will_topic","will_msg" "version",(0),(3,4) "aliauth",(0),"productkey","devicename","devicesecret" OK **If there is any error, response:** ERROR
Maximum Response Time	5s
Parameter Saving Mode	NO_SAVE

Parameter

< tcpconnectID >	Integer type	
	MQTT socket identifier, the value is 0	
<echomode>	Integer type; whether to echo the input datato UART in data mode (**Not Supported**)	
	0	Don't echo the input data to UART
	1	Echo the input data to UART
<send_format>	Integer type; The format of sent data	
	0	Text mode
	1	Hex mode
<recv_format>	Integer type; The format of received data(**Not Supported**)	
	0	Text mode
	1	Hex mode
<keep-alive time>	Integer type	
	The range is 0-3600. The default value is 120. Unit: second. It defines the maximum	

	time interval between messages received from a client. If the server does not receive a message from the client within 1.5 times of the keep-alive time period, it disconnects the client as if the client has sent a DISCONNECT message.	
<clean_session>	Integer type	
	0	The server must store the subscriptions of the client after it disconnects.
	1	The server must discard any previously maintained information about the client
<pkt_timeout>	Integer type	
	Timeout of the packet delivery. The range is 1-60. The default value is 10. Unit: second.	
<retry_times>	Integer type.(**Not Support**)	
	Retry times when packet delivery times out. The range is 0-10. The default value is 3.	
<timeout_notice>	Integer type.(**Not Support**)	
	0	Not report timeout message when transmitting packet
	1	Report timeout message when transmitting packet
<will_fg>	Integer type. Configure the Will flag	
	The Will message defines the content of the message that is published to the will topic if the client is unexpectedly disconnected. It can be a zero-length message.	
<will_qos>	Integer type; Quality of service for message delivery	
	0	at most once
	1	at least once
	2	Exactly once
<will_retain>	Integer type. The Will retain flag is only used on PUBLISH messages.	
	0	When a client sends a PUBLISH message to a server, the server will not hold on to the message after it has been delivered to the current subscribers
	1	When a client sends a PUBLISH message to a server, the server should hold on to the message after it has been delivered to the current subscribers
<will_topic>	String type	
	Will topic string	
<version>	Integer type Version of MQTT protocol, the default is MQTT v3.1.1	
	3	MQTT V3.1
	4	MQTT V3.1.1
<product_key**>**	String type	
	Product key issued by AliCloud	
<device_name**>**	String type	
	Device name issued by AliCloud	
<device_secret**>**	String type	
	Device secret key issued by AliCloud	

Example

3.5.2 AT+ECMTOPEN

The command is used to open a network for MQTT client.

AT+ECMTOPEN	
Set Command AT+ECMTOPEN=<tcpconnectID>,"<hostname>",<port>	**Response** OK +ECMTOPEN: <tcpconnectID>,<result> **If there is any error, response:** ERROR
Test Command AT+ECMTOPEN=?	**Response** +ECMTOPEN: (list of supported <tcpconnectID>s), "<hostname>",list of supported <port>s OK
Read Command AT+ECMTOPEN?	**Response** [+ECMTOPEN: <tcpconnectID>,"<host_name>",<port>] OK
Maximum Response Time	5s
Parameter Saving Mode	NO_SAVE

Parameter

<tcpconnectID>	Integer type
	MQTT socket identifier, the value is 0
<host_name>	String type
	The address of the server. It could be an IP address or a domain name. The maximum size is 100 bytes
<port>	Integer type
	The port of the server. The range is 1-65535
<result>	Result of the command execution

	-1	Failed to open network
	0	Opened network successfully

Example

3.5.3 AT+ECMTCLOSE

The command send MQTT subscribe packet.

AT+ECMTCLOSE	
Set Command AT+ECMTCLOSE=<tcpconnectID>	**Response** OK +ECMTCLOSE: <tcpconnectID>,<result> **If there is any error, response:** ERROR
Test Command AT+ECMTCLOSE=?	**Response** +ECMTCLOSE: (list of supported <tcpconnectID>) OK
Maximum Response Time	5s
Parameter Saving Mode	NO_SAVE

Parameter

<tcpconnectID>	Integer type	
	MQTT socket identifier, the value is 0	
<result>	Result of the command execution	
	-1	Failed
	0	Successfully

Example

3.5.4 AT+ECMTCONN

The command send MQTT subscribe packet.

AT+ECMTCONN	
Set Command AT+ECMTCONN=<tcpconnectID>,"<clientID>"[,"<username>"[,"<password>"]]	**Response** OK +ECMTCONN: <tcpconnectID>,<result>[,<ret_code>] **If there is any error, response:** ERROR
Test Command AT+ECMTCONN=?	**Response** +ECMTCONN: (list of supported <tcpconnectID>s),"<clientID>"[,"<username>"[,"<password>"]] OK
Read Command AT+ECMTCONN?	**Response** [+ECMTCONN: <tcpconnectID>,<state>] OK
Maximum Response Time	5s
Parameter Saving Mode	NO_SAVE

Parameter

<tcpconnectID>	Integer type
	MQTT socket identifier, the value is 0.
<clientID>	String type
	The client identifier.
<username>	String type
	User name of the client. It can be used for authentication.
<password>	String type
	Password corresponding to the user name of the client. It can be used for authentication.

Example

3.5.5 AT+ECMTDISC

The command send MQTT subscribe packet.

AT+ECMTDISC	
Set Command AT+ECMTDISC=<tcpconnectID>	**Response** OK +ECMTDISC: <tcpconnectID>,<result> **If there is any error, response:** ERROR
Test Command AT+ECMTDISC=?	**Response** +ECMTDISC: (list of supported <tcpconnectID>s) OK
Maximum Response Time	5s
Parameter Saving Mode	NO_SAVE

Parameter

<tcpconnectID>	Integer type
	MQTT socket identifier, the value is 0.

Example

3.5.6 AT+ECMTSUB

The command send MQTT subscribe packet.

AT+ECMTSUB	
Set Command AT+ECMTSUB=<tcpconnectID>,<msgID>, "<topic>",<qos>	**Response** OK +ECMTSUB: <tcpconnectID>,<msgID>,<result>[,<value

	>] **If there is any error, response:** ERROR
Test Command AT+ECMTSUB=?	**Response** +ECMTSUB: (list of supported <tcpconnectID>s), (list of supported <msgID>s),”<topic>”,(list of supported <qos>s) OK
Maximum Response Time	5s
Parameter Saving Mode	NO_SAVE

Parameter

<tcpconnectID>	Integer type
	MQTT socket identifier, the value is 0.
<msgID>	Integer type
	Message identifier of packet. The range is 1-65535
<topic>	String type
	Topic that the client wants to subscribe to or unsubscribe from. The maximum length is 255 bytes
<qos>	Integer type
	Message QoS, can be 0,1 or 2

Example

3.5.7 AT+ECMTUNS

The command send MQTT unsubscribe packet.

AT+ECMTSUB	
Set Command AT+ECMTUNS=<tcpconnectID>,<msgID>,”<topic>”	**Response** OK +ECMTUNS: <tcpconnectID>,<msgID>,<result>

	If there is any error, response: ERROR
Test Command AT+ECMTUNS=?	**Response** +ECMTUNS: (list of supported <tcpconnectID>s), (list of supported <msgID>s),"<topic>" OK
Maximum Response Time	5s
Parameter Saving Mode	NO_SAVE

Parameter

<tcpconnectID>	Integer type
	MQTT socket identifier, the value is 0.
<msgID>	Integer type
	Message identifier of packet. The range is 1-65535
<topic>	String type
	Topic that the client wants to subscribe to or unsubscribe from. The maximum length is 255 bytes

Example

3.5.8 AT+ECMTPUB

The command send MQTT publish packet .

AT+ECMTPUB	
Set Command AT+ECMTPUB=<tcpconnectID>,<msgID>,<qos>,<retain>,"<topic>","<payload>"	**Response** OK +ECMTPUB: <tcpconnectID>,<msgID>,<result>[,<value>] **If there is any error, response:** ERROR
Test Command AT+ECMTPUB=?	**Response** +ECMTPUB: (list of supported

	<tcpconnectID>s), (list of supported <msgID>s"), (list of supported <qos>s"), (list of supported <retain>s"),"<topic>", "payload" OK
Maximum Response Time	5s
Parameter Saving Mode	NO_SAVE

Parameter

<tcpconnectID>	Integer type	
	MQTT socket identifier, the value is 0.	
<msgID>	Integer type	
	Message identifier of packet. The range is 0-65535. It will be 0 only when <qos>=0	
<qos>	Integer type	
	Message QoS, can be 0,1 or 2	
<retain>	Integer type	
	0	Server should not retain the message
	1	Server should retain the message
<topic>	String type	
	Topic that needs to be published. The maximum length is 255 bytes	
<payload>	String type	
	Message that needs to be published. Maximum length is 700 bytes. If in data mode, the maximum length is 1024 bytes	

Example

3.5.9 +ECMTSTAT

This is an unsolicited message to indicate MQTT client receive data from MQTT server.

+ECMTSTAT
+ECMTSTAT: <tcpconnectID>,<err_code>

Parameter

<tcpconnectID>	Integer type
	MQTT socket identifier, the value is 0.

<err_code>	Integer type. Error code	
	1	Connection is closed or reset by peer

Example

```
```

3.5.10 +ECMTRECV

This is an unsolicited message to indicate MQTT client receive data from MQTT server.

+ECMTRECV
+ECMTRECV: <tcpconnectID>, <msgID>,<topic>,<data>

Parameter

<tcpconnectID>	Integer type
	MQTT socket identifier, the value is 0.
<msgID>	String type
	The message identifier of packet
<topic>	string type
	The topic that received from MQTT server
<data>	String type
	Receive data from server

Example

```
```

3.6 HTTP Commands

3.6.1 AT+HTTPCREATE

Set command creates a http or https client instance. Configure host, server certification, etc. It can create most 5 instance at one time.

Test command returns values supported as a compound value.

Note: only one instance and http was fully verified. https and multiple instances will be test later.

AT+HTTPCREATE	
Set Command AT+HTTPCREATE=<flag>,<host>[,<authuser>,<authpasswd>[,<totalCaCertlen>,<currentCaCertlen>,<caCert>[,<clientCertlen>,<clientCert>[,<clientPklen>,<clientPk>]]]	**Response** **If there are more commands need to enter:** +HTTP CMD: CONTIUE ENTER CMD **If all commands has enter:** +HTTPCREATE: <httpclientId> **If there is any error, response:** +HTTP ERROR: <err>
Test Command AT+HTTPCREATE=?	**Response** +HTTPCREATE: (list of supported< flag >s), "<host>" , "<authuser>" , "<authpasswd>" ,(range of supported< totalCaCertlen >), (range of supported< currentCaCertlen >), "<caCert>" ,(range of supported< clientCertlen >), "<clientCert>" ,(range of supported< clientPklen >), "<clientPk>" OK
Maximum Response Time	5s
Parameter Saving Mode	NO_SAVE

Parameter

<flag>	Integer type
	1 not last part of command
	0 last part of command
<host>	string type
	http server's host name
<authuser>	String type
	Authentication user name

<authpasswd>	String type
	Authentication password
<totalCaCertlen>	Integer type
	Total length of server certification, no larger than 4096 bytes
<currentCaCertlen>	Integer type
	Current length of server certification, each part no larger than 512 bytes
<caCert>	String type
	server certification in hex string
<clientCertlen>	Integer type
	length of client certification, no larger than 4096 bytes
<clientCert>	String type
	client certification in hex string
<clientPktlen>	Integer type
	length of client private key, no larger than 4096 bytes
<clientPk>	String type
	Client private key in hex string
< httpclientId >	Integer type
	http Client Id ,0-4

Example

```
AT+HTTPCREATE=0,”http://api.openweathermap.org:80”
+HTTPCREATE: 0
OK
```

3.6.2 AT+HTTPCON

Set command creates a socket and connects with a http server. Then creates a task to receive data come from http server.

Test command returns values supported as a compound value.

AT+HTTPCON	
Set Command AT+HTTPCON=<httpclientId>	**Response** OK **If there is any error, response:** +HTTP ERROR: <err>
Test Command AT+HTTPCON=?	**Response** +HTTPCON: (list of supported< httpclientId >) OK

Maximum Response Time	40s
Parameter Saving Mode	NO_SAVE

Parameter

<httpclientId>	Integer type
	http client id returned by +HTTPCREATE

Example

```
AT+HTTPCON=0
OK
```

3.6.3 AT+HTTPDESTROY

Set command closes a socket, stops receive data from the http server and free the memory that was allocated by the client when creation.

Test command returns values supported as a compound value.

AT+HTTPDESTROY	
Set Command AT+HTTPDESTROY=<httpclientId>	**Response** OK **If there is any error, response:** +HTTP ERROR: <err>
Test Command AT+HTTPDESTROY=?	**Response** +HTTPDESTROY: (list of supported< httpclientId >) OK
Maximum Response Time	5s
Parameter Saving Mode	NO_SAVE

Parameter

<httpclientId>	Integer type
	http client id returned by +HTTPCREATE

Example

```
AT+HTTPDESTROY=0
OK
```

3.6.4 AT+HTTPSEND

Set command sends data to the http server.

Test command returns values supported as a compound value.

AT+HTTPSEND	
Set Command AT+HTTPSEND=<httpclientId>,<method>,<pathlen>,<path>,<customheaderlen>,<customheader>,<contentTypelen>,<contentType>,<contentlen>,<content>	**Response** OK **If there is any error, response:** +HTTP ERROR: <err>
Test Command AT+HTTPSEND=?	**Response** +HTTPSEND: (list of supported<httpclientId>),(list of supported<method>), (range of supported< pathlen>), "<path>" , (range of supported<customheaderlen>), "<customheader>" ,(range of supported<contentTypelen>), "<contentType>" ,(range of supported<contentlen>), "<content>" OK
Maximum Response Time	5s
Parameter Saving Mode	NO_SAVE

Parameter

<httpclientId>	Integer type	
	http client id returned by +HTTPCREATE	
<method>	Integer type; http method	
	0	GET
	1	POST
	2	PUT
	3	DELETE
	4	HEAD
<pathlen>	Integer type	
	Lenth of path,0-255	
<path>	string type	
	Path	
<customheaderlen>	Integer type	

	Length of custom header,0-255
<customheader>	string type
	Customheader in hex string
<contentTypelen>	Integer type
	Length of content type,0-64
<contentType>	string type
	Content type
<contentlen>	Integer type,0-4095
	Length of content
<content>	String type
	User data need to send in hex string

Example

```
AT+HTTPSEND=0,0,90, "/data2.5/weather?q=shanghai&
appid=c592e14137c3471fa9627b44f6649db4&mode=xml&units=metric”

OK
```

3.6.5 +HTTPRESPH

This is an unsolicited message to represent response header.

+HTTPRESPH
+HTTPRESPH: <clientId>,<responseCode>,<headerlen>,<header>

Parameter

<clientId>	Integer type
	http client id returned by +HTTPCREATE
<responseCode>	Integer type
	http response code
<headerlen>	Integer type
	Length of http response header
<header>	string type
	Header

3.6.6 +HTTPRESPC

This is an unsolicited message to represent response content.

+HTTPRESPC

```
+HTTPRESPC: <clientId>,<flag>,<contentlength>,<blockcontentlen>,<content>
```

Parameter

<clientId>	Integer type	
	http client id returned by +HTTPCREATE	
<flag>	Integer type; if has more data	
	0	No more data
	1	Has more data
<contentlength>	Integer type	
	Length of content	
<blockcontentlen>	Integer type	
	Current block length	
<content>	string type	
	content	

3.6.7 +HTTPERR

This is an unsolicited message to represent error message when error happen.

+HTTPERR

```
+HTTPERR: <clientId>,<errorcode>
```

Parameter

<clientId>	Integer type	
	http client id returned by +HTTPCREATE	
<errorcode>	Integer type;if has more data	
	2	URL parse error
	4	Protocol error
	8	Connection timeout
	9	Connection error
	10	Connection fatal error
	11	Connection closed
	12	Need get more data

13	Buffer overflow error
14	Ssl fail

3.7 OneNET Extension Commands

3.7.1 AT+MIPLCONFIG

This command is used to enable/disable bootstrap mode and configure bootstrap server address or access server address.

Note: the command should be executed before the communication suite instance is created with AT+MIPLCREATE command.

AT+MIPLCONFIG	
Set Command AT+MIPLCONFIG=<bsMode>,<ip>,<port>	**Response** OK **If there is any error, response:** +CIS ERROR: <err>
Test Command AT+MIPLCONFIG=?	**Response** +MIPLCONFIG: (list of supported< bsMode>s, "<ip>","<port>" OK
Maximum Response Time	5s
Parameter Saving Mode	AUTO_SAVE

Parameter

<bsMode>	Integer type
	0 Disable bootstrap
	1 Enable bootstrap
	The default value is 0
<ip>	When <bsMode>=0,<ip> represents access server IP When <bsMode>=1,<ip> represents bootstrap server IP
<port>	When <bsMode>=0,<port> represents access server port When <bsMode>=1,<port> represents bootstrap server port

Example

```
AT+MIPLCONFIG=1,”183.230.40.39”,”5683”
OK
```

3.7.2 AT+MIPLCREATE

This command creates an instance of communication to CMIoT OneNET platform.

AT+MIPLCREATE	
Execution Command AT+MIPLCREATE	**Response** +MIPLCREATE:<ref> OK **If there is any error, response:** +CIS ERROR: <err>
Maximum Response Time	5s
Parameter Saving Mode	NO_SAVE

Example

3.7.3 AT+MIPLDELETE

This command deletes a specified OneNET communication instance.

AT+MIPLDELETE	
Set Command AT+MIPLDELETE=<ref>	**Response** OK **If there is any error, response:** +CIS ERROR: <err>
Test Command AT+MIPLDELETE=?	**Response** +MIPLDELETE: (list of supported<ref>) OK
Maximum Response Time	5s
Parameter Saving Mode	NO_SAVE

Parameter

<ref>	Integer type
	a specified OneNET communication instance id returned by +MIPLCREATE

Example

3.7.4 AT+MIPLOPEN

This command send login request to OneNET.

AT+MIPLOPEN	
Set Command AT+MIPLOPEN=<ref>,<lifetime>	**Response** OK **If there is any error, response:** +CIS ERROR: <err>
Test Command AT+MIPLOPEN=?	**Response** +MIPLOPEN: (list of supported<ref>),(range of supported<lifetime>) OK
Maximum Response Time	5s
Parameter Saving Mode	NO_SAVE

Parameter

<ref>	Integer type
	a specified OneNET communication instance id returned by +MIPLCREATE
<lifetime>	Integer type
	Instance's life time, in this time instance need send update msg to OneNET

Example

3.7.5 AT+MIPLCLOSE

This command send logout request to OneNET.

AT+MIPLCLOSE	
Set Command AT+MIPLCLOSE=<ref>	**Response** OK **If there is any error, response:** +CIS ERROR: <err>
Test Command AT+MIPLCLOSE=?	**Response** +MIPLCLOSE: (list of supported<ref>) OK

Maximum Response Time	5s
Parameter Saving Mode	NO_SAVE

Parameter

<ref>	Integer type
	a specified OneNET communication instance id returned by +MIPLCREATE

Example

3.7.6 AT+MIPLADDOBJ

This command add an object from a specified OneNET communication instance.

AT+MIPLADDOBJ	
Set Command AT+MIPLADDOBJ=<ref>,<objectid>,<instanceid>,<instancebitmap>,<attributecount>,<actioncount>	**Response** OK **If there is any error, response:** +CIS ERROR: <err>
Test Command AT+MIPLADDOBJ=?	**Response** +MIPLADDOBJ: (list of supported<ref>),(range of supported< objectid >),(range of supported< instanceid >), "<instanceBitmap>",(range of supported< attributecount >), (range of supported< actioncount >) OK
Maximum Response Time	5s
Parameter Saving Mode	NO_SAVE

Parameter

<ref>	Integer type
	a specified OneNET communication instance id returned by +MIPLCREATE
<objectid>	Integer type
	object id
<instancecount>	Integer type

	Number of instances
<instancebitmap>	string type
	Instance bitmap, each bit represents an instance, 0 means not available, 1 means available
<attributecount>	Integer type
	Number of attributes
<actioncount>	Integer type
	Number of actions

Example

```
AT+MIPLADDOBJ=0,3306,1,"1",7,1

OK
```

3.7.7 AT+MIPLDELOBJ

This command deletes an object from a specified OneNET communication instance.

AT+MIPLDELOBJ	
Set Command AT+MIPLDELOBJ=<ref>,<objectid>	**Response** OK **If there is any error, response:** +CIS ERROR: <err>
Test Command AT+MIPLDELOBJ=?	**Response** +MIPLDELOBJ: (list of supported<ref>), (range of supported< objectid >) OK
Maximum Response Time	5s
Parameter Saving Mode	NO_SAVE

Parameter

<ref>	Integer type
	a specified OneNET communication instance id returned by +MIPLCREATE
<objectid>	Integer type
	object id

Example

```

```

3.7.8 AT+MIPLNOTIFY

This command notify OneNET that specified values changed.

AT+MIPLNOTIFY	
Set Command AT+MIPLNOTIFY=<ref>,<msgid>,<objectid>,<instanceid>,<resourceid>,<valuetype>,<len>,<value>,<index>,<flag>[,<ackid>[,raiflag]]	**Response** OK **If there is any error, response:** +CIS ERROR: <err>
Test Command AT+MIPLNOTIFY=?	**Response** +MIPLNOTIFY: (list of supported<ref>),(range of supported<msgid>),(range of supported<objectid>),(range of supported<instanceid>),(range of supported<resourceid>),(range of supported<valuetype>),(range of supported<len>),"<value>",(list of supported<index>),(list of supported<flag>),(list of supported<ackid>),(list of supported<raiflag>) OK
Maximum Response Time	5s
Parameter Saving Mode	NO_SAVE

Parameter

<ref>	Integer type	
	a specified OneNET communication instance id returned by +MIPLCREATE	
<msgid>	Integer type	
	Msg id, msg id carried in +MIPLOBSERVE	
<objectid>	Integer type	
	object id	
<instanceid>	Integer type	
	Instance id	
<resourceid>	Integer type	
	Resource id	
<valuetype>	Integer type	
	1	string
	2	opaque

	3	Integer
	4	float
	5	bool
<len>	Integer type	
	Value type	len
	string	string len
	opaque	byte len
	integer	The number of bytes occupied by the interger. Option 2,4,8
	float	the number of bytes occupied by the float. Option 4,8
	bool	1
<value>	string type	
	value type	len
	string	string
	opaque	hex string
	integer	integer
	float	string
	bool	0:false; 1:ture
<index>	Integer type	
	serial number, for one command may not be able to send the entire content. It's from N~0. 0 is last part of content.	
<flag>	Integer type; if there are multiple msg	
	1	First msg
	2	Middle msg
	0	Last msg
<ackid>	Integer type	
	0	This notify will be sent in Non-confirmable(NON) message
	1-65535	This notify will be sent in Confirmable(CON) message and the value will indicated by "+MIPLEVENT"
<raiflag>	Integer type	
	0 (PS_SOCK_RAI_NO_INFO) disable RAI	
	1 (PS_SOCK_RAI_NO_UL_DL_FOLLOWED) enable RAI, release after send	
	2 (PS_SOCK_ONLY_DL_FOLLOWED) enable RAI, release after ACK has received	

Example

```
AT+MIPLNOTIFY=0,msgid,3306,0,5750,1,5,"hello",0,0,116
+MIPLEVENT: 0,26,116
OK
```

3.7.9 AT+MIPLREADRSP

After receive +MIPLREAD, This command return the read result to OneNET.

AT+MIPLREADRSP	
Set Command AT+MIPLREADRSP=<ref>,<msgid>,<result>[,<objectid>,<instanceid>,<resourceid>,<valuetype>,<len>,<value>,<index>,<flag>[,raiflag]]	**Response** OK **If there is any error, response:** +CIS ERROR: <err>
Test Command AT+MIPLREADRSP=?	**Response** +MIPLREADRSP: (list of supported<ref>),(range of supported<msgid>),(range of supported<objectid>),(range of supported<instanceid>),(range of supported<resourceid>),(range of supported<valuetype>),(range of supported<len>),"<value>",(list of supported<index>),(list of supported<flag>),(list of supported<ackid>),(list of supported<raiflag>) OK
Maximum Response Time	5s
Parameter Saving Mode	NO_SAVE

Parameter

<ref>	Integer type	
	a specified OneNET communication instance id returned by +MIPLCREATE	
<msgid>	Integer type	
	Msg id, msg id carried in +MIPLREAD	
<result>	Integer type; Result of read command	
	1	205 content, read success
	11	400 bad request
	12	401 unauthorized
	13	404 not found
	14	405 method not allowed
	15	406 not acceptable
<objectid>	Integer type	

	object id
<instanceid>	Integer type
	Instance id
<resourceid>	Integer type
	Resource id
<valuetype>	Integer type
	1 string
	2 opaque
	3 Integer
	4 float
	5 bool
<len>	Integer type
	Value length
<value>	string type
	value
<index>	Integer type
	serial number, for one command may not be able to send the entire content. It's from N~0. 0 is last part of content.
<Flag>	Integer type
	1 First msg
	2 Middle msg
	0 Last msg
<raiflag>	Integer type
	0 (PS_SOCK_RAI_NO_INFO) disable RAI
	1 (PS_SOCK_RAI_NO_UL_DL_FOLLOWED) enable RAI，release after send
	2 (PS_SOCK_ONLY_DL_FOLLOWED) enable RAI，release after ACK has received

Example

```
AT+MIPLREADRSP=0,msgid,1,3306,0,5750,1,5,”hello”,0,0
OK
```

3.7.10 AT+MIPLWRITERSP

After receive +MIPLWRITE, This command return the write result to OneNET.

AT+MIPLWRITERSP	
Set Command	**Response**
AT+MIPLWRITERSP=<ref>,<msgid>,<result>[,raiflag]	OK

	If there is any error, response: +CIS ERROR: <err>
Test Command AT+MIPLWRITERSP=?	**Response** +MIPLWRITERSP: (list of supported<ref>),(range of supported<msgid>),(list of supported<result>),(list of supported<raiflag>) OK
Maximum Response Time	5s
Parameter Saving Mode	NO_SAVE

Parameter

<ref>	Integer type
	a specified OneNET communication instance id returned by +MIPLCREATE
<msgid>	Integer type
	Msg id, msg id carried in +MIPLWRITE
<result>	Integer type; Result of read command
	1 205 content
	2 204 changed, write success
	11 400 bad request
	12 401 unauthorized
	13 404 not found
	14 405 method not allowed
	15 406 not acceptable
<raiflag>	Integer type
	0 (PS_SOCK_RAI_NO_INFO) disable RAI
	1 (PS_SOCK_RAI_NO_UL_DL_FOLLOWED) enable RAI，release after send
	2 (PS_SOCK_ONLY_DL_FOLLOWED) enable RAI，release after ACK has received

Example

```
AT+MIPLWRITERSP=0,msgid,2
OK
```

3.7.11 AT+MIPLEXECUTERSP

After receive +MIPLEXECUTE, This command return the execute result to OneNET.

AT+MIPLEXECUTERSP	
Set Command AT+MIPLEXECUTERSP=<ref>,<msgid>,<result>[,raiflag]	**Response** OK **If there is any error, response:** +CIS ERROR: <err>
Test Command AT+MIPLEXECUTERSP=?	**Response** +MIPLEXECUTERSP: (list of supported<ref>),(range of supported<msgid>),(list of supported<result>),(list of supported<raiflag>) OK
Maximum Response Time	5s
Parameter Saving Mode	NO_SAVE

Parameter

<ref>	Integer type	
	a specified OneNET communication instance id returned by +MIPLCREATE	
<msgid>	Integer type	
	Msg id, msg id carried in +MIPLEXECUTE	
<result>	Integer type; Result of read command	
	1	205 content
	2	204 changed, execute success
	11	400 bad request
	12	401 unauthorized
	13	404 not found
	14	405 method not allowed
	15	406 not acceptable
<raiflag>	Integer type	
	0 (PS_SOCK_RAI_NO_INFO) disable RAI	
	1 (PS_SOCK_RAI_NO_UL_DL_FOLLOWED) enable RAI，release after send	
	2 (PS_SOCK_ONLY_DL_FOLLOWED) enable RAI，release after ACK has received	

Example

```
AT+MIPLEXECUTERSP=0,msgid,2
OK
```

3.7.12 AT+MIPLOBSERVERSP

After receive +MIPLOBSERVE, This command return the observe result to OneNET. It also can response the cancel observe.

AT+MIPLOBSERVERSP	
Set Command AT+MIPLOBSERVERSP=<ref>,<msgid>,<result>	**Response** OK **If there is any error, response:** +CIS ERROR: <err>
Test Command AT+MIPLOBSERVERSP=?	**Response** +MIPLOBSERVERSP: (list of supported<ref>),(range of supported<msgid>),(list of supported<result>),(range of supported<raiflag>) OK
Maximum Response Time	5s
Parameter Saving Mode	NO_SAVE

Parameter

<ref>	Integer type
	a specified OneNET communication instance id returned by +MIPLCREATE
<msgid>	Integer type
	Msg id, msg id carried in +MIPLOBSERVE
<result>	Integer type; Result of read command
	1 205 content
	2 204 changed, execute success
	11 400 bad request
	12 401 unauthorized
	13 404 not found
	14 405 method not allowed
	15 406 not acceptable
<raiflag>	Integer type
	0 (PS_SOCK_RAI_NO_INFO) disable RAI
	1 (PS_SOCK_RAI_NO_UL_DL_FOLLOWED) enable RAI，release after send
	2 (PS_SOCK_ONLY_DL_FOLLOWED) enable RAI，release after ACK has received

Example

```
AT+MIPLOBSERVERSP=0,msgid,1
OK
```

3.7.13 AT+MIPLDISCOVERRSP

After receive +MIPLDISCOVER, This command return all resources of object id specified in +MIPLDISCOVER.

AT+MIPLDISCOVERRSP	
Set Command AT+MIPLDISCOVERRSP=<ref>,<msgid>,<result>,<length>,<valuestring>[,<raiflag>]	**Response** OK **If there is any error, response:** +CIS ERROR: <err>
Test Command AT+MIPLDISCOVERRSP=?	**Response** +MIPLDISCOVERRSP: (list of supported<ref>),(range of supported<msgid>),(list of supported<result>),(range of supported<length>),\"<valuestring>\",(list of supported<raiflag>) OK
Maximum Response Time	5s
Parameter Saving Mode	NO_SAVE

Parameter

<ref>	Integer type	
	a specified OneNET communication instance id returned by +MIPLCREATE	
<msgid>	Integer type	
	Msg id, msg id carried in +MIPLDISCOVER	
<result>	Integer type; Result of read command	
	1	205 content, discover success
	2	204 changed
	11	400 bad request
	12	401 unauthorized
	13	404 not found
	14	405 method not allowed
	15	406 not acceptable

<len>	Integer type
	Value length
<valuestring>	string type
	Object's resource id, multiple resources separated by ";"
<raiflag>	Integer type
	0 (PS_SOCK_RAI_NO_INFO) disable RAI
	1 (PS_SOCK_RAI_NO_UL_DL_FOLLOWED) enable RAI, release after send
	2 (PS_SOCK_ONLY_DL_FOLLOWED) enable RAI, release after ACK has received

Example

```
AT+MIPLDISCOVERRSP=0,msgid,1,14,"5750;5850;5851"
OK
```

3.7.14 AT+MIPLPARAMETERRSP

After receive +MIPLPARAMETER, This command return the execute result to OneNET.

AT+MIPLPARAMETERRSP	
Set Command AT+MIPLPARAMETERRSP=<ref>,<msgid>,<result>[,<raiflag>]	**Response** OK **If there is any error, response:** +CIS ERROR: <err>
Test Command AT+MIPLPARAMETERRSP=?	**Response** +MIPLPARAMETERRSP: (list of supported<ref>),(range of supported<msgid>),(list of supported<result>),(list of supported<raiflag>) OK
Maximum Response Time	5s
Parameter Saving Mode	NO_SAVE

Parameter

<ref>	Integer type
	a specified OneNET communication instance id returned by +MIPLCREATE
<msgid>	Integer type
	Msg id, msg id carried in +MIPLPARAMETER
<result>	Integer type; Result of read command
	1 205 content

	2	204 changed, execute success
	11	400 bad request
	12	401 unauthorized
	13	404 not found
	14	405 method not allowed
	15	406 not acceptable
<raiflag>	Integer type	
	0 (PS_SOCK_RAI_NO_INFO) disable RAI	
	1 (PS_SOCK_RAI_NO_UL_DL_FOLLOWED) enable RAI，release after send	
	2 (PS_SOCK_ONLY_DL_FOLLOWED) enable RAI，release after ACK has received	

Example

```
AT+MIPLPARAMETERRSP=0,msgid,2
OK
```

3.7.15 AT+MIPLUPDATE

This command updates the register information,such as lifetime

AT+MIPLUPDATE	
Set Command AT+MIPLUPDATE=<ref>,<lifetime>,<withobjectflag>[,<raiflag>]	**Response** OK **If there is any error, response:** +CIS ERROR: <err>
Test Command AT+MIPLUPDATE=?	**Response** +MIPLUPDATE: (list of supported<ref>),(range of supported<lifetime>),(list of supported<withobjectflag>),(list of supported<raiflag>) OK
Maximum Response Time	5s
Parameter Saving Mode	NO_SAVE

Parameter

<ref>	Integer type
	a specified OneNET communication instance id returned by +MIPLCREATE
<lifetime>	Integer type

	Update lifetime value in second of the client	
<withobjectflag>	Integer type;	
	1	Update object info at the same time
	0	Don't update object info

Example

```
AT+MIPLUPDATE=0,3600,0
OK
```

3.7.16 AT+MIPLVER

This command return the version

AT+MIPLUPDATE	
Read Command AT+MIPLVER?	**Response** +MIPLVER: <ver> OK
Maximum Response Time	5s
Parameter Saving Mode	NO_SAVE

Parameter

<ver>	string type	
	Version of OneNET protocol	

Example

```
AT+MIPLVER?
+MIPLVER: 2.2.0
OK
```

3.7.17 +MIPLREAD

This is an unsolicited message to represent remote reading command. OneNET request device to upload specified resource value.

+MIPLREAD

```
+MIPLREAD: <ref>,<msgid>,<objectid>,<instanceid>,<resourceid>
```

Parameter

<ref>	Integer type
	a specified OneNET communication instance id returned by +MIPLCREATE
<msgid>	Integer type
	Message id of this message
<objectid>	Integer type
	object id
<instanceid>	Integer type
	Instance id, -1 means read all the instance belong to the object
<resourceid>	Integer type
	Resource id, -1 means read all the resources belong to the instance

Example

3.7.18 +MIPLWRITE

This is an unsolicited message to represent remote writing command. OneNET request device to modify specified resource value.

+MIPLWRITE

```
+MIPLWRITE:
<ref>,<msgid>,<objectid>,<instanceid>,<resourceid>,<valuetype>,<len>,
<value>,<flag>,<index>
```

Parameter

<ref>	Integer type
	a specified OneNET communication instance id returned by +MIPLCREATE
<msgid>	Integer type
	Message id of this message
<objectid>	Integer type
	object id
<instanceid>	Integer type
	Instance id
<resourceid>	Integer type

	Resource id	
<valuetype>	Integer type	
	1	string
	2	opaque
	3	Integer
	4	float
	5	bool
<len>	Integer type	
	Value length	
<value>	string type	
	value	
<Flag>	Integer type	
	1	First msg
	2	Middle msg
	0	Last msg
<index>	Integer type	
	serial number, for one command may not be able to send the entire content. It's from N~0. 0 is last part of content.	

Example

3.7.19 +MIPLEXECUTE

This is an unsolicited message to represent remote executing command. OneNET request device to execute some predefined operations on specified resource.

+MIPLEXECUTE

```
+MIPLEXECUTE:
<ref>,<msgid>,<objectid>,<instanceid>,<resourceid>,<len>,<arguments>
```

Parameter

<ref>	Integer type
	a specified OneNET communication instance id returned by +MIPLCREATE
<msgid>	Integer type
	Message id of this message
<objectid>	Integer type
	object id
<instanceid>	Integer type
	Instance id
<resourceid>	Integer type

	Resource id
<len>	Integer type
	Value length
<arguments>	string type
	Represent the operation

Example

3.7.20 +MIPLOBSERVE

This is an unsolicited message to represent remote observe command. OneNET request device to upload the values of specified resource or all resources of specified instance when they change.

+MIPLOBSERVE

```
+MIPLOBSERVE: <ref>,<msgid>,<flag>,<objectid>,<instanceid>,<resourceid>
```

Parameter

<ref>	Integer type
	a specified OneNET communication instance id returned by +MIPLCREATE
<msgid>	Integer type
	Message id of this message
<Flag>	Integer type
	1 Observe add
	0 Observe cancel
<objectid>	Integer type
	object id,
<instanceid>	Integer type
	Instance id, -1 means observing all the instances and resources belong to the object
<resourceid>	Integer type
	Resource id, -1 means observing all the resources belong to the instance

Example

3.7.20 +MIPLDISCOVER

This is an unsolicited message to represent remote discover command. OneNET request device to upload the resource id list of specified object id.

+MIPLDISCOVER

```
+MIPLDISCOVER: <ref>,<msgid>,<objectid>
```

Parameter

<ref>	Integer type
	a specified OneNET communication instance id returned by +MIPLCREATE
<msgid>	Integer type
	Message id of this message
<objectid>	Integer type
	object id

Example

3.7.21 +MIPLPARAMETER

This is an unsolicited message to represent remote parameter command. with this message, OneNET can set interval and threshold value of specified resource.

+MIPLPARAMETER

```
+MIPLPARAMETER: <ref>,<msgid>,<objectid>,<instanceid>,<resourceid>,<len>,
<parameter>
```

Parameter

<ref>	Integer type
	a specified OneNET communication instance id returned by +MIPLCREATE
<msgid>	Integer type
	Message id of this message
<objectid>	Integer type
	object id
<instanceid>	Integer type
	Instance id , -1 means observing all the instances and resources belong to the object

<resourceid>	Integer type
	Resource id, -1 means observing all the resources belong to the instance
<len>	Integer type
	Value length
<arguments>	string type
	Strategy parameter, like pmin=xxx;pmax=xxx;lt=xxx;gt=xx;st=xxx

Example

3.7.22 +MIPLEVENT

This is an unsolicited message to report status event.

+MIPLEVENT

```
+MIPLEVENT: <ref>,<evtid>[,<ackid>]
```

Parameter

<ref>	Integer type	
	a specified OneNET communication instance id returned by +MIPLCREATE	
<eventid>	Integer type	
	1	Bootstrap start
	2	Bootstrap success
	3	Bootstrap failed
	4	Connect success
	5	Connect failed
	6	Registration success
	7	Registration failed
	8	Registration timeout
	9	Lifetime timeout
	10	Status halt
	11	Update success
	12	Update failed
	13	Update timeout
	14	Update need
	15	Un-registration done
	20	Response failed
	21	Response success
	25	Notify failed
	26	Notify success

<ackid>	Integer type
	ackid carrying by notify success

Example

```
```

3.7.23 Summary of <err> Codes

<err> Codes	Description
100	unknown error
601	parameter error
602	status error
651	not support
652	sdk error
653	no instance
654	malloc fail
655	network not ready
656	configuration error

3.8 OceanConnect Extension Commands

3.8.1 AT+CTM2MVER

This command sends request to get CTM2M version.

AT+CTM2MVER	
Read Command AT+CTM2MVER?	**Response** +CTM2MVER: <lwm2m>,<ctm2m>,<ctmt>,<ctmv> OK **If there is any error, response:** +CTM2M ERROR: <err>
Maximum Response Time	5s
Parameter Saving Mode	NO_SAVE

Parameter

<lwm2m>	String type
	Version of LWM2M
<ctm2m>	String type
	Version of CTM2M
<ctmt>	String type
	Chip type: EC616 etc.
<ctmv>	String type
	Software version of module

Example

```
AT+CTM2MVER?
+CTM2MVER: 1.0,1.1.0,EC616,V1.0
OK
```

3.8.2 AT+CTM2MSETMOD

This command sends command to set UE working mode.

AT+CTM2MSETMOD	
Set Command AT+CTM2MSETMOD=<MOD_ID>,<MOD_DATA>	**Response** OK **If there is any error, response:**

	+CTM2M ERROR: <err>
Read Command AT+CTM2MSETMOD?	**Response** +CTM2MSETMOD: <idAuthMode>, <autoUpdate>, <onUQMode>, <level2Policy>,<autoHeartbeat>, <wakeupPolicy>, <protocolMode> OK
Test Command AT+CTM2MSETMOD=?	**Response** +CTM2MSETMOD:(list of supported <MOD_ID>s), (list of supported < MOD_DATA>s) OK
Maximum Response Time	5s
Parameter Saving Mode	SAVE

Parameter

<MOD_ID>	Integer type
	1---IDAuth_Mode 2--- Auto_TAUTimer_Update 3--- ON_UQMode 4--- ON_CELevel2Policy 5--- Auto_Heartbeat 6--- Wakeup_Notify 7--- Protocol_Mode
<MOD_DATA>	Integer type
	If MOD_ID=1: 1: default value, no authentication string 2: SIMD authentication string from outside of module 3: SM9 authentication string from outside of module 4: SIMD authentication string from inside of module 5: SM9 authentication string from inside of module If MOD_ID=2: 1: default no action 2: notify MCU 3: not notify MCU, auto update inside of module If MOD_ID=3: 1: UQ mode off 2: UQ mode on If MOD_ID=4: 1: default send under CE level2

	2: not send under CE level2 If MOD_ID=5: 1: no auto heartbeat 2: default auto heartbeat If MOD_ID=6: 1: not notify to MCU 2: default notify to MCU If MOD_ID=7: 1: normal 2: enhance

Example

```
AT+CTM2MSETMOD=1,1
OK
```

3.8.3 AT+CTM2MSETPM

This command sends command to set the parameters to register to the China Telecom iot platform.

AT+CTM2MSETPM	
Set Command AT+CTM2MSETPM =<Sever_IP>,<Port>,<Lifetime>[,<Object_Instance_List>]	**Response** OK **If there is any error, response:** +CTM2M ERROR: <err>
Read Command AT+CTM2MSETPM?	**Response** +CTM2MSETPM: <Sever_IP>,<Port>,<Lifetime>[,<Object_Instance_List>] OK
Test Command AT+CTM2MSETPM =?	**Response** +CTM2MSETPM: (list of supported <Sever_IP>), (range of supported<Port>),(range of supported<Lifetime>),(list of supported <Object_Instance_List>) OK
Maximum Response Time	5s
Parameter Saving Mode	SAVE

Parameter

<Sever_IP>	String type
	IP address for LWM2M server.
<Port>	Integer type
	Port number for LWM2M server.
<Lifetime>	Integer type
	Lifetime for LWM2M server and unit is second with minimum value 300
<Object_Instance_List>	String type
	Object&instance list supported by MCU, with format like "</3303/0>,</3303/1>"

Example

```
AT+CTM2MSETPM=180.101.147.115,5683,86400,"<3303/0>,<3303/1>"
OK
```

3.8.4 AT+CTM2MREG

This command sends command to register to the China Telecom iot platform.

AT+CTM2MREG	
Execution Command AT+CTM2MREG	**Response** OK **If there is any error, response:** +CTM2M ERROR: <err>
Read Command AT+CTM2MREG?	**Response** +CTM2MREG:<status> OK
Maximum Response Time	5s
Parameter Saving Mode	NO_SAVE

Example

```
AT+CTM2MREG
OK
```

3.8.4 AT+CTM2MUPDATE

This command sends command to update binding mode to China Telecom iot platform.

AT+CTM2MUPDATE	
Set Command AT+CTM2MUPDATE[=<Binding_Mode>[,<Object List>]]	**Response** +CTM2MUPDATE:<msgID> OK **If there is any error, response:** +CTM2M ERROR: <err>
Test Command AT+CTM2MUPDATE=?	**Response** +CTM2MUPDATE: (list of supported <Binding_Mode>), (list of supported <ObjectList>) OK
Maximum Response Time	5s
Parameter Saving Mode	NO_SAVE

Parameter

<Binding_Mode>	Integer type
	0---Not set 1---UQ mode 2---U mode
<ObjectList>	String type
	NULL---Not set

Example

```
AT+CTM2MUPDATE
OK
+CTM2M: update,0,298
```

3.8.5 AT+CTM2MDEREG

This command sends command to exit China Telecom iot platform.

AT+CTM2MDEREG	
Execution Command AT+CTM2MDEREG	**Response** OK +CTM2M: dereg,0 **If there is any error, response:** +CTM2M ERROR: <err>
Maximum Response Time	5s
Parameter Saving Mode	NO_SAVE

Example

```
AT+CTM2MDEREG
OK
+CTM2M: dereg,0
```

3.8.6 AT+CTM2MSEND

This command sends command to send business data to China Telecom iot platform.

AT+CTM2MSEND	
Set Command AT+CTM2MSEND=<data>[,<mode>]	**Response** +CTM2MSEND: <msgID> OK **If there is any error, response:** +CTM2M ERROR: <err>
Test Command AT+CTM2MSEND=?	**Response** +CTM2MSEND: (list of supported <data>), (list of supported <mode>) OK
Maximum Response Time	5s
Parameter Saving Mode	NO_SAVE

Parameter

<data>	String type
	Length should be less than 1024
<mode>	Integer type
	0---CON mode
	1---NON mode
	2---NON with RAI flag
	3---CON with RAI flag

Example

```
AT+CTM2MSEND=0131323334
+CTM2MSEND: 554
OK

+CTM2M: send,31,554
```

3.8.7 AT+CTM2MCMDRSP

This command sends response to China Telecom iot platform.

AT+CTM2MCMDRSP	
Set Command AT+CTM2MCMDRSP =<Msgid>,<Token>,<Rspcode>, <Uri_str >,<Observe>[,<Dataformat>,<Data>]	**Response** OK **If there is any error, response:** +CTM2M ERROR: <err>
Test Command AT+CTM2MCMDRSP=?	**Response** +CTM2MCMDRSP: (list of supported <Msgid>), (list of supported <Token>),(list of supported <Rspcode>), (list of supported <Uri_str>),(list of supported <Observe>), (list of supported <Dataformat>),(list of supported <Data>) OK
Maximum Response Time	5s
Parameter Saving Mode	NO_SAVE

Parameter

<Msgid>	Integer type
	It should be conform to the msgid of received message which needs to be replied.
<Token>	String type
	It should be conform to the token value of received message which needs to be replied.
<Rspcode>	Integer type
	Message response code. Something like 204, 205 etc.
<Uri_str >	String type
	/object id/instance id/resource id
<Observe>	Integer type
	0--- set observe and no following parameters
	1---cancel observe and no following parameters
	8---just for response case
	9---there are <Dataformat>,<data> parameters following
<Dataformat>	Integer type
	1--- TLV format(application/vnd.oma.lwm2m+tlv)
	2---opaque format(application/octet-stream)
	7---TEXT format(text/plain)
	8---JSON format(application/vnd.oma.lwm2m+json)
	9---Core Link Param format(application/link-format)
<Data>	String type
	Length should be less than 1024

Example

```
AT+CTM2MCMDRSP=51209,962AB03A,205,/3/0,0
OK
+CTM2M: send,31,51209
```

3.8.8 +CTM2MRECV

This is an unsolicited message to represent received data message from China Telecom iot platform.

+CTM2MRECV

```
+CTM2MRECV: <len>,<data>
```

Parameter

<len>	Integer type
	datalen
<data>	String type

	Received data from platform

Example

```
+CTM2MRECV: 10,651A320502DA0034710B
```

3.8.9 +CTM2M

This is an unsolicited message to represent received async notification from China Telecom iot platform.

+CTM2M

```
+CTM2M: <operation>,<status code>[,<data1>,<data2>,<data3>]
```

Parameter

<operation>	String type
	Can be one type of below operations: reg/obsrv/update/ping/dereg/send/lwstatus
<status code>	Integer type
	0---success 1---timeout 2---not send out packet 9---receive platform RST packet and mean can't send UL to platform 10---parameter error 11---other errors 13---authentication error 14---UE not login 22---iot protocol or lwm2m version mismatch 24---lwm2m session invalid 25---session load failure when quitting from sleep or after reboot 26---Engine abnormal, need reboot by MCU 28---TAU is due 31---packet is already sent out 32---object 19 not exist
<data1,2,3>	Integer type
	data1---msgID data2---N/A data3---N/A

Example

```
+CTM2M: lwstatus,29
```

3.8.10 +CTM2MCMD

This is a notification message from China Telecom iot platform to notify UE that some object operation is triggered and need to be replied.

+CTM2MCMD
`+CTM2MCMD:<msgid>,<cmdtype>,<token>,<uri_str>[,<observe>(,<dataformat>,<data>)]`

Parameter

<msgid>	Integer type
	Message ID
<cmdtype>	Integer type
	0---Read 1---Observe 2---Write 3---Write-Partial 4---Write-Attribute 5---Discover 6---Execute 7---Create 8---Delete
<token>	Hex String type
	Async message response ID
<uri_str>	String type
	/objectid/instanceid/resourceid
<observe>	Integer type
	It is must when cmdtype=1 0---Set 1---Cancel
<dataformat>	Integer type
	1---TLV 6---CoAP Param 7---text/plain 8---JSON
<data>	String type
	It is must when cmdtype=2、3、4、6、7 It is null when execute cmd has no arguments

Example

```
+CTM2MCMD: 3423,1,4AA33F97,/3/0,0
```

3.9 Socket Commands(Solution B)

3.9.1 AT+ECSOCR

This command creates a socket on the UE and associates with specified protocol. If the port is set, receiving is enabled and "+ECSONMI" unsolicited messages will appear for any message that is received on that port.

AT+ECSOCR	
Set Command AT+ECSOCR=<type>,< protocol>,<listen_port>[,<recevice_control>[,<af_type>[,<ip_address>]]]	**Response** <socket_id> OK **If there is any error, response:** +CME ERROR: <err>
Test Command AT+ECSOCR=?	**Response** +ECSOCR: (list of supported <**type**>), (list of supported < **protocol**>),(list of supported < **listen_port**>) , (list of supported < **recevice_control** >), (list of supported < **af_type**>), (list of supported < **ip_address**>) OK
Maximum Response Time	5s
Parameter Saving Mode	SAVE

Parameter

<type>	String type	
	DGRAM	UDP
	STREAM	TCP
<protocol>	Integer type	
	Standard internet protocol definition. UDP is 17, and TCP is 6.	
<listen_port>	Integer type	
	A number in the range of 0-65535 except 5683, 5684, 56830, 56831 and 56833. This is the local port that will be included in sent messages and on which messages will be received. If it is 0 or omitted, the module will assign a random <listen_port> for this socket.	

<recevice_control>	Integer type; standard internet protocol definition	
	0	the incoming messages will be ignored
	1	the default value,the incoming messages will be received
<af_type>	String type	
	AF_INET	the default value IPV4
	AF_INET6	IPV6
<ip_address>	String type	
	IP address.The IP address of the network assigned to UE.	
<socket_id>	Integer type	
	1-7 It is an integer greater than 1.A maximum of 5 sockets are supported ,but other serviced may reduce this number.	

Example

```
AT+ECSOCR=DGRAM,17,1,1
1
OK
```

3.9.2 AT+ECSOST

Send a UDP datagram containing length bytes of data to <remote_port> on <remote_addr> .

The command sends a UDP datagram containing length bytes of data to the specified host and port. It will return with the socket that it was sent on, and the number of bytes of data sent. If the amount of data is larger than the largest datagram that can be sent, return value of AT+ECSOST will indicate how much of the data was successfully sent.

AT+ECSOST	
Set Command AT+ECSOST=<socket_id>, remote_addr>,<remote_port>,<length>,<data>, [<sequence>[,<segment_id>[,<segment_num>]]]	**Response** <socket_id>,<length> OK **If there is any error, response:** +CME ERROR: <err>
Test Command AT+ECSOST=?	**Response** +ECSOCR: (list of supported <**socket_id**>), (list of supported <**remote_addr**>),(list of supported <**remote_port**>), (list of supported <**length**>), <**data**>, (list of supported <**sequence**>), (list of supported

	<**segment_id**>), (list of supported <**segment_num**>) OK
Maximum Response Time	5s
Parameter Saving Mode	SAVE

Parameter

<socket_id>	Integer type
	1-7 socket_id returned by AT+ECSOCR
<remote_addr>	String type
	Remote IP address
<remote_port>	Integer type
	0-65535 This is the remote port on which messages will be received
<length>	Integer type
	1-950 Decimal length of data to be sent
<data>	String type
	data be sent in hex string format
<sequence>	Integer type
	1-255 if it is omitted,data sent will not to be reported.if not omitted, when datagram is sent over RF or is discarded,then the result will be reported:+ECSOSTR:<socket_id>,<sequence>,<status>
<segment_id>	Integer type
	1-4 One segment index of a segment message.
<segment_num>	Integer type
	2-4 The total number which the messages will fragment.Range:1-4

Example

```
AT+ECSOST=1,47.105.44.99,1002,3,333132
1,3
OK
```

3.9.3 AT+ECSOSTF

Send a UDP datagram containing length bytes of data to <remote_port> on <remote_addr> and allows meta-data flags to be sent.

The command sends a UDP datagram containing length bytes of data to the specified host:port. It will return with the socket that it was sent on, and the number of bytes of data sent. If the amount of data is larger than the largest datagram that can be sent, return value of AT+ECSOSTF will indicate how much of the data was successfully sent.

AT+ECSOSTF

Set Command	Response
AT+ECSOSTF=<socket_id>, remote_addr>,<remote_port>,<flag>,<length>, <data>,<sequence>[,<segment_id>[,<segment_num>]]	<socket_id>,<length> OK **If there is any error, response:** +CME ERROR: <err>
Test Command AT+ECSOSTF=?	**Response** +ECSOCR: (list of supported <**socket_id**>), (list of supported <**remote_addr**>),(list of supported <**remote_port**>), (list of supported <**flag**>), (list of supported <**length**>), <**data**>, (list of supported <**sequence**>), (list of supported <**segment_id**>), (list of supported <**segment_num**>) OK
Maximum Response Time	5s
Parameter Saving Mode	SAVE

Parameter

<socket_id>	Integer type
	1-7 socket_id returned by AT+ECSOCR
<remote_addr>	String type
	Remote IP address
<remote_port>	Integer type
	0-65535 This is the remote port on which messages will be received
<flag>	Integer type
	Specifies the type of message transmission. Values of this argument are in hex format and are formed by logically OR'ing zero or more of the following flags: 0x100 Exception Message: Send messages with high priority 0x200 Release Indicator: indicate release after next message 0x400 Release Indicator: indicate release after next messages has been replied to If no flags are set, a value of 0 should be provided
<length>	Integer type

	1-950	Decimal length of data to be sent
<data>	String type data be sent in hex string format	
<sequence>	Integer type	
	1-255	if it is omitted,data sent sill not to be reported.if not omitted, when datagram is sent over RF or is discarded,then the result will be reported:+ECSOSTR:<socket_id>,<sequence>,<status>
<segment_id>	Integer type	
	1-4	One segment index of a segment message.
<segment_num>	Integer type	
	2-4	The total number which the messages will fragment.Range:1-4

Example

```
AT+ECSOSTF=1,47.105.44.99,1002,0x100,3,333132
1,3
OK
```

3.9.4 AT+ECQSOS

The command queries the list of the pending upstream message.

AT+ECQSOS	
Set Command AT+ECQSOS=<socket_id>[,<socket_id>[,<socket_id>[…]]]	**Response** [+ECQSOS:<socket_id>,<sequence>] OK **If there is any error, response:** +CME ERROR: <err>
Test Command AT+ECQSOS=?	**Response** +ECQSOS:(list of supported <**socket_id**>) OK
Maximum Response Time	5s
Parameter Saving Mode	SAVE

Parameter

<socket_id>	Integer type	
	1-7	socket_id returned by AT+ECSOCR

<sequence>	Integer type
	1-255 if it is omitted,data sent will not to be reported.if not omitted, when datagram is sent over RF or is discarded,then the result will be reported:+ECSOSTR:<socket_id>,<sequence>,<status>

Example

```
AT+ECQSOS=1
+ECQSOS:1,3
OK
```

3.9.5 AT+ECSORF

The command can read up to <req_length> characters of data from <socket>, and returned length is the actual number of characters returned.

The command is use to receive data on a socket. When data arrives, a "+ECSONMI" response will be generated to indicate the socket the message was received on and also the amount of data. The AT+ECSORF command takes a length, which is the maximum amount of data that will be returned.
If the requested length is larger than the actual size of the returned data, only the length of returned data is provided, and the remaining length is returned as 0. If the requested length is less than the amount of data returned, only the requested amount of data will be returned, plus an indication of the number of bytes remaining. Once a message has been fully read, a new "+ECSONMI" notification will be sent if there is another message to process.
If messages arrive faster than they are read, and the internal message buffer is full, the most recent message will be discarded.

AT+ECSORF	
Set Command AT+ECSORF=<socket_id>,<req_length>	**Response** <socket_id>,<ip_addr>,<port>,<length>,<data>,<remaining_length> OK **If there is any error, response:** +CME ERROR: <err>
Test Command AT+ECSORF=?	Response +ECSORF:(list of supported <**socket_id**>),(list of supported <**req_length**>) OK
Maximum Response Time	5s
Parameter Saving Mode	SAVE

Parameter

<socket_id>	Integer type
	1-7 socket_id returned by AT+ECSOCR
<ip_addr>	String type
	Remote IP address
<port>	Integer type
	0-65535 This is the remote port on which messages will be sent from
<req_length>	Integer type
	1-1357 Decimal length of data which want to read
<length>	Integer type
	1-1357 Decimal length of data to be read
<data>	Integer type
	data be sent in hex string format
<remaining_length>	Integer type
	1-1357 Amount of data left to read for this messages as a decimal byte length.Remaining length is always 0;The remaining data is readable.

Example

```
AT+ECSORF=1,4
1,”47.105.44.99”,1010,4,”31323334”,0
OK
```

3.9.6 AT+ECSOCO

The command connect a TCP server to the specified host and port.

AT+ECSOCO	
Set Command AT+ECSOCO=<socket_id>,<remote_addr>,<remote_port>	**Response** OK If there is any error, response: +CME ERROR: <err>
Test Command AT+ECSOCO=?	**Response** +ECSOCO:(list of supported <**socket_id**>),(list of supported <**remote_addr**>),(list of supported <**remote_port**>)

	OK
Maximum Response Time	30s
Parameter Saving Mode	SAVE

Parameter

<socket_id>	Integer type	
	1-7	socket_id returned by AT+ECSOCR
<remote_addr>	String type	
	Remote IP address	
<remote_port>	Integer type	
	0-65535	This is the remote port to be connected to

Example

```
AT+ECSOCO=1,”47.105.44.99”,1010
OK
```

3.9.7 AT+ECSOSD

The command sends a TCP datagram to the TCP server. It will return with the socket that it was sent on, and the number of bytes of data sent. If the amount of data is larger than the largest datagram that can be sent, then AT+ECSOSD return value will indicate how much the data was successfully sent.
The If <sequence> is not omitted, when datagram is asked for by the server or is discarded by UE, the result will be reported.

AT+ECSOSD	
Set Command AT+ECSOSD=<socket_id>,<length>,<data>[,<flag>[,<sequence>]]	**Response** <socket_id>,<length> OK **If there is any error, response:** +CME ERROR: <err>
Test Command AT+ECSOSD=?	**Response** +ECSOSD:(list of supported <**socket_id**>),(list of supported <**length**>),<**data**>,(list of supported <**flag**>),(list of supported <**sequence**>)

	OK
Maximum Response Time	5s
Parameter Saving Mode	SAVE

Parameter

<socket_id>	Integer type
	1-7 socket_id returned by AT+ECSOCR
<length>	Integer type
	1-950 Decimal length of data to be sent
<data>	Integer type
	data be sent in hex string format
<flag>	Integer type
	Specifies the type of message transmission. Values of this argument are in hex format and are formed by logically OR'ing zero or more of the following flags: 0x100 Exception Message: Send messages with high priority 0x200 Release Indicator: indicate release after next message 0x400 Release Indicator: indicate release after next messages has been replied to If no flags are set, a value of 0 should be provided
<sequence>	Integer type
	1-255 if it is omitted,data sent sill not to be reported.if not omitted, when datagram is sent over RF or is discarded,then the result will be reported:+ECSOSTR:<socket_id>,<sequence>,<status>

Example

```
AT+ECSOSD=1,2,”3132”
1,2
OK
```

3.9.8 AT+ECSOCL

The command sends a TCP datagram to the TCP server. It will return with the socket that it was sent on, and the number of bytes of data sent. If the amount of data is larger than the largest datagram that can be sent, then AT+ECSOSD return value will indicate how much the data was successfully sent.
The If <sequence> is not omitted, when datagram is asked for by the server or is discarded by UE, the result will be reported.

AT+ECSOSD	
Set Command	**Response**

AT+ECSOCL=<socket_id>	OK **If there is any error, response:** +CME ERROR: <err>
Test Command AT+ECSOCL=?	**Response** +ECSOCL:(list of supported <**socket_id**>) OK
Maximum Response Time	5s
Parameter Saving Mode	SAVE

Parameter

<socket_id>	Integer type	
	1-7	socket_id returned by AT+ECSOCR

Example

```
AT+ECSOCL=1
OK
```

3.9.9 AT+ECSONMI

The write command is used to set the unsolicited result code "+ECSONMI" to indicate arrived socket messages(the socket were not configure as private socket by AT+ECSONMIE command):

If <mode>=1, the UE will receive an unsolicited result code:
"+ECSONMI:<socket_id>,<length>".
If <mode>=2, the UE will receive an unsolicited result code:
"+ECSONMI:<socket_id>,<remote_addr>,<remote_port>,<length>,<data>".
If <mode>=3, the UE will receive an unsolicited result code:
"+ECSONMI: <socket_id>,<length>,<data>".

The read command returns the current setting of the command.

The write command is also used to set the public max downlink buffer size and the public max messages number

AT+ECSONMI	
Set Command AT+ECSONMI=<mode>[,<max_public_dl_buffer>[,<max_public_dl_pkg_num>]]	**Response** OK **If there is any error, response:**

	+CME ERROR: <err>
Test Command AT+ECSONMI=?	**Response** +ECSONMI:(list of supported <**mode**>),(list of supported <**max_public_dl_buffer**>),(list of supported <**max_public_dl_pkg_num**>) OK
Read Command AT+ECSONMI?	**Response** +ECSONMI: <**mode**>,<**max_public_dl_buffer**>,<**max_public_dl_pkg_num**> OK
Maximum Response Time	5s
Parameter Saving Mode	SAVE

Parameter

<mode>	Integer type	
	Control downlink data format	
	0	Disable indication messages unsolicited result code.
	1	Enable indication messages unsolicited result code: "+ECSONMI: <socket_id>,<length>"
	2	Enable indication messages unsolicited result code: "+ECSONMI: <socket_id>,<remote_addr>,<remote_port>,<length>,<data>"
	3	Enable indication messages unsolicited result code: "+ECSONMI: <socket_id>,<length>,<data>"
<socket_id>	Integer type	
	1-7	socket_id returned by AT+ECSOCR
<length>	Integer type	
	1-1358	Number of bytes of data in the first message
<remote_addr>	String type	
	Address of system sending the message.	
<remote_port>	Integer type	
	0-65535	This is the remote port on which messages will be received.
<data>	Integer type	
	Data received in hex string format. Maximum length of received data is 1358	
<max_public_dl_buffer>	Integer type	

	1358-3072 The maximum downlink buffer total size for all sockets created by AT+ECSOCR which is not configure as private mode socket with the command “AT+ECSONMIE”. The default value is 2048
<max_public_dl_pkg_num>	Integer type
	8-16 The maximum downlink buffer message total number for all sockets created by AT+ECSOCR which is not configure as private mode socket with the command “AT+ECSONMIE”. The default value is 8

Example

```
AT+ECSONMI=2,1500,9
OK
```

3.9.10 AT+ECSONMIE

The write command is used to set the unsolicited result code “+ECSONMI” to indicate arrived messages of a specified socket:

If <mode>=1, the UE will receive an unsolicited result code:
"+ECSONMI:<socket_id>,<length>".
If<mode>=2,the UE will receive an unsolicited result code:
"+ECSONMI: <socket_id>,<remote_ad dr>,<remote_port>,<length>,<data>".
If<mode>=3,the UE will receive an unsolicited result code:
"+ECSONMI: <socket_id>,<length>,<data>".

The read command returns the current setting of the command.

The write command is also used to set the max downlink buffer size and the max messages number for the specified socket

AT+ECSONMIE	
Set Command AT+ECSONMIE=<socket_id>,<mode>[,<max_public_dl_buffer>[,<max_public_dl_pkg_num>]]	**Response** OK **If there is any error, response:** +CME ERROR: <err>
Test Command AT+ECSONMIE=?	**Response** +ECSONMIE: (list of supported <**socket_id**>s), (list of supported <**mode**>),(list of

	supported <**max_public_dl_buffer**>),(list of supported <**max_public_dl_pkg_num**>) OK
Read Command AT+ECSONMIE?	**Response** [+ECSONMIE: <**socket_id**>,<**mode**>,<**max_public_dl_buffer**>,<**max_public_dl_pkg_num**>][…] OK
Maximum Response Time	5s
Parameter Saving Mode	SAVE

Parameter

<mode>	Integer type	
	Control downlink data format	
	0	Disable indication messages unsolicited result code.
	1	Enable indication messages unsolicited result code: "+ECSONMI: <socket_id>,<length>"
	2	Enable indication messages unsolicited result code: "+ECSONMI: <socket_id>,<remote_addr>,<remote_port>,<length>,<data>"
	3	Enable indication messages unsolicited result code: "+ECSONMI: <socket_id>,<length>,<data>"
<socket_id>	Integer type	
	1-7	socket_id returned by AT+ECSOCR
<length>	Integer type	
	1-1358	Number of bytes of data in the first message
<remote_addr>	String type	
	Address of system sending the message.	
<remote_port>	Integer type	
	0-65535	This is the remote port on which messages will be received.
<data>	Integer type	
	Data received in hex string format. Maximum length of received data is 1358	
<max_public_dl_buffer>	Integer type	
	1358-2048	The maximum downlink buffer size for the specified

	socket. The default value is 1358
<max_public_dl_pkg_num>	Integer type
	1-8 The maximum downlink buffer message number for the specified socket. The default value is 4

Example

```
AT+ECSONMI=2,1500,6
OK
```

3.9.11 +ECSOCLI

This is an unsolicited message to notify that a socket has been closed. It returns the socket number and error code

+ECSOCLI

```
+ECSOCLI: <socket_id>,<errno>
```

Parameter

<socket_id>	Integer type	
	1-7	socket_id returned by AT+ECSOCR
<errno>	Integer type	（Posix Errno defines）
	12	Out of memory error
	105	No buffer space available
	62	Timer expired
	113	No route to host
	115	Operation now in progress
	22	Invalid argument
	11	Operation would block
	107	Transport endpoint is not connected
	103	Software caused connection abort
	104	Connection reset by peer

Example

```
+ECSOCLI:1,104
```

3.9.12 +ECSOSTR

This is an unsolicited message to notify that one uplink datagram sent status with sequence

+ECSOSTR
+ECSOSTR: <socket_id>,<sequence>,<status>

Parameter

<socket_id>	Integer type	
	1-7	socket_id returned by AT+ECSOCR
<sequence>	Integer type	
	1-255	If it is omitted, data sent will not be reported. If not omitted, when datagram is sent over RF or is discarded, then the result will be reported
<status>	Integer type	
	0	the sent status of datagram is fail
	1	the sent status of datagram is success

Example

```
+ECSOSTR:1,101,1
```

3.9.13 Summary of <err> Codes(Socket solution B)

<err> Codes	Description
1	Parameter invalid
2	Too much socket instance
3	Create socket error
4	operation not supported
5	Cannot find the socket
6	Socket Connect fail
7	Socket bind fail
8	Send data fail
9	The socket status is not connected
10	The socket status is already connected
11	The socket status is invalid
12	The socket connect timeout
13	The socket close fail
14	The socket happen fatal error
15	Can not allocate more memory
16	SIM PUK2 required
17	No more DL buffer resource
18	The socket is connecting
19	UL sequence is invalid
20	Unknown error

4 Error Values

If the AT command not implemented or format dose not match, it will output “ERROR”.

For general control commands compliant with the 3Gpp specifications. Please refer to 3GPP TS 27007 V14.5.0, sub-clause 9.2 for all possible <err> values. If an error occurs, it will output “+CME ERROR: <err>”. Some common values are listed in the table below.

General Eorros(27.007)	
Code of <err>	Description
1	MT not connection
2	MT 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 call only
40	network personalisation PIN required
41	network personalisation PUK required
42	network subset personalisation PIN required
43	network subset personalisation PUK required

44	service provider personalisation PIN required
45	service provider personalisation PUK required
46	corporate personalisation PIN required
47	corporate personalisation PUK required
48	hidden key required
49	EAP method not support
50	incorrect Parameters
51	command implemented but currently disabled
52	command aborted by user
53	not attached to network due to MT functionality restrictions
54	modem not allowed - MT restricted to emergency calls only
55	operation not allowed because of MT functionality restrictions
56	fixed dial number only allowed - called number is not a fixed dial number
57	temporarily out of service due to other MT usage
58	language/alphabet not supported
59	unexpected data value
60	system failure
61	data missing
62	call barred
63	message waiting indication subscription failure
100	unknown
103	illegal MS
106	illegal ME
107	GPRS services not allowed
108	GPRS services and non GPRS services not allowed
111	PLMN not allowed
112	location area not allowed
113	roaming not allowed in this location area
114	GPRS services not allowed in this plmn
115	No suitable cells in location area
122	Congestion
126	Insufficient resources
127	Mission or unknown APN
128	Unknown PDP address or PDP type
129	User authentication failed
130	Active reject by GGSN services gw or PDN gw
131	Active reject unspecified
132	service option not supported
133	requested service option not subscribed
134	service option temporarily out of order
140	Feature not supported

141	Semantic errors in the TFT operation
142	Syntactical errors in the TFT operation
143	Unknown PDP context
144	Semantic errors in packet filters
145	Syntactical errors in packet filters
146	PDP context without TFT already activated
148	unspecified GPRS error
149	PDP authentication failure
150	invalid mobile class
171	Last PDN disconnection not allowed
172	Semantically incorrect message
173	Mandatory information element error
174	Information element non existent or not implemented
175	Conditional ie error
176	Protocol error unspecified
177	Operator determined barring
178	Max number of PDP contexts reached
179	Requested APN not supported in current rat and plmn combination
180	Request rejected bearer control mode violation
181	Unsupported oci value
182	User data transmission via control plane is congested
301	Internal error base
302	UE busy
303	Not power on
304	PDN not active
305	PDN not valid
306	PDN invalid type
307	PDN no parameter
308	UE fail

For general control commands compliant with 3GPP TS 27005. If an error occurs, it will output “+CMS ERROR: <err>”. Some common values are listed in the table below.

General Errors(27.005)	
Code of <err>	Description
300	ME failure
301	SMS service of ME reserved
302	operation not allowed
303	operation not supported
304	invalid PDU mode parameter
305	invalid text mode parameter
310	USIM not inserted
311	USIM PIN required

312	PH-(U)SIM PIN required
313	USIM failure
314	USIM busy
315	USIM wrong
316	USIM PUK required
317	USIM PIN2 required
318	USIM PUK2 required
320	memory failure
321	invalid memory index
322	memory full
330	SMSC address unknown
331	no network service
332	network timeout
340	no +CNMA acknowledgement expected
500	unknown error