nRF93M1 Cellular AT Commands

Command Reference Guide

v0.9

NORDIC® SEMICONDUCTOR

Contents

1 Introduction

This document describes the *AT command*s used to control the nRF93M1 and 3GPP Release 14 LTE Cat 1 bis modules. The AT command API is exposed on the nRF93M1 serial interfaces to provide an external MCU or computer access to the module API.

The AT commands described in this document apply to all versions of the nRF93M1 modules. If a command applies only to a specific version of the module, it is mentioned in the command description. The module version is printed on the module label.

2 AT command syntax

The AT Commands have standardized syntax rules.

Words enclosed in <angle brackets> are references to syntactical elements. Words enclosed in [square brackets] represent optional items which may be left out from the command line at the specified point. The brackets are not used when the words appear in the command line.

<CR>, <LF>, and terminating NUL are allowed in an AT command sent by an application, but are not mandatory when using an interface where both the command string and length of command string are provided.

All 3GPP standard AT commands for controlling the module feature begin with a plus sign (+), whereas Nordic-proprietary commands begin with a percent sign (%).

A string type parameter input should be enclosed between quotation marks ("").

For more information, see 3GPP 27.007 AT command set for User Equipment (UE) and 3GPP 27.005 Use of Data Terminal Equipment - Data Circuit terminating Equipment (DTE - DCE) interface for Short Message Service (SMS) and Cell Broadcast Service (CBS).

2.1 Set command <CMD>[=...]

Set commands set values or perform actions.

Example:

```
AT+CMD=1
```

where

- AT is the command line prefix
- + is the prefix for extended commands
- CMD is the body of a basic command
- 1 is a subparameter (multiple subparameters are separated by commas)

2.2 Read command <CMD>?

Read commands check the current values of subparameters.

Example:

```
AT+CMD?
```

where

- AT is the command line prefix
- + is the prefix for extended commands
- CMD is the body of a basic command
- ? represents a read command

2.3 Test command <CMD>=?

Test commands test the existence of the command and provide information about the type of its subparameters. Some test commands have also other functionality, which is described in the command-specific chapters.

Example:

```
AT+CMD=?
```

where

- AT is the command line prefix
- + is the prefix for extended commands
- CMD is the body of a basic command
- =? represents a test command for checking possible subparameter values

2.4 Response

AT responds to all commands with a final response.

The response is one of the following:

```
OK<CR><LF>
ERROR<CR><LF>
+CME ERROR: <cause_value><CR><LF>
+CMS ERROR: <cause_value><CR><LF>
```

"CMS ERROR:" is used as an error response for SMS related commands specified in *3GPP 27.005*.

Some commands may also produce a varying number of information response lines before the final response. An information response can be received only when a command-specific response syntax is specified. An information response line usually starts with a prefix, which is the command entered:

```
+CMD: [...]<CR><LF>
```

Some commands may also produce notifications, which do not start with the command prefix:

```
AT+CGSN
490154203237518
OK
```

3 Module firmware versions and hardware variants

This document describes AT commands used in all versions of the nRF93M1 module firmware.

The firmware versions that support a command are marked in the command description with the following version tag: vx.x.x

If a parameter is not supported by all versions of the firmware that support the command, the firmware versions that support the parameter are marked after the parameter with the version tag.

The version tags are read as follows:

- If a command or parameter is marked v1.0.x, it is supported by module firmware versions where the first two digits are 1 and 0.
- If a command is marked v1.0.x v1.1.x v1.2.x, it is supported by module firmware versions where the first two digits are 1 and 0, 1 and 1, or 1 and 2.
- If a command or parameter is marked v1.1.x≥3, it is supported by module firmware versions where the first two digits are 1 and 1 and the third digit is greater than or equal to 3.
- If a command or parameter is marked v1.1.3, it is supported only by module firmware version 1.1.3.

4 ITU-T V.250 commands

ITU-T Recommendation V.250 commands are used to control the TE-TA interface.

4.1 Attention AT

The **AT** command is used to check communication with the module. v1.4.x

For reference, see *ITU-T V.250 Ch. 5.2.1*.

4.1.1 Set command

The set command verifies that the module is responding to AT commands.

Syntax:

```
AT
```

The command has no parameters. If the module is ready, it responds with `OK`.

The following command example verifies that the module is responding:

```
AT
OK
```

4.1.2 Read command

The read command is not supported.

4.1.3 Test command

The test command is not supported.

4.2 Request packet domain service ATD

The **ATD** command requests packet domain service and causes the *User Equipment (UE)* to enter the V.250 online data state. v1.4.x

4.2.1 Set command

The set command requests packet domain service.

Syntax:

```
ATD*<GPRS_SC>[*[<called_address>][*[<L2P>][*[<cid>]]]]#
```

The set command parameters and their defined values are the following:

<GPRS_SC>

String. Maximum length 2 characters. *General Packet Radio Services (GPRS)* service code.

99 – Identifies a request to use the packet domain service.

<called_address>

String. The called party in the address space applicable to the *Packet Data Protocol (PDP)*. Ignored.

<L2P>

String. Maximum length 3 characters. Layer 2 protocol for the data link.

PPP – Use PPP (default).

1 – Use PPP (default).

<cid>

Integer, 1–15. Specifies a particular *Packet Data Protocol (PDP) Context* definition.

The set command returns `CONNECT` when a connection has been established successfully and `NO CARRIER` when a connection has not been established. When a connection is established, AT commands are no longer interpreted as the modem being in V.250 online data state.

The following command example requests packet domain service:

```
ATD*99#
CONNECT
```

4.2.2 Read command

The read command is not supported.

4.2.3 Test command

The test command is not supported.

4.3 Set baud rate +IPR

The **+IPR** command sets the baud rate at which the *UE* accepts AT commands. v1.4.x

For reference, see *ITU-T V.250 Ch. 6.2.10*.

4.3.1 Set command

The set command sets the baud rate at which the *UE* accepts commands.

The configuration is stored to *Non-volatile Memory (NVM)* when the module is set to minimum functionality.

Syntax:

```
+IPR=<rate>
```

The set command parameter and its defined values are the following:

<rate>

Integer, 0–921600. Baud rate in bits per second.

0 – Enable automatic baud rate detection. The character format is forced to 8 data bits with no parity and 1 stop bit.

600 – 600 bps.

1200 – 1200 bps.

2400 – 2400 bps.

4800 – 4800 bps.

9600 – 9600 bps.

19200 – 19200 bps.

38400 – 38400 bps.

57600 – 57600 bps.

115200 – 115200 bps (default).

230400 – 230400 bps.

460800 – 460800 bps.

921600 – 921600 bps.

The following command example sets the baud rate to `115200` bps:

```
AT+IPR=115200
OK
```

4.3.2 Read command

The read command returns the current baud rate.

Response syntax:

```
+IPR: <rate>
```

The read command parameter and its defined value are the following:

<rate>

Integer, 0–921600. Baud rate in bits per second.

0 – Enable automatic baud rate detection. The character format is forced to 8 data bits with no parity and 1 stop bit.

600 – 600 bps.

1200 – 1200 bps.

2400 – 2400 bps.

4800 – 4800 bps.

9600 – 9600 bps.

19200 – 19200 bps.

38400 – 38400 bps.

57600 – 57600 bps.

115200 – 115200 bps (default).

230400 – 230400 bps.

460800 – 460800 bps.

921600 – 921600 bps.

The following command example reads the current baud rate:

```
AT+IPR?
+IPR: 115200
OK
```

4.3.3 Test command

The test command returns the supported baud rates.

Response syntax:

```
+IPR: (list of supported auto-detectable <rate>s),(list of supported fixed-only <rate>s)
```

The test command parameter and its defined value are the following:

<rate>

Integer, 0–921600. Baud rate in bits per second.

0 – Enable automatic baud rate detection. The character format is forced to 8 data bits with no parity and 1 stop bit.

600 – 600 bps.

1200 – 1200 bps.

2400 – 2400 bps.

4800 – 4800 bps.

9600 – 9600 bps.

19200 – 19200 bps.

38400 – 38400 bps.

57600 – 57600 bps.

115200 – 115200 bps (default).

230400 – 230400 bps.

460800 – 460800 bps.

921600 – 921600 bps.

The following command example returns the supported baud rates:

```
AT+IPR=?
+IPR: (600,1200,2400,4800,9600,19200,38400,57600,115200,230400,460800,921600),
(0,600,1200,2400,4800,9600,19200,38400,57600,115200,230400,460800,921600)
OK
```

4.4 Set local flow control +IFC

The **+IFC** command sets the local flow control between the *Terminal Equipment (TE)* and *Terminal Adapter (TA)*. v1.4.x

For reference, see *ITU-T V.250 Ch. 6.2.12*.

4.4.1 Set command

The set command sets the local flow control method.

Syntax:

```
+IFC=<DCE_by_DTE>,<DTE_by_DCE>
```

The set command parameters and their defined values are the following:

<DCE_by_DTE>

Integer, 0 and 2. Method used by the DTE to control the flow of received data from the DCE.

0 – None.

2 – RTS.

<DTE_by_DCE>

Integer, 0 and 2. Method used by the DCE to control the flow of transmitted data from the DTE.

0 – None.

2 – CTS.

The following command example disables flow control:

```
AT+IFC=0,0
OK
```

4.4.2 Read command

The read command returns the current local flow control settings.

Response syntax:

```
+IFC: <DCE_by_DTE>,<DTE_by_DCE>
```

The read command parameters and their defined values are the following:

<DCE_by_DTE>

Integer, 0 and 2. Method used by the DTE to control the flow of received data from the DCE.

0 – None.

2 – RTS.

<DTE_by_DCE>

Integer, 0 and 2. Method used by the DCE to control the flow of transmitted data from the DTE.

0 – None.

2 – CTS.

The following command example reads the current flow control settings:

```
AT+IFC?
+IFC: 0,0
OK
```

4.4.3 Test command

The test command returns the supported flow control methods.

Response syntax:

```
+IFC: (list of supported <DCE_by_DTE>s),(list of supported <DTE_by_DCE>s)
```

The test command parameters and their defined values are the following:

<DCE_by_DTE>

Integer, 0 and 2. Method used by the DTE to control the flow of received data from the DCE.

0 – None.

2 – RTS.

<DTE_by_DCE>

Integer, 0 and 2. Method used by the DCE to control the flow of transmitted data from the DTE.

0 – None.

2 – CTS.

The following command example returns the supported flow control methods:

```
AT+IFC=?
+IFC: (0,2),(0,2)
OK
```

5 General commands

The general commands are for identifying and configuring various features of the module.

For reference, see *3GPP 27.007 Ch. 5*.

5.1 Manufacturer identification +CGMI

The **+CGMI** command requests manufacturer identification. v1.4.x

For reference, see *3GPP 27.007 Ch. 5.1*.

5.1.1 Set command

The set command requests manufacturer identification.

Syntax:

```
+CGMI
```

Response syntax:

```
<manufacturer_ID>
```

The <manufacturer_ID> parameter returns a string of up to 2048 characters followed by <CR><LF>.

The following command example reads the manufacturer ID:

```
AT+CGMI
Nordic Semiconductor ASA
OK
```

5.1.2 Read command

The read command is not supported.

5.1.3 Test command

The test command returns the manufacturer identification.

Response syntax:

```
<manufacturer_ID>
```

The response parameter and its defined value are the following:

<manufacturer_ID>

String. Maximum length 2048 characters. Manufacturer identification.

The following command example returns the manufacturer identification:

```
AT+CGMI=?
<manufacturer_ID>
OK
```

5.2 Revision identification +CGMR

The **+CGMR** command requests modem firmware revision identification. v1.4.x

For reference, see *3GPP 27.007 Ch. 5.3*.

5.2.1 Set command

The set command requests revision identification.

Syntax:

```
+CGMR
```

Response syntax:

```
<revision>
```

The <revision> parameter returns a string of up to 2048 characters followed by <CR><LF>OK.

The following command example reads the revision ID:

```
AT+CGMR
mfw_nrf93m1_1.4.1
OK
```

5.2.2 Read command

The read command is not supported.

5.2.3 Test command

The test command is not supported.

5.3 Cell information %BCINFO

The proprietary **%BCINFO** command returns basic serving cell information and neighbor cell information which is mainly used for location services. v1.4.x

Note:

- If the modem is not registered, the cell with the best signal quality or *Reference Signal Received Power (RSRP)* is regarded as the serving cell.
- If the modem is registered, only cells with the same registered *Public Land Mobile Network (PLMN)* or EPLMN are returned.

5.3.1 Set command

The set command returns basic cell information for the serving cell and optionally for neighbor cells.

Syntax:

```
%BCINFO[=<mode>[,<timeoutS>[,<save_for_later>[,<max_cell_number>[,<report_mode>[,<freq_list>[,<band_list>
```

Response syntax for serving cell:

```
%BCINFOSC: <earfcn>,<pci>,<rsrp>,<rsrq>,<mcc>,<mnc>,<cellid>,<tac>
```

Response syntax for neighbor cells:

```
%BCINFONC: <earfcn>,<pci>,<rsrp>,<rsrq>
```

+CME ERROR code

50 – Incorrect parameters.

The set command parameters and their defined values are the following:

<mode>

Integer, 0–5. Measurement mode.

0 – Measure neighbor cell *RSRP* and *Reference Signal Received Quality (RSRQ)* without acquiring SIB1. The neighbor cell response does not include location information (default).

1 – Search cells including SIB1 acquisition to obtain the cell ID, *PLMN*, and <tac>. The neighbor cell response includes location information.

2 – Read saved cell information. Requires <save_for_later> to be set to `1` in a previous command.

3 – Search cells including SIB1 acquisition to obtain the cell ID, PLMN, and <tac>. The neighbor cell response includes location information. Higher priority than a PLMN search. If a PLMN search is ongoing, it is aborted.

4 – Search cells including SIB1 acquisition. Higher priority than a Wi-Fi scan. If a Wi-Fi scan is ongoing, it is aborted.

5 – Search cells including SIB1 acquisition. Higher priority than a PLMN search and a Wi-Fi scan. If either one is ongoing, it is aborted.

<timeoutS>

Integer, 4–300. Maximum measurement time in seconds. The default is `8`.

<save_for_later>

Integer, 0–1. Save cell information for later retrieval.

0 – Do not save (default).

1 – Save cell information.

<max_cell_number>

Integer, 1–21. Maximum number of cells to measure or search including the serving cell. The default is `21`. For <mode>=`2`, the maximum is `7`.

<report_mode>

Integer, 0–1. Report mode.

0 – Synchronous mode. Report cell information in an AT response (default).

1 – Asynchronous mode. Command returns immediately. Cell information is returned as *Unsolicited Result Code (URC)*s.

<freq_list>

String. Maximum length 100 bytes. Comma-separated list of *E-UTRA Absolute Radio Frequency Channel Number (EARFCN)*s to measure. If omitted, all frequencies are measured.

<band_list>

String. Maximum length 100 bytes. Comma-separated list of bands to measure. If omitted, all bands are measured.

The response parameters and their defined values are the following:

<earfcn>

Integer, 0–262143. EARFCN of the cell.

<pci>

Integer, 0–503. Physical cell ID.

<rsrp>

Integer, –156 to –44. RSRP in dBm.

<rsrq>

Integer, –34–25. RSRQ in dB.

<mcc>

String. Maximum length 3 bytes. Mobile country code.

<mnc>

String. Maximum length 3 bytes. Mobile network code.

<cellid>

String. Maximum length 4 bytes. *Evolved Terrestrial Radio Access Network (E-UTRAN)* cell ID in hexadecimal format.

<tac>

String. Maximum length 2 bytes. Tracking area code in hexadecimal format.

The following command example returns serving cell information:

```
AT%BCINFO=0
%BCINFOSC: 1598,140,-96,-11,"244","12","001FE503","0138"
OK
```

The following command example returns serving cell and neighbor cell information with location data:

```
AT%BCINFO=1
%BCINFOSC: 1598,140,-95,-9,"244","12","001FE503","0138"
%BCINFONC: 2850,110,-103,-10,"244","12","001B8B0D","0138"
%BCINFONC: 300,110,-108,-15,"244","12","001B8B17","0138"
%BCINFONC: 300,56,-109,-17,"244","12","0030F817","0141"
OK
```

5.3.2 Read command

The read command is not supported.

5.3.3 Test command

The test command returns supported values as compound values.

Response syntax:

```
%BCINFO: (range of supported <mode>s),(range of supported <timeoutS>s),(list of supported
 <save_for_later>s),
(range of supported <max_cell_number>s),(list of supported
 <report_mode>s),<freq_list>,<band_list>
```

The test command parameters and their defined values are the following:

<mode>

Integer, 0–5. Measurement mode.

0 – Measure neighbor cell *RSRP* and *RSRQ* without acquiring SIB1. The neighbor cell response does not include location information (default).

1 – Search cells including SIB1 acquisition to obtain the cell ID, *PLMN*, and <tac>. The neighbor cell response includes location information.

2 – Read saved cell information. Requires <save_for_later> to be set to `1` in a previous command.

3 – Search cells including SIB1 acquisition to obtain the cell ID, PLMN, and <tac>. The neighbor cell response includes location information. Higher priority than a PLMN search. If a PLMN search is ongoing, it is aborted.

4 – Search cells including SIB1 acquisition. Higher priority than a Wi-Fi scan. If a Wi-Fi scan is ongoing, it is aborted.

5 – Search cells including SIB1 acquisition. Higher priority than a PLMN search and a Wi-Fi scan. If either one is ongoing, it is aborted.

<timeoutS>

Integer, 4–300. Maximum measurement time in seconds. The default is `8`.

<save_for_later>

Integer, 0–1. Save cell information for later retrieval.

0 – Do not save (default).

1 – Save cell information.

<max_cell_number>

Integer, 1–21. Maximum number of cells to measure or search including the serving cell. The default is `21`. For <mode>=`2`, the maximum is `7`.

<report_mode>

Integer, 0–1. Report mode.

0 – Synchronous mode. Report cell information in an AT response (default).

1 – Asynchronous mode. Command returns immediately. Cell information is returned as *URC*s.

<freq_list>

String. Maximum length 100 bytes. Comma-separated list of *EARFCN*s to measure. If omitted, all frequencies are measured.

<band_list>

String. Maximum length 100 bytes. Comma-separated list of bands to measure. If omitted, all bands are measured.

The following command example returns supported values as compound values:

```
AT%BCINFO=?
%BCINFO: (0-5),(4-300),(0,1),(1-21),(0,1),<freq_list>,<band_list>
OK
```

5.4 Wi-Fi scan %WIFISCAN

The proprietary **%WIFISCAN** command detects Wi-Fi SSID information for location services. v1.4.x

For the most optimal result, execute a Wi-Fi scan when *UE* is in RRC IDLE mode with *Discontinuous Reception (DRX)* or *Extended Discontinuous Reception (eDRX)*. A scan can also be executed in *Power Saving Mode (PSM)* or when cellular functionality is disabled by entering Minimum functionality mode or Flight mode.

5.4.1 Set command

The set command executes a Wi-Fi scan with the specified options.

Syntax:

```
%WIFISCAN=[<time>][,<round>][,<maxbssidnum>][,<scantimeout>][,<priority>]
[,<channelRecLen>][,<channelCount>][,<channelId1>][,<channelId2>]...[,<channelId14>]
[,<specific MAC address>]
```

Response syntax:

```
%WIFISCAN: (<ecn>,<ssid>,<rssi>,<mac>,<channel>)
```

The set command parameters and their defined values are the following:

<time>

Integer, 4000–255000. Maximum response time of a Wi-Fi scan in milliseconds. The default is `12000`.

<round>

Integer, 1–3. Maximum number of Wi-Fi scan rounds. The default is `1`.

<maxbssidnum>

Integer, 4–40. Maximum required Wi-Fi *Service Set Identifier (SSID)* number. The default is `5`.

<scantimeout>

Integer, 1–255. Maximum search time for each Wi-Fi scan round in seconds. The default is `5`.

<priority>

Integer, 0–1. Priority of a Wi-Fi scan.

0 – Data preferred. Wi-Fi scan waits for *Radio Resource Control (RRC)* IDLE state (default). If the module is in RRC CONNECTED state, the Wi-Fi scan waits until the module returns to RRC IDLE state.

1 – Wi-Fi preferred. Connection is released immediately to start a scan. If the module is in RRC CONNECTED state, the connection is released immediately and a Wi-Fi scan starts.

<channelRecLen>

Integer, 100–280. Maximum scan time on each frequency channel in milliseconds. The default is `280`.

<channelCount>

Integer, 1–14. Number of frequency channels to scan. If <channelCount> is `1` and <channelId1> is `0`, all frequencies are scanned. The default value is `1`.

<channelId1>

Integer, 0–14. Specific frequency channel to scan. When the value is `0`, no channel is specified. The default value is `0`.

<channelId2> – <channelId14>

Integer, 1–14. Specific frequency channel to scan. Channels are scanned in the provided order.

<specific MAC address>

String. Maximum length 17 bytes. Specific BSSID to scan. If it is specified, only this BSSID is reported. If the BSSID is not found, an empty result is returned. The default is empty.

The response parameters and their defined values are the following:

<ecn>

String. Encryption type.

<ssid>

String. Maximum length 32 bytes. SSID name.

<rssi>

Integer, –100–0. Signal strength in dBm.

<mac>

String. Maximum length 17 bytes. BSSID of the access point.

<channel>

Integer, 1–13. Wi-Fi channel number.

The <time> value should not be less than <round> multiplied by <scantimeout>. If the required number of SSIDs is detected, the scan stops and reports the result. Duplicate SSIDs are removed and the results are sorted by signal strength in descending order.

Note: The way the SSIDs are displayed can be configured with the **`%MODULECFG`** command's <wifiSsidMode> parameter.

The following command example scans for up to five Wi-Fi networks:

```
AT%WIFISCAN=12000,1,5,3,0
%WIFISCAN: (-,,-28,"00:BE:D5:20:EA:54",11)
%WIFISCAN: (-,,-38,"00:BE:D5:20:EA:27",3)
%WIFISCAN: (-,,-39,"00:BE:D5:20:EA:27",6)
%WIFISCAN: (-,,-44,"00:BE:D5:20:EA:27",9)
%WIFISCAN: (-,,-44,"00:BE:D5:20:EC:23",1)
OK
```

The following command example scans channels `7`, `4`, and `11`:

```
AT%WIFISCAN=12000,1,5,3,0,280,3,7,4,11
%WIFISCAN: (-,,-28,"00:BE:D5:20:EA:54",11)
%WIFISCAN: (-,,-38,"00:BE:D5:20:EA:27",7)
%WIFISCAN: (-,,-39,"00:BE:D5:20:EA:27",4)
%WIFISCAN: (-,,-44,"00:BE:D5:20:EA:27",11)
%WIFISCAN: (-,,-44,"00:BE:D5:20:EC:23",4)
OK
```

5.4.2 Read command

The read command returns the current Wi-Fi scan parameters.

Response syntax:

```
%WIFISCAN: <time>,<round>,<maxbssidnum>,<scantimeout>,<priority>,<channelRecLen>,
<channelCount>,<channelId1>[,...]
```

The read command parameters and their defined values are the following:

<time>

Integer, 4000–255000. Maximum response time of a Wi-Fi scan in milliseconds. The default is `12000`.

<round>

Integer, 1–3. Maximum number of Wi-Fi scan rounds. The default is `1`.

<maxbssidnum>

Integer, 4–40. Maximum required Wi-Fi *SSID* number. The default is `5`.

<scantimeout>

Integer, 1–255. Maximum search time for each Wi-Fi scan round in seconds. The default is `5`.

<priority>

Integer, 0–1. Priority of a Wi-Fi scan.

0 – Data preferred. Wi-Fi scan waits for *RRC* IDLE state (default). If the module is in RRC CONNECTED state, the Wi-Fi scan waits until the module returns to RRC IDLE state.

1 – Wi-Fi preferred. Connection is released immediately to start a scan. If the module is in RRC CONNECTED state, the connection is released immediately and a Wi-Fi scan starts.

<channelRecLen>

Integer, 100–280. Maximum scan time on each frequency channel in milliseconds. The default is `280`.

<channelCount>

Integer, 1–14. Number of frequency channels to scan. If <channelCount> is `1` and <channelId1> is `0`, all frequencies are scanned. The default value is `1`.

<channelId1>

Integer, 0–14. Specific frequency channel to scan. When the value is `0`, no channel is specified. The default value is `0`.

<channelId2> – <channelId14>

Integer, 1–14. Specific frequency channel to scan. Channels are scanned in the provided order.

<specific MAC address>

String. Maximum length 17 bytes. Specific BSSID to scan. If it is specified, only this BSSID is reported. If the BSSID is not found, an empty result is returned. The default is empty.

The following command example reads the current Wi-Fi scan parameters:

```
AT%WIFISCAN?
%WIFISCAN: 12000,1,5,5,0,280,1,0
OK
```

5.4.3 Test command

The test command returns supported values as compound values.

Response syntax:

```
%WIFISCAN: (range of supported <time>s),(range of supported <round>s),(range of supported
 <maxbssidnum>s),
(range of supported <scantimeout>s),(range of supported <priority>s),(range of supported
 <channelRecLen>s),
(range of supported <channelCount>s),(range of supported <channelId1>s),...,"<MAC format>"
```

The test command parameters and their defined values are the following:

<time>

Integer, 4000–255000. Maximum response time of a Wi-Fi scan in milliseconds. The default is `12000`.

<round>

Integer, 1–3. Maximum number of Wi-Fi scan rounds. The default is `1`.

<maxbssidnum>

Integer, 4–40. Maximum required Wi-Fi *SSID* number. The default is `5`.

<scantimeout>

Integer, 1–255. Maximum search time for each Wi-Fi scan round in seconds. The default is `5`.

<priority>

Integer, 0–1. Priority of a Wi-Fi scan.

0 – Data preferred. Wi-Fi scan waits for *RRC* IDLE state (default). If the module is in RRC CONNECTED state, the Wi-Fi scan waits until the module returns to RRC IDLE state.

1 – Wi-Fi preferred. Connection is released immediately to start a scan. If the module is in RRC CONNECTED state, the connection is released immediately and a Wi-Fi scan starts.

<channelRecLen>

Integer, 100–280. Maximum scan time on each frequency channel in milliseconds. The default is `280`.

<channelCount>

Integer, 1–14. Number of frequency channels to scan. If <channelCount> is `1` and <channelId1> is `0`, all frequencies are scanned. The default value is `1`.

<channelId1>

Integer, 0–14. Specific frequency channel to scan. When the value is `0`, no channel is specified. The default value is `0`.

<channelId2> – <channelId14>

Integer, 1–14. Specific frequency channel to scan. Channels are scanned in the provided order.

<specific MAC address>

String. Maximum length 17 bytes. Specific BSSID to scan. If it is specified, only this BSSID is reported. If the BSSID is not found, an empty result is returned. The default is empty.

The following command example returns supported values as compound values:

```
AT%WIFISCAN=?
%WIFISCAN:(4000-255000),(1-3),(4-40),(1-255),(0-1),(100-280),(1-14),(0-14),(1-14),
(1-14),(1-14),(1-14),(1-14),(1-14),(1-14),(1-14),(1-14),(1-14),(1-14),(1-14),
(1-14),"AA:BB:CC:DD:EE:FF"
OK
```

5.5 RF non-signaling test %RFTEST

The proprietary **%RFTEST** command is used for non-signaling RF testing of the module. v1.4.x

The command allows production-line testing of RF transmission (TX) and reception (RX) performance without establishing a network connection.

Warning: Ensure the **ANT** port is correctly connected to an antenna or test instrument before high-power transmission. Mismatch at the **ANT** port during high-power transmission risks damaging the RF module.

Before using the **%RFTEST** command, execute `AT+CFUN=0`.

Ensure that **%PRODDONE** is set to `0`. If **%PRODDONE** is set to `1`, the **%RFTEST** command is blocked.

5.5.1 Set command

The set command executes non-signaling RF test operations.

Syntax:

```
%RFTEST=<test>,<operation>[,<param0>,<param1>,<param2>]
```

After the command is accepted, the module returns `OK`. The module outputs also the response string `MT000000000000` to indicate an execution.

To convert RF test payload into dBm after receiving the `MT` response, split the payload into groups of 8 hexadecimal characters and reverse the byte order of each group before converting it to a numeric value. The first 8 hexadecimal characters after `MT` identify the response type. The following groups carry response data, such as measurement values. RSSI measurement fields are signed 32-bit integers, and the decoded value must be divided by `16` to obtain dBm.

For example, for the `MT000004001EFCFFFF0000` response, extract the measurement field `1EFCFFFF`, reverse it to `FFFFFC1E`, interpret it as `-994` as a signed 32-bit integer, and divide it by `16` to get `-62.125` dBm.

The set command parameters and their defined values are the following:

<test>

Integer, 0–3. Selects the function of the RF test.

0 – Enter or exit non-signaling test mode.

1 – Start or stop single-tone TX.

2 – Execute an RSSI RX test.

3 – Obtain the RF test status.

<operation>

Integer, 0–1. Operation for the selected test.

0 – If <test> is set to `0`, exit non-signaling test mode. If <test> is set to `1`, stop single-tone TX.

1 – If <test> is set to `0`, enter non-signaling test mode. If <test> is set to `1`, start single-tone TX.

If <test> is set to `2` or `3`, the parameter is ignored.

<param0>

Integer, 1–66. LTE band number.

<param1>

Integer, 6000–26900. Frequency in units of 100 kHz. For example, `8365` means 836.5 MHz.

<param2>

Integer, 0–5. The function of this parameter depends on the function of the RF test selected with <test>.

If <test> is set to `1`, configure TX power in dBm. Range 0–23.

If <test> is set to `2`, select RX bandwidth. Range 0–5.

0 – 1.4 MHz.

1 – 3 MHz.

2 – 5 MHz.

3 – 10 MHz.

4 – 15 MHz.

5 – 20 MHz.

The following command example performs a single-tone TX test on band `5` at `836.5` MHz with `23` dBm:

```
AT+CFUN=0
OK
AT%RFTEST=0,1
OK
MT000000000000

AT%RFTEST=1,1,5,8365,23
OK
MT000000000000

AT%RFTEST=1,0
OK
MT000000000000

AT%RFTEST=0,0
OK
MT000000000000
```

The following command example performs an RSSI RX test on band `3` at `1825.0` MHz with `15` MHz bandwidth:

```
AT+CFUN=0
OK
AT%RFTEST=0,1
OK
MT000000000000

AT%RFTEST=2,1,3,18250,4
OK
MT000004001EFCFFFF0000

AT%RFTEST=0,0
OK
MT000000000000
```

5.5.2 Read command

The read command is not supported.

5.5.3 Test command

The test command returns the parameter format supported by **%RFTEST**.

Response syntax:

```
%RFTEST: <test>,<operation>,<param0>,<param1>,<param2>
```

The test command parameters and their defined values are the following:

<test>

Integer, 0–3. Selects the function of the RF test.

0 – Enter or exit non-signaling test mode.

1 – Start or stop single-tone TX.

2 – Execute an RSSI RX test.

3 – Obtain the RF test status.

<operation>

Integer, 0–1. Operation for the selected test.

0 – If <test> is set to `0`, exit non-signaling test mode. If <test> is set to `1`, stop single-tone TX.

1 – If <test> is set to `0`, enter non-signaling test mode. If <test> is set to `1`, start single-tone TX.

If <test> is set to `2` or `3`, the parameter is ignored.

<param0>

Integer, 1–66. LTE band number.

<param1>

Integer, 6000–26900. Frequency in units of 100 kHz. For example, `8365` means 836.5 MHz.

<param2>

Integer, 0–5. The function of this parameter depends on the function of the RF test selected with <test>.

If <test> is set to `1`, configure TX power in dBm. Range 0–23.

If <test> is set to `2`, select RX bandwidth. Range 0–5.

0 – 1.4 MHz.

1 – 3 MHz.

2 – 5 MHz.

3 – 10 MHz.

4 – 15 MHz.

5 – 20 MHz.

The following command example returns the parameter format:

```
AT%RFTEST=?
%RFTEST: <test>,<operation>,<param0>,<param1>,<param2>
OK
```

5.6 Production test complete %PRODDONE

The proprietary **%PRODDONE** command reads and updates the non-volatile RF conducted test complete flag. v1.4.x

5.6.1 Set command

The set command sets the RF conducted test status to done.

Syntax:

```
%PRODDONE
```

The following command example sets the RF conducted test status to done:

```
AT%PRODDONE
OK
```

5.6.2 Read command

The read command returns the current RF conducted test complete status.

Syntax:

```
%PRODDONE?
```

Response syntax:

```
%PRODDONE: <value>
```

The response parameter and its defined values are the following:

<value>

Integer, 0–1. RF conducted test complete status.

0 – RF conducted test process is not marked complete.

1 – RF conducted test process is marked complete.

The following command example reads the RF conducted test status:

```
AT%PRODDONE?
%PRODDONE: 1
OK
```

5.6.3 Test command

The test command is not supported.

5.7 Channel multiplexing mode +CMUX

The **+CMUX** command configures channel multiplexing mode. v1.4.x

The GSM 0710 Multiplexer Protocol (CMUX) enables multiple data streams to be multiplexed through a single serial link and sets up one channel for each data stream. For example, it can be used to exchange AT data and have a Point-to-Point Protocol (PPP) link up at the same time on a single UART.

For reference, see *3GPP 27.007 Ch. 5.7*.

5.7.1 Set command

The set command configures channel multiplexing mode.

Syntax:

```
+CMUX=<transparency>[,<subset>[,<port_speed>[,<N1>[,<T1>[,<N2>[,<T2>[,<T3>[,<k>]]]]]]]]
```

The set command parameters and their defined values are the following:

<transparency>

Integer, 0–1. Multiplexer transparency mechanism. The default is `0`.

0 – Basic option.

1 – Advanced option.

<subset>

Integer, 0–2. Multiplexer control channel frame type.

0 – UIH frames are used only.

1 – UI frames are used only.

2 – I frames are used only.

<port_speed>

Integer, 1–8. Transmission rate. The default is implementation-specific.

1 – 9600 bps.

2 – 19200 bps.

3 – 38400 bps.

4 – 57600 bps.

5 – 115200 bps.

6 – 230400 bps.

7 – 460800 bps.

8 – 921600 bps.

<N1>

Integer, 1–32768. Maximum frame size. The default for the basic <transparency> option is `31`. The default for the advanced <transparency> option is `64`.

<T1>

Integer, 1–255. Acknowledgement timer in units of ten milliseconds. The default is `10` (100 ms).

<N2>

Integer, 0–100. Maximum number of retransmissions. The default is `3`.

<T2>

Integer, 2–255. Response timer for the multiplexer control channel in units of ten milliseconds. The default is `30` (300 ms).

<T3>

Integer, 1–255. Wakeup response timer in seconds. The default is `10`.

<k>

Integer, 1–7. Window size for the advanced <transparency> option with Error-Recovery Mode. The default is `2`.

The following command example starts channel multiplexing in basic mode:

```
AT+CMUX=0
OK
```

5.7.2 Read command

The read command returns the current channel multiplexing configuration.

Response syntax:

```
+CMUX: <transparency>,[<subset>],<port_speed>,<N1>,<T1>,<N2>,<T2>,<T3>[,<k>]
```

The read command parameters and their defined values are the following:

<transparency>

Integer, 0–1. Multiplexer transparency mechanism. The default is `0`.

0 – Basic option.

1 – Advanced option.

<subset>

Integer, 0–2. Multiplexer control channel frame type.

0 – UIH frames are used only.

1 – UI frames are used only.

2 – I frames are used only.

<port_speed>

Integer, 1–8. Transmission rate. The default is implementation-specific.

1 – 9600 bps.

2 – 19200 bps.

3 – 38400 bps.

4 – 57600 bps.

5 – 115200 bps.

6 – 230400 bps.

7 – 460800 bps.

8 – 921600 bps.

<N1>

Integer, 1–32768. Maximum frame size. The default for the basic <transparency> option is `31`. The default for the advanced <transparency> option is `64`.

<T1>

Integer, 1–255. Acknowledgement timer in units of ten milliseconds. The default is `10` (100 ms).

<N2>

Integer, 0–100. Maximum number of retransmissions. The default is `3`.

<T2>

Integer, 2–255. Response timer for the multiplexer control channel in units of ten milliseconds. The default is `30` (300 ms).

<T3>

Integer, 1–255. Wakeup response timer in seconds. The default is `10`.

<k>

Integer, 1–7. Window size for the advanced <transparency> option with Error-Recovery Mode. The default is `2`.

The following command example returns the current channel multiplexing configuration:

```
AT+CMUX?
+CMUX: 0, 0, 5, 31, 10, 3, 30, 10, 2
```

5.7.3 Test command

The test command returns a list of supported channel multiplexing parameters.

Syntax:

```
+CMUX: (list of supported <transparency>s),(list of supported <subset>s),(list of supported
 <port_speed>s),(list of supported <N1>s),(list of supported <T1>s),(list of supported
 <N2>s),(list of supported <T2>s),(list of supported <T3>s),(list of supported <k>s)
```

The test command parameters and their defined values are the following:

<transparency>

Integer, 0–1. Multiplexer transparency mechanism. The default is `0`.

0 – Basic option.

1 – Advanced option.

<subset>

Integer, 0–2. Multiplexer control channel frame type.

0 – UIH frames are used only.

1 – UI frames are used only.

2 – I frames are used only.

<port_speed>

Integer, 1–8. Transmission rate. The default is implementation-specific.

1 – 9600 bps.

2 – 19200 bps.

3 – 38400 bps.

4 – 57600 bps.

5 – 115200 bps.

6 – 230400 bps.

7 – 460800 bps.

8 – 921600 bps.

<N1>

Integer, 1–32768. Maximum frame size. The default for the basic <transparency> option is `31`. The default for the advanced <transparency> option is `64`.

<T1>

Integer, 1–255. Acknowledgement timer in units of ten milliseconds. The default is `10` (100 ms).

<N2>

Integer, 0–100. Maximum number of retransmissions. The default is `3`.

<T2>

Integer, 2–255. Response timer for the multiplexer control channel in units of ten milliseconds. The default is `30` (300 ms).

<T3>

Integer, 1–255. Wakeup response timer in seconds. The default is `10`.

<k>

Integer, 1–7. Window size for the advanced <transparency> option with Error-Recovery Mode. The default is `2`.

The following command example lists the supported parameters:

```
AT+CMUX=?
+CMUX: 0,[(0,1)],<1-8>,<1-32768>,<1-255>,<0-100>,<2-255>,<1-255>[,<1-7>]
```

5.8 SSL/TLS configuration %SSLCFG

The proprietary **%SSLCFG** command configures SSL/TLS parameters for secure connections. v1.4.x

5.8.1 Set command

The set command configures an SSL/TLS parameter for a specified context. If the optional value is omitted, the current configuration for that parameter is returned.

Syntax:

```
%SSLCFG=<option>,<SSL_ctxID>[,<value>]
```

Response syntax when value is omitted:

```
%SSLCFG: <option>,<SSL_ctxID>,<value>
```

The set command parameters and their defined values are the following:

<option>

String. Name of the configuration parameter.

<SSL_ctxID>

Integer, 0–5. SSL context identifier.

The following configuration parameters are supported:

"ciphersuite"

String. Maximum length 6 characters. SSL cipher suite in hexadecimal notation in the format `0xXXXX`.

0xC0AD – TLS_ECDHE_ECDSA_WITH_AES_256_CCM.

0xC09F – TLS_DHE_RSA_WITH_AES_256_CCM.

"cacert"

String. Maximum length 47 characters. CA certificate file path.

"clientcert"

String. Maximum length 47 characters. Client certificate file path.

"clientkey"

String. Maximum length 47 characters. Client private key file path.

"psk_identity"

String. Maximum length 47 characters. File path to the PSK identity used in PSK-based TLS authentication.

"psk_key"

String. Maximum length 47 characters. File path to the preshared key used in PSK-based TLS authentication.

"seclevel"

Integer, 0–2. Authentication mode.

0 – No authentication (default).

1 – Perform server authentication.

2 – Perform server and client authentication if required by the remote server.

"session_cache"

Integer, 0–1. Session resumption.

0 – Disabled.

1 – Enabled (default).

"sni"

Integer, 0–1. Server Name Indication.

0 – Disabled (default).

1 – Enabled.

"ignorelocaltime"

Integer, 0–1. Ignore local time for certificate validation.

0 – Disabled.

1 – Enabled (default).

"debug"

Integer, 0–1. SSL debug logging.

0 – Disabled (default).

1 – Enabled.

Note: PSK-based TLS is enabled only when both "psk_identity" and "psk_key" are configured for the same SSL context.

The following command example sets the security level to server authentication for SSL context 0:

```
AT%SSLCFG="seclevel",0,1
OK
```

The following command example sets the CA certificate path for SSL context 0:

```
AT%SSLCFG="cacert",0,"ca.pem"
OK
```

The following command example sets the PSK identity and key file paths for SSL context 0:

```
AT%SSLCFG="psk_identity",0,"identity.txt"
OK
AT%SSLCFG="psk_key",0,"key.txt"
OK
```

5.8.2 Read command

The read command returns the current SSL/TLS configuration for all contexts.

Response syntax:

```
<SSL_ctxID>:<cacert>,<clientcert>,<clientkey>,"seclevel":<seclevel>,"cache":<session_cache>,
"ignore":<ignorelocaltime>,"debug":<debug>,"psk_identity":<psk_identity_path>,"psk_key":<psk_key_path>
```

The response parameters and their defined values are the following:

<SSL_ctxID>

Integer, 0–5. SSL context identifier.

"cacert"

String. Maximum length 47 characters. CA certificate file path.

"clientcert"

String. Maximum length 47 characters. Client certificate file path.

"clientkey"

String. Maximum length 47 characters. Client private key file path.

"seclevel"

Integer, 0–2. Authentication mode.

0 – No authentication (default).

1 – Perform server authentication.

2 – Perform server and client authentication if required by the remote server.

"session_cache"

Integer, 0–1. Session resumption.

0 – Disabled.

1 – Enabled (default).

"ignorelocaltime"

Integer, 0–1. Ignore local time for certificate validation.

0 – Disabled.

1 – Enabled (default).

"debug"

Integer, 0–1. SSL debug logging.

0 – Disabled (default).

1 – Enabled.

"psk_identity"

String. Maximum length 47 characters. File path to the PSK identity used in PSK-based TLS authentication.

"psk_key"

String. Maximum length 47 characters. File path to the preshared key used in PSK-based TLS authentication.

The following command example reads the current SSL/TLS configuration:

```
AT%SSLCFG?
0:,,,"seclevel":0,"cache":1,"ignore":1,"debug":0,"psk_identity":"","psk_key":""
OK
```

5.8.3 Test command

The test command returns the supported configuration parameters and their value ranges.

Response syntax:

```
%SSLCFG: "<option>":(range of supported <SSL_ctxID>s),<value_range>
[%SSLCFG: "<option>":(range of supported <SSL_ctxID>s),<value_range>
[...]]
```

The test command parameters and their defined values are the following:

<option>

String. Name of the configuration parameter.

<SSL_ctxID>

Integer, 0–5. SSL context identifier.

<value_range>

Range or list of supported values for the option.

The following command example returns the supported parameters and their ranges:

```
AT%SSLCFG=?
%SSLCFG: "ciphersuite":(0-5),<cipher_suites>
%SSLCFG: "cacert":(0-5),<cacertpath>
%SSLCFG: "clientcert":(0-5),<client_cert_path>
%SSLCFG: "clientkey":(0-5),<client_key_path>
%SSLCFG: "seclevel":(0-5),(0-2)
%SSLCFG: "session_cache":(0-5),(0,1)
%SSLCFG: "sni":(0-5),(0,1)
%SSLCFG: "psk_identity":(0-5),<psk_identity_path>
%SSLCFG: "psk_key":(0-5),<psk_key_path>
%SSLCFG: "ignorelocaltime":(0-5),(0,1)
%SSLCFG: "debug":(0-5),(0,1)
OK
```

6 File commands

The file commands are for managing files in the module's filesystem.

6.1 Open file %FOPEN

The proprietary **%FOPEN** command opens a file and returns a file handle. v1.4.x

6.1.1 Set command

The set command opens a file and returns a file handle.

Syntax:

```
%FOPEN=<filename>[,<mode>]
```

Response syntax:

```
%FOPEN: <filehandle>
```

%FILE ERROR

PARAMETER ERROR – The command is malformed or out of range.

OPEN ERROR – The filesystem failed to open the file.

FILE NOT FOUND – The file does not exist. Returned when <mode> is 2.

FILE ALREADY OPEN – The file is already open. Close the existing handle first.

NO FREE HANDLES – All 15 file handle slots are in use.

The set command parameters and their defined values are the following:

<filename>

String. Maximum length 62 bytes. Name of the file to be opened.

<filehandle>

Integer, 1–15. File handle.

<mode>

Integer, 0–2. File open mode.

0 – If the file does not exist, a new file is created. If the file exists, it is opened directly. The file is readable and writable (default).

1 – If the file does not exist, a new file is created. If the file exists, it is cleared and overwritten. The file is readable and writable.

2 – If the file exists, it is opened as read-only. If the file does not exist, an error is returned.

The following command example opens a file for writing:

```
AT%FOPEN="config.txt",1
%FOPEN: 1

OK
```

6.1.2 Read command

The read command returns information about all currently open files.

One or more lines are returned with one line for each open file.

Response syntax:

```
%FOPEN: <filename>,<filehandle>,<mode>,<size>
```

The read command parameters and their defined values are the following:

<filename>

String. Maximum length 62 bytes. Name of the file to be opened.

<filehandle>

Integer, 1–15. File handle.

<mode>

Integer, 0–2. File open mode.

0 – If the file does not exist, a new file is created. If the file exists, it is opened directly. The file is readable and writable (default).

1 – If the file does not exist, a new file is created. If the file exists, it is cleared and overwritten. The file is readable and writable.

2 – If the file exists, it is opened as read-only. If the file does not exist, an error is returned.

<size>

String. File data length displayed in bytes.

The following command example returns information about an open file:

```
AT%FOPEN?
%FOPEN: config.txt,1,2,26B
OK
```

6.1.3 Test command

The test command returns the command syntax.

Response syntax:

```
%FOPEN: <filename>[,<mode>]
```

The test command parameters and their defined values are the following:

<filename>

String. Maximum length 62 bytes. Name of the file to be opened.

<mode>

Integer, 0–2. File open mode.

0 – If the file does not exist, a new file is created. If the file exists, it is opened directly. The file is readable and writable (default).

1 – If the file does not exist, a new file is created. If the file exists, it is cleared and overwritten. The file is readable and writable.

2 – If the file exists, it is opened as read-only. If the file does not exist, an error is returned.

The following command example returns the command syntax:

```
AT%FOPEN=?
%FOPEN: <filename>[,<mode>]

OK
```

6.2 Read file %FREAD

The proprietary **%FREAD** command reads data from an open file. v1.4.x

6.2.1 Set command

The set command reads data from an open file. The modem switches to data mode to transfer the file contents.

When the total data size exceeds <length>, the modem switches back to command mode.

Syntax:

```
%FREAD=<filehandle>[,<length>]
```

Response syntax:

```
CONNECT <read_length>
```

%FILE ERROR

PARAMETER ERROR – The command is malformed or the file handle is invalid or out of range.

READ ERROR – The filesystem failed to read the file.

OUT OF MEMORY – Memory allocation for the read buffer failed.

The set command parameters and their defined values are the following:

<filehandle>

Integer, 1–15. File handle.

<length>

Integer, 0–6000. Expected file length to be read. If it is omitted or 0, the entire file is read.

<read_length>

Integer. Actual length of data read from the file.

The following command example reads all data from file handle `1`:

```
AT%FREAD=1
CONNECT 26
Hello, this is a test file

OK
```

6.2.2 Read command

The read command is not supported.

6.2.3 Test command

The test command returns the command syntax.

Response syntax:

```
%FREAD:<filehandle>[,<length>]
```

The test command parameters and their defined values are the following:

<filehandle>

Integer, 1–15. File handle.

<length>

Integer, 0–6000. Expected file length to be read. If it is omitted or `0`, the entire file is read.

The following command example returns the command syntax:

```
AT%FREAD=?
%FREAD:<filehandle>[,<length>]

OK
```

6.3 Write file %FWRITE

The proprietary **%FWRITE** command writes data to an open file. v1.4.x

6.3.1 Set command

The set command writes data to an open file. The modem switches to data mode to receive the data.

Syntax:

```
%FWRITE=<filehandle>[,<length>[,<timeout>]]
```

Response syntax:

```
CONNECT
```

%FILE ERROR

PARAMETER ERROR – The command is malformed or the file handle is invalid or out of range.

WRITE ERROR – The filesystem failed to write the requested amount of data.

OPERATION NOT ALLOWED IN THIS MODE – The file handle was opened read-only.

The modem switches to data mode. When the total data size exceeds <length>, the modem switches back to command mode and returns the following:

```
%FWRITE: <written_length>,<total_length>
```

The set command parameters and their defined values are the following:

<filehandle>

Integer, 1–15. File handle.

<length>

Integer, 0–10240. Maximum number of bytes to accept in data mode. The default is `1000`.

<timeout>

Integer, 0–100. Waiting time for input data in seconds. The default is `5`.

<written_length>

Integer. Length of data actually written.

<total_length>

Integer. Total file length after write.

The following command example writes `26` bytes to file handle `1` with a `40`second timeout:

```
AT%FWRITE=1,26,40
CONNECT
Hello, this is a test file.
%FWRITE: 26,26

OK
```

6.3.2 Read command

The read command is not supported.

6.3.3 Test command

The test command returns the command syntax.

Response syntax:

```
%FWRITE:<filehandle>[,<length>[,<timeout>]]
```

The test command parameters and their defined values are the following:

<filehandle>

Integer, 1–15. File handle.

<length>

Integer, 0–10240. Maximum number of bytes to accept in data mode. The default is `1000`.

<timeout>

Integer, 0–100. Waiting time for input data in seconds. The default is `5`.

The following command example returns the command syntax:

```
AT%FWRITE=?
%FWRITE:<filehandle>[,<length>[,<timeout>]]

OK
```

6.4 Delete file %FDELETE

The proprietary **%FDELETE** command deletes a file referenced by its file handle. v1.4.x

6.4.1 Set command

The set command deletes a file referenced by its file handle or all files.

Syntax for deleting a file referenced by its file handle:

```
%FDELETE=<filehandle>
```

Syntax for deleting all files:

```
%FDELETE="all"
```

%FILE ERROR

PARAMETER ERROR – The command is malformed or the file handle is invalid or out of range.

DELETE ERROR – The filesystem failed to remove the file.

The set command parameters and their defined values are the following:

<filehandle>

Integer, 1–15. File handle.

"all"

String. Closes all open file handles and deletes all associated files.

The following command example deletes the file associated with file handle `1`:

```
AT%FDELETE=1
OK
```

6.4.2 Read command

The read command is not supported.

6.4.3 Test command

The test command returns the command syntax.

Response syntax:

```
%FDELETE:(<filehandle>,"all")
```

The test command parameters and their defined values are the following:

<filehandle>

Integer, 1–15. File handle.

"all"

String. Closes all open file handles and deletes all associated files.

The following command example returns the command syntax:

```
AT%FDELETE=?
%FDELETE:(<filehandle>,"all")
OK
```

6.5 Close file %FCLOSE

The proprietary **%FCLOSE** command closes an open file referenced by its file handle. v1.4.x

6.5.1 Set command

The set command closes an open file.

Syntax:

```
%FCLOSE=<filehandle>
```

%FILE ERROR

PARAMETER ERROR – The command is malformed or the file handle is invalid or out of range.

CLOSE ERROR – The filesystem failed to close the file.

The set command parameter and its defined values are the following:

<filehandle>

Integer, 1–15. File handle.

The following command example closes the file associated with file handle `1`:

```
AT%FCLOSE=1
OK
```

6.5.2 Read command

The read command is not supported.

6.5.3 Test command

The test command returns the command syntax.

Response syntax:

```
%FCLOSE:<filehandle>
```

The test command parameter and its defined values are the following:

<filehandle>

Integer, 1–15. File handle.

The following command example returns the command syntax:

```
AT%FCLOSE=?
%FCLOSE:<filehandle>

OK
```

6.6 SHA-256 hash of file %FSHA256

The proprietary **%FSHA256** command computes the SHA-256 hash of a file on the module's filesystem and returns it as a 64-character lowercase hexadecimal string. v1.4.x

6.6.1 Set command

The set command computes the SHA-256 hash of a file in the module's filesystem and returns it as a 64-character lowercase hexadecimal string.

Syntax:

```
%FSHA256=<filename>
```

Response syntax:

```
%FSHA256: <hash>
```

%FILE ERROR

PARAMETER ERROR – The command is malformed or out of range.

FILE NOT FOUND – The file does not exist.

The set command parameters and their defined values are the following:

<filename>

String. Maximum length 62 bytes. Name of the file to hash.

<hash>

String. SHA-256 hash as 64 lowercase hexadecimal characters.

The following command example computes the SHA-256 hash of a file:

```
AT%FSHA256="test.bin"
%FSHA256: 84d89877f0d4041efb6bf91a16f0248f2fd573e6af05c19f96bedb9f882f7882

OK
```

6.6.2 Read command

The read command is not supported.

6.6.3 Test command

The test command returns the command syntax.

Response syntax:

```
%FSHA256: <filename>
```

The test command parameter and its defined value are the following:

<filename>

String. Maximum length 62 bytes. Name of the file to hash.

The following command example returns the command syntax:

```
AT%FSHA256=?
%FSHA256: <filename>

OK
```

7 HTTP and HTTPS commands

The HTTP and HTTPS commands are for sending HTTP and HTTPS requests and reading responses.

7.1 Configure HTTP parameters %HTTPCFG

The proprietary **%HTTPCFG** command configures HTTP and HTTPS related parameters. v1.4.x

7.1.1 Set command

The set command configures an HTTP or HTTPS parameter. If the optional value is omitted, the current configuration for that parameter is returned.

Syntax:

```
%HTTPCFG=<param>[,<value>]
```

Response syntax when value is omitted:

```
%HTTPCFG: <param>,<value>
```

The set command parameters and their defined values are the following:

<param>

String. Name of the configuration parameter.

<value>

Integer. Value for the parameter. The range depends on <param>.

The supported configuration parameters and their defined values are the following:

"contextid"

Integer, 1–15. PDP context ID. The default is `1`.

"requestheader"

Integer, 0–1. Enable or disable custom HTTP request headers.

0 – Disabled (default).

1 – Enabled.

"responseheader"

Integer, 0–1. Enable or disable output of HTTP response headers.

0 – Disabled (default).

1 – Enabled.

"sslctxid"

Integer, 0–5. SSL context ID. The default is `0`.

"contenttype"

Integer, 0–3. HTTP content type.

0 – application/x-www-form-urlencoded (default).

1 – text/plain.

2 – application/octet-stream.

3 – multipart/form-data.

The following command example sets the PDP context ID to 2:

```
AT%HTTPCFG="contextid",2
OK
```

7.1.2 Read command

The read command returns the current setting of each HTTP and HTTPS configuration parameter.

Response syntax:

```
%HTTPCFG: <param>,<value>
```

The read command parameters and their defined values are the following:

<param>

String. Name of the configuration parameter.

<value>

Integer. Value for the parameter. The range depends on <param>.

The following command example returns the current HTTP and HTTPS configuration:

```
AT%HTTPCFG?
%HTTPCFG: "contextid",1
%HTTPCFG: "requestheader",0
%HTTPCFG: "responseheader",0
%HTTPCFG: "sslctxid",0
%HTTPCFG: "contenttype",0

OK
```

7.1.3 Test command

The test command returns supported values as a compound value for each parameter.

Response syntax:

```
%HTTPCFG: "<param>":(range)
```

The test command parameter and its defined value are the following:

<param>

String. Name of the configuration parameter.

The following command example returns the supported parameters and their value ranges:

```
AT%HTTPCFG=?
%HTTPCFG: "contextid":(1-15)
%HTTPCFG: "requestheader":(0-1)
%HTTPCFG: "responseheader":(0-1)
%HTTPCFG: "sslctxid":(0-5)
%HTTPCFG: "contenttype":(0-3)

OK
```

7.2 Set HTTP URL %HTTPURL

The proprietary **%HTTPURL** command sets the URL for HTTP and HTTPS requests. v1.4.x

7.2.1 Set command

The set command sets the URL for HTTP and HTTPS requests. The module switches to data mode to receive the URL.

Syntax:

```
%HTTPURL=<URL_length>[,<timeout>]
```

Response syntax:

```
CONNECT
```

The set command parameters and their defined values are the following:

<URL_length>

Integer, 1–2048. Length of URL in bytes.

When the input data reaches <URL_length>, the module switches back to command mode.

<timeout>

Integer, 1–65535. Maximum input time for the URL in seconds. The default is `60`.

The following command example sets the URL for an HTTP `GET` request:

```
AT%HTTPURL=22,60
CONNECT
http://httpbin.org/get
OK
```

7.2.2 Read command

The read command returns the currently configured URL.

Response syntax:

```
%HTTPURL: <URL>
```

The read command parameter and its defined value are the following:

<URL>

String. Currently configured URL.

The following command example returns the currently configured URL:

```
AT%HTTPURL?
%HTTPURL: http://httpbin.org/get

OK
```

7.2.3 Test command

The test command returns the supported ranges for URL length and timeout.

Response syntax:

```
%HTTPURL: (range of supported <URL_length>),(range of supported <timeout>)
```

The test command parameters and their defined values are the following:

<URL_length>

Integer, 1–2048. Length of URL in bytes.

When the input data reaches <URL_length>, the module switches back to command mode.

<timeout>

Integer, 1–65535. Maximum input time for the URL in seconds. The default is `60`.

The following command example returns the supported ranges:

```
AT%HTTPURL=?
%HTTPURL: (1-2048),(1-65535)

OK
```

7.3 HTTP GET request %HTTPGET

The proprietary **%HTTPGET** command sends an HTTP `GET` request to the configured URL. v1.4.x

7.3.1 Set command

The set command sends an HTTP `GET` request to the configured URL.

If the "requestheader" parameter is set to `0` with the **%HTTPCFG** command, the **%HTTPGET** command takes no parameters. If the "requestheader" parameter is set to `1`, the module switches to data mode to receive custom request headers.

For HTTP and HTTPS error codes and server response codes, see HTTP and HTTPS error codes on page 59.

Syntax when "`requestheader`" is set to `0`:

```
%HTTPGET
```

Syntax when "requestheader" is set to 1:

```
%HTTPGET=<data_length>[,<input_time>]
```

The set command parameters and their defined values are the following:

<data_length>

Integer, 1–2048. Length of HTTP request headers in bytes.

<input_time>

Integer, 1–65535. Maximum input time for HTTP request headers in seconds. The default is 60.

After the response is received, the following *URC* is sent:

```
%HTTPGET: <err>,<httprspcode>,<content_length>
```

<content_length>

Integer. Length of the HTTP response's body in bytes.

The following command example sends a GET request with default request headers:

```
AT%HTTPGET
OK
```

7.3.2 Read command

The read command is not supported.

7.3.3 Test command

The test command returns the supported ranges for the length and input time of the request headers.

Response syntax:

```
%HTTPGET: (range of supported <data_length>),(range of supported <input_time>)
```

The test command parameters and their defined values are the following:

<data_length>

Integer, 1–2048. Length of HTTP request headers in bytes.

<input_time>

Integer, 1–65535. Maximum input time for HTTP request headers in seconds. The default is 60.

The following command example returns the supported ranges:

```
AT%HTTPGET=?
%HTTPGET: (1-2048),(1-65535)

OK
```

7.4 HTTP range GET request %HTTPGETEX

The proprietary **%HTTPGETEX** command sends a range GET request to retrieve data from a specified position and length from the HTTP or HTTPS server. v1.4.x

7.4.1 Set command

The set command sends a range GET request to retrieve data from a specified position and length. The command can be used only when the "requestheader" parameter is set to 0 with the **%HTTPCFG** command.

Syntax:

```
%HTTPGETEX=<start_position>,<read_len>
```

The set command parameters and their defined values are the following:

<start_position>

Integer. Start position of data to retrieve starting from 0.

<read_len>

Integer. Length of data to retrieve.

The following command example sends a range GET request for the first 50 bytes:

```
AT%HTTPGETEX=0,50
OK
```

7.4.2 Read command

The read command is not supported.

7.4.3 Test command

The test command returns the command syntax.

Response syntax:

```
%HTTPGETEX: <start_position>,<read_len>
```

The test command parameters and their defined values are the following:

<start_position>

Integer. Start position of data to retrieve starting from 0.

<read_len>

Integer. Length of data to retrieve.

The following command example returns the command syntax:

```
AT%HTTPGETEX=?
%HTTPGETEX: <start_position>,<read_len>

OK
```

7.5 HTTP POST request %HTTPPOST

The proprietary **%HTTPPOST** command sends an HTTP POST request with data entered through the serial interface. v1.4.x

7.5.1 Set command

The set command sends an HTTP POST request. The module switches to data mode to receive the request data.

Syntax:

```
%HTTPPOST=<data_length>[,<input_time>]
```

Response syntax:

```
CONNECT
```

The set command parameters and their defined values are the following:

<data_length>

Integer, 1–4096. Length of POST request data in bytes.

If the "requestheader" parameter is set to 0 with the **%HTTPCFG** command, the input data includes the body of the POST request.

If the "requestheader" is set to 1 with the **%HTTPCFG** command, the input data includes the headers and body of the POST request.

When the data length reaches <data_length>, the module switches back to command mode.

<input_time>

Integer, 1–65535. Maximum input time for the POST request data in seconds. The default is 60.

After the response is received, the following *URC* is sent:

```
%HTTPPOST: <err>,<httprspcode>[,<content_length>]
```

<content_length>

Integer. HTTP response content length in bytes.

The following command example sends a POST request with 13 bytes of body data:

```
AT%HTTPPOST=13,60
CONNECT
Hello, World!
OK
```

7.5.2 Read command

The read command is not supported.

7.5.3 Test command

The test command returns the supported ranges for data length and input time.

Response syntax:

```
%HTTPPOST: (range of supported <data_length>),(range of supported <input_time>)
```

The test command parameters and their defined values are the following:

<data_length>

Integer, 1–4096. Length of `POST` request data in bytes.

If the "`requestheader`" parameter is set to `0` with the **`%HTTPCFG`** command, the input data includes the body of the `POST` request.

If the "`requestheader`" is set to `1` with the **`%HTTPCFG`** command, the input data includes the headers and body of the `POST` request.

When the data length reaches <data_length>, the module switches back to command mode.

<input_time>

Integer, 1–65535. Maximum input time for the `POST` request data in seconds. The default is `60`.

The following command example returns the supported ranges:

```
AT%HTTPPOST=?
%HTTPPOST: (1-4096),(1-65535)

OK
```

7.6 HTTP POST request from file %HTTPPOSTFILE

The proprietary **`%HTTPPOSTFILE`** command sends an HTTP `POST` request with the content of a file.
v1.4.x

7.6.1 Set command

The set command sends an HTTP `POST` request with the content of a file.

If the "`requestheader`" parameter is set to `0` with the **`%HTTPCFG`** command, the file contains the body of the `POST` request. If the "`requestheader`" is set to `1`, the file contains the headers and body of the `POST` request.

For HTTP and HTTPS error codes and server response codes, see HTTP and HTTPS error codes on page 59.

Syntax:

```
%HTTPPOSTFILE=<file_name>
```

The set command parameter and its defined values are the following:

<file_name>

String. Maximum length 62 bytes. File name.

After the response is received, the following *URC* is sent:

```
%HTTPPOSTFILE: <err>,<httprspcode>[,<content_length>]
```

<content_length>

Integer. Length of HTTP response content in bytes.

The following command example sends a `POST` request from a file:

```
AT%HTTPPOSTFILE="data.txt"
OK
```

7.6.2 Read command

The read command is not supported.

7.6.3 Test command

The test command returns the command syntax.

Response syntax:

```
%HTTPPOSTFILE: <file_name>
```

The test command parameter and its defined value are the following:

<file_name>

String. Maximum length 62 bytes. File name.

The following command example returns the command syntax:

```
AT%HTTPPOSTFILE=?
%HTTPPOSTFILE: <file_name>

OK
```

7.7 Read HTTP response %HTTPREAD

The proprietary **%HTTPREAD** command reads the response from the HTTP or HTTPS server through the serial interface. v1.4.x

7.7.1 Set command

The set command reads the HTTP or HTTPS server's response through the serial interface. This command is used after the **%HTTPGET**, **%HTTPGETEX**, **%HTTPPOST**, or **%HTTPPOSTFILE** command.

Syntax:

```
%HTTPREAD
```

If the connection is successful, the modem responds with `CONNECT` and outputs the response data. If the `"responseheader"` parameter is set to `1` with the **%HTTPCFG** command, the response includes the HTTP response headers.

The following command example sets a URL, sends a `GET` request, and reads the response:

```
AT%HTTPURL=22,60
CONNECT
http://httpbin.org/get
OK

AT%HTTPGET
OK

AT%HTTPREAD
OK

CONNECT

%HTTPGET: 0,200,226
{
"args": {},
"headers": {
"Content-Length": "0",
"Host": "httpbin.org"
},
"origin": "85.76.162.158",
"url": "http://httpbin.org/get"
}

%HTTPGET: 100,200,226
```

7.7.2 Read command

The read command is not supported.

7.7.3 Test command

The test command is not supported.

7.8 Read HTTP response to file %HTTPREADFILE

The proprietary **%HTTPREADFILE** command saves the HTTP or HTTPS server's response to a file. v1.4.x

7.8.1 Set command

The set command saves the HTTP or HTTPS server's response to a file.

For HTTP and HTTPS error codes and server response codes, see HTTP and HTTPS error codes on page 59.

Syntax:

```
%HTTPREADFILE=<file_name>
```

The set command parameter and its defined value are the following:

<file_name>

String. Maximum length 62 bytes. File name.

The following command example saves the response to a file:

```
AT%HTTPREADFILE="response.txt"
OK

%HTTPGET: 0,200,226

%HTTPGET: 100,200,226
```

7.8.2 Read command

The read command is not supported.

7.8.3 Test command

The test command returns the command syntax.

Response syntax:

```
%HTTPREADFILE: <file_name>
```

The test command parameter and its defined value are the following:

<file_name>

String. Maximum length 62 bytes. File name.

The following command example returns the command syntax:

```
AT%HTTPREADFILE=?
%HTTPREADFILE: <file_name>

OK
```

7.9 Cancel HTTP request %HTTPSTOP

The proprietary **%HTTPSTOP** command cancels the current HTTP `GET` or `POST` request and disconnects from the HTTP or HTTPS server. v1.4.x

7.9.1 Set command

The set command cancels the current HTTP request and disconnects from the HTTP or HTTPS server.

For HTTP and HTTPS error codes and server response codes, see HTTP and HTTPS error codes on page 59.

Syntax:

```
%HTTPSTOP
```

The following command example cancels the current HTTP request:

```
AT%HTTPSTOP
OK
```

7.9.2 Read command

The read command is not supported.

7.9.3 Test command

The test command is not supported.

7.10 Firmware upgrade through HTTP or HTTPS %HTTPFOTADL

The proprietary **%HTTPFOTADL** command downloads a firmware upgrade package from an HTTP or HTTPS server and triggers a firmware upgrade. v1.4.x

7.10.1 Set command

The set command downloads a firmware upgrade package from an HTTP or HTTPS server and triggers a firmware upgrade.

For HTTP and HTTPS error codes and server response codes, see HTTP and HTTPS error codes on page 59.

Syntax:

```
%HTTPFOTADL=<HTTP_URL>[,<download_URC_max>]
```

The set command parameters and their defined values are the following:

<HTTP_URL>

String. Maximum length 255 characters. URL of the firmware upgrade package. Must start with `http://` or `https://`.

<download_URC_max>

Integer, 0 and 50–100. Maximum download progress reporting.

0 – Do not report download progress.

50–100 – Maximum download progress percentage to report.

During download, the following *URC*s are sent:

```
%HTTPURC: "FOTA","HTTPSTART"
%HTTPURC: "FOTA","DOWNLOADING",<percent>
%HTTPURC: "FOTA","DOWNLOADED"
```

If the download fails, one of the following URCs is sent instead:

```
%HTTPURC: "FOTA","DOWNLOADERROR"
%HTTPURC: "FOTA","PACKAGE MISMATCH"
```

<percent>

Integer, 0–100. Download progress.

After the download completes successfully, the module reboots and applies the delta update during the bootloader phase.

The following command example downloads a firmware upgrade package with progress reporting:

```
AT%HTTPFOTADL="http://example.com:8080/delta.binpkg",100
OK
%HTTPURC: "FOTA","HTTPSTART"
%HTTPURC: "FOTA","DOWNLOADING",10
%HTTPURC: "FOTA","DOWNLOADING",30
%HTTPURC: "FOTA","DOWNLOADING",50
%HTTPURC: "FOTA","DOWNLOADING",80
%HTTPURC: "FOTA","DOWNLOADING",100
%HTTPURC: "FOTA","DOWNLOADED"
```

7.10.2 Read command

The read command is not supported.

7.10.3 Test command

The test command is not supported.

7.11 HTTP and HTTPS error codes

The HTTP and HTTPS commands return `%HTTP ERROR: <err>` when an error occurs.

The error codes and their descriptions are the following:

<err>

Integer. Error code.

0 – Operation successful.

1 – Parameter error.

2 – Input timeout.

3 – URL is not set.

4 – Operation not supported.

5 – No GET/POST requests.

6 – Request processing.

7 – Filename too long.

8 – File not found.

9 – Memory not enough.

10 – SSL configuration error.

11 – URL parse error.

12 – DNS resolution failed.

13 – Protocol error.

14 – Socket error.

15 – PDP context bind failed.

16 – Socket connect timeout.

17 – Socket connect error.

18 – Socket connect close.

19 – SSL error.

20 – Request timeout.

21 – Internal error.

22 – URL format error.

100 – Data receive complete.

The HTTP and HTTPS server response codes are the following:

<httprspcode>

Integer. HTTP response code.

200 – OK.

400 – Bad request.

403 – Forbidden.

404 – Not found.

405 – Method not allowed.

409 – Conflict.

411 – Length required.

500 – Internal server error.

8 Cloud commands

The cloud commands are for device registration, authentication, communication, firmware updates, and cloud-based services with *nRF Cloud*.

8.1 Read device UUID %DEVICEUUID

The proprietary **%DEVICEUUID** command returns the device-specific *UUID*. v1.4.x

8.1.1 Set command

The set command returns the 36-character device *UUID*. If the UUID does not exist, one is created and stored in *NVM*.

Syntax:

```
%DEVICEUUID
```

Response syntax:

```
%DEVICEUUID: <uuid>
```

The set command parameter and its defined value are the following:

<uuid>

String. 36-character device *UUID* in the following format: `xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx`.

If the UUID does not exist in NVM, the module creates one and returns `%DEVICEUUID: creating device uuid...` before the UUID.

The following command example returns the device UUID:

```
AT%DEVICEUUID
%DEVICEUUID: xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx

OK
```

8.1.2 Read command

The read command is not supported.

8.1.3 Test command

The test command is not supported.

8.2 Cloud access public key %CLOUDACCESSKEY

The proprietary **%CLOUDACCESSKEY** command returns the device's cloud access public key in Base64-encoded *DER* format. v1.4.x

8.2.1 Set command

The set command returns the device's cloud access public key as a Base64-encoded *DER* string. If a key pair does not exist, a P-256 ECC key pair is generated and stored in NVM.

Syntax:

```
%CLOUDACCESSKEY
```

Response syntax:

```
%CLOUDACCESSKEY: <public_key_der_base64>
```

The set command parameter and its defined value are the following:

<public_key_der_base64>

String. P-256 ECC public key in Base64-encoded DER format. The key is uploaded to the cloud service for device registration.

If a key pair does not exist in NVM, the modem generates one and returns `%CLOUDACCESSKEY: creating device keys...` before the public key.

The following command example returns the cloud access public key:

```
AT%CLOUDACCESSKEY
%CLOUDACCESSKEY: xxxxx...xxxxx==

OK
```

8.2.2 Read command

The read command is not supported.

8.2.3 Test command

The test command returns the command syntax.

Response syntax:

```
%CLOUDACCESSKEY: <public_key_der_base64>
```

The test command parameter and its defined value are the following:

<public_key_der_base64>

String. P-256 ECC public key in Base64-encoded DER format. The key is uploaded to the cloud service for device registration.

The following command example returns the command syntax:

```
AT%CLOUDACCESSKEY=?
%CLOUDACCESSKEY: <public_key_der_base64>

OK
```

8.3 Registration JWT %REGJWT

The proprietary **%REGJWT** command generates a signed *JSON Web Token (JWT)* for device registration with a cloud service. v1.4.x

8.3.1 Set command

The set command generates a *JWT* signed with the device private key. The JWT payload contains the device *UUID* as `"sub"` and the provided identifier as `"id"`.

Before using the command, execute the **%DEVICEUUID** command to ensure the device UUID exists and the **%CLOUDACCESSKEY** command to ensure the signing key pair exists.

Syntax:

```
%REGJWT=<ID>
```

Response syntax:

```
%REGJWT: <jwt>
```

The set command parameters and their defined values are the following:

<ID>

String, 1–127 characters. Identifier.

<jwt>

String. The generated JWT in `header.payload.signature` format.

The following command example generates a registration JWT:

```
AT%REGJWT="test"
%REGJWT: xxxxx.xxxxx.xxxxx

OK
```

8.3.2 Read command

The read command is not supported.

8.3.3 Test command

The test command returns the command syntax.

Response syntax:

```
%REGJWT:<ID>
```

The test command parameter and its defined value are the following:

<ID>

String, 1–127 characters. Identifier.

The following command example returns the command syntax:

```
AT%REGJWT=?
%REGJWT:<ID>

OK
```

8.4 Creating JWT %JWT

The proprietary **%JWT** command generates a short-lived signed *JWT* for authenticating with cloud APIs. v1.4.x

8.4.1 Set command

The set command generates a *JWT* signed with the device private key. The JWT payload contains the device *UUID* as "sub" and an expiration time as "exp". The token expires after one hour.

Before using this command, execute the **%DEVICEUUID** command to ensure the device UUID exists, execute the **%CLOUDACCESSKEY** command to ensure the signing key pair exists, and verify that the modem has a valid system time.

Syntax:

```
%JWT
```

Response syntax:

```
%JWT: <jwt>
```

The set command parameter and its defined value are the following:

<jwt>

String. The generated JWT in header.payload.signature format.

The following command example generates a JWT:

```
AT%JWT
%JWT: xxxxx.xxxxx.xxxxx

OK
```

8.4.2 Read command

The read command is not supported.

8.4.3 Test command

The test command is not supported.

8.5 Cloud location %NRFCLOUDLOCATION

The proprietary **%NRFCLOUDLOCATION** command triggers a cloud-based location flow. v1.4.x

8.5.1 Set command

The set command starts a cloud-based location flow.

Syntax:

```
%NRFCLOUDLOCATION=<method>,<fetch_result>
```

The set command parameters and their defined values are the following:

<method>

Integer, 1–7. Data collection method.

1 – Single-cell.

2 – Multicell.

3 – Single-cell and multicell.

4 – Wi-Fi only.

5 – Single-cell and Wi-Fi.

6 – Multicell and Wi-Fi.

7 – All.

<fetch_result>

Integer, 0–1. Specifies whether the modem reports a location result after sending the collected data to *nRF Cloud*.

0 – Report an acknowledgment without returning a location result.

1 – Report the location result.

When <fetch_result> is `0`, a *URC* is sent with an acknowledgment, such as the following:

```
%NRFCLOUDLOCATION: ACK RECEIVED
```

When <fetch_result> is `1`, a URC is sent with the location result:

```
%NRFCLOUDLOCATION: <lat>,<lon>,<unc>,<fulfilled_method>
```

The response parameters and their defined values are the following:

<lat>

String. Latitude.

<lon>

String. Longitude.

<unc>

Integer. Uncertainty in meters.

<fulfilled_method>

Integer. Method used by the cloud to determine the location.

The fulfilled method can differ from the requested <method> when multiple data sources are allowed.

1 – Single-cell.

2 – Multicell.

4 – Wi-Fi.

Note: During the cloud transaction, unsolicited **%COAP** result codes can be emitted.

Warning: Multicell and Wi-Fi positioning do not work if the module is in RRC Connected mode.

The following command example collects single-cell data without sending it to the cloud:

```
AT%NRFCLOUDLOCATION=1,0
OK
%COAP: RESPONSE,2.05
%NRFCLOUDLOCATION: ACK RECEIVED
```

The following command example collects single-cell data and requests a location from the cloud:

```
AT%NRFCLOUDLOCATION=1,1
OK
%COAP: RESPONSE,2.05
%COAP: DATA,HEX,bf01fb404ebf3c5f2bdf8202fb4037c68853f6f81a03186404655343454c4cff
%NRFCLOUDLOCATION: 61.494029,23.775517,100,1
```

The following command example allows single-cell and multicell positioning. The cloud fulfills the request with multicell positioning:

```
AT%NRFCLOUDLOCATION=3,1
OK
%COAP: RESPONSE,2.05
%COAP: DATA,HEX,bf01fb404ebf3c5f2bdf8202fb4037c68853f6f81a03186404654d43454c4cff
%NRFCLOUDLOCATION: 61.494029,23.775517,100,2
```

8.5.2 Read command

The read command is not supported.

8.5.3 Test command

The test command returns the supported parameter ranges.

Response syntax:

```
%NRFCLOUDLOCATION: (range of supported <method>),(range of supported <fetch_result>)
```

The test command parameters and their defined values are the following:

<method>

Integer, 1–7. Data collection method.

1 – Single-cell.

2 – Multicell.

3 – Single-cell and multicell.

4 – Wi-Fi only.

5 – Single-cell and Wi-Fi.

6 – Multicell and Wi-Fi.

7 – All.

<fetch_result>

Integer, 0–1. Specifies whether the modem reports a location result after sending the collected data to *nRF Cloud*.

0 – Report an acknowledgment without returning a location result.

1 – Report the location result.

The following command example returns the supported parameter ranges:

```
AT%NRFCLOUDLOCATION=?
%NRFCLOUDLOCATION: (1-7),(0-1)

OK
```

8.6 Cloud message %NRFCLOUDMESSAGE

The proprietary **%NRFCLOUDMESSAGE** command sends a JSON message to *nRF Cloud* using CoAP. v1.4.x

8.6.1 Set command

The set command sends a JSON message to *nRF Cloud*.

Syntax:

```
%NRFCLOUDMESSAGE=<message>
```

Response syntax:

```
%NRFCLOUDMESSAGE: SENT
```

The set command parameter and its defined value are the following:

<message>

String. Maximum length 511 characters. JSON object or JSON array sent as `application/json`.

Note: After the command is accepted, the server response is reported separately with unsolicited **%COAP** result codes.

The following command example sends a device-to-cloud (D2C) `BUTTON` *Application Identifier (appId)* message with data `1` to nRF Cloud:

```
AT%NRFCLOUDMESSAGE={"appId":"BUTTON","data":"1"}
%NRFCLOUDMESSAGE: SENT
OK
%COAP: RESPONSE,2.01
```

8.6.2 Read command

The read command is not supported.

8.6.3 Test command

The test command returns the command syntax.

Response syntax:

```
%NRFCLOUDMESSAGE: <message>
```

The test command parameter and its defined value are the following:

<message>

String. Maximum length 511 characters. JSON object or JSON array sent as `application/json`.

The following command example returns the command syntax:

```
AT%NRFCLOUDMESSAGE=?
%NRFCLOUDMESSAGE: <message>

OK
```

8.7 Cloud shadow %NRFCLOUDSHADOW

The proprietary **%NRFCLOUDSHADOW** command gets or sets a device shadow value in *nRF Cloud*. v1.4.x

8.7.1 Set command

The set command gets or sets a device shadow value in *nRF Cloud*.

Syntax:

```
%NRFCLOUDSHADOW=<path>[,<value>]
```

Response syntax:

```
%NRFCLOUDSHADOW: <path>,<value>
```

The set command parameters and their defined values are the following:

<path>

String. Maximum length 127 characters. Dotted shadow path.

<value>

String. Maximum length 255 characters. Value to write.

When only <path> is provided, the module reads the desired shadow value from `state/desired`.

When both <path> and <value> are provided, the module sends the value into `state/reported`.

> **Note:** After the command is accepted, the server response is reported separately with unsolicited **%COAP** result codes.

The following command example sets a Boolean value:

```
AT%NRFCLOUDSHADOW=config.switch,true
OK
%COAP: RESPONSE,2.05
%COAP: DATA,2,{}
```

The following command example sets a string value:

```
AT%NRFCLOUDSHADOW=config.name,Vlad
OK
%COAP: RESPONSE,2.05
%COAP: DATA,2,{}
```

The following command example reads a desired shadow value:

```
AT%NRFCLOUDSHADOW=config.name
OK
%COAP: RESPONSE,2.05
%COAP: DATA,6,"Lisa"
```

8.7.2 Read command

The read command is not supported.

8.7.3 Test command

The test command returns the command syntax.

Response syntax:

```
%NRFCLOUDSHADOW: <path>[,<value>]
```

The test command parameters and their defined values are the following:

<path>

String. Maximum length 127 characters. Dotted shadow path.

<value>

String. Maximum length 255 characters. Value to write.

The following command example returns the command syntax:

```
AT%NRFCLOUDSHADOW=?
%NRFCLOUDSHADOW: <path>[,<value>]

OK
```

8.8 Cloud FOTA %NRFCLOUDFOTA

The proprietary **%NRFCLOUDFOTA** command checks for host application or modem firmware updates in *nRF Cloud*. It can also retrieve host *Over-the-Air (OTA)* chunks or apply a modem update. v1.4.x

8.8.1 Set command

The set command checks for firmware updates in *nRF Cloud*, downloads host *OTA* chunks, or applies a modem firmware update based on the selected mode.

Syntax for <mode> 0:

```
%NRFCLOUDFOTA=<mode>,<projectKey>,<hwVer>,<swType>,<swVer>
```

Syntax for <mode> 1:

```
%NRFCLOUDFOTA=<mode>,<chunk_idx>
```

Syntax for <mode> 2 or 3:

```
%NRFCLOUDFOTA=<mode>,<projectKey>
```

Syntax for <mode> 4:

```
%NRFCLOUDFOTA=<mode>
```

The set command parameters and their defined values are the following:

<mode>

Integer, 0–4. Operation mode.

0 – Check for a host application OTA update and cache the download URL.

1 – Download one host OTA chunk from the cached URL.

2 – Check for a modem firmware update.

3 – Check for and apply a modem firmware update.

4 – Clear the cached host OTA download URL.

<projectKey>

String. Maximum length 47 characters. Project key used by the cloud proxy.

<hwVer>

String. Maximum length 31 characters. Hardware version reported for host application OTA checks.

<swType>

String. Maximum length 31 characters. Software type reported for host application OTA checks.

<swVer>

String. Maximum length 47 characters. Current software version reported for host application OTA checks.

<chunk_idx>

Integer. Zero-based host OTA chunk index.

When <mode> is 1 or 2, the modem reports an unsolicited result code:

```
%FOTA: <base64>,<crc16>,<has_more>
```

The host OTA chunk response parameters and their defined values are the following:

<base64>

String. Standard Base64 encoding of the host OTA chunk payload.

<crc16>

String. CRC-16/CCITT-FALSE of the raw host OTA chunk payload encoded as hexadecimal.

<has_more>

Integer, 0–1. Indicates whether more host OTA chunks are available.

0 – The final chunk.

1 – Additional chunks are available.

Note: During the cloud transaction, unsolicited **%COAP** result codes can be emitted.

The following command example checks for a host application OTA update and caches the download URL:

```
AT%NRFCLOUDFOTA=0,IZAAAFTDABsTQEAAEQDCIEw,0.3,app,1.0.0
OK
%FOTA: CHECKING
%COAP: RESPONSE,2.05
%COAP: DATA,153,{"data":{"url":"https://ota-cdn.memfault.com/..."}}
%FOTA: UPDATE_AVAILABLE
```

The following command example downloads the first host OTA chunk:

```
AT%NRFCLOUDFOTA=1,0
OK
%FOTA: SGVsbG8=,D26E,1
```

The following command example checks whether a modem firmware update is available:

```
AT%NRFCLOUDFOTA=2,IZAAAFTDABsTQEAAEQDCIEw
OK
%FOTA: CHECKING
%FOTA: UPDATE_AVAILABLE
```

The following command example checks for and applies a modem firmware update:

```
AT%NRFCLOUDFOTA=3,IZAAAFTDABsTQEAAEQDCIEw
OK
%FOTA: CHECKING
%FOTA: ZONE,1,458752
%FOTA: ERASING
%FOTA: DOWNLOADING
%FOTA: DOWNLOAD_DONE,53248
%FOTA: VERIFYING
%FOTA: VERIFIED
%FOTA: UPGRADING
```

The following command example clears the cached host OTA download URL:

```
AT%NRFCLOUDFOTA=4
OK
```

8.8.2 Read command

The read command is not supported.

8.8.3 Test command

The test command returns the command syntax.

Response syntax:

```
%NRFCLOUDFOTA: (0,<projectKey>,<hwVer>,<swType>,<swVer>)|(1,<chunk_idx>)|(2,<projectKey>)|
(3,<projectKey>)|(4)
```

The test command parameters and their defined values are the following:

<mode>

Integer, 0–4. Operation mode.

0 – Check for a host application OTA update and cache the download URL.

1 – Download one host OTA chunk from the cached URL.

2 – Check for a modem firmware update.

3 – Check for and apply a modem firmware update.

4 – Clear the cached host OTA download URL.

<projectKey>

String. Maximum length 47 characters. Project key used by the cloud proxy.

<hwVer>

String. Maximum length 31 characters. Hardware version reported for host application OTA checks.

<swType>

String. Maximum length 31 characters. Software type reported for host application OTA checks.

<swVer>

String. Maximum length 47 characters. Current software version reported for host application OTA checks.

<chunk_idx>

Integer. Zero-based host OTA chunk index.

The following command example returns the command syntax:

```
AT%NRFCLOUDFOTA=?
%NRFCLOUDFOTA: (0,<projectKey>,<hwVer>,<swType>,<swVer>)|(1,<chunk_idx>)|(2,<projectKey>)|
(3,<projectKey>)|(4)

OK
```

8.9 Memfault chunk upload %NRFCLOUDOBSUPLOAD

The proprietary **%NRFCLOUDOBSUPLOAD** command drains the on-device Memfault chunk queue and uploads each chunk to the Memfault ingestion endpoint through CoAP. v1.4.x

8.9.1 Set command

The set command drains the on-device Memfault chunk queue and uploads all pending chunks to the Memfault ingestion endpoint through CoAP.

Syntax:

```
%NRFCLOUDOBSUPLOAD[=<projectKey>]
```

The set command parameter and its defined values are the following:

<projectKey>

String. Maximum length 127 characters. Memfault project key. If omitted, the project key compiled into the firmware is used.

When the upload succeeds, the following response is returned:

```
%NRFCLOUDOBSUPLOAD: UPLOADED,<n>
OK
```

When the upload fails, the following response is returned:

```
%NRFCLOUDOBSUPLOAD: UPLOAD_ERROR,<coap_code>
ERROR
```

The response parameters and their defined values are the following:

<n>

Integer. Number of chunks uploaded in this invocation.

<coap_code>

Integer. CoAP response code returned on a failed upload. For a list of codes, see CoAP response codes on page 76.

Note: During the CoAP transaction, unsolicited **%COAP** result codes can be emitted.

The following command example uploads all pending Memfault chunks:

```
AT%NRFCLOUDOBSUPLOAD
%NRFCLOUDOBSUPLOAD: UPLOADED,3
OK
```

The following command example uploads chunks using a specific project key:

```
AT%NRFCLOUDOBSUPLOAD="myProjectKey"
%NRFCLOUDOBSUPLOAD: UPLOADED,1
OK
```

8.9.2 Read command

The read command is not supported.

8.9.3 Test command

The test command returns the command syntax.

Response syntax:

```
%NRFCLOUDOBSUPLOAD: ["<projectKey>"]
```

The test command parameter and its defined values are the following:

<projectKey>

String. Maximum length 127 characters. Memfault project key. If omitted, the project key compiled into the firmware is used.

The following command example returns the command syntax:

```
AT%NRFCLOUDOBSUPLOAD=?
%NRFCLOUDOBSUPLOAD: ["<projectKey>"]
OK
```

8.10 Memfault heartbeat %NRFCLOUDOBSHEARTBEAT

The proprietary **%NRFCLOUDOBSHEARTBEAT** command immediately triggers a Memfault metrics heartbeat event and serializes all currently accumulated metric values into the chunk queue. v1.4.x

8.10.1 Set command

The set command immediately triggers a Memfault metrics heartbeat event.

Syntax:

```
%NRFCLOUDOBSHEARTBEAT
```

The command has no parameters.

The following command example triggers a Memfault metrics heartbeat:

```
AT%NRFCLOUDOBSHEARTBEAT
OK
```

8.10.2 Read command

The read command is not supported.

8.10.3 Test command

The test command is not supported.

8.11 Memfault chunk forwarding %NRFCLOUDOBSFORWARD

The proprietary **%NRFCLOUDOBSFORWARD** command uploads a single pre-encoded Memfault chunk supplied by the host to the Memfault ingestion endpoint through CoAP. v1.4.x

8.11.1 Set command

The set command uploads a single pre-encoded Memfault chunk supplied by the host to the Memfault ingestion endpoint through CoAP.

Syntax:

```
%NRFCLOUDOBSFORWARD=<base64_chunk>
```

+CME ERROR code

50 – Incorrect parameters. Returned when the parameter is missing or the base64 string is invalid.

The set command parameter and its defined values are the following:

<base64_chunk>

String. Base64-encoded binary Memfault chunk from the host. The modem decodes the string and uploads the raw chunk to the Memfault ingestion endpoint over CoAP.

When the upload fails, the following response is returned:

```
%NRFCLOUDOBSFORWARD: FORWARD_ERROR,<coap_code>
ERROR
```

The response parameter and its defined value are the following:

<coap_code>

Integer. CoAP response code returned on a failed upload. For a list of codes, see CoAP response codes on page 76.

Note: During the CoAP transaction, unsolicited **%COAP** result codes can be emitted.

The following command example forwards a Memfault chunk to the ingestion endpoint:

```
AT%NRFCLOUDOBSFORWARD="AQIDBAU="
OK
```

8.11.2 Read command

The read command is not supported.

8.11.3 Test command

The test command returns the command syntax.

Response syntax:

```
%NRFCLOUDOBSFORWARD: "<base64_chunk>"
```

The test command parameter and its defined value are the following:

<base64_chunk>

String. Base64-encoded binary Memfault chunk from the host. The modem decodes the string and uploads the raw chunk to the Memfault ingestion endpoint over CoAP.

The following command example returns the command syntax:

```
AT%NRFCLOUDOBSFORWARD=?
%NRFCLOUDOBSFORWARD: "<base64_chunk>"
OK
```

8.12 CoAP response codes

CoAP response codes indicate the result of a CoAP request made by the modem to a cloud endpoint.

A CoAP response code is a single integer that encodes both a class and a detail value using the `integer = (class × 32) + detail` formula conventionally written as `c.dd`. A response is successful when the class is `2`. Any other class indicates an error.

To decode an integer code, divide it by 32 to get the class. The remainder is the detail. For example, `129 = (4 × 32) + 1` gives `4.01 Unauthorized`.

The <coap_code> parameter and its defined values are the following:

<coap_code>

Integer. CoAP response code.

-1 – No CoAP response received. Connection or transport failure before the server replied.

65 (2.01) – Created. Chunk accepted, new resource created.

68 (2.04) – Changed. Chunk accepted.

69 (2.05) – Content. Request succeeded with response body.

128 (4.00) – Bad request. Malformed chunk data.

129 (4.01) – Unauthorized. Invalid or missing project key.

131 (4.03) – Forbidden.

132 (4.04) – Not found. Wrong endpoint URL.

141 (4.13) – Request entity too large. Chunk exceeded server limit.

160 (5.00) – Internal server error.

163 (5.03) – Service unavailable. Memfault backend temporarily unavailable.

164 (5.04) – Gateway timeout.

9 Mobile termination control and status commands

Mobile termination control and status commands are used for mobile-terminated power and indicator handling. Commands are listed for accessing *Subscriber Identity Module (SIM)* or *Universal Integrated Circuit Card (UICC)* database records.

9.1 Functional mode +CFUN

The **+CFUN** command sets and reads the module's functional mode. v1.4.x

For reference, see *3GPP 27.007 Ch. 8.2*.

9.1.1 Set command

The set command sets the functional mode to Minimum functionality, Normal, or Offline (Flight) mode.

Setting the module to Minimum functionality or Offline mode might take some time if signaling with the network is needed. The command can also be used to reset the module and start up in Normal mode.

Syntax:

```
+CFUN=<fun>[,<rst>]
```

+CME ERROR code

3 – Operation not allowed.

4 – Operation not supported.

50 – Incorrect parameters.

The set command parameters and their defined values are the following:

<fun>

Integer, 0, 1, and 4. Modem functionality mode.

0 – Sets the module to minimum functionality. Disables transmit RF circuit and deactivates LTE services (default).

1 – Sets the module to full functionality.

4 – Sets the module to Flight mode. Disables both transmit and receive RF circuits and deactivates LTE services.

<rst>

Integer, 0–1. Reset the module after <cfun> change.

0 – Do not reset the module before setting it to <fun> functional mode (default).

1 – Reset the module and start up in Normal mode regardless of the current mode. <fun> is currently ignored.

Note: If a **%PALARM** voltage or temperature alarm is active, setting <fun> to 1 is rejected with LOW VOLT or HIGH THERM followed by +CME ERROR: 3. Wait for the recovery **%PALARM** *URC* before retrying. To disable this behavior, see the **%MODULECFG** command's "disableAlarm" option.

The following command example activates the module's Normal mode:

```
AT+CFUN=1
OK
```

The following command example resets the module and activates Normal mode:

```
AT+CFUN=1,1
OK
```

9.1.2 Read command

The read command reads the current functional mode.

Response syntax:

```
+CFUN: <fun>
```

The read command parameter and its defined value are the following:

<fun>

Integer, 0, 1, and 4. Modem functionality mode.

0 – Sets the module to minimum functionality. Disables transmit RF circuit and deactivates LTE services (default).

1 – Sets the module to full functionality.

4 – Sets the module to Flight mode. Disables both transmit and receive RF circuits and deactivates LTE services.

The following command example reads the current functional mode:

```
AT+CFUN?
+CFUN: 1
OK
```

9.1.3 Test command

The test command lists supported functional modes.

Response syntax:

```
+CFUN: (list of supported <fun>s),(list of supported <rst>s)
```

The response parameters and their defined values are the following:

<fun>

Integer, 0, 1, and 4. Modem functionality mode.

0 – Sets the module to minimum functionality. Disables transmit RF circuit and deactivates LTE services (default).

1 – Sets the module to full functionality.

4 – Sets the module to Flight mode. Disables both transmit and receive RF circuits and deactivates LTE services.

<rst>

Integer, 0–1. Reset the module after <cfun> change.

0 – Do not reset the module before setting it to <fun> functional mode (default).

1 – Reset the module and start up in Normal mode regardless of the current mode. <fun> is currently ignored.

The following command example returns the supported functional modes:

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

9.2 List all available AT commands +CLAC

The **+CLAC** command returns a list of all available AT commands. v1.4.x

For reference, see *3GPP 27.007 Ch. 8.37*.

9.2.1 Set command

The set command returns a list of all available AT commands.

Syntax:

```
+CLAC
```

Response syntax:

```
<AT Command1>[<CR><LF><AT Command2>[...]]
```

+CME ERROR code

518 – Not allowed in active state.

The following command example lists the supported AT commands:

```
AT+CLAC
AT+CFUN
AT+COPS
...
OK
```

9.2.2 Read command

The read command is not supported.

9.2.3 Test command

The test command is not supported.

9.3 Extended signal quality +CESQ

The **+CESQ** command returns received signal quality parameters. The command issues a valid response only when the modem is activated. v1.4.x

For reference, see *3GPP 27.007 Ch. 8.69*.

> **Note:** The command can be executed even when a *Universal Subscriber Identity Module (USIM)* is not inserted.

9.3.1 Set command

The set command returns received signal quality parameters.

Syntax:

```
+CESQ
```

Response syntax:

```
+CESQ: <rxlev>,<ber>,<rscp>,<ecno>,<rsrq>,<rsrp>
```

The set command parameters and their defined values are the following:

<rxlev>

Integer, 99. Received signal strength level.

99 – Not known or not detectable.

<ber>

Integer, 99. Channel bit error rate.

99 – Not known or not detectable.

<rscp>

Integer, 255. Received signal code power.

255 – Not known or not detectable.

<ecno>

Integer, 255. UMTS energy per chip to noise ratio.

255 – Not known or not detectable.

<rsrq>

Integer, 0–34 and 255. Reported *RSRQ* value.

0 – When measured RSRQ < −19.5 dB.

1 – When −19.5 dB ≤ measured RSRQ < −19 dB.

2 – When −19 dB ≤ measured RSRQ < −18.5 dB.

...

32 – When −4 dB ≤ measured RSRQ < −3.5 dB.

33 – When −3.5 dB ≤ measured RSRQ < −3 dB.

34 – When −3 dB ≤ measured RSRQ.

255 – Not known or not detectable.

The lower dB boundary of this range can be determined using the reported RSRQ value and the following formula: (reported RSRQ value – 40)/2=dB. For example, (32–40)/2=–4 dB. Thus, the range is –4 dB ≤ measured RSRQ < –3.5 dB.

<rsrp>

Integer, 0–97 and 255. Reported *RSRP* value.

0 – When measured RSRP < −140 dBm.

1 – When −140 dBm ≤ measured RSRP < −139 dBm.

2 – When −139 dBm ≤ measured RSRP < −138 dBm.

...

95 – When −46 dBm ≤ measured RSRP < −45 dBm.

96 – When −45 dBm ≤ measured RSRP < −44 dBm.

97 – When −44 dBm ≤ measured RSRP.

255 – Not known or not detectable.

The lower dB boundary of this range can be determined using the reported RSRP value and the following formula: reported RSRP value–141=dBm. For example, 95–141=–46 dBm. Thus, the range is –46 dBm ≤ measured RSRP < –45 dBm.

The following command example reads the current signal quality, mapped RSRQ 31, and RSRP 62:

```
AT+CESQ
+CESQ: 99,99,255,255,31,62
OK
```

9.3.2 Read command

The read command is not supported.

9.3.3 Test command

The test command returns supported values as compound values.

Response syntax:

```
+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)
```

The following command example returns supported values as compound values:

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

9.4 Band lock %BAND

The proprietary **%BAND** command configures band lock.

9.4.1 Set command

The set command configures the bands to be used.

The configuration is stored to *NVM* when the module is set to minimum functionality.

Syntax:

```
%BAND=<band1>[,<band2>,...,<band32>]
```

+CME ERROR code

50 – Incorrect parameters.

The set command parameter and its defined value are the following:

<band>

Integer, 0–255. Band number. If the first value is `0`, all available bands are used.

The following command example enables bands `5` and `8`:

```
AT%BAND=5,8
OK
```

9.4.2 Read command

The read command returns a list of enabled bands.

Response syntax:

```
%BAND: <band1>[,<band2>[,<band3>…]]
```

The read command parameter and its defined value are the following:

<band>

Integer, 0–255. Band number.

The following command example returns a list of enabled bands:

```
AT%BAND?
%BAND: 1,3,5,8,38,40,41
OK
```

9.4.3 Test command

The test command returns a list of all supported bands.

Response syntax:

```
%BAND: (list of supported <band>s)
```

The test command parameter and its defined value are the following:

<band>

Integer, 0–255. Band number.

The following command example returns a list of all supported bands:

```
AT%BAND=?
%BAND: (1,3,5,7,8,20,28,38,40,41)
```

9.5 Clock +CCLK

The **+CCLK** command sets or reads the real-time clock of the *Mobile Termination (MT)*. v1.4.x

For reference, see *3GPP 27.007 Ch. 8.15*.

9.5.1 Set command

The set command sets the real-time clock of the *MT*.

Syntax:

```
+CCLK=<time>
```

The set command parameter and its defined value are the following:

<time>

String. Current date and time in the format `"yyyy/MM/dd,hh:mm:ss±zz"`, where yyyy is the year, MM is the month, dd is the day, hh is the hour, mm is the minutes, ss is the seconds, and ±zz is the time zone offset expressed in quarters of an hour from GMT (range –96 to +96).

The following command example sets the clock to April 13, 2026 at 14:05:15 GMT+3:

```
AT+CCLK="2026/04/13,14:05:15+12"
OK
```

9.5.2 Read command

The read command returns the current setting of the clock.

Response syntax:

```
+CCLK: <time>
```

The read command parameter and its defined value are the following:

<time>

String. Current date and time in the format `"yy/MM/dd,hh:mm:ss±zz"`, where yy is the year (two last digits), MM is the month, dd is the day, hh is the hour, mm is the minutes, ss is the seconds, and ±zz is the time zone offset expressed in quarters of an hour from GMT (range –96 to +96).

The following command example reads the current clock:

```
AT+CCLK?
+CCLK: "26/04/13,14:05:15+12"
OK
```

9.5.3 Test command

The test command is not supported.

9.6 Module configuration %MODULECFG

The proprietary **%MODULECFG** command sets and reads module configuration parameters. v1.4.x

9.6.1 Set command

The set command sets a module configuration parameter.

The configuration is stored to *NVM* when the module is set to minimum functionality.

Syntax:

```
%MODULECFG=<option>,<value>
```

The set command parameters and their defined values are the following:

<option>

String. Name of the configuration parameter.

<value>

Integer. The value to set for the parameter.

The following configuration parameters are supported:

"faultAction"

Integer, 0–4. Hardfault action mode.

0 – Dump full exception info to flash and tool, then halt.

1 – Print necessary exception info, then reset.

2 – Dump full exception info to flash, then reset.

3 – Dump full exception info to flash and tool, then reset.

4 – Reset directly (default). Recommended for production.

"uartDumpPort"

Integer, 0–255. *Universal Asynchronous Receiver/Transmitter (UART)* port index for dump info output. The default is `1`.

255 – Disabled.

"startWDT"

Integer, 0–1. Watchdog timer mode. Takes effect after reset.

0 – Stop watchdog.

1 – Start watchdog (default).

"logCtrl"

Integer, 0–2. Log control mode. Takes effect after reset.

0 – Logging disabled.

1 – Software logging enabled.

2 – All logging enabled (default).

"logLevel"

Integer, 0–5. Log print level.

0 – Debug (default).

1 – Info.

2 – Value.

3 – Signal.

4 – Warning.

5 – Error.

"logOwnerId&logOwnerLevel"

Integer, 0–0xFFFFFF. Combined owner ID and level bitmask. Each owner uses 3 bits. The default is `0`.

"logBaudrate"

Integer, 921600–6000000. Log output baud rate. Takes effect after reset. The default is `6000000`.

"logPortSel"

Integer, 0–3. Log output port. Takes effect after reset.

0 – USB (default).

1 – UART0.

2 – USB and UART.

3 – Flash dump area.

"slpWaitTime"

Integer, 0–65535. Sleep wait time after wakeup. The default is `0`.

"pmuInCdrx"

Integer, 0–1. Sleep in connected *DRX* state. Takes effect after reset.

0 – Do not allow sleep in connected DRX.

1 – Allow sleep in connected DRX (default).

"wfiMode"

Integer, 0–2. Idle mode.

0 – Doze mode (default).

1 – WFI with low CPU frequency.

2 – WFI with normal CPU frequency.

"pwrKeyMode"

Integer, 0–2. POWERKEY pin work mode. Takes effect after reset.

0 – Internal pull-up without debounce.

1 – Internal pull-up with debounce (default).

2 – Without internal pull-up.

"usbCtrl"

Integer, 0–2. USB control mode. Takes effect after reset.

0 – USB enabled, RNDIS enumerated (default).

1 – USB enabled, RNDIS not enumerated.

2 – USB disabled.

"usbSwTrace"

Integer, 0–1. USB software trace. Takes effect after reset.

0 – Disabled (default).

1 – Enabled.

"usbSlpMask"

Integer, 0–1. USB sleep mask.

0 – USB votes for sleep (default).

1 – Mask USB sleep vote.

"usbSlpThd"

Integer, 0–65535. USB sleep threshold. The *UE* does not sleep when sleep time is less than this threshold. The default is `0`.

"usbNet"

Integer, 0–1. Network interface type. Takes effect after reset.

0 – RNDIS (default).

1 – ECM.

"fotaUrcBaudrate"

Integer. *Firmware-Over-The-Air (FOTA)* URC output baud rate. This value is set by the **`+IPR`** command. Read only.

"fotaUrcPortSel"

Integer. FOTA URC output port. Takes effect after reset. The default is `0x11` (UART, port index 1).

Bit 4–7 – Port type. 0 – USB. 1 – UART.

Bit 0–3 – Port index.

"ledMode"

Integer, 0–1. STATUS and NETWORK LEDs function.

0 – Disabled.

1 – Enabled (default). A *General-Purpose Input/Output (GPIO)* pin indicates the network connection status through different LED blinking patterns.

Network status	LED ON (ms)	LED OFF (ms)	Description
Initializing	200	1800	Slow blink
Connected	1800	200	Mostly on
Disconnected	200	1800	Slow blink
Out of service	200	1800	Slow blink
Data transfer	125	125	Fast blink

Table 1: LED blinking patterns

Note: The LED function increases power consumption in active state.

"icmpEchoReply"

Integer, 0–3. ICMP echo reply configuration. Takes effect after reset.

0 – ICMP echo disabled for IPv4 and IPv6.

1 – ICMP echo enabled for IPv4 and IPv6 (default).

2 – ICMP echo enabled for IPv4 only.

3 – ICMP echo enabled for IPv6 only.

"wifiSsidMode"

Integer, 0–2. Configure Wi–Fi SSID display mode.

0 – Display SSIDs without stripping non-ASCII characters (default).

1 – Do not display SSIDs.

2 – Display SSIDs with non-ASCII characters removed.

"disableAlarm"

Integer, 0–1. Controls whether a **%PALARM** automatically sets the modem to Minimum functionality mode and blocks re-enabling full functionality mode.

0 – Operation limit enabled (default).

1 – Operation limit disabled.

The following command example enables the watchdog timer:

```
AT%MODULECFG="startWDT",1
OK
```

The following command example sets the log level to warning:

```
AT%MODULECFG="logLevel",4
OK
```

9.6.2 Read command

The read command returns the current settings of all module configuration parameters.

Response syntax:

```
%MODULECFG: <option>:<value>
[%MODULECFG: <option>:<value>
[...]]
```

The read command parameters and their defined values are the following:

<option>

String. Name of the configuration parameter.

<value>

Integer. The value to set for the parameter.

The following command example reads all module configuration parameters:

```
AT%MODULECFG?
%MODULECFG: "faultAction":4
%MODULECFG: "uartDumpPort":1
%MODULECFG: "startWDT":1
%MODULECFG: "logCtrl":2
%MODULECFG: "logLevel":0
%MODULECFG: "logOwnerId&logOwnerLevel":0x00000000
%MODULECFG: "logBaudrate":3000000
%MODULECFG: "slpWaitTime":0
%MODULECFG: "logPortSel":2
%MODULECFG: "usbCtrl":0
%MODULECFG: "usbSwTrace":0
%MODULECFG: "usbSlpMask":0
%MODULECFG: "usbSlpThd":0
%MODULECFG: "pwrKeyMode":1
%MODULECFG: "usbNet":0
%MODULECFG: "fotaUrcBaudrate":115200
%MODULECFG: "fotaUrcPortSel":17
%MODULECFG: "pmuInCdrx":1
%MODULECFG: "wfiMode":0
%MODULECFG: "cpSlpTest":0
%MODULECFG: "ledMode":1
%MODULECFG: "icmpEchoReply":1
%MODULECFG: "wifiSsidMode":0
OK
```

9.6.3 Test command

The test command returns the command syntax.

Response syntax:

```
%MODULECFG: <option>,<setting>
```

The test command parameters and their defined values are the following:

<option>

String. Name of the configuration parameter.

<value>

Integer. The value to set for the parameter.

The following command example returns the command syntax:

```
AT%MODULECFG=?
%MODULECFG: <option>,<setting>
OK
```

9.7 Power management configuration %PMUCFG

The proprietary **%PMUCFG** command allows to manually control the nRF93M1's *Power Management Unit (PMU)*. It is intended for developing and testing power consumption on the nRF93M1. It is not intended to be used in commercial deployments since power management and sleep are automatically managed by the nRF93M1. v1.4.x

9.7.1 Set command

The set command enables power management control and selects power mode.

The configuration is stored to *NVM* when the module is set to minimum functionality.

Syntax:

```
%PMUCFG=<enable>[,<mode>]
```

The set command parameters and their defined values are the following:

<enable>

Integer. Specifies whether power management control is enabled.

0 – Disable power management control. The <mode> parameter is ignored, and the module enters idle mode.

1 – Enable power management control (default).

<mode>

Integer. Specifies power mode. Applicable only when <enable> is `1`.

0 – Active ON.

1 – System ON IDLE (default).

2 – System ON IDLE2.

3 – System ON IDLE3.

4 – System OFF.

The following command example enables power management control with System OFF mode:

```
AT%PMUCFG=1,4
OK
```

The following command example disables power management control:

```
AT%PMUCFG=0
OK
```

9.7.2 Read command

The read command returns the current *PMU* configuration.

Response syntax:

```
%PMUCFG: <enable>[,<mode>]
```

Note: If power management control is disabled with <enable>=0, the <mode> parameter is not returned.

The read command parameters and their defined values are the following:

<enable>

Integer. Specifies whether power management control is enabled.

0 – Disable power management control. The <mode> parameter is ignored, and the module enters idle mode.

1 – Enable power management control (default).

<mode>

Integer. Specifies power mode. Applicable only when <enable> is 1.

0 – Active ON.

1 – System ON IDLE (default).

2 – System ON IDLE2.

3 – System ON IDLE3.

4 – System OFF.

The following command example reads the current power management control configuration:

```
AT%PMUCFG?
%PMUCFG: 1,1
OK
```

9.7.3 Test command

The test command returns the supported parameter values.

Response syntax:

```
%PMUCFG: (range of supported <enable>s),(list of supported <mode>s)
```

The test command parameters and their defined values are the following:

<enable>

Integer. Specifies whether power management control is enabled.

0 – Disable power management control. The <mode> parameter is ignored, and the module enters idle mode.

1 – Enable power management control (default).

<mode>

Integer. Specifies power mode. Applicable only when <enable> is `1`.

0 – Active ON.

1 – System ON IDLE (default).

2 – System ON IDLE2.

3 – System ON IDLE3.

4 – System OFF.

The following command example returns the supported parameter values:

```
AT%PMUCFG=?
%PMUCFG: (0-1),(0-4)
OK
```

9.8 Modem configuration %MODEMCFG

The proprietary **%MODEMCFG** command sets and reads modem configuration parameters. v1.4.x

The command provides access to various modem configuration options including protocol stack settings, power optimization, network behavior, and feature toggles.

> **Note:** Many parameters require the modem to be in Minimum functionality mode (**AT+CFUN=0**) or Flight mode (**AT+CFUN=4**) to take effect.

9.8.1 Set command

The set command configures the modem configuration parameter.

The configuration is stored to *NVM* when the module is set to minimum functionality.

Syntax:

```
%MODEMCFG=<param>,<value>[,<value2>]
```

The set command parameters and their defined values are the following:

<param>

String. Name of the configuration parameter.

<value>

Integer. The value to set for the parameter.

<value2>

Integer. Optional second value for parameters that require two values.

The following configuration parameters are supported:

"T3324MaxValueS"

Integer, 0–16777215. User-controlled T3324 value in seconds. The default is `16777215`.

"IgnoreEmmCause"

Integer, 0–1. Ignore *EPS Mobility Management (EMM)* cause in ATTACH or *Tracking Area Update (TAU)* ACCEPT.

0 – Do not ignore.

1 – Ignore (default).

"DisableNCellMeas"

Integer, 0–1. Disable neighbor cell measurement in *RRC* IDLE state.

0 – Enabled (default).

1 – Disabled.

"CfunInit"

Initial <fun> state at power-on and <max_delay>.

<fun>

Integer, 0, 1, and 4. Modem functionality mode.

0 – Sets the module to minimum functionality. Disables transmit RF circuit and deactivates LTE services (default).

1 – Sets the module to full functionality.

4 – Sets the module to Flight mode. Disables both transmit and receive RF circuits and deactivates LTE services.

<max_delay>

Integer, 0–65535. Delay in seconds before PS power-on. The default is `0`.

The following command example sets the maximum T3324 value:

```
AT%MODEMCFG="T3324MaxValueS",3600
OK
```

The following command example sets the initial modem mode to full functionality with a delay of up to `60` seconds:

```
AT%MODEMCFG="CfunInit",1,60
OK
```

9.8.2 Read command

The read command returns the current settings of all configuration parameters.

Response syntax:

```
%MODEMCFG: <param>,<value>[,<value2>]
```

The read command parameters and their defined values are the following:

<param>

String. Name of the configuration parameter.

<value>

Integer. The value to set for the parameter.

<value2>

Integer. Optional second value for parameters that require two values.

The following command example reads all configuration parameters:

```
AT%MODEMCFG?
%MODEMCFG: "T3324MaxValueS",16777215
%MODEMCFG: "IgnoreEmmCause",1
%MODEMCFG: "DisableNCellMeas",0
%MODEMCFG: "CfunInit",0,0
OK
```

9.8.3 Test command

The test command returns a list of the names of the supported configuration parameters.

Response syntax:

```
(list of supported <param>s)
```

The test command parameter and its defined value are the following:

<param>

String. Name of the configuration parameter.

The following command example returns a list of supported parameters:

```
AT%MODEMCFG=?
("T3324MaxValueS","IgnoreEmmCause","DisableNCellMeas","CfunInit")
OK
```

9.9 Power down module %POWD

The proprietary **%POWD** command powers down the module and switches it to System Disabled mode.
v1.4.x

9.9.1 Set command

The set command powers down the module.

Once the module is powered down, it enters System Disabled mode and only a negative edge on the POWERKEY pin can power on and wake up the module.

Syntax:

```
%POWD=<option>
```

The set command parameter and its defined values are the following:

<option>

Integer, 0–1. Power–off mode.

0 – Power down immediately.

1 – Set function mode to minimum functionality and power down.

The following command example powers down the module:

```
AT%POWD=1
OK
POWERED DOWN
```

9.9.2 Read command

The read command is not supported.

9.9.3 Test command

The test command returns a list of supported <option> values.

Response syntax:

```
%POWD: (list of supported <option>s)
```

The test command parameter and its defined values are the following:

<option>

Integer, 0–1. Power–off mode.

0 – Power down immediately.

1 – Set function mode to minimum functionality and power down.

The following command example reads the supported power down command parameters:

```
AT%POWD=?
%POWD: (0,1)
OK
```

9.10 Reset module %RST

The proprietary **%RST** command performs a reset of the module. v1.4.x

9.10.1 Set command

The set command resets the module.

After execution, the module restarts and outputs boot information followed by `RDY` when ready.

Syntax:

```
%RST
```

Response syntax:

```
OK
```

The following command example restarts the modem:

```
AT%RST
OK
RDY
```

9.10.2 Read command

The read command is not supported.

9.10.3 Test command

The test command is not supported.

9.11 Factory reset %FACTORYRESET

The proprietary **%FACTORYRESET** command performs a factory reset by erasing the user file system and the module configuration region in flash memory. v1.4.x

Warning: This command permanently erases all data in the file system and platform configuration areas. This action cannot be undone. The device should be rebooted after executing this command.

9.11.1 Set command

The set command performs a factory reset.

The command restores the device to its default factory state by clearing all user-stored files and module configuration data.

Syntax:

```
%FACTORYRESET
```

+CME ERROR code

60 – System failure. Flash erase operation failed.

The following command example performs a factory reset:

```
AT%FACTORYRESET
OK
AT%RST
OK
RDY
```

9.11.2 Read command

The read command is not supported.

9.11.3 Test command

The test command is not supported.

9.12 URC reporting %PURC

The proprietary **%PURC** command enables or disables the module's *URC* reporting for sleep mode transitions. v1.4.x

Note: This command is for test purposes only and subject to change in a future firmware release.

9.12.1 Set command

The set command enables or disables the module's *URC* reporting for a specified sleep mode.

The configuration is stored to *NVM*.

Syntax:

```
%PURC=<Mode>,<value>
```

The set command parameters and their defined values are the following:

<Mode>

String. Name of the sleep mode URC to configure.

"SYSTEMOFF" – URC reporting when entering System OFF.

"SYSONIDLE3" – URC reporting when entering System ON IDLE3.

"SYSONIDLE2" – URC reporting when entering System ON IDLE2.

<value>

Integer, 0–1. Enable or disable URC reporting for the specified sleep mode.

0 – Disabled (default).

1 – Enabled.

See Sleep mode URC %SLEEPMODE on page 97 for the corresponding URC values.

The following command example enables URC reporting when the module enters System OFF state:

```
AT%PURC="SYSTEMOFF",1
OK
```

9.12.2 Read command

The read command returns the module's current *URC* reporting settings.

Response syntax:

```
%PURC: "SYSTEMOFF":<value>, "SYSONIDLE3":<value>, "SYSONIDLE2":<value>
```

The response parameters and their defined values are the following:

<value>

Integer, 0–1. Enable or disable URC reporting for the specified sleep mode.

0 – Disabled (default).

1 – Enabled.

The following command example reads the module's current URC reporting settings:

```
AT%PURC?
%PURC: "SYSTEMOFF":0, "SYSONIDLE3":0, "SYSONIDLE2":0
OK
```

9.12.3 Test command

The test command returns the supported value range.

Response syntax:

```
%PURC: "Mode":(range of supported <value>s)
```

The test command parameters and their defined values are the following:

<Mode>

String. Name of the sleep mode URC to configure.

"SYSTEMOFF" – URC reporting when entering System OFF.

"SYSONIDLE3" – URC reporting when entering System ON IDLE3.

"SYSONIDLE2" – URC reporting when entering System ON IDLE2.

<value>

Integer, 0–1. Enable or disable URC reporting for the specified sleep mode.

0 – Disabled (default).

1 – Enabled.

The following command example returns the supported value range:

```
AT%PURC=?
%PURC: "Mode":(0-1)
OK
```

9.12.4 Sleep mode URC %SLEEPMODE

The proprietary **%SLEEPMODE** *URC* indicates that the module is entering a sleep mode or exiting sleep to active mode.

Response syntax:

```
%SLEEPMODE: <mode>
```

The URC parameter and its defined values are the following:

<mode>

Integer, 0–4. Sleep transition indication.

0 – Entering System ON IDLE.

2 – Entering System ON IDLE2.

3 – Entering System ON IDLE3.

4 – Entering System OFF.

The following command example enables System OFF reporting and shows the module entering System OFF:

```
AT%PURC="SYSTEMOFF",1
OK
%SLEEPMODE: 4
```

The following command example shows the module entering System ON IDLE:

```
%SLEEPMODE: 0
```

9.13 ADC readout %ADC

The proprietary **%ADC** command triggers a single conversion of the internal ADC channels and returns the decoded result. v1.4.x

9.13.1 Set command

The set command samples one or more internal ADC channels and returns the decoded result.

Syntax:

```
%ADC=<option>
```

Response syntax:

```
%ADC: <result_body>
```

%ADC ERROR

PARAMETER ERROR – Incorrect <option> is given.

The set command parameter and its defined values are the following:

<option>

String. ADC channel selection.

"all" – Sample die temperature and battery voltage.

"temp" – Sample die temperature only.

"vbat" – Sample battery voltage only.

The response parameter and its defined values are the following:

<result_body>

String. ADC result body.

TEMP,<temp_cdeg> – Die temperature in Celsius degrees.

VBAT,<vbat_mV> – Battery voltage in millivolts.

TEMP,<temp_cdeg>, VBAT,<vbat_mV> – Die temperature in Celsius degrees and battery voltage in millivolts.

<temp_cdeg>

Integer. Die temperature in Celsius degrees.

<vbat_mV>

Integer. Battery voltage in millivolts.

The following command example samples die temperature:

```
AT%ADC="temp"
%ADC: TEMP, 24
OK
```

The following command example samples battery voltage:

```
AT%ADC="vbat"
%ADC: VBAT, 3865
OK
```

The following command example samples die temperature and battery voltage:

```
AT%ADC="all"
%ADC: TEMP, 24, VBAT, 3865
OK
```

9.13.2 Read command

The read command is not supported.

9.13.3 Test command

The test command returns the supported ADC channel options.

Response syntax:

```
%ADC: (list of supported <option>s)
```

The test command parameter and its defined values are the following:

<option>

String. ADC channel selection.

"all" – Sample die temperature and battery voltage.

"temp" – Sample die temperature only.

"vbat" – Sample battery voltage only.

The following command example returns the supported ADC channel options:

```
AT%ADC=?
%ADC: ("all","temp","vbat")
OK
```

9.14 Platform alarm URC %PALARM

The proprietary **%PALARM** *URC* indicates that a voltage or die temperature alarm threshold has been crossed.

Syntax for voltage alarm:

```
%PALARM: Volt,<voltAlarm>
```

Syntax for thermal alarm:

```
%PALARM: Therm,<thermAlarm>
```

The URC parameters and their defined values are the following:

<voltAlarm>

Integer, 0–1. Voltage alarm state.

0 – Voltage dropped below the threshold of 2788 mV.

1 – Voltage recovered above the threshold.

<thermAlarm>

Integer, 0–1. Die temperature alarm state.

0 – Die temperature dropped back below the threshold.

1 – Die temperature rose above the configured threshold of 100 degrees Celsius.

Note: When a danger condition is detected, the modem sends a **%PALARM** URC to all AT channels and sets the radio to minimum functionality if the **%MODULECFG**command's "disableAlarm" is 0. The host must issue **AT+CFUN=1** to restore full functionality once the conditions improve. The command is rejected when any alarm is active.

The following URC example shows a voltage dropping below the threshold:

```
%PALARM: Volt,0
```

The following URC example shows the die temperature rising above the threshold:

```
%PALARM: Therm,1
```

10 Packet domain commands

Commands for the packet domain include commands that control packet-switched services.

10.1 Define PDP context +CGDCONT

The **+CGDCONT** command defines *Packet Data Protocol (PDP) Context*. v1.4.x

> **Note:** Operator configurations can override values set by the user.

For reference, see *3GPP 27.007 Ch. 10.1.1*.

10.1.1 Set command

The set command configures connection parameters.

Syntax:

```
+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>]]]]]]]]]]]
```

The set command parameters and their defined values are the following:

<cid>

Integer, 1–15. Specifies a particular *Packet Data Protocol (PDP) Context* definition. The parameter is local to the device and used in other PDP context-related commands.

<PDP_type>

String. Specifies the type of packet data protocol.

IP – Internet Protocol (default).

IPV6 – Internet Protocol version 6.

IPV4V6 – Virtual type of dual IP stack.

Non-IP – Transfer of non-IP data to external packet data network. See *3GPP TS 23.401 [82]*.

<APN>

String. *Access Point Name (APN)*. Maximum length 99 characters. The default is network-provided or previously configured.

<PDP_addr>

String. PDP address. Ignored.

<d_comp>

Integer. PDP Data compression. Ignored.

<h_comp>

Integer. PDP Header compression. Ignored.

<IPv4AddrAlloc>

Integer, 0. IPv4 Address Allocation Method.

0 – IPv4 address through *Non-access Stratum (NAS)* signaling (default).

<request_type>

Integer, 0–2. Bearer Request Type.

0 – PDP context is for a new PDP context establishment or for a handover from a non-3GPP access network (default).

1 – PDP context is for emergency bearer services (not supported).

2 – PDP context is for a new PDP context establishment.

<P-CSCF_discovery>

Integer, 0–2. Sets preference for obtaining P-CSCF address.

0 – P-CSCF address discovery is not affected by **+CGDCONT** (default).

1 – P-CSCF address discovery through NAS signaling is preferred (not supported).

2 – P-CSCF address discovery through DHCP is preferred (not supported).

<IM_CN_Signalling_Flag_Ind>

Integer, 0–1. Indicates to the network whether the PDP context is only for IM CN subsystem-related signaling.

0 – UE indicates that the PDP context is not for IM CN subsystem-related signaling only (default).

1 – UE indicates that the PDP context is for IM CN subsystem-related signaling only (not supported).

<NSLPI>

Integer, 0–1. Indicates the *Non-access Stratum (NAS) Signalling Low Priority Indication (NSLPI)* requested for this PDP context.

0 – NAS signaling low priority value from configuration is used (default).

1 – Value "`Not configured`" for NAS signaling low priority.

<securePCO>

Integer, 0–1. Specifies if security protected transmission of *Protocol Configuration Options (PCO)* is requested or not.

0 – Protected transmission of PCO is not requested (default).

1 – Protected transmission of PCO is requested (not supported).

<IPv4_MTU_discovery>

Integer, 0–1. Sets preference for requesting the IPv4 MTU size.

0 – MTU size discovery is not affected by **+CGDCONT** (default).

1 – MTU size discovery through NAS signaling is preferred. Supported only for `<PDP_type>`: `IP` and `IPv4V6`.

Note: `+CGDCONT=<cid>` causes the values for context number <cid> to become undefined.

A maximum of four contexts can be defined. In some network configurations, the maximum can be less than four.

The following command example configures CID 1 to use IPv4 and access point "internet":

```
AT+CGDCONT=1,"IP","internet"
OK
```

10.1.2 Read command

The read command reads the list of defined contexts.

Response syntax:

```
+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>]]]]]]]]]
```

The read command parameters and their defined values are the following:

<cid>

Integer, 1–15. Specifies a particular *Packet Data Protocol (PDP) Context* definition. The parameter is local to the device and used in other PDP context-related commands.

<PDP_type>

String. Specifies the type of packet data protocol.

IP – Internet Protocol (default).

IPV6 – Internet Protocol version 6.

IPV4V6 – Virtual type of dual IP stack.

Non-IP – Transfer of non-IP data to external packet data network. See *3GPP TS 23.401 [82]*.

<APN>

String. *APN*. Maximum length 99 characters. The default is network-provided or previously configured.

<PDP_addr>

String. IP address of the context. Empty if not yet assigned.

<d_comp>

Integer, 0. PDP Data compression.

0 – Compression not supported.

<h_comp>

Integer, 0. PDP Header compression.

0 – Compression not supported.

<IPv4AddrAlloc>

Integer, 0. IPv4 Address Allocation Method.

0 – IPv4 address through *NAS* signaling (default).

<request_type>

Integer, 0–2. Bearer Request Type.

0 – PDP context is for a new PDP context establishment or for a handover from a non-3GPP access network (default).

1 – PDP context is for emergency bearer services (not supported).

2 – PDP context is for a new PDP context establishment.

<P-CSCF_discovery>

Integer, 0–2. Sets preference for obtaining P-CSCF address.

0 – P-CSCF address discovery is not affected by **+CGDCONT** (default).

1 – P-CSCF address discovery through NAS signaling is preferred (not supported).

2 – P-CSCF address discovery through DHCP is preferred (not supported).

<IM_CN_Signalling_Flag_Ind>

Integer, 0–1. Indicates to the network whether the PDP context is only for IM CN subsystem-related signaling.

0 – UE indicates that the PDP context is not for IM CN subsystem-related signaling only (default).

1 – UE indicates that the PDP context is for IM CN subsystem-related signaling only (not supported).

<NSLPI>

Integer, 0–1. Indicates the *NSLPI* requested for this PDP context.

0 – NAS signaling low priority value from configuration is used (default).

1 – Value "`Not configured`" for NAS signaling low priority.

<securePCO>

Integer, 0–1. Specifies if security protected transmission of *PCO* is requested or not.

0 – Protected transmission of PCO is not requested (default).

1 – Protected transmission of PCO is requested (not supported).

<IPv4_MTU_discovery>

Integer, 0–1. Sets preference for requesting the IPv4 MTU size.

0 – MTU size discovery is not affected by **+CGDCONT** (default).

1 – MTU size discovery through NAS signaling is preferred. Supported only for `<PDP_type>`: `IP` and `IPv4V6`.

The following command example reads configured default bearers:

```
AT+CGDCONT?
+CGDCONT: 1,"IP","internet","10.132.176.29",0,0
+CGDCONT: 2,"IPV6","internet","2001:0999:0038:8ED2:9A69:FA75:99B2:B542",0,0
+CGDCONT: 3,"IPV4V6","internet","10.196.234.198
 2001:0999:0408:D2BA:2138:D370:1E85:5BDF",0,0
OK
```

10.1.3 Test command

The test command returns a list of supported context parameters.

Response syntax:

```
+CGDCONT: (range of supported <cid>s),<PDP_type>,,,,,(list of supported <IPv4AddrAlloc>s),
(list of supported <request_type>s),(list of supported <P-CSCF_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)
```

The test command parameters and their defined values are the following:

<cid>

Integer, 1–15. Specifies a particular *Packet Data Protocol (PDP) Context* definition. The parameter is local to the device and used in other PDP context-related commands.

<PDP_type>

String. Specifies the type of packet data protocol.

IP – Internet Protocol (default).

IPV6 – Internet Protocol version 6.

IPV4V6 – Virtual type of dual IP stack.

Non-IP – Transfer of non-IP data to external packet data network. See *3GPP TS 23.401 [82]*.

<APN>

String. *APN*. Maximum length 99 characters. The default is network-provided or previously configured.

<PDP_addr>

String. IP address of the context. Empty if not yet assigned.

<d_comp>

Integer, 0. PDP Data compression.

0 – Compression not supported.

<h_comp>

Integer, 0. PDP Header compression.

0 – Compression not supported.

<IPv4AddrAlloc>

Integer, 0. IPv4 Address Allocation Method.

0 – IPv4 address through *NAS* signaling (default).

<request_type>

Integer, 0–2. Bearer Request Type.

0 – PDP context is for a new PDP context establishment or for a handover from a non-3GPP access network (default).

1 – PDP context is for emergency bearer services (not supported).

2 – PDP context is for a new PDP context establishment.

<P-CSCF_discovery>

Integer, 0–2. Sets preference for obtaining P-CSCF address.

0 – P-CSCF address discovery is not affected by **+CGDCONT** (default).

1 – P-CSCF address discovery through NAS signaling is preferred (not supported).

2 – P-CSCF address discovery through DHCP is preferred (not supported).

<IM_CN_Signalling_Flag_Ind>

Integer, 0–1. Indicates to the network whether the PDP context is only for IM CN subsystem-related signaling.

0 – UE indicates that the PDP context is not for IM CN subsystem-related signaling only (default).

1 – UE indicates that the PDP context is for IM CN subsystem-related signaling only (not supported).

<NSLPI>

Integer, 0–1. Indicates the *NSLPI* requested for this PDP context.

0 – NAS signaling low priority value from configuration is used (default).

1 – Value "`Not configured`" for NAS signaling low priority.

<securePCO>

Integer, 0–1. Specifies if security protected transmission of *PCO* is requested or not.

0 – Protected transmission of PCO is not requested (default).

1 – Protected transmission of PCO is requested (not supported).

<IPv4_MTU_discovery>

Integer, 0–1. Sets preference for requesting the IPv4 MTU size.

0 – MTU size discovery is not affected by **+CGDCONT** (default).

1 – MTU size discovery through NAS signaling is preferred. Supported only for `<PDP_type>`: `IP` and `IPv4V6`.

The following command example returns a list of supported context parameters:

```
AT+CGDCONT=?
+CGDCONT: (1-15),"IP",,,,,(0),(0,2),(0),(0),(0,1),(0),(0,1)
+CGDCONT: (1-15),"IPV6",,,,,(0),(0,2),(0),(0),(0,1),(0),(0)
+CGDCONT: (1-15),"IPV4V6",,,,,(0),(0,2),(0),(0),(0,1),(0),(0,1)
+CGDCONT: (1-15),"Non-IP",,,,,(0),(0,2),(0),(0),(0,1),(0),(0)
OK
```

10.2 Packet domain event reporting +CGEREP

The **+CGEREP** command enables or disables the sending of unsolicited **+CGEV** result codes from *Mobile Equipment (ME)* to *TE* when certain events occur in the packet domain *MT* or the network.

For reference, see *3GPP 27.007 Ch. 10.1.19*.

10.2.1 Set command

The set command enables or disables sending of unsolicited **+CGEV** result codes from *MT* to *TE* when certain events occur in the packet domain MT or the network.

Syntax:

```
+CGEREP=<mode>[,<bfr>]
```

The set command parameters and their defined values are the following:

<mode>

Integer, 0–1. Controls the processing of *URC*s.

0 – Buffer URCs in the MT. If the MT's result code buffer is full, the oldest codes can be discarded. No codes are forwarded to the TE (default).

1 – Discard URCs when the MT-TE link is reserved, for example, in on-line data mode. Otherwise, forward them directly to the TE.

<bfr>

Integer, 0. Controls the effect on buffered codes when <mode> is set to `1` or `2`.

0 – The MT's defined buffer for URCs is cleared when <mode> is set to `1` or `2`.

The following command example enables the forwarding of **+CGEV** codes and clears the buffer when <mode> is set to `1`:

```
AT+CGEREP=1,0
OK
```

10.2.2 Read command

The read command returns the current mode and buffer settings.

Response syntax:

```
+CGEREP: <mode>,<bfr>
```

The read command parameters and their defined values are the following:

<mode>

Integer, 0–1. Controls the processing of *URC*s.

0 – Buffer URCs in the MT. If the MT's result code buffer is full, the oldest codes can be discarded. No codes are forwarded to the TE (default).

1 – Discard URCs when the MT-TE link is reserved, for example, in on-line data mode. Otherwise, forward them directly to the TE.

<bfr>

Integer, 0. Controls the effect on buffered codes when <mode> is set to 1 or 2.

0 – The MT's defined buffer for URCs is cleared when <mode> is set to 1 or 2.

The following command example returns the current mode and buffer settings:

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

10.2.3 Test command

The test command returns the modes and buffer settings supported by the *MT* as compound values.

Response syntax:

```
+CGEREP: (list of supported <mode>s),(list of supported <bfr>s)
```

The test command parameters and their defined values are the following:

<mode>

Integer, 0–1. Controls the processing of *URC*s.

0 – Buffer URCs in the MT. If the MT's result code buffer is full, the oldest codes can be discarded. No codes are forwarded to the TE (default).

1 – Discard URCs when the MT-TE link is reserved, for example, in on-line data mode. Otherwise, forward them directly to the TE.

<bfr>

Integer, 0. Controls the effect on buffered codes when <mode> is set to 1 or 2.

0 – The MT's defined buffer for URCs is cleared when <mode> is set to 1 or 2.

The following command example returns the supported modes and buffer settings:

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

10.3 Packet domain event notification +CGEV

Unsolicited message to indicate EPS *Packet Data Network (PDN)* connection and bearer resources operations status. Notifications are sent when a packet data context is activated, deactivated, or modified.
v1.4.x

For reference, see *3GPP 27.007 Ch. 10.1.19*.

These notifications are subscribed using the **+CGEREP** command.

Syntax descriptions are as follows:

The *ME* has activated a context:

```
+CGEV: ME PDN ACT <cid>[,<pdnreason>]
```

The network has forced a context deactivation:

```
+CGEV: NW PDN DEACT <cid>
```

The mobile termination has forced a context deactivation:

```
+CGEV: ME PDN DEACT <cid>
```

The network has modified a context:

```
+CGEV: NW MODIFY <cid>,<change_reason>,<event_type>
```

The mobile termination has modified a context:

```
+CGEV: ME MODIFY <cid>,<change_reason>,<event_type>
```

The notification parameters and their defined values are the following:

<cid>

Integer, 1–15. Context identifier for an associated context.

<pdnreason>

Integer, 0–4. Indicates why the context activation request for *PDP* type IPv4v6 was not granted.

0 – IPv4 allowed only.

1 – IPv6 allowed only.

2 – Single-address bearer allowed only.

3 – Single-address bearer allowed only and active second bearer failed.

4 – No reason.

<change_reason>

Integer. A bitmap that indicates the type of change that occurred. The value is determined by summing up all the applicable bits.

Bit 1 – TFT changed.

Bit 2 – *Quality of Service (QoS)* changed.

Bit 3 – WLAN offload changed.

<event_type>

Integer. Not supported.

The following notification example shows that the ME has activated a context:

```
+CGEV: ME PDN ACT 1
```

10.4 Activate PDP context +CGACT

The **+CGACT** command activates or deactivates a *PDN* connection. v1.4.x

For reference, see *3GPP 27.007 Ch. 10.1.10*.

10.4.1 Set command

The set command activates or deactivates a *PDN* connection.

The *Packet Data Protocol (PDP) Context* must first be defined with the **+CGDCONT** command. If a PDP context is already in the requested state when the **+CGACT** command is executed, the state for that context remains unchanged. If the requested state for a specified context cannot be achieved, a `+CME ERROR` response is returned.

If the device is not attached to the network when the activation command is executed, the device performs an attach and then attempts to activate the specified contexts. If the attach attempt fails, the device responds with a `+CME ERROR` response. If an attempt is made to disconnect the last PDN connection, the device responds with a `+CME ERROR` response.

The activation request for an *Evolved Packet System (EPS)* bearer resource is answered by the network by either an EPS dedicated bearer activation or EPS bearer modification request. The request must be accepted by the device before the PDP context can be set into established state.

Syntax:

```
+CGACT=<state>,<cid>
```

+CME ERROR code

50 – Incorrect <cid>.

55 – Operation not allowed because of MT functionality restrictions.

143 – Unknown PDP context.

173 – Initial PDN connection <cid> `1` cannot be deactivated.

The set command parameters and their defined values are the following:

<state>

Integer, 0–1. PDP context activation state.

0 – Deactivated state.

1 – Activated state.

<cid>

Integer, 1–15. PDP context identifier.

The following command example activates a bearer configured with CID `1`:

```
AT+CGACT=1,1
OK
```

10.4.2 Read command

The read command reads a list of *PDN* connections and states.

Response syntax:

```
+CGACT: <cid>,<state>
```

The read command parameters and their defined values are the following:

<state>

Integer, 0–1. PDP context activation state.

0 – Deactivated state.

1 – Activated state.

<cid>

Integer, 1–15. PDP context identifier.

The following command example returns a list of connections with states:

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

10.4.3 Test command

The test command returns a list of supported states.

Response syntax:

```
+CGACT: (list of supported <state>s)
```

The test command parameter and its defined values are the following:

<state>

Integer, 0–1. PDP context activation state.

0 – Deactivated state.

1 – Activated state.

The following command example returns a list of supported states:

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

10.5 Read PDP context dynamic parameters +CGCONTRDP

The **+CGCONTRDP** command returns information for an active non-secondary *PDP* context. This command issues a valid response only when the modem is activated. v1.4.x

For reference, see *3GPP 27.007 Ch. 10.1.23*.

10.5.1 Set command

The set command returns information for an active non-secondary *PDP* context.

The command can be executed with an optional <cid> parameter. If the parameter is omitted, the command returns information for all active non-secondary PDP contexts.

If the *MT* has dual stack capabilities, at least one pair of lines is returned for each <cid>. The first line contains the IPv4 parameters, and the second line contains the IPv6 parameters. If the MT indicates more than two IP addresses of DNS servers, multiple pairs of such lines are returned.

Syntax:

```
+CGCONTRDP[=<cid>]
```

Response syntax:

```
+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>[
,<Serving_PLMN_rate_control_value>]]]]]]]]]]]]]
```

The set command parameters and their defined values are the following:

<cid>

Integer, 1–15. Context identifier for a non-secondary PDP context.

<bearer_id>

Integer, 0 or 5–15. EPS bearer identifier. `0` when bearer is not activated.

<apn>

String. Logical name of the *PDN*.

<local_addr_and_subnet_mask>

String, 0–255. IP address and subnet mask of the *MT*. Dot-separated numeric value.

<gw_addr>

String, 0–255. Gateway address. Dot-separated numeric value.

<DNS_prim_addr>, <DNS_sec_addr>

String. IP address of the primary and secondary DNS server.

<PCSCF_prim_addr>, <PCSCF_sec_addr>

String. IP address of the primary and secondary P-CSCF server.

<IM_CN_Signalling_Flag>

Integer, 0–1. `0` – PDP context is not for IM CN subsystem-related signaling only.

`1` – PDP context is for IM CN subsystem-related signaling only.

<LIPA_indication>

Integer. Not supported.

<IPv4_MTU>

Integer, 0–65535. IPv4 MTU size in octets.

<WLAN_Offload>

Integer. Not supported.

<Local_Addr_Ind>

Integer. Not supported.

<Serving_PLMN_rate_control_value>

Integer, 0–65535. Maximum number of uplink messages the *UE* is allowed to send in a 6 minute interval.

An error results in a `+CME ERROR: <err>` response.

The following command example returns dynamic parameters for context `1`:

```
AT+CGCONTRDP=1
+CGCONTRDP: 1,5,"onomondo","100.117.83.235.255.0.0.0",,"172.30.8.5","172.30.8.6",,,0,,1360
OK
```

10.5.2 Read command

The read command is not supported.

10.5.3 Test command

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

Response syntax:

```
+CGCONTRDP: (list of <cid>s associated with active contexts)
```

The test command parameter and its defined value are the following:

<cid>

Integer, 1–15. Context identifier for an active non-secondary PDP context.

The following command example returns a list of active context identifiers:

```
AT+CGCONTRDP=?
+CGCONTRDP: (1)
OK
```

10.6 PS attach or detach +CGATT

The **+CGATT** command attaches the *MT* to the Packet Domain services or detaches the MT from them.
v1.4.x

For reference, see *3GPP 27.007 Ch. 10.1.9*.

> **Note:** The **+CGATT** command is for test purposes only.

10.6.1 Set command

The set command attaches the *UE* to the Packet Domain services or detaches the UE from them.

> **Note:** The UE performs an attach automatically when activated. In normal operation there is no need to execute the **+CGATT** command.

Syntax:

```
+CGATT=<state>
```

+CME ERROR code

13 – SIM failure.

30 – No network service.

31 – Network timeout.

50 – Incorrect parameters.

55 – Operation not allowed because of MT functionality restrictions.

100 – Unknown.

The set command parameter and its defined values are the following:

<state>

Integer, 0–1. PS attach state.

0 – Detached state.

1 – Attached state.

The following command example performs an *EPS* attach:

```
AT+CGATT=1
OK
```

10.6.2 Read command

The read command reads the state.

Response syntax:

```
+CGATT: <state>
```

The response parameters and their defined values are the following:

<state>

Integer, 0–1. PS attach state.

0 – Detached state.

1 – Attached state.

The following command example reads the state in *EPS* attach state:

```
AT+CGATT?
+CGATT: 1
OK
```

10.6.3 Test command

The test command returns a list of supported states.

Response syntax:

```
+CGATT: (list of supported <state>s)
```

The test command parameter and its defined values are the following:

<state>

Integer, 0–1. PS attach state.

0 – Detached state.

1 – Attached state.

The following command example returns a list of supported states:

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

10.7 DNS lookup %DNS

The proprietary **%DNS** command resolves a domain name to an *Internet Protocol (IP)* address. v1.4.x

10.7.1 Set command

The set command resolves a domain name and returns the corresponding *IP* address.

Syntax:

```
%DNS=<url>
```

Response syntax:

```
%DNS: <ipaddr>
```

The set command parameters and their defined values are the following:

<url>

String. Maximum length 255 characters. Domain name to resolve.

<ipaddr>

String. Resolved IP address.

The following command example resolves a domain name:

```
AT%DNS="www.google.fi"
%DNS: "192.178.25.67"
OK
```

10.7.2 Read command

The read command is not supported.

10.7.3 Test command

The test command is not supported.

10.8 ICMP echo request %PING

The proprietary **%PING** command sends ICMP packets to a specified host address. v1.4.x

The command initiates the sending of ping packets with a specified payload size to the target address. If the remote system is connected and responding to ping packets, a response is returned. The command continues to send ping packets until the specified count is reached or the timeout period expires.

10.8.1 Set command

The set command sends ICMP ping packets to the specified host or stops an ongoing ping operation.

Syntax:

```
%PING=<destination>[,<count>[,<size>[,<timeout>]]]
```

Syntax to stop an ongoing ping operation:

```
%PING=0
```

The following *URC*s are returned during the ping operation:

```
%PING: SUCC, dest: <dest_ip_addr>, RTT: <rtt_time> ms
%PING: FAIL, dest: <dest_ip_addr>, time out: <timeout> ms
%PING: ERROR, cause: <cause>
%PING: DONE
%PING: dest: <dest_ip_addr>, <count> packets transmitted, <reply_count> received,
 <lost_percent> % packet loss
 rtt min/avg/max = <rtt_min> / <rtt_avg> / <rtt_max> ms
```

The set command parameters and their defined values are the following:

<destination>

String. Maximum length 255 characters. *IP* address or URL of the target host.

0

Integer, 0. Stop an ongoing ping operation.

<count>

Integer, 1–255. Number of ping packets to send. The default is `4`.

255 – Ping continuously until stopped.

<size>

Integer, 1–1500. Payload size in bytes. The default is `32`.

<timeout>

Integer, 1000–600000. Ping reply timeout in milliseconds. The default is `20000`.

<dest_ip_addr>

String. Maximum length 39 characters. Destination IP address in the response.

<rtt_time>

Integer. Round-trip time in milliseconds.

<rtt_min>

Integer. Minimum round-trip time in milliseconds.

<rtt_avg>

Integer. Average round-trip time in milliseconds.

<rtt_max>

Integer. Maximum round-trip time in milliseconds.

<reply_count>

Integer. Number of successful ping replies received.

<lost_percent>

Integer. Percentage of packets lost.

The following command example pings `8.8.8.8` ten times with 32-byte payload and 5-second timeout:

```
AT%PING="8.8.8.8",10,32,5000
OK

%PING: SUCC, dest: 8.8.8.8, RTT: 132 ms

%PING: SUCC, dest: 8.8.8.8, RTT: 82 ms

%PING: SUCC, dest: 8.8.8.8, RTT: 79 ms

%PING: SUCC, dest: 8.8.8.8, RTT: 88 ms

%PING: SUCC, dest: 8.8.8.8, RTT: 81 ms

%PING: SUCC, dest: 8.8.8.8, RTT: 78 ms

%PING: SUCC, dest: 8.8.8.8, RTT: 80 ms

%PING: SUCC, dest: 8.8.8.8, RTT: 79 ms

%PING: SUCC, dest: 8.8.8.8, RTT: 79 ms

%PING: SUCC, dest: 8.8.8.8, RTT: 79 ms

%PING: DONE

%PING: dest: 8.8.8.8, 10 packets transmitted, 10 received, 0 % packet loss
 rtt min/avg/max = 78 / 85 / 132 ms
```

The following command example stops an ongoing ping operation:

```
AT%PING=0
OK
```

10.8.2 Read command

The read command is not supported.

10.8.3 Test command

The test command returns the supported parameter values.

Response syntax:

```
%PING: (<destination>,0),(range of supported <count>s),(range of supported <size>s),(range
 of supported <timeout>s)
```

The test command parameters and their defined values are the following:

<destination>

String. Maximum length 255 characters. *IP* address or URL of the target host.

0

Integer, 0. Stop an ongoing ping operation.

<count>

Integer, 1–255. Number of ping packets to send. The default is `4`.

255 – Ping continuously until stopped.

<size>

Integer, 1–1500. Payload size in bytes. The default is `32`.

<timeout>

Integer, 1000–600000. Ping reply timeout in milliseconds. The default is `20000`.

<dest_ip_addr>

String. Maximum length 39 characters. Destination IP address in the response.

<rtt_time>

Integer. Round-trip time in milliseconds.

<rtt_min>

Integer. Minimum round-trip time in milliseconds.

<rtt_avg>

Integer. Average round-trip time in milliseconds.

<rtt_max>

Integer. Maximum round-trip time in milliseconds.

<reply_count>

Integer. Number of successful ping replies received.

<lost_percent>

Integer. Percentage of packets lost.

The following command example returns the supported parameter values:

```
AT%PING=?
%PING: ("IP ADDR\URL",0),(1-255),(1-1500),(1000-600000)
OK
```

10.9 Show PDP addresses +CGPADDR

The **+CGPADDR** command returns a list of *PDP* addresses for the specified context identifiers. v1.4.x

For reference, see *3GPP 27.007 Ch. 10.1.14*.

10.9.1 Set command

The set command returns a list of *PDP* addresses for the specified context identifiers.

Syntax:

```
+CGPADDR[=<cid>]
```

If <cid> is omitted, the addresses for all defined contexts are returned.

Response syntax:

```
+CGPADDR: <cid>[,<PDP_addr_1>[,<PDP_addr_2>]]
```

The set command parameters and their defined values are the following:

<cid>

Integer, 1–15. PDP context identifier.

<PDP_addr_1>

String. For IPv4, a dot-separated numeric value (0–255). For IPv6, a colon-separated hexadecimal value (0x0000–0xFFFF).

<PDP_addr_2>

String. Given as a colon-separated hexadecimal value (0x0000–0xFFFF). Included when both IPv4 and IPv6 addresses are assigned.

The following command example returns the IP address for context `1`:

```
AT+CGPADDR=1
+CGPADDR: 1,"100.85.235.28"
OK
```

10.9.2 Read command

The read command is not supported.

10.9.3 Test command

The test command returns a list of defined <cid> values.

Response syntax:

```
+CGPADDR: (list of defined <cid>s)
```

The test command parameter and its defined value are the following:

<cid>

Integer, 1–15. PDP context identifier.

The following command example returns a list of defined <cid> values:

```
AT+CGPADDR=?
+CGPADDR: (1)
OK
```

11 Socket commands

Network socket commands can be used for sending and receiving data.

11.1 Create socket %SKTCREATE

The proprietary **%SKTCREATE** command creates a socket on the *UE* and associates it with the specified protocol. v1.4.x

11.1.1 Set command

The set command creates a socket on the *UE* and associates it with the specified protocol. The UE supports up to 12 TCP or UDP sockets simultaneously.

Syntax:

```
%SKTCREATE=<domain>,<type>,<protocol>
```

Response syntax:

```
%SKTCREATE: <fd>
```

+CME ERROR code

402 – Parameter error.

403 – No free socket context.

404 – Create socket failed.

416 – Out of memory.

420 – Send request failed.

429 – Internal error.

The set command parameters and their defined values are the following:

<domain>

Integer, 1–2. Address family.

1 – IPv4 (Default).

2 – IPv6.

<type>

Integer, 1–2. Socket type.

1 – TCP (Default).

2 – UDP.

<protocol>

Integer, 6 and 17. Transport protocol.

6 – TCP.

17 – UDP.

<fd>

Integer, 0–11. Socket file descriptor.

The following command example creates a UDP socket:

```
AT%SKTCREATE=1,2,17
%SKTCREATE: 0
OK
```

11.1.2 Read command

The read command is not supported.

11.1.3 Test command

The test command returns supported values as a compound value.

Response syntax:

```
%SKTCREATE: (list of supported <domain>s),(list of supported <type>s),(list of supported
 <protocol>s)
```

The test command parameters and their defined values are the following:

<domain>

Integer, 1–2. Address family.

1 – IPv4 (Default).

2 – IPv6.

<type>

Integer, 1–2. Socket type.

1 – TCP (Default).

2 – UDP.

<protocol>

Integer, 6 and 17. Transport protocol.

6 – TCP.

17 – UDP.

The following command example returns the supported values:

```
AT%SKTCREATE=?
%SKTCREATE: (1,2),(1,2),(6,17)
OK
```

11.2 Connect socket %SKTCONNECT

The proprietary **%SKTCONNECT** command connects a socket to a remote address and port. v1.4.x

11.2.1 Set command

The set command connects the TCP socket to a remote address and port or stores the connection parameters for a UDP socket.

Syntax:

```
%SKTCONNECT=<fd>,<address>,<port>
```

+CME ERROR code

402 – Parameter error.

406 – Socket not found.

407 – Connect failed.

412 – Invalid state.

413 – Connect timeout.

416 – Out of memory.

418 – Connect in progress.

420 – Send request failed.

423 – DNS resolve failed.

429 – Internal error.

The set command parameters and their defined values are the following:

<fd>

Integer, 0–11. Socket file descriptor returned by the **%SKTCREATE** command.

<address>

String. Remote address to connect or send to. *IP* address or hostname of the target host. The maximum size is 63 characters

<port>

Integer, 1–65535. Remote port to connect or send to.

The following command example connects the socket:

```
AT%SKTCONNECT=0,"172.248.12.114",20084
OK
```

11.2.2 Read command

The read command is not supported.

11.2.3 Test command

The test command returns supported values.

Response syntax:

```
%SKTCONNECT: (range of supported <fd>s),(<address>),(range of supported <port>s)
```

The test command parameters and their defined values are the following:

<fd>

Integer, 0–11. Socket file descriptor returned by the **%SKTCREATE** command.

<address>

String. Remote address to connect or send to. *IP* address or hostname of the target host. The maximum size is 63 characters

<port>

Integer, 1–65535. Remote port to connect or send to.

The following command example returns the supported values:

```
AT%SKTCONNECT=?
%SKTCONNECT: (0-11),"IP ADDR\URL",(1-65535)
OK
```

11.3 Bind socket to local endpoint %SKTBIND

The proprietary **%SKTBIND** command binds a socket to a local address and port. v1.4.x

11.3.1 Set command

The set command binds a socket to a local address and port.

Syntax:

```
%SKTBIND=<fd>,<address>,<port>
```

+CME ERROR code

402 – Parameter error.

406 – Socket not found.

408 – Bind failed.

416 – Out of memory.

420 – Send request failed.

429 – Internal error.

The set command parameters and their defined values are the following:

<fd>

Integer, 0–11. Socket file descriptor returned by the **%SKTCREATE** command.

<address>

String. Local address. If empty, the socket accepts packets on any local address.

<port>

Integer, 1–65535. Local port.

The following command example binds the socket to a local endpoint:

```
AT%SKTBIND=0,"172.248.12.114",20084
OK
```

11.3.2 Read command

The read command is not supported.

11.3.3 Test command

The test command lists supported values.

Response syntax:

```
%SKTBIND: (range of supported <fd>s),(<address>),(range of supported <port>s)
```

The test command parameters and their defined values are the following:

<fd>

Integer, 0–11. Socket file descriptor returned by the **%SKTCREATE** command.

<address>

String. Local address. If empty, the socket accepts packets on any local address.

<port>

Integer, 1–65535. Local port.

The following command example lists the supported values:

```
AT%SKTBIND=?
%SKTBIND: (0-11),"IP ADDR",(1-65535)
OK
```

11.4 Send data %SKTSEND

The proprietary **%SKTSEND** command sends data to a remote address and port through a socket. v1.4.x

11.4.1 Set command

The set command sends data to the remote address and port configured for the socket.

Syntax:

```
%SKTSEND=<fd>,<data_len>,<data>[,<rai_info>[,<except_info>]]
```

+CME ERROR code

402 – Parameter error.

406 – Socket not found.

409 – Send failed.

410 – Socket not connected.

416 – Out of memory.

420 – Send request failed.

429 – Internal error.

The set command parameters and their defined values are the following:

<fd>

Integer, 0–11. Socket file descriptor returned by the **%SKTCREATE** command.

<data_len>

Integer, 1–1400. Length of data in bytes.

<data>

String. Maximum length 2800 characters. Data in hexadecimal format. Each byte is represented as two hexadecimal characters.

<rai_info>

Integer, 0–2. Release assistance indication.

0 – No release assistance indication (default).

1 – No further uplink or downlink 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.

<except_info>

Integer, 0–1. Mark transmitted data as NAS exception data.

0 – Disabled (default).

1 – Enabled.

The following command example sends the five–byte ASCII string "23456" in hexadecimal format:

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

11.4.2 Read command

The read command is not supported.

11.4.3 Test command

The test command returns supported values.

Response syntax:

```
%SKTSEND: (range of supported <fd>s),(range of supported <data_len>s),(<data>),(range of
 supported <rai_info>s),(list of supported <except_info>s)
```

The test command parameters and their defined values are the following:

<fd>

Integer, 0–11. Socket file descriptor returned by the **%SKTCREATE** command.

<data_len>

Integer, 1–1400. Length of data in bytes.

<data>

String. Maximum length 2800 characters. Data in hexadecimal format. Each byte is represented as two hexadecimal characters.

<rai_info>

Integer, 0–2. Release assistance indication.

0 – No release assistance indication (default).

1 – No further uplink or downlink 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.

<except_info>

Integer, 0–1. Mark transmitted data as NAS exception data.

0 – Disabled (default).

1 – Enabled.

The following command example returns the supported values:

```
AT%SKTSEND=?
%SKTSEND: (0-11),(1-1400),"data",(0-2),(0,1)

OK
```

11.5 Send data in transparent mode %SKTSENDT

The proprietary **%SKTSENDT** command sends data to a remote address and port using transparent mode.
v1.4.x

11.5.1 Set command

The set command sends data to the remote address and port using transparent mode. The modem prompts for hexadecimal data input after the command is accepted.

If the <data_len> parameter is specified, the modem outputs > and waits for exactly the <data_len> bytes of data in hexadecimal format. When the expected amount of data is received, the modem sends the data and returns OK.

If <data_len> is omitted, the modem outputs > and accepts data in hexadecimal format until the input is terminated with **Ctrl+Z** (0x1A).

Syntax:

```
%SKTSENDT=<fd>[,<data_len>[,<rai_info>[,<except_info>]]]
```

+CME ERROR code

402 – Parameter error.

406 – Socket not found.

409 – Send failed.

410 – Socket not connected.

416 – Out of memory.

420 – Send request failed.

429 – Internal error.

The set command parameters and their defined values are the following:

<fd>

Integer, 0–11. Socket file descriptor returned by the **%SKTCREATE** command.

<data_len>

Integer, 1–1400. Length of data in bytes. If omitted, the input must be terminated with **Ctrl+Z**.

<rai_info>

Integer, 0–2. Release assistance indication.

0 – No release assistance indication (default).

1 – No further uplink or downlink 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.

<except_info>

Integer, 0–1. Mark transmitted data as NAS exception data.

0 – Disabled (default).

1 – Enabled.

The following command example sends 5 bytes with a specified length:

```
AT%SKTSENDT=0,5,0,0
> 68656C6C6F
OK
```

The following command example sends data without a specified length and terminated with **Ctrl+Z**:

```
AT%SKTSENDT=0
> 3233343536\1A
OK
```

11.5.2 Read command

The read command is not supported.

11.5.3 Test command

The test command returns supported values.

Response syntax:

```
%SKTSENDT: (range of supported <fd>s),(range of supported <data_len>s),(range of supported
 <rai_info>s),
(list of supported <except_info>s)
```

The test command parameters and their defined values are the following:

<fd>

Integer, 0–11. Socket file descriptor returned by the **%SKTCREATE** command.

<data_len>

Integer, 1–1400. Length of data in bytes. If omitted, the input must be terminated with **Ctrl+Z**.

<rai_info>

Integer, 0–2. Release assistance indication.

0 – No release assistance indication (default).

1 – No further uplink or downlink 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.

<except_info>

Integer, 0–1. Mark transmitted data as NAS exception data.

0 – Disabled (default).

1 – Enabled.

The following command example returns the supported values:

```
AT%SKTSENDT=?
%SKTSENDT: (0-11),(1-1400),(0-2),(0,1)
OK
```

11.6 Delete socket %SKTDELETE

The proprietary **%SKTDELETE** command deletes a socket. v1.4.x

11.6.1 Set command

The set command deletes a socket or all sockets.

Syntax for deleting a socket:

```
%SKTDELETE=<fd>
```

Syntax for deleting all sockets:

```
%SKTDELETE="all"
```

+CME ERROR code

402 – Parameter error.

406 – Socket not found.

414 – Delete failed.

416 – Out of memory.

420 – Send request failed.

429 – Internal error.

The set command parameters and their defined values are the following:

<fd>

Integer, 0–11. Socket file descriptor returned by **%SKTCREATE**.

"all"

String. Closes all open socket file descriptors and deletes all associated sockets.

The following command example deletes socket `0`:

```
AT%SKTDELETE=0
OK
```

11.6.2 Read command

The read command is not supported.

11.6.3 Test command

The test command returns supported values.

Response syntax:

```
%SKTDELETE: (range of supported <fd>s,"all")
```

The test command parameter and its defined values are the following:

<fd>

Integer, 0–11. Socket file descriptor returned by **%SKTCREATE**.

"all"

String. Closes all open socket file descriptors and deletes all associated sockets.

The following command example returns the supported values:

```
AT%SKTDELETE=?
%SKTDELETE: (0-11,"all")
OK
```

11.7 Socket status %SKTSTATUS

The proprietary **%SKTSTATUS** command returns the status of a socket. v1.4.x

11.7.1 Set command

The set command returns the connection status of a socket.

Syntax:

```
%SKTSTATUS=<fd>
```

Response syntax:

```
%SKTSTATUS: <status>
```

+CME ERROR code

402 – Parameter error.

406 – Socket not found.

416 – Out of memory.

420 – Send request failed.

429 – Internal error.

The set command parameters and their defined values are the following:

<fd>

Integer, 0–11. Socket file descriptor returned by **%SKTCREATE**.

<status>

Integer, 1–3. Socket connection status.

1 – Not connected.

2 – Connecting.

3 – Connected.

The following command example returns the status of socket 0:

```
AT%SKTSTATUS=0
%SKTSTATUS: 3

OK
```

11.7.2 Read command

The read command is not supported.

11.7.3 Test command

The test command returns supported values.

Response syntax:

```
%SKTSTATUS: (list of supported <fd>s)
```

The test command parameter and its defined value are the following:

<fd>

Integer, 0–11. Socket file descriptor returned by **%SKTCREATE**.

The following command example returns the supported values:

```
AT%SKTSTATUS=?
%SKTSTATUS: (0-11)
OK
```

11.8 Socket receive notification %SKTRECV

The proprietary **%SKTRECV** *URC* indicates that data is received on a socket. v1.4.x

Notification syntax:

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

The notification parameters and their defined values are the following:

<fd>

Integer, 0–11. Socket file descriptor that received the data.

<len>

Integer. Received data length in bytes.

<data>

String. Maximum length 2800 characters. Received data in hexadecimal format.

The following notification example shows `5` bytes are received on socket `0`:

```
%SKTRECV: 0,5,"68656C6C6F"
```

11.9 Socket error notification %SKTERR

The proprietary **%SKTERR** *URC* indicates that a socket error has occurred. v1.4.x

Notification syntax:

```
%SKTERR: <fd>,<errno>
```

The notification parameters and their defined values are the following:

<fd>

Integer, 0–11. Socket file descriptor where the error occurred.

<errno>

Integer. POSIX error number.

5 – Invalid argument.

6 – Out of memory error.

9 – Bad file descriptor.

11 – Operation would block.

23 – Too many open files in system.

34 – I/O error.

88 – Socket operation on non-socket.

89 – Destination address required.

90 – Message too long.

91 – Protocol wrong type for socket.

92 – Protocol not available.

93 – Protocol not supported.

95 – Operation not supported.

97 – Address family not supported.

98 – Address already in use.

99 – Cannot assign requested address.

100 – Network is down.

101 – Network unreachable.

102 – Network dropped connection on reset.

103 – Connection aborted.

104 – Connection reset by peer.

105 – No buffer space available.

106 – Socket already connected.

107 – Socket not connected or closed.

108 – Cannot send after transport endpoint shutdown.

110 – Connection timed out.

111 – Connection refused.

112 – Host is down.

113 – No route to host.

114 – Connection already in progress.

115 – Operation in progress.

155 – Low-level interface out of service.

Note: If an error occurs on a connected socket, the socket is closed automatically.

11.10 Listen on socket %SKTLISTEN

The proprietary **%SKTLISTEN** command puts a bound TCP socket into the listening state so that it can accept incoming client connections. v1.4.x

11.10.1 Set command

The set command marks a bound TCP socket as passive so that it can accept incoming client connections.

Syntax:

```
%SKTLISTEN=<fd>[,<backlog>]
```

+CME ERROR code

402 – Parameter error.

405 – Operation not supported.

406 – Socket not found.

416 – Out of memory.

420 – Send request failed.

424 – Listen failed.

429 – Internal error.

The set command parameters and their defined values are the following:

<fd>

Integer, 0–11. Socket file descriptor returned by **%SKTCREATE**. The socket must be a TCP socket that has already been bound with **%SKTBIND**.

<backlog>

Integer, 1–4. Maximum number of half-open connections queued by the TCP stack before new SYN requests are rejected. The default is `1`.

When a remote client completes the TCP handshake, the module reports a `%SKTACCEPT: <fd>` *URC*, where <fd> is the listening socket. The host must execute **%SKTACCEPT** with that file descriptor to promote the pending client into a usable accepted socket.

The following command example enables listening on TCP socket `0` with backlog `4`:

```
AT%SKTCREATE=1,1,6
%SKTCREATE: 0
OK
AT%SKTBIND=0,,5000
OK
AT%SKTLISTEN=0,4
OK
```

11.10.2 Read command

The read command is not supported.

11.10.3 Test command

The test command returns the supported parameter values.

Response syntax:

```
%SKTLISTEN: (0-11),(1-4)
```

The test command parameters and their defined values are the following:

<fd>

Integer, 0–11. Socket file descriptor returned by **%SKTCREATE**. The socket must be a TCP socket that has already been bound with **%SKTBIND**.

<backlog>

Integer, 1–4. Maximum number of half-open connections queued by the TCP stack before new SYN requests are rejected. The default is `1`.

The following command example returns the supported parameter values:

```
AT%SKTLISTEN=?
%SKTLISTEN: (0-11),(1-4)
OK
```

11.11 Accept pending client %SKTACCEPT

The proprietary **%SKTACCEPT** command promotes a pending TCP client connection into an accepted socket that can be used for data transfer. v1.4.x

11.11.1 Set command

The set command dequeues the next pending client from a listening TCP socket and returns a new socket for data transfer.

Syntax:

```
%SKTACCEPT=<fd>
```

Response syntax:

```
%SKTACCEPT: <client_fd>,"<client_ip>",<client_port>
```

+CME ERROR code

402 – Parameter error.

406 – Socket not found.

412 – Invalid state.

416 – Out of memory.

420 – Send request failed.

429 – Internal error.

The set command parameters and their defined values are the following:

<fd>

Integer, 0–11. File descriptor of the listening socket.

<client_fd>

Integer, 0–11. File descriptor of the accepted client socket.

<client_ip>

String. Peer IP address in IPv4 dotted-decimal or IPv6 text format.

<client_port>

Integer. Peer TCP port.

After the client is promoted, all subsequent data transfer uses <client_fd>. Incoming data for the accepted socket is reported with the **%SKTRECV** unsolicited result code. Any data already buffered for the client can be reported immediately after the accept operation.

The following command example accepts a pending client from listener `0`:

```
AT%SKTACCEPT=0
%SKTACCEPT: 1,"172.30.8.4",33180
OK
%SKTRECV: 1,5,"68656C6C6F"
%SKTERR: 1,107
```

11.11.2 Read command

The read command is not supported.

11.11.3 Test command

The test command returns the supported parameter values.

Response syntax:

```
%SKTACCEPT: (0-11)
```

The test command parameter and its defined value are the following:

<fd>

Integer, 0–11. File descriptor of the listening socket.

The following command example returns the supported parameter values:

```
AT%SKTACCEPT=?
%SKTACCEPT: (0-11)
OK
```

11.12 Send data to a destination %SKTSENDTO

The proprietary **%SKTSENDTO** command sends a single UDP datagram to an explicitly specified destination address and port without a prior **%SKTCONNECT** similarly to posix sendto(). v1.4.x

11.12.1 Set command

The set command sends a single UDP datagram to the specified destination address and port.

Syntax:

```
%SKTSENDTO=<fd>,<address>,<port>,<data_len>,<data>[,<rai_info>[,<except_info>]]
```

+CME ERROR code

402 – Parameter error.

406 – Socket not found.

409 – Send failed.

412 – Invalid state.

416 – Out of memory.

420 – Send request failed.

423 – DNS resolve failed.

429 – Internal error.

The set command parameters and their defined values are the following:

<fd>

Integer, 0–11. Socket file descriptor returned by the **%SKTCREATE** command.

<address>

String. Remote address to connect or send to. *IP* address or hostname of the target host. The maximum size is 63 characters

<port>

Integer, 1–65535. Remote port to connect or send to.

<data_len>

Integer, 1–1400. Length of data in bytes.

<data>

String. Maximum length 2800 characters. Data in hexadecimal format. Each byte is represented as two hexadecimal characters.

<rai_info>

Integer, 0–2. Release assistance indication.

0 – No release assistance indication (default).

1 – No further uplink or downlink 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.

<except_info>

Integer, 0–1. Mark transmitted data as NAS exception data.

0 – Disabled (default).

1 – Enabled.

The following command example sends the five–byte ASCII string "23456" in hexadecimal format to the specified destination:

```
AT%SKTSENDTO=0,"8.8.8.8",80,5,3233343536
OK
```

11.12.2 Read command

The read command is not supported.

11.12.3 Test command

The test command returns supported values.

Response syntax:

```
%SKTSENDTO: (range of supported <fd>s),(<address>),(range of supported <port>s),
(range of supported <data_len>s),(<data>),(range of supported <rai_info>s),(list of
 supported <except_info>s)
```

The test command parameters and their defined values are the following:

<fd>

Integer, 0–11. Socket file descriptor returned by the **%SKTCREATE** command.

<address>

String. Remote address to connect or send to. *IP* address or hostname of the target host. The maximum size is 63 characters

<port>

Integer, 1–65535. Remote port to connect or send to.

<data_len>

Integer, 1–1400. Length of data in bytes.

<data>

String. Maximum length 2800 characters. Data in hexadecimal format. Each byte is represented as two hexadecimal characters.

<rai_info>

Integer, 0–2. Release assistance indication.

0 – No release assistance indication (default).

1 – No further uplink or downlink 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.

<except_info>

Integer, 0–1. Mark transmitted data as NAS exception data.

0 – Disabled (default).

1 – Enabled.

The following command example returns the supported values:

```
AT%SKTSENDTO=?
%SKTSENDTO: (0-11),"IP ADDR\URL",(1-65535),(1-4096),"data",(0-2),(0,1)
OK
```

12 Network service related commands

For reference, see *3GPP 27.007 Ch. 7*.

12.1 PLMN selection +COPS

The **+COPS** command selects a *PLMN* automatically or manually and reads and searches the current mobile network. v1.4.x

For reference, see *3GPP 27.007 Ch. 7.3*.

12.1.1 Set command

The set command selects a mobile network automatically or manually. The selection is stored to NVRAM automatically, and it is not lost if the module is rebooted. If the selected operator network is not available, no other operator shall be selected except when <mode> is set to `4`.

The selected mode affects all further network registrations. For example, when <mode> is set to `2`, *MT* shall be unregistered until <mode> `0` or `1` is selected.

Syntax when <mode> is `0`:

```
+COPS=<mode>
```

Syntax when <mode> is `1` or `4`. <format> must be `2`:

```
+COPS=<mode>,<format>,<oper>[,<AcT>]
```

Syntax when <mode> is `3`:

```
+COPS=<mode>,<format>
```

+CME ERROR code

50 – Incorrect parameters.

The set command parameters and their defined values are the following:

<mode>

Integer, 0–4. Network registration mode.

0 – Automatic network selection. <oper> is ignored (default).

1 – Manual network selection. <oper> shall be present. <AcT> is optional.

2 – Force an attempt to deregister from network immediately.

3 – Set <format> for the **+COPS** read command response.

4 – Manual or automatic. <oper> shall be present. If manual selection fails, automatic network selection mode `0` is used.

<format>

Integer, 0–2. Operator name format.

0 – Long alphanumeric <oper> format.

1 – Short alphanumeric <oper> format.

2 – Numeric *Mobile Country Code (MCC)* and *Mobile Network Code (MNC)* <oper> format.

<oper>

String in double quotes. Maximum length 16 characters. Operator name or MCC and MNC values.

<AcT>

Integer, 7. Preferred access technology.

7 – *E-UTRAN*.

Note: The selected operator name format applies also to **+COPS** commands.

Note: On the first boot or after a factory reset, a **+COPS** command that is sent before initializing *SIM* is overridden. **+COPS** commands should be sent only after `+CFUN=1` or `+CFUN=4` has been executed. Once the *International Mobile Subscriber Identity (IMSI)* is stored in *NVM*, a **+COPS** command can be sent at any time.

For manual selection, only the numeric string format is supported and <oper> is mandatory.

The following command example selects automatic network selection:

```
AT+COPS=0
OK
```

The following command manually selects network `24407`:

```
AT+COPS=1,2,"24407"
OK
```

12.1.2 Read command

The read command reads the current mobile network.

Response syntax:

```
+COPS: <mode>[,<format>,<oper>,[<AcT>]]
```

The read command parameters and their defined values are the following:

<mode>

Integer, 0–4. Network registration mode.

0 – Automatic network selection. <oper> is ignored (default).

1 – Manual network selection. <oper> shall be present. <AcT> is optional.

2 – Force an attempt to deregister from network immediately.

3 – Set <format> for the **+COPS** read command response.

4 – Manual or automatic. <oper> shall be present. If manual selection fails, automatic network selection mode `0` is used.

<format>

Integer, 0–2. Operator name format.

0 – Long alphanumeric <oper> format.

1 – Short alphanumeric <oper> format.

2 – Numeric *MCC* and *MNC* <oper> format.

<oper>

String in double quotes. Maximum length 16 characters. Operator name or MCC and MNC values.

<AcT>

Integer, 7. Preferred access technology.

7 – *E-UTRAN*.

The following command example reads the current network selection mode and network:

```
AT+COPS?
+COPS: 0,2,"24491",7
OK
```

The following command example reads the current network selection mode and network with the operator name in alphanumeric format:

```
AT+COPS?
+COPS: 0,0,"Telia FI",7
OK
```

12.1.3 Test command

The test command searches the mobile network and returns a list of found operators. If the search is interrupted, the search returns existing results and the list can be incomplete.

If the modem is registered to a network without failures and camped on a cell, the search is restricted to the currently active system.

Response syntax:

```
+COPS: [(<stat>,long alphanumeric <oper>,short alphanumeric <oper>,numeric
 <oper>[,<AcT>])s],,
(list of supported <mode>s),(list of supported <format>s)
```

The test command parameters and their defined values are the following:

<stat>

Integer, 0–4. Operator availability status.

0 – Unknown status.

1 – Available for registration.

2 – Currently registered.

3 – Forbidden.

<mode>

Integer, 0–4. Network registration mode.

0 – Automatic network selection. <oper> is ignored (default).

1 – Manual network selection. <oper> shall be present. <AcT> is optional.

2 – Force an attempt to deregister from network immediately.

3 – Set <format> for the **+COPS** read command response.

4 – Manual or automatic. <oper> shall be present. If manual selection fails, automatic network selection mode `0` is used.

<format>

Integer, 0–2. Operator name format.

0 – Long alphanumeric <oper> format.

1 – Short alphanumeric <oper> format.

2 – Numeric *MCC* and *MNC* <oper> format.

<oper>

String in double quotes. Maximum length 16 characters. Operator name or MCC and MNC values.

<AcT>

Integer, 7. Preferred access technology.

7 – *E-UTRAN*.

Note:

- The time required to perform a network search depends on device configuration and network conditions. The maximum is `305` seconds.
- If the selected operator name format is unavailable, an empty field is returned.

The following command example is used for a manual network search:

```
AT+COPS=?
+COPS: (0,"","","24445",7),(0,"","","24405",7),(0,"","","24446",7),(0,"","","24491",7),
(0,"","","24412",7),,(0-4),(0-2)
OK
```

12.2 Power saving mode setting +CPSMS

The **+CPSMS** command controls *PSM* settings. v1.4.x

For reference, see *3GPP 27.007 Ch. 7.38*.

12.2.1 Set command

The set command sets the power saving mode. It sets activity timer and *PSM* period after *NAS* signaling connection release.

Syntax:

```
+CPSMS=[<mode>[,<Requested_Periodic-RAU>,<Requested_GPRS-READY-timer>,
<Requested_Periodic-TAU-ext>[,<Requested_Active-Time>]]]
```

The set command parameters and their defined values are the following:

<mode>

Integer, 0–2. PSM mode.

0 – Disable power saving mode (default).

1 – Enable power saving mode.

2 – Disable power saving mode and discard all parameters.

<Requested_Periodic-RAU>

String. Requested extended periodic RAU. Not supported.

<Requested_GPRS-READY-timer>

String. Requested GPRS READY timer value. Not supported.

<Requested_Periodic-TAU-ext>

String. 1 byte in 8-bit format.

Indicates the extended periodic TAU value (T3412_EXT extended value) allocated to the device in E-UTRAN. For the coding and value range, see the GPRS Timer 3 IE in *3GPP TS 24.008* Table 10.5.163a/3GPP TS 24.008.

GPRS Timer 3 value (octet 3).

Bits 5 to 1 represent the binary coded timer value.

Bits 8 to 6 define the timer value unit for the *GPRS* timer as follows:

Bits

8 7 6

0 0 0 – Value is incremented in multiples of 10 min.

0 0 1 – Value is incremented in multiples of 1 h.

0 1 0 – Value is incremented in multiples of 10 h.

0 1 1 – Value is incremented in multiples of 2 s.

1 0 0 – Value is incremented in multiples of 30 s.

1 0 1 – Value is incremented in multiples of 1 min.

1 1 0 – Value is incremented in multiples of 320 h.

> **Note:** Interpreted as 1 hour if the request sent to the network is not integrity protected. After registration, check the value given by the network (see **+CEREG**). If 1 hour unit is given, disable and enable PSM using the `+CPSMS=0` and `+CPSMS=1` commands.

1 1 1 – Value indicates that the timer is deactivated.

> **Note:** If the *USIM* profile in use is a Verizon one, the minimum value for <Requested_Periodic-TAU-ext> is 190 minutes.

<Requested_Active-Time>

String. 1 byte in 8-bit format.

Optional. Timer value updated if present. If not present, the value of the requested Active-Time is set to the manufacturer-specific default. For the coding and value range, see the GPRS Timer 2 IE in *3GPP TS 24.008* Table 10.5.163/3GPP TS 24.008.

GPRS Timer 2 value (octet 3).

Bits 5 to 1 represent the binary coded timer value.

Bits 8 to 6 define the timer value unit for the GPRS timer as follows:

Bits

8 7 6

0 0 0 – Value is incremented in multiples of 2 s.

0 0 1 – Value is incremented in multiples of 1 min.

0 1 0 – Value is incremented in multiples of 6 min.

1 1 1 – Value indicates that the timer is deactivated.

The following command example enables power saving mode with Periodic-TAU set to `10` minutes and Active-Time set to `10` seconds:

```
AT+CPSMS=1,,,"10101010","00000101"
OK
```

The following command example disables power saving mode:

```
AT+CPSMS=0
OK
```

The following command example disables power saving mode and discards all parameters:

```
AT+CPSMS=2
OK
```

12.2.2 Read command

The read command reads the current *PSM* settings.

Response syntax:

```
+CPSMS: <mode>,[<Requested_Periodic-RAU>],[<Requested_GPRS-READY-timer>],
[<Requested_Periodic-TAU-ext>],
[<Requested_Active-Time>]
```

The read command parameters and their defined values are the following:

<mode>

Integer, 0–2. PSM mode.

0 – Disable power saving mode (default).

1 – Enable power saving mode.

2 – Disable power saving mode and discard all parameters.

<Requested_Periodic-RAU>

String. Requested extended periodic RAU. Not supported.

<Requested_GPRS-READY-timer>

String. Requested GPRS READY timer value. Not supported.

<Requested_Periodic-TAU-ext>

String. 1 byte in 8-bit format.

Indicates the extended periodic TAU value (T3412_EXT extended value) allocated to the device in E-UTRAN. For the coding and value range, see the GPRS Timer 3 IE in *3GPP TS 24.008* Table 10.5.163a/3GPP TS 24.008.

GPRS Timer 3 value (octet 3).

Bits 5 to 1 represent the binary coded timer value.

Bits 8 to 6 define the timer value unit for the *GPRS* timer as follows:

Bits

8 7 6

0 0 0 – Value is incremented in multiples of 10 min.

0 0 1 – Value is incremented in multiples of 1 h.

0 1 0 – Value is incremented in multiples of 10 h.

0 1 1 – Value is incremented in multiples of 2 s.

1 0 0 – Value is incremented in multiples of 30 s.

1 0 1 – Value is incremented in multiples of 1 min.

1 1 0 – Value is incremented in multiples of 320 h.

> **Note:** Interpreted as 1 hour if the request sent to the network is not integrity protected. After registration, check the value given by the network (see **+CEREG**). If 1 hour unit is given, disable and enable PSM using the `+CPSMS=0` and `+CPSMS=1` commands.

1 1 1 – Value indicates that the timer is deactivated.

> **Note:** If the *USIM* profile in use is a Verizon one, the minimum value for <Requested_Periodic-TAU-ext> is 190 minutes.

<Requested_Active-Time>

String. 1 byte in 8-bit format.

Optional. Timer value updated if present. If not present, the value of the requested Active-Time is set to the manufacturer-specific default. For the coding and value range, see the GPRS Timer 2 IE in *3GPP TS 24.008* Table 10.5.163/3GPP TS 24.008.

GPRS Timer 2 value (octet 3).

Bits 5 to 1 represent the binary coded timer value.

Bits 8 to 6 define the timer value unit for the GPRS timer as follows:

Bits

8 7 6

0 0 0 – Value is incremented in multiples of 2 s.

0 0 1 – Value is incremented in multiples of 1 min.

0 1 0 – Value is incremented in multiples of 6 min.

1 1 1 – Value indicates that the timer is deactivated.

The following command example reads the current power saving mode settings:

```
AT+CPSMS?
+CPSMS: 1,,,"10101111","01101100"
OK
```

12.2.3 Test command

The test command returns the supported values.

Response syntax:

```
+CPSMS: (range of supported <mode>s),(list of supported <Requested_Periodic-RAU>s),
(list of supported <Requested_GPRS-READY-timer>s),(range of supported <Requested_Periodic-
TAU-ext>s),
(range of supported <Requested_Active-Time>s)
```

The test command parameters and their defined values are the following:

<mode>

Integer, 0–2. PSM mode.

0 – Disable power saving mode (default).

1 – Enable power saving mode.

2 – Disable power saving mode and discard all parameters.

<Requested_Periodic-RAU>

String. Requested extended periodic RAU. Not supported.

<Requested_GPRS-READY-timer>

String. Requested GPRS READY timer value. Not supported.

<Requested_Periodic-TAU-ext>

String. 1 byte in 8-bit format.

Indicates the extended periodic TAU value (T3412_EXT extended value) allocated to the device in E-UTRAN. For the coding and value range, see the GPRS Timer 3 IE in *3GPP TS 24.008* Table 10.5.163a/3GPP TS 24.008.

GPRS Timer 3 value (octet 3).

Bits 5 to 1 represent the binary coded timer value.

Bits 8 to 6 define the timer value unit for the *GPRS* timer as follows:

Bits

8 7 6

0 0 0 – Value is incremented in multiples of 10 min.

0 0 1 – Value is incremented in multiples of 1 h.

0 1 0 – Value is incremented in multiples of 10 h.

0 1 1 – Value is incremented in multiples of 2 s.

1 0 0 – Value is incremented in multiples of 30 s.

1 0 1 – Value is incremented in multiples of 1 min.

1 1 0 – Value is incremented in multiples of 320 h.

> **Note:** Interpreted as 1 hour if the request sent to the network is not integrity protected. After registration, check the value given by the network (see **+CEREG**). If 1 hour unit is given, disable and enable PSM using the `+CPSMS=0` and `+CPSMS=1` commands.

1 1 1 – Value indicates that the timer is deactivated.

> **Note:** If the *USIM* profile in use is a Verizon one, the minimum value for <Requested_Periodic-TAU-ext> is 190 minutes.

<Requested_Active-Time>

String. 1 byte in 8-bit format.

Optional. Timer value updated if present. If not present, the value of the requested Active-Time is set to the manufacturer-specific default. For the coding and value range, see the GPRS Timer 2 IE in *3GPP TS 24.008* Table 10.5.163/3GPP TS 24.008.

GPRS Timer 2 value (octet 3).

Bits 5 to 1 represent the binary coded timer value.

Bits 8 to 6 define the timer value unit for the GPRS timer as follows:

Bits

8 7 6

0 0 0 – Value is incremented in multiples of 2 s.

0 0 1 – Value is incremented in multiples of 1 min.

0 1 0 – Value is incremented in multiples of 6 min.

1 1 1 – Value indicates that the timer is deactivated.

The following command example returns the supported values:

```
AT+CPSMS=?
+CPSMS: (0-2),,,("00000000"-"11111111"),("00000000"-"11111111")
OK
```

12.3 eDRX setting +CEDRXS

The **+CEDRXS** command controls the setting of *eDRX* parameters. v1.4.x

For reference, see *3GPP 27.007 Ch. 7.40*.

12.3.1 Set command

The set command sets the requested *eDRX* parameters.

Deactivating the modem clears the unsolicited notification requests and, therefore, the configuration for the unsolicited notification must be set again.

When an eDRX parameter is changed, the default *Paging Time Window (PTW)* is set. If the default PTW is not used, the **%XPTW** command must be sent after the **+CEDRXS** command.

To ensure a proper amount of paging occasions and time for reception with all possible DRX cycle lengths, it is recommended to set the PTW length to ≥5.12 s.

> **Note:** Use a shorter PTW length than the eDRX length provided by the network.

Syntax:

```
+CEDRXS=[<mode>,[,<AcT-type>[,<Requested_eDRX_value>]]]
```

URC syntax:

```
+CEDRXP: <AcT-type>[,<Requested_eDRX_value>[,<NW-provided_eDRX_value>
 [,<Paging_time_window>]]]
```

The set command parameters and their defined values are the following:

<mode>

Integer, 0–3. Controls whether eDRX is enabled and whether the **+CEDRXP** URC is sent on changes.

0 – Disable the use of eDRX and disable the URC (default).

1 – Enable the use of eDRX and disable the URC.

2 – Enable the use of eDRX and enable the URC.

3 – Disable the use of eDRX and disable the URC. Discard all parameters for eDRX.

<AcT-type>

Integer, 0 and 4. Radio Access technology for which eDRX is configured. The default is `4`.

0 – Current cell is not using eDRX or network configuration does not support eDRX. Used only in the URC.

4 – *E-UTRAN* (WB-S1 mode).

<Requested_eDRX_value>

String. Requested eDRX cycle length as a 4–bit binary string. Refers to bits 4–1 of octet 3 of the Extended DRX parameters (see *3GPP TS 24.008, subclause 10.5.5.32*). If omitted, the value of the requested eDRX period is set to the manufacturer-specific default.

Bit

4 3 2 1 – E-UTRAN eDRX cycle length duration in seconds.

0 0 0 0 – 5.12 s.

0 0 0 1 – 10.24 s.

0 0 1 0 – 20.48 s.

0 0 1 1 – 40.96 s.

0 1 0 0 – 61.44 s.

0 1 0 1 – 81.92 s.

0 1 1 0 – 102.4 s.

0 1 1 1 – 122.88 s.

1 0 0 0 – 143.36 s.

1 0 0 1 – 163.84 s.

1 0 1 0 – 327.68 s.

1 0 1 1 – 655.36 s.

1 1 0 0 – 1310.72 s.

1 1 0 1 – 2621.44 s.

1 1 1 0 – 5242.88 s.

1 1 1 1 – 10485.76 s.

<NW-Provided_eDRX_value>

String. The eDRX cycle length granted by the network as a 4–bit binary string. Refers to bits 4–1 of octet 3 of the Extended DRX parameters (see *3GPP TS 24.008, subclause 10.5.5.32*).

Bit

4 3 2 1 – E-UTRAN eDRX cycle length duration in seconds.

0 0 0 0 – 5.12 s

0 0 0 1 – 10.24 s

0 0 1 0 – 20.48 s.

0 0 1 1 – 40.96 s.

0 1 0 0 – 61.44 s

0 1 0 1 – 81.92 s.

0 1 1 0 – 102.4 s

0 1 1 1 – 122.88 s

1 0 0 0 – 143.36 s

1 0 0 1 – 163.84 s.

1 0 1 0 – 327.68 s.

1 0 1 1 – 655.36 s.

1 1 0 0 – 1310.72 s.

1 1 0 1 – 2621.44 s.

1 1 1 0 – 5242.88 s.

1 1 1 1 – 10485.76 s.

<Paging_time_window>

String. The network assigned paging time window duration as a 4–bit binary string. Refers to bit 8 to 5 of octet 3 of the Extended DRX parameters information element (see *3GPP TS 24.008, subclause 10.5.5.32*).

Bit

4 3 2 1 – PTW length in seconds.

0 0 0 0 – 1.28 s.

0 0 0 1 – 2.56 s.

0 0 1 0 – 3.84 s.

0 0 1 1 – 5.12 s.

0 1 0 0 – 6.4 s.

0 1 0 1 – 7.68 s.

0 1 1 0 – 8.96 s.

0 1 1 1 – 10.24 s.

1 0 0 0 – 11.52 s.

1 0 0 1 – 12.8 s.

1 0 1 0 – 14.08 s.

1 0 1 1 – 15.36 s.

1 1 0 0 – 16.64 s.

1 1 0 1 – 17.92 s.

1 1 1 0 – 19.2 s.

1 1 1 1 – 20.48 s.

The following command example enables eDRX and sets the requested eDRX value:

```
AT+CEDRXS=1,4,"1000"
OK
```

The unsolicited notification when <mode> `2` is used:

```
+CEDRXP: 4,"1000","0101","1011"
OK
```

12.3.2 Read command

The read command is used to read the requested *eDRX* parameters.

If eDRX has been disabled by `+CEDRXS=0` or `+CEDRXS=3`, the modem replies with an empty information response `OK`.

Response syntax:

```
+CEDRXS: <AcT-type>,<Requested_eDRX_value>
```

The read command parameters and their defined values are the following:

<AcT-type>

Integer, 0 and 4. Radio Access technology for which eDRX is configured. The default is `4`.

0 – Current cell is not using eDRX or network configuration does not support eDRX. Used only in the URC.

4 – *E-UTRAN* (WB-S1 mode).

<Requested_eDRX_value>

String. Requested eDRX cycle length as a 4–bit binary string. Refers to bits 4–1 of octet 3 of the Extended DRX parameters (see *3GPP TS 24.008, subclause 10.5.5.32*). If omitted, the value of the requested eDRX period is set to the manufacturer-specific default.

Bit

4 3 2 1 – E-UTRAN eDRX cycle length duration in seconds.

0 0 0 0 – 5.12 s.

0 0 0 1 – 10.24 s.

0 0 1 0 – 20.48 s.

0 0 1 1 – 40.96 s.

0 1 0 0 – 61.44 s.

0 1 0 1 – 81.92 s.

0 1 1 0 – 102.4 s.

0 1 1 1 – 122.88 s.

1 0 0 0 – 143.36 s.

1 0 0 1 – 163.84 s.

1 0 1 0 – 327.68 s.

1 0 1 1 – 655.36 s.

1 1 0 0 – 1310.72 s.

1 1 0 1 – 2621.44 s.

1 1 1 0 – 5242.88 s.

1 1 1 1 – 10485.76 s.

Note: The actual eDRX cycle length granted by the network might differ from the requested length. Follow **+CEDRXS** URC for the actually used values.

The following command example reads the requested eDRX value:

```
AT+CEDRXS?
+CEDRXS: 4,"0011"
OK
```

12.3.3 Test command

The test command returns the supported *eDRX* parameters as compound values.

Response syntax:

```
+CEDRXS: (list of supported <mode>s),(list of supported <AcT-type>s),(range of supported
 <Requested_eDRX_value>s)
```

The test command parameters and their defined values are the following:

<mode>

Integer, 0–3. Controls whether eDRX is enabled and whether the **+CEDRXP** URC is sent on changes.

0 – Disable the use of eDRX and disable the URC (default).

1 – Enable the use of eDRX and disable the URC.

2 – Enable the use of eDRX and enable the URC.

3 – Disable the use of eDRX and disable the URC. Discard all parameters for eDRX.

<AcT-type>

Integer, 0 and 4. Radio Access technology for which eDRX is configured. The default is `4`.

0 – Current cell is not using eDRX or network configuration does not support eDRX. Used only in the URC.

4 – *E-UTRAN* (WB-S1 mode).

<Requested_eDRX_value>

String. Requested eDRX cycle length as a 4–bit binary string. Refers to bits 4–1 of octet 3 of the Extended DRX parameters (see *3GPP TS 24.008, subclause 10.5.5.32*). If omitted, the value of the requested eDRX period is set to the manufacturer-specific default.

Bit

4 3 2 1 – E-UTRAN eDRX cycle length duration in seconds.

0 0 0 0 – 5.12 s.

0 0 0 1 – 10.24 s.

0 0 1 0 – 20.48 s.

0 0 1 1 – 40.96 s.

0 1 0 0 – 61.44 s.

0 1 0 1 – 81.92 s.

0 1 1 0 – 102.4 s.

0 1 1 1 – 122.88 s.

1 0 0 0 – 143.36 s.

1 0 0 1 – 163.84 s.

1 0 1 0 – 327.68 s.

1 0 1 1 – 655.36 s.

1 1 0 0 – 1310.72 s.

1 1 0 1 – 2621.44 s.

1 1 1 0 – 5242.88 s.

1 1 1 1 – 10485.76 s.

The following command example reads the supported parameter values:

```
AT+CEDRXS=?
+CEDRXS: (0,1,2,3),(4),("0000"-"1111")
OK
```

12.4 EDRX dynamic parameters +CEDRXRDP

The **+CEDRXRDP** command reads dynamic *eDRX* parameters. v1.4.x

For reference, see *3GPP 27.007 Ch. 7.41*.

12.4.1 Set command

The set command reads dynamic *eDRX* parameters.

Syntax:

```
+CEDRXRDP
```

Response syntax:

```
+CEDRXRDP: <AcT-type>[,<Requested_eDRX_value>[,<NW-
provided_eDRX_value>[,<Paging_time_window>]]]
```

<AcT-type>

Integer, 0 and 4. Radio Access technology for which eDRX is configured.

0 – Current cell is not using eDRX or network configuration does not support eDRX.

4 – *E-UTRAN* (WB-S1 mode).

<Requested_eDRX_value>

String. Requested eDRX cycle length as a 4–bit binary string. Refers to bits 4–1 of octet 3 of the Extended DRX parameters (see *3GPP TS 24.008, subclause 10.5.5.32*). If omitted, the value of the requested eDRX period is set to the manufacturer-specific default.

Bit

4 3 2 1 – E-UTRAN eDRX cycle length duration in seconds.

0 0 0 0 – 5.12 s.

0 0 0 1 – 10.24 s.

0 0 1 0 – 20.48 s.

0 0 1 1 – 40.96 s.

0 1 0 0 – 61.44 s.

0 1 0 1 – 81.92 s.

0 1 1 0 – 102.4 s.

0 1 1 1 – 122.88 s.

1 0 0 0 – 143.36 s.

1 0 0 1 – 163.84 s.

1 0 1 0 – 327.68 s.

1 0 1 1 – 655.36 s.

1 1 0 0 – 1310.72 s.

1 1 0 1 – 2621.44 s.

1 1 1 0 – 5242.88 s.

1 1 1 1 – 10485.76 s.

<NW-Provided_eDRX_value>

String. The eDRX cycle length granted by the network as a 4–bit binary string. Refers to bits 4–1 of octet 3 of the Extended DRX parameters (see *3GPP TS 24.008, subclause 10.5.5.32*).

Bit

4 3 2 1 – E-UTRAN eDRX cycle length duration in seconds.

0 0 0 0 – 5.12 s

0 0 0 1 – 10.24 s

0 0 1 0 – 20.48 s.

0 0 1 1 – 40.96 s.

0 1 0 0 – 61.44 s

0 1 0 1 – 81.92 s.

0 1 1 0 – 102.4 s

0 1 1 1 – 122.88 s

1 0 0 0 – 143.36 s

1 0 0 1 – 163.84 s.

1 0 1 0 – 327.68 s.

1 0 1 1 – 655.36 s.

1 1 0 0 – 1310.72 s.

1 1 0 1 – 2621.44 s.

1 1 1 0 – 5242.88 s.

1 1 1 1 – 10485.76 s.

<Paging_time_window>

String. The network assigned paging time window duration as a 4–bit binary string. Refers to bit 8 to 5 of octet 3 of the Extended DRX parameters information element (see *3GPP TS 24.008, subclause 10.5.5.32*).

Bit

4 3 2 1 – PTW length in seconds.

0 0 0 0 – 1.28 s.

0 0 0 1 – 2.56 s.

0 0 1 0 – 3.84 s.

0 0 1 1 – 5.12 s.

0 1 0 0 – 6.4 s.

0 1 0 1 – 7.68 s.

0 1 1 0 – 8.96 s.

0 1 1 1 – 10.24 s.

1 0 0 0 – 11.52 s.

1 0 0 1 – 12.8 s.

1 0 1 0 – 14.08 s.

1 0 1 1 – 15.36 s.

1 1 0 0 – 16.64 s.

1 1 0 1 – 17.92 s.

1 1 1 0 – 19.2 s.

1 1 1 1 – 20.48 s.

The following command example reads eDRX parameters:

```
AT+CEDRXRDP
+CEDRXRDP: 4,"0011","0010","1001"
OK
```

12.4.2 Read command

The read command is not supported.

12.4.3 Test command

The test command is not supported.

12.5 EPS network registration status +CEREG

The **+CEREG** command subscribes unsolicited *EPS* network registration status notifications. v1.4.x

For reference, see *3GPP 27.007 Ch. 10.1.22*.

12.5.1 Set command

The set command subscribes unsolicited network status notifications.

Syntax:

```
+CEREG=<n>
```

The set command parameter and its defined values are the following:

<n>

Integer, 0–5. *URC* subscription level controlling.

0 – Unsubscribe URCs.

1 – Subscribe URCs `+CEREG:<stat>`.

2 – Subscribe URCs `+CEREG:<stat>[,<tac>,<ci>,<AcT>]`.

3 – Subscribe URCs `+CEREG:<stat>[,<tac>,<ci>,<AcT>[,<cause_type>,<reject_cause>]]`.

4 – Subscribe URCs `+CEREG: <stat>[,[<tac>],[<ci>],[<AcT>][,,[,[<Active-Time>],[<Periodic-TAU-ext>]]]]`.

5 – Subscribe URCs `+CEREG: <stat>[,[<tac>],[<ci>],[<AcT>][,[<cause_type>],[<reject_cause>][,[<Active-Time>],[<Periodic-TAU-ext>]]]]`.

The notification parameters and their defined values are the following:

<stat>

Integer 0–5. EPS registration status.

0 – Not registered. *UE* is not currently searching for an operator to register to.

1 – Registered, home network.

2 – Not registered, but UE is currently trying to attach or searching an operator to register to.

3 – Registration denied.

4 – Unknown.

5 – Registered, roaming.

<tac>

String in hexadecimal format. 2-byte *Tracking Area Code (TAC)*.

<ci>

String in hexadecimal format. 4-byte *E-UTRAN* cell ID.

<AcT>

Integer, 7. Access technology of the serving cell.

7 – E-UTRAN.

<cause_type>

Integer, 0. Type of <reject_cause>.

0 – <reject_cause> contains an *EMM* cause value. See *3GPP TS 24.301 Annex A*.

<reject_cause>

Integer. Cause of the failed registration. The value type is defined by <cause_type>. For *EMM* cause values, see *3GPP TS 24.301 Annex A*.

<Active-Time>

String. 1 byte in 8-bit format.

Indicates the Active Time value (T3324) allocated to the device in E-UTRAN. For the coding and value range, see the GPRS Timer 2 IE in *3GPP TS 24.008* Table 10.5.163/3GPP TS 24.008.

The device can enter *PSM* state in LTE idle mode when the <Active-Time> parameter has a valid value where at least one of bits 6–8 is set to `0`. The length of PSM sleep is the value that remains from the value configured for periodic TAU after active time has elapsed, which is T3324 subtracted from T3412.

GPRS Timer 2 value (octet 3).

Bits 5 to 1 represent the binary coded timer value.

Bits 8 to 6 define the timer value unit for the GPRS timer as follows:

Bits

8 7 6

0 0 0 – Value is incremented in multiples of 2 s.

0 0 1 – Value is incremented in multiples of 1 min.

0 1 0 – Value is incremented in multiples of 6 min.

1 1 1 – Value indicates that the timer is deactivated.

<Periodic-TAU-ext>

String. 1 byte in 8-bit format.

Indicates the extended periodic *TAU* value (T3412ext) allocated to the device in E-UTRAN. For the coding and value range, see the GPRS Timer 3 IE in *3GPP TS 24.008* Table 10.5.163a/3GPP TS 24.008.

A deactivated value where bits 6–8 are set to `1` after a successful LTE Attach or TAU means that the network configures the periodic TAU or PSM cycle length using a non-extended format with *GPRS* timer instead of GPRS timer 3.

GPRS Timer 3 value (octet 3).

Bits 5 to 1 represent the binary coded timer value.

Bits 8 to 6 define the timer value unit for the GPRS timer as follows:

Bits

8 7 6

0 0 0 – Value is incremented in multiples of 10 min.

0 0 1 – Value is incremented in multiples of 1 h.

0 1 0 – Value is incremented in multiples of 10 h.

0 1 1 – Value is incremented in multiples of 2 s.

1 0 0 – Value is incremented in multiples of 30 s.

1 0 1 – Value is incremented in multiples of 1 min.

1 1 0 – Value is incremented in multiples of 320 h.

1 1 1 – Value indicates that the timer is deactivated.

The following command example subscribes notifications with level 2:

```
AT+CEREG=2
OK
```

The following example shows an unsolicited notification with level 1 trying to attach:

```
+CEREG: 2
```

The following example shows an unsolicited notification with level 2 registered:

```
+CEREG: 1,"002F","0012BEEF",7
```

12.5.2 Read command

The read command reads the current *EPS* network registration status.

Response syntax when <n> is 0, 1, 2, or 3:

```
+CEREG: <n>,<stat>[,[<tac>],[<ci>],[<AcT>[,<cause_type>,<reject_cause>]]]
```

Response syntax when <n> is 4 or 5:

```
+CEREG: <n>,<stat>[,[<tac>],[<ci>],[<AcT>][,[<cause_type>],[<reject_cause>][,[<Active-
Time>],[<Periodic-TAU-ext>]]]]
```

The response parameters and their defined values are the following:

<n>

Integer, 0–5. *URC* subscription level controlling.

0 – Unsubscribe URCs.

1 – Subscribe URCs +CEREG:<stat>.

2 – Subscribe URCs +CEREG:<stat>[,<tac>,<ci>,<AcT>].

3 – Subscribe URCs +CEREG:<stat>[,<tac>,<ci>,<AcT>[,<cause_type>,<reject_cause>]].

4 – Subscribe URCs +CEREG: <stat>[,[<tac>],[<ci>],[<AcT>][,,[,[<Active-Time>],[<Periodic-TAU-ext>]]]].

5 – Subscribe URCs +CEREG: <stat>[,[<tac>],[<ci>],[<AcT>][,[<cause_type>],[<reject_cause>][,[<Active-Time>],[<Periodic-TAU-ext>]]]].

<stat>

Integer 0–5. EPS registration status.

0 – Not registered. *UE* is not currently searching for an operator to register to.

1 – Registered, home network.

2 – Not registered, but UE is currently trying to attach or searching an operator to register to.

3 – Registration denied.

4 – Unknown.

5 – Registered, roaming.

<tac>

String in hexadecimal format. 2-byte *TAC*.

<ci>

String in hexadecimal format. 4-byte *E-UTRAN* cell ID.

<AcT>

Integer, 7. Access technology of the serving cell.

7 – E-UTRAN.

<cause_type>

Integer, 0. Type of <reject_cause>.

0 – <reject_cause> contains an *EMM* cause value. See *3GPP TS 24.301 Annex A*.

<reject_cause>

Integer. Cause of the failed registration. The value type is defined by <cause_type>. For *EMM* cause values, see *3GPP TS 24.301 Annex A*.

<Active-Time>

String. 1 byte in 8-bit format.

Indicates the Active Time value (T3324) allocated to the device in E-UTRAN. For the coding and value range, see the GPRS Timer 2 IE in *3GPP TS 24.008* Table 10.5.163/3GPP TS 24.008.

The device can enter *PSM* state in LTE idle mode when the <Active-Time> parameter has a valid value where at least one of bits 6–8 is set to `0`. The length of PSM sleep is the value that remains from the value configured for periodic TAU after active time has elapsed, which is T3324 subtracted from T3412.

GPRS Timer 2 value (octet 3).

Bits 5 to 1 represent the binary coded timer value.

Bits 8 to 6 define the timer value unit for the GPRS timer as follows:

Bits

8 7 6

0 0 0 – Value is incremented in multiples of 2 s.

0 0 1 – Value is incremented in multiples of 1 min.

0 1 0 – Value is incremented in multiples of 6 min.

1 1 1 – Value indicates that the timer is deactivated.

<Periodic-TAU-ext>

String. 1 byte in 8-bit format.

Indicates the extended periodic *TAU* value (T3412ext) allocated to the device in E-UTRAN. For the coding and value range, see the GPRS Timer 3 IE in *3GPP TS 24.008* Table 10.5.163a/3GPP TS 24.008.

A deactivated value where bits 6–8 are set to `1` after a successful LTE Attach or TAU means that the network configures the periodic TAU or PSM cycle length using a non-extended format with *GPRS* timer instead of GPRS timer 3.

GPRS Timer 3 value (octet 3).

Bits 5 to 1 represent the binary coded timer value.

Bits 8 to 6 define the timer value unit for the GPRS timer as follows:

Bits

8 7 6

0 0 0 – Value is incremented in multiples of 10 min.

0 0 1 – Value is incremented in multiples of 1 h.

0 1 0 – Value is incremented in multiples of 10 h.

0 1 1 – Value is incremented in multiples of 2 s.

1 0 0 – Value is incremented in multiples of 30 s.

1 0 1 – Value is incremented in multiples of 1 min.

1 1 0 – Value is incremented in multiples of 320 h.

1 1 1 – Value indicates that the timer is deactivated.

The following command example reads the current registration status:

```
AT+CEREG?
+CEREG: 2,1,"002F","0012BEEF",7
OK
```

12.5.3 Test command

The test command returns the supported *URC* levels as a compound value.

Response syntax:

```
+CEREG: (supported <n> values)
```

The test command parameter and its defined values are the following:

<n>

Integer, 0–5. *URC* subscription level controlling.

0 – Unsubscribe URCs.

1 – Subscribe URCs `+CEREG:<stat>`.

2 – Subscribe URCs `+CEREG:<stat>[,<tac>,<ci>,<AcT>]`.

3 – Subscribe URCs `+CEREG:<stat>[,<tac>,<ci>,<AcT>[,<cause_type>,<reject_cause>]]`.

4 – Subscribe URCs `+CEREG: <stat>[,[<tac>],[<ci>],[<AcT>][,,[,[<Active-Time>],[<Periodic-TAU-ext>]]]]`.

5 – Subscribe URCs `+CEREG: <stat>[,[<tac>],[<ci>],[<AcT>][,[<cause_type>],[<reject_cause>][,[<Active-Time>],[<Periodic-TAU-ext>]]]]`.

The following command example shows supported <n> values:

```
AT+CEREG=?
+CEREG: (0,1,2,3,4,5)
OK
```

12.6 Paging time window and eDRX %PTWEDRXS

The proprietary **%PTWEDRXS** command controls the *UE*'s paging time window and *eDRX* parameters.
v1.4.x

12.6.1 Set command

The set command controls whether the *UE* applies paging time window and *eDRX* and sets the requested values.

The command also controls the presentation of the *URC* when <mode> is set to 2 and there is a change of the paging time window and eDRX parameters provided by the network:

```
%PTWEDRXP: <AcT-type>[,<Requested_Paging_time_window>[,<Requested_eDRX_value>[,<NW-provided_eDRX_value>[,<Paging_time_window>]]]]
```

Syntax:

```
%PTWEDRXS=<mode>,<AcT-type>[,<Requested_Paging_time_window>[,<Requested_eDRX_value>]]
```

The set command parameters and their defined values are the following:

<mode>

Integer, 0–3. Controls whether eDRX is enabled and whether the **%PTWEDRXS** URC is sent on changes.

0 – Disable the use of eDRX and disable the URC (default).

1 – Enable the use of eDRX and disable the URC.

2 – Enable the use of eDRX and enable the URC.

3 – Disable the use of eDRX and disable the URC. Discard all parameters for eDRX.

<AcT-type>

Integer, 0 and 4. Radio Access technology for which eDRX is configured. The default is `4`.

0 – Current cell is not using eDRX or network configuration does not support eDRX. Used only in the URC.

4 – *E-UTRAN* (WB-S1 mode).

<Requested_Paging_time_window>

String. Requested PTW length as a 4-bit binary string. Refers to bit 8 to 5 of octet 3 of the Extended DRX parameters information element (see *3GPP TS 24.008, subclause 10.5.5.32*).

Bit

4 3 2 1 – PTW length in seconds.

0 0 0 0 – 1.28 s.

0 0 0 1 – 2.56 s.

0 0 1 0 – 3.84 s.

0 0 1 1 – 5.12 s.

0 1 0 0 – 6.4 s.

0 1 0 1 – 7.68 s.

0 1 1 0 – 8.96 s.

0 1 1 1 – 10.24 s.

1 0 0 0 – 11.52 s.

1 0 0 1 – 12.8 s.

1 0 1 0 – 14.08 s.

1 0 1 1 – 15.36 s.

1 1 0 0 – 16.64 s.

1 1 0 1 – 17.92 s.

1 1 1 0 – 19.2 s.

1 1 1 1 – 20.48 s.

<Requested_eDRX_value>

String. Requested eDRX cycle length as a 4–bit binary string. Refers to bits 4–1 of octet 3 of the Extended DRX parameters (see *3GPP TS 24.008, subclause 10.5.5.32*). If omitted, the value of the requested eDRX period is set to the manufacturer-specific default.

Bit

4 3 2 1 – E-UTRAN eDRX cycle length duration in seconds.

0 0 0 0 – 5.12 s.

0 0 0 1 – 10.24 s.

0 0 1 0 – 20.48 s.

0 0 1 1 – 40.96 s.

0 1 0 0 – 61.44 s.

0 1 0 1 – 81.92 s.

0 1 1 0 – 102.4 s.

0 1 1 1 – 122.88 s.

1 0 0 0 – 143.36 s.

1 0 0 1 – 163.84 s.

1 0 1 0 – 327.68 s.

1 0 1 1 – 655.36 s.

1 1 0 0 – 1310.72 s.

1 1 0 1 – 2621.44 s.

1 1 1 0 – 5242.88 s.

1 1 1 1 – 10485.76 s.

<NW-Provided_eDRX_value>

String. The eDRX cycle length granted by the network as a 4–bit binary string. Refers to bits 4–1 of octet 3 of the Extended DRX parameters (see *3GPP TS 24.008, subclause 10.5.5.32*).

Bit

4 3 2 1 – E-UTRAN eDRX cycle length duration in seconds.

0 0 0 0 – 5.12 s

0 0 0 1 – 10.24 s

0 0 1 0 – 20.48 s.

0 0 1 1 – 40.96 s.

0 1 0 0 – 61.44 s

0 1 0 1 – 81.92 s.

0 1 1 0 – 102.4 s

0 1 1 1 – 122.88 s

1 0 0 0 – 143.36 s

1 0 0 1 – 163.84 s.

1 0 1 0 – 327.68 s.

1 0 1 1 – 655.36 s.

1 1 0 0 – 1310.72 s.

1 1 0 1 – 2621.44 s.

1 1 1 0 – 5242.88 s.

1 1 1 1 – 10485.76 s.

<Paging_time_window>

String. The network assigned paging time window duration as a 4–bit binary string. Refers to bit 8 to 5 of octet 3 of the Extended DRX parameters information element (see *3GPP TS 24.008, subclause 10.5.5.32*).

Bit

4 3 2 1 – PTW length in seconds.

0 0 0 0 – 1.28 s.

0 0 0 1 – 2.56 s.

0 0 1 0 – 3.84 s.

0 0 1 1 – 5.12 s.

0 1 0 0 – 6.4 s.

0 1 0 1 – 7.68 s.

0 1 1 0 – 8.96 s.

0 1 1 1 – 10.24 s.

1 0 0 0 – 11.52 s.

1 0 0 1 – 12.8 s.

1 0 1 0 – 14.08 s.

1 0 1 1 – 15.36 s.

1 1 0 0 – 16.64 s.

1 1 0 1 – 17.92 s.

1 1 1 0 – 19.2 s.

1 1 1 1 – 20.48 s.

The following command example enables a requested paging time window of `2.56` seconds and eDRX of `163.84` seconds:

```
AT%PTWEDRXS=1,4,"0001","1001"
OK
```

12.6.2 Read command

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

Response syntax:

```
%PTWEDRXS: <AcT-type>,<Requested_Paging_time_window>,<Requested_eDRX_value>[,<NW-
provided_eDRX_value>[,<Paging_time_window>]]
```

The read command parameters and their defined values are the following:

<AcT-type>

Integer, 0 and 4. Radio Access technology for which eDRX is configured. The default is `4`.

0 – Current cell is not using eDRX or network configuration does not support eDRX. Used only in the URC.

4 – *E-UTRAN* (WB-S1 mode).

<Requested_Paging_time_window>

String. Requested PTW length as a 4-bit binary string. Refers to bit 8 to 5 of octet 3 of the Extended DRX parameters information element (see *3GPP TS 24.008, subclause 10.5.5.32*).

Bit

4 3 2 1 – PTW length in seconds.

0 0 0 0 – 1.28 s.

0 0 0 1 – 2.56 s.

0 0 1 0 – 3.84 s.

0 0 1 1 – 5.12 s.

0 1 0 0 – 6.4 s.

0 1 0 1 – 7.68 s.

0 1 1 0 – 8.96 s.

0 1 1 1 – 10.24 s.

1 0 0 0 – 11.52 s.

1 0 0 1 – 12.8 s.

1 0 1 0 – 14.08 s.

1 0 1 1 – 15.36 s.

1 1 0 0 – 16.64 s.

1 1 0 1 – 17.92 s.

1 1 1 0 – 19.2 s.

1 1 1 1 – 20.48 s.

<Requested_eDRX_value>

String. Requested eDRX cycle length as a 4–bit binary string. Refers to bits 4–1 of octet 3 of the Extended DRX parameters (see *3GPP TS 24.008, subclause 10.5.5.32*). If omitted, the value of the requested eDRX period is set to the manufacturer-specific default.

Bit

4 3 2 1 – E-UTRAN eDRX cycle length duration in seconds.

0 0 0 0 – 5.12 s.

0 0 0 1 – 10.24 s.

0 0 1 0 – 20.48 s.

0 0 1 1 – 40.96 s.

0 1 0 0 – 61.44 s.

0 1 0 1 – 81.92 s.

0 1 1 0 – 102.4 s.

0 1 1 1 – 122.88 s.

1 0 0 0 – 143.36 s.

1 0 0 1 – 163.84 s.

1 0 1 0 – 327.68 s.

1 0 1 1 – 655.36 s.

1 1 0 0 – 1310.72 s.

1 1 0 1 – 2621.44 s.

1 1 1 0 – 5242.88 s.

1 1 1 1 – 10485.76 s.

<NW-Provided_eDRX_value>

String. The eDRX cycle length granted by the network as a 4–bit binary string. Refers to bits 4–1 of octet 3 of the Extended DRX parameters (see *3GPP TS 24.008, subclause 10.5.5.32*).

Bit

4 3 2 1 – E-UTRAN eDRX cycle length duration in seconds.

0 0 0 0 – 5.12 s

0 0 0 1 – 10.24 s

0 0 1 0 – 20.48 s.

0 0 1 1 – 40.96 s.

0 1 0 0 – 61.44 s

0 1 0 1 – 81.92 s.

0 1 1 0 – 102.4 s

0 1 1 1 – 122.88 s

1 0 0 0 – 143.36 s

1 0 0 1 – 163.84 s.

1 0 1 0 – 327.68 s.

1 0 1 1 – 655.36 s.

1 1 0 0 – 1310.72 s.

1 1 0 1 – 2621.44 s.

1 1 1 0 – 5242.88 s.

1 1 1 1 – 10485.76 s.

<Paging_time_window>

String. The network assigned paging time window duration as a 4–bit binary string. Refers to bit 8 to 5 of octet 3 of the Extended DRX parameters information element (see *3GPP TS 24.008, subclause 10.5.5.32*).

Bit

4 3 2 1 – PTW length in seconds.

0 0 0 0 – 1.28 s.

0 0 0 1 – 2.56 s.

0 0 1 0 – 3.84 s.

0 0 1 1 – 5.12 s.

0 1 0 0 – 6.4 s.

0 1 0 1 – 7.68 s.

0 1 1 0 – 8.96 s.

0 1 1 1 – 10.24 s.

1 0 0 0 – 11.52 s.

1 0 0 1 – 12.8 s.

1 0 1 0 – 14.08 s.

1 0 1 1 – 15.36 s.

1 1 0 0 – 16.64 s.

1 1 0 1 – 17.92 s.

1 1 1 0 – 19.2 s.

1 1 1 1 – 20.48 s.

The following command example reads the current settings and shows a requested paging time window of `2.56` seconds and eDRX of `163.84` seconds:

```
AT%PTWEDRXS?
%PTWEDRXS: 4,"0001","1001"
OK
```

12.6.3 Test command

The test command returns supported values as a compound value.

Response syntax:

```
%PTWEDRXS: (list of supported <mode>s),(list of supported <AcT-type>s),
(range of supported <Requested_Paging_time_window>s),(range of supported
 <Requested_eDRX_value>s)
```

The test command parameters and their defined values are the following:

<mode>

Integer, 0–3. Controls whether eDRX is enabled and whether the **%PTWEDRXS** URC is sent on changes.

0 – Disable the use of eDRX and disable the URC (default).

1 – Enable the use of eDRX and disable the URC.

2 – Enable the use of eDRX and enable the URC.

3 – Disable the use of eDRX and disable the URC. Discard all parameters for eDRX.

<AcT-type>

Integer, 0 and 4. Radio Access technology for which eDRX is configured. The default is `4`.

0 – Current cell is not using eDRX or network configuration does not support eDRX. Used only in the URC.

4 – *E-UTRAN* (WB-S1 mode).

<Requested_Paging_time_window>

String. Requested PTW length as a 4-bit binary string. Refers to bit 8 to 5 of octet 3 of the Extended DRX parameters information element (see *3GPP TS 24.008, subclause 10.5.5.32*).

Bit

4 3 2 1 – PTW length in seconds.

0 0 0 0 – 1.28 s.

0 0 0 1 – 2.56 s.

0 0 1 0 – 3.84 s.

0 0 1 1 – 5.12 s.

0 1 0 0 – 6.4 s.

0 1 0 1 – 7.68 s.

0 1 1 0 – 8.96 s.

0 1 1 1 – 10.24 s.

1 0 0 0 – 11.52 s.

1 0 0 1 – 12.8 s.

1 0 1 0 – 14.08 s.

1 0 1 1 – 15.36 s.

1 1 0 0 – 16.64 s.

1 1 0 1 – 17.92 s.

1 1 1 0 – 19.2 s.

1 1 1 1 – 20.48 s.

<Requested_eDRX_value>

String. Requested eDRX cycle length as a 4–bit binary string. Refers to bits 4–1 of octet 3 of the Extended DRX parameters (see *3GPP TS 24.008, subclause 10.5.5.32*). If omitted, the value of the requested eDRX period is set to the manufacturer-specific default.

Bit

4 3 2 1 – E-UTRAN eDRX cycle length duration in seconds.

0 0 0 0 – 5.12 s.

0 0 0 1 – 10.24 s.

0 0 1 0 – 20.48 s.

0 0 1 1 – 40.96 s.

0 1 0 0 – 61.44 s.

0 1 0 1 – 81.92 s.

0 1 1 0 – 102.4 s.

0 1 1 1 – 122.88 s.

1 0 0 0 – 143.36 s.

1 0 0 1 – 163.84 s.

1 0 1 0 – 327.68 s.

1 0 1 1 – 655.36 s.

1 1 0 0 – 1310.72 s.

1 1 0 1 – 2621.44 s.

1 1 1 0 – 5242.88 s.

1 1 1 1 – 10485.76 s.

The following command example returns the supported parameter ranges:

```
AT%PTWEDRXS=?
%PTWEDRXS: (0,1,2,3),(4),("0000"-"1111"),("0000"-"1111")
OK
```

13 Mobile termination errors

For reference, see *3GPP 27.007 Ch. 9*.

13.1 Mobile termination error notification +CMEE

The **+CMEE** command disables or enables the use of the final result code +CME ERROR. v1.4.x

+CME ERROR codes are listed in command descriptions.

For reference, see *3GPP 27.007 Ch. 9.1*.

13.1.1 Set command

The set command disables or enables the use of the final result code +CME ERROR. When enabled, MT related errors cause a +CME ERROR: <err> final result code instead of the regular ERROR final result code. ERROR is returned normally when the error is related to syntax, invalid parameters, or AT command parser.

Syntax:

```
+CMEE=[<n>]
```

The set command parameter and its defined values are the following:

<n>

Integer, 0–2. Enable or disable final result code +CME ERROR.

0 – Disable and use ERROR instead (default).

1 – Enable +CME ERROR: <err> result code and use numeric <err> values.

2 – Enable +CME ERROR: <err> result code and use verbose <err> values.

<err> values are specified in *3GPP TS 27.007 Ch. 9.2*. Vendor-specific values are listed separately for each command with the value range starting from 512.

The following command example enables error codes in responses:

```
AT+CMEE=1
OK
```

13.1.2 Read command

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

Response syntax:

```
+CMEE: <n>
```

The read command parameter and its defined values are the following:

<n>

Integer, 0–2. Enable or disable final result code +CME ERROR.

0 – Disable and use ERROR instead (default).

1 – Enable +CME ERROR: <err> result code and use numeric <err> values.

2 – Enable +CME ERROR: <err> result code and use verbose <err> values.

The following command example reads the current error code setting:

```
AT+CMEE?
+CMEE: 1
OK
```

13.1.3 Test command

The test command returns supported values as a compound value.

Response syntax:

```
+CMEE: (list of supported <n>s)
```

The test command parameter and its defined values are the following:

<n>

Integer, 0–2. Enable or disable final result code +CME ERROR.

0 – Disable and use ERROR instead (default).

1 – Enable +CME ERROR: <err> result code and use numeric <err> values.

2 – Enable +CME ERROR: <err> result code and use verbose <err> values.

The following command example returns the supported values:

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

14 SMS commands

For reference, see *3GPP 27.005 Ch. 3*.

14.1 New message indications +CNMI

The **+CNMI** command subscribes and configures new message indications. v1.4.x

For reference, see *3GPP 27.005 Ch. 3.4.1*.

14.1.1 Set command

The set command registers or unregisters an SMS client. Only one AT client can be registered as an SMS client. An existing registration must be released before registering a new client.

Syntax:

```
+CNMI=[<mode>[,<mt>[,<bm>[,<ds>[,<bfr>]]]]]
```

+CMS ERROR code

302 – Operation not allowed.

The set command parameters and their defined values are the following:

<mode>

Integer, 0–3. Controls the processing of *URC*s.

0 – Buffer URCs in the modem. If the buffer is full, the oldest indications can be discarded (default).

1 – Discard indication and reject the new received message with the URCs when *TA-TE* link is reserved. Otherwise, forward them directly to the TE.

2 – Buffer URCs when TA-TE link is reserved and flush them to the TE after reservation. Otherwise, forward them directly to the TE.

3 – Forward URCs directly to the TE.

<mt>

Integer, 0–3. Controls SMS-DELIVER routing.

0 – No SMS-DELIVER indications are routed to the TE (default).

1 – If SMS-DELIVER is stored, indication of the memory location is routed to the TE using a **+CMTI** notification (`+CMTI: <mem>,<index>`).

2 – SMS-DELIVER indications except class 2 and message waiting indication group are routed directly to the TE with **+CMT**. The TE must ACK with the **+CNMA** command.

3 – Class 3 SMS-DELIVER indications are routed directly to the TE using the **+CMT** command. Messages of other data coding schemes result in indication as defined in <mt>=1.

<bm>

Integer. Cell Broadcast Message routing. Not supported.

<ds>

Integer, 0–2. Controls SMS-STATUS-REPORT routing.

0 – No SMS-STATUS-REPORT indications are routed to the TE (default).

1 – SMS-STATUS-REPORT indications are routed to the TE using the unsolicited **+CDS** result code. The TE is expected to acknowledge the received status report with the **+CNMA** command when required by the message service mode.

2 – If SMS-STATUS-REPORT is stored, indication of the memory location is routed to the TE using **+CDSI** (`+CDSI: <mem>,<index>`).

<bfr>

Integer. Buffer handling. Not supported.

The following command example enables direct routing of SMS-DELIVERs to the TE:

```
AT+CNMI=2,2,0,0,0
OK
```

The following command example enables storage notification for received messages:

```
AT+CNMI=2,1,0,0,0
OK

+CMTI: "ME",2
```

14.1.2 Read command

The read command is used to query how new messages are indicated.

Response syntax:

```
+CNMI: <mode>,<mt>,<bm>,<ds>,<bfr>
```

The read command parameters and their defined values are the following:

<mode>

Integer, 0–3. Controls the processing of *URC*s.

0 – Buffer URCs in the modem. If the buffer is full, the oldest indications can be discarded (default).

1 – Discard indication and reject the new received message with the URCs when *TA-TE* link is reserved. Otherwise, forward them directly to the TE.

2 – Buffer URCs when TA-TE link is reserved and flush them to the TE after reservation. Otherwise, forward them directly to the TE.

3 – Forward URCs directly to the TE.

<mt>

Integer, 0–3. Controls SMS-DELIVER routing.

0 – No SMS-DELIVER indications are routed to the TE (default).

1 – If SMS-DELIVER is stored, indication of the memory location is routed to the TE using a **+CMTI** notification (`+CMTI: <mem>,<index>`).

2 – SMS-DELIVER indications except class 2 and message waiting indication group are routed directly to the TE with **+CMT**. The TE must ACK with the **+CNMA** command.

3 – Class 3 SMS-DELIVER indications are routed directly to the TE using the **+CMT** command. Messages of other data coding schemes result in indication as defined in <mt>=`1`.

<bm>

Integer. Cell Broadcast Message routing. Not supported.

<ds>

Integer, 0–2. Controls SMS-STATUS-REPORT routing.

0 – No SMS-STATUS-REPORT indications are routed to the TE (default).

1 – SMS-STATUS-REPORT indications are routed to the TE using the unsolicited **+CDS** result code. The TE is expected to acknowledge the received status report with the **+CNMA** command when required by the message service mode.

2 – If SMS-STATUS-REPORT is stored, indication of the memory location is routed to the TE using **+CDSI** (`+CDSI: <mem>,<index>`).

<bfr>

Integer. Buffer handling. Not supported.

The following command example reads the configuration defining how new messages are indicated:

```
AT+CNMI?
+CNMI: 2,2,0,0,0
OK
```

14.1.3 Test command

The test command returns the supported values for the configuration of new message indication.

Response syntax:

```
+CNMI: (range of supported <mode>s),(range of supported <mt>s),(range of supported <bm>s),
(range of supported <ds>s),(range of supported <bfr>s)
```

The test command parameters and their defined values are the following:

<mode>

Integer, 0–3. Controls the processing of *URC*s.

0 – Buffer URCs in the modem. If the buffer is full, the oldest indications can be discarded (default).

1 – Discard indication and reject the new received message with the URCs when *TA-TE* link is reserved. Otherwise, forward them directly to the TE.

2 – Buffer URCs when TA-TE link is reserved and flush them to the TE after reservation. Otherwise, forward them directly to the TE.

3 – Forward URCs directly to the TE.

<mt>

Integer, 0–3. Controls SMS-DELIVER routing.

0 – No SMS-DELIVER indications are routed to the TE (default).

1 – If SMS-DELIVER is stored, indication of the memory location is routed to the TE using a **+CMTI** notification (`+CMTI: <mem>,<index>`).

2 – SMS-DELIVER indications except class 2 and message waiting indication group are routed directly to the TE with **+CMT**. The TE must ACK with the **+CNMA** command.

3 – Class 3 SMS-DELIVER indications are routed directly to the TE using the **+CMT** command. Messages of other data coding schemes result in indication as defined in <mt>=1.

<bm>

Integer. Cell Broadcast Message routing. Not supported.

<ds>

Integer, 0–2. Controls SMS-STATUS-REPORT routing.

0 – No SMS-STATUS-REPORT indications are routed to the TE (default).

1 – SMS-STATUS-REPORT indications are routed to the TE using the unsolicited **+CDS** result code. The TE is expected to acknowledge the received status report with the **+CNMA** command when required by the message service mode.

2 – If SMS-STATUS-REPORT is stored, indication of the memory location is routed to the TE using **+CDSI** (`+CDSI: <mem>,<index>`).

<bfr>

Integer. Buffer handling. Not supported.

The following command example returns the supported values:

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

14.2 Send message +CMGS

The **+CMGS** command sends a message in *Protocol Data Unit (PDU)* mode. v1.4.x

The message reference value <mr> is returned on successful message delivery. This value can be used to identify the message in unsolicited delivery status report result codes. If sending fails, the final result code `+CMS ERROR: <err>` is returned.

For reference, see *3GPP 27.005 Ch. 3.5.1* and *3GPP 27.005 Ch. 4.3*.

14.2.1 Set command

The set command sends an SMS message to the network.

Note: This command can be executed only by a client registered with the **+CNMI** command.

Syntax:

```
+CMGS=<length><CR><pdu><ctrl-Z>
```

Response syntax:

```
+CMGS: <mr>[,<ackpdu>]
```

+CMS ERROR code

300 – ME failure.

304 – Invalid PDU mode parameter.

The set command parameters and their defined values are the following:

<length>

Integer, 1–100. Number of octets coded in the TP layer data unit. SMSC address octets are excluded.

<pdu>

String. Hexadecimal numbers containing two *International Reference Alphabet (IRA)* characters per octet.

<mr>

Integer, 0–255. Message reference value.

<ackpdu>

RP-User-Data element of RP-ACK PDU.

The following command example sends an SMS in PDU mode:

```
AT+CMGS=19
> 0011000C91531832547698000000B05D4F29C1E02<ctrl-Z>
+CMGS: 28
OK
```

14.2.2 Read command

The read command is not supported.

14.2.3 Test command

The test command is not supported.

14.3 Received SMS notification +CMT

The **+CMT** command indicates through an unsolicited notification the reception of an SMS-DELIVER message routed directly to the *TE*. v1.4.x

The TE is expected to acknowledge the received message with the **+CNMA** command.

For reference, see *3GPP 27.005 Ch. 3.4.1*.

The notification is subscribed using the **+CNMI** command with <mt> set to 2.

+CMS ERROR code

524 – SMS client has been unregistered due to acknowledgment timeout.

Notification syntax:

```
+CMT: [<alpha>],<length><CR><LF><pdu>
```

The notification parameters and their defined values are the following:

<alpha>

String. Alphanumeric representation of originating address from phonebook, if available.

<length>

Integer. The length of the PDU in octets excluding SMSC address.

<pdu>

String. The SMS-DELIVER PDU in hexadecimal format.

The following notification example shows a received SMS notification:

```
AT+CNMI=2,2,0,0,0
OK

+CMT:,18
0011000C91534810325476000000B04D4F29C0E

AT+CNMA=1
OK
```

14.4 New message ACK, PDU mode +CNMA

The **+CNMA** command sends a positive RP-ACK or negative RP-ERROR acknowledgment in *PDU* mode. v1.4.x

For reference, see *3GPP 27.005 Ch. 4.7*.

14.4.1 Set command

The set command sends a new message or delivery status ACK. A client receiving unsolicited notifications for new messages and delivery status is mandated to acknowledge those. This command can be used only when the modem is activated.

Note: This command can be executed only by a client registered with the **+CNMI** command.

If the *UE* does not get an acknowledgement within the required time (network timeout), it should respond as specified in *3GPP TS 24.011* and the UE or *TA* shall automatically disable routing to the *TE* by setting the <mt> and <ds> values of **+CNMI** to zero. That is, the SMS client gets unregistered. In that case, the *ME* informs the TE client with a `CMS ERROR: 524` notification.

Syntax:

```
+CNMA[=<n>[,<length>[<CR>PDU is given<ctrl-Z/ESC>]]]
```

+CMS ERROR code

300 – ME failure.

302 – Operation not allowed.

The set command parameters and their defined values are the following:

<n>

Integer, 0–2. Defines whether a positive or negative acknowledgment is sent to the network.

0 – Send RP-ACK without PDU data (default).

1 – Send RP-ACK to the network with optional PDU data.

2 – Send RP-ERROR. The ME or TA sends SMS-DELIVER-REPORT with *3GPP TS 23.040* TP-FCS value set to unspecified error cause `FF`.

<length>

Integer, 1–100. Number of octets coded in the TP layer data unit. SMSC address octets are excluded.

<pdu>

Hexadecimal numbers containing two *IRA* characters per octet.

The following command example confirms the reception of a message with the timestamp 13/04/2026 13:11:05:

```
AT+CNMA=1,9<CR>0100624031311150 21<Ctrl-Z>
OK
```

14.4.2 Read command

The read command is not supported.

14.4.3 Test command

The test command lists supported <n>s.

Response syntax:

```
+CNMA: (range of supported <n>s)
```

The test command parameter and its defined values are the following:

<n>

Integer, 0–2. Defines whether a positive or negative acknowledgment is sent to the network.

0 – Send RP-ACK without PDU data (default).

1 – Send RP-ACK to the network with optional PDU data.

2 – Send RP-ERROR. The ME or TA sends SMS-DELIVER-REPORT with *3GPP TS 23.040* TP-FCS value set to unspecified error cause `FF`.

The following command example lists the supported delivery statuses:

```
AT+CNMA=?
+CNMA: (0-2)
OK
```

14.5 Message service failure result code +CMS ERROR

Message service failure result code **+CMS** is sent as an error response to SMS-related commands. v1.4.x

For reference, see *3GPP 27.005 Ch. 3.2.5*.

Response syntax:

```
+CMS ERROR: <err>
```

The parameter and the values used by common messaging commands are the following:

<err>

0–127 – *3GPP TS 24.011 clause E.2* values.

128–255 – *3GPP TS 23.040 clause 9.2.3.22* values.

300–511 – *3GPP TS 27.005 Ch. 3.2.5*.

14.6 Service center address +CSCA

The **+CSCA** command sets or reads the SMS Service Center address. v1.4.x

The SMSC address is used for transmitting SMS messages. This setting is used only when the length of the SMSC address coded into the <pdu> parameter equals zero.

For reference, see *3GPP 27.005 Ch. 3.3.1*.

14.6.1 Set command

The set command updates the SMS Service Center address.

Syntax:

```
+CSCA=<sca>[,<tosca>]
```

+CMS ERROR code

303 – Operation not supported.

The set command parameters and their defined values are the following:

<sca>

String. Maximum length 40 characters. Service Center Address in string format. BCD numbers or GSM 7-bit default alphabet characters converted to characters.

<tosca>

Integer, 129 and 145. Type of Service Center address.

129 – Unknown type. ISDN or telephone numbering plan. Default when the first character of <sca> is not +.

145 – International type. ISDN or telephone numbering plan. Default when the first character of <sca> is +.

The following command example sets the Service Center address:

```
AT+CSCA="+358449900000"
OK
```

14.6.2 Read command

The read command returns the current SMS Service Center address.

Response syntax:

```
+CSCA: <sca>,<tosca>
```

The response parameters and their defined values are the following:

<sca>

String. Maximum length 40 characters. Service Center Address in string format. BCD numbers or GSM 7-bit default alphabet characters converted to characters.

<tosca>

Integer, 129 and 145. Type of Service Center address.

129 – Unknown type. ISDN or telephone numbering plan. Default when the first character of <sca> is not +.

145 – International type. ISDN or telephone numbering plan. Default when the first character of <sca> is +.

The following command example reads the current Service Center address:

```
AT+CSCA?
+CSCA: "+358449900000",145
OK
```

14.6.3 Test command

The test command is not supported.

14.7 New message indication +CMTI

The **+CMTI** *URC* indicates that a new SMS-DELIVER message has been stored in message storage.

v1.4.x

The notification is subscribed using the **+CNMI** command with <mt> set to 1.

Notification syntax:

```
+CMTI: <mem>,<index>
```

The notification parameters and their defined values are the following:

<mem>

String. Message storage containing the received message.

<index>

Integer. Index of the stored message in <mem>.

The following notification example shows a new message stored in the <mem> message storage:

```
AT+CNMI=2,1,0,0,0
OK
+CMTI: "ME",0
```

14.8 Status report notification +CDS

The **+CDS** *URC* indicates that an `SMS-STATUS-REPORT` message has been routed directly to the *TE* in *PDU* mode. v1.4.x

The notification is subscribed using the **+CNMI** command with <ds> set to `1`.

The *TE* is expected to acknowledge the received status report with the **+CNMA** command when required by the message service mode.

A directly routed status report can be triggered by sending a mobile-originated PDU SMS with TP-SRR set to `1`.

Notification syntax:

```
+CDS: <length><CR><LF><pdu>
```

+CMS ERROR code

524 – SMS client has been unregistered due to acknowledgment timeout.

The notification parameters and their defined values are the following:

<length>

Integer. Length of the TPDU in octets.

<pdu>

String. Status report PDU in hexadecimal format including the SMSC address followed by the TPDU.

The following notification example shows a directly routed SMS status report in PDU mode:

```
AT+CNMI=2,1,0,1,0
OK
+CDS: 25
0791...
AT+CNMA=1
OK
```

14.9 Status report storage indication +CDSI

The **+CDSI** *URC* indicates that an SMS-STATUS-REPORT message has been stored in message storage.
v1.4.x

The notification is subscribed using the **+CNMI** command with <ds> set to 2.

A stored status report can be triggered by sending a mobile-originated *PDU* SMS with TP-SRR set to 1.

Notification syntax:

```
+CDSI: <mem>,<index>
```

The notification parameters and their defined values are the following:

<mem>

String. Message storage containing the status report.

<index>

Integer. Index of the stored status report in <mem>.

The following notification example shows a stored SMS status report:

```
AT+CNMI=2,1,0,2,0
OK
+CDSI: "ME",1
```

15 Security commands

Security commands can be used for device and data security.

15.1 PIN code +CPIN

The **+CPIN** command enters and checks the required *Personal Identification Number (PIN)*. v1.4.x

For reference, see *3GPP 27.007 Ch. 8.3*.

15.1.1 Set command

The set command enters the *PIN*.

Syntax:

```
+CPIN=<pin>[,<newpin>]
```

The set command parameters and their defined values are the following:

<pin>

String of digits, 4–8 characters. PIN.

<newpin>

String of digits, 4–8 characters. Mandatory if the required code is *SIM Personal Unblocking Key (PUK)* or SIM PUK2.

Note: If no PIN is required, the response code is ERROR.

The following command example enters PIN 1234:

```
AT+CPIN="1234"
OK
```

15.1.2 Read command

The read command checks if a *PIN* is needed or if a personalization lock is blocking the device start-up.

Response syntax:

```
+CPIN: <code>
```

The read command parameter and its defined values are the following:

<code>

READY – No PIN required

SIM PIN – PIN code required

SIM PUK – PUK code required

SIM PIN2 – PIN2 code required

SIM PUK2 – PUK2 code required

PH-SIM PIN – *USIM* depersonalization required

PH-NET PIN – Network depersonalization required

PH-NETSUB PIN – Network subset depersonalization required

PH-SP PIN – Service provider depersonalization required

PH-CORP PIN – Corporate depersonalization required

The following command example checks whether a PIN code is required with the response indicating that a PIN code is required:

```
AT+CPIN?
+CPIN: "SIM PIN"
OK
```

Note: Use **%XUSIMLCK** when facility lock depersonalization is required.

15.1.3 Test command

The test command is not supported.

15.2 Change password +CPWD

The **+CPWD** command changes the password for the facility lock. v1.4.x

For reference, see *3GPP 27.007 Ch. 7.5*.

15.2.1 Set command

The set command changes the password for the facility lock.

Syntax:

```
+CPWD=<fac>,<oldpwd>,<newpwd>
```

The set command parameters and their defined values are the following:

<fac>

SC – *SIM PIN*

<oldpwd>,<newpwd>

String, 4–8 characters. Password.

The following command example changes the SIM PIN:

```
AT+CPWD="SC","1234","5678"
OK
```

15.2.2 Read command

The read command is not supported.

15.2.3 Test command

The test command returns the supported facilities and password length.

Response syntax:

```
+CPWD: list of supported (<fac>,<pwdlength>)s
```

The test command parameters and their defined values are the following:

<fac>

SC – *SIM PIN*

<pwdlength>

Integer. Maximum length of the password.

The following command example returns a list of supported facilities and password lengths:

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

16 UICC access commands

UICC access commands can be used to access UICC and receive notifications of the UICC state.

16.1 Request IMSI +CIMI

The **+CIMI** command reads the *IMSI* from the *USIM* card. v1.4.x

For reference, see *3GPP 27.007 Ch. 5.6*.

16.1.1 Set command

The set command reads the *IMSI* from the *SIM* card.

Syntax:

```
+CIMI
```

Response syntax:

```
<IMSI>
```

The response parameter and its defined value are the following:

<IMSI>

String without double quotes. IMSI.

Note: ERROR is returned if IMSI is not available.

The following command example reads the IMSI string:

```
AT+CIMI
284011234567890
OK
```

16.1.2 Read command

The read command is not supported.

16.1.3 Test command

The test command is not supported.

16.2 Subscriber number +CNUM

The **+CNUM** command returns the subscriber *Mobile Station International Subscriber Directory Number (MSISDN)*. v1.4.x

For reference, see *3GPP 27.007 Ch. 7.1*.

16.2.1 Set command

The set command returns the subscriber *MSISDN*.

Syntax:

```
+CNUM
```

Response syntax:

```
+CNUM: ,<number1>,<type1>
```

An ERROR response is returned if MSISDN is not available on the *SIM* card or if the SIM card is not initialized.

The set command parameters and their defined values are the following:

<numberx>

String. Maximum length 24 characters. Phone number of format specified by <typex>.

<typex>

Integer. Type of address octet (see *3GPP TS 24.008, subclause 10.5.4.7*).

The following command example reads the subscriber number stored in the SIM:

```
AT+CNUM
+CNUM: ,"+1234567891234",145
OK
```

16.2.2 Read command

The read command is not supported.

16.2.3 Test command

The test command is not supported.

16.3 SIM configuration %SIMCFG

The proprietary **%SIMCFG** command sets *SIM* configuration parameters. v1.4.x

16.3.1 Set command

The set command configures a *SIM* configuration parameter.

Syntax:

```
%SIMCFG=<param>,<value>
```

The set command parameters and their defined values are the following:

<param>

String. Name of the configuration parameter. The maximum number of parameters in a single set command is 8.

<value>

Integer or string. Value for the parameter. Supported range depends on the parameter.

The configuration parameters and their defined values are the following:

"SimSimulator"

Integer, 0–1. Enable or disable virtual SIM card for instrument test. Virtual SIM or physical SIM card takes effect after terminal reset.

0 – Disabled (default).

1 – Enabled.

"SimPowerSave"

Integer, 0–1. Enable or disable SIM power saving mode.

0 – Disabled.

1 – Enabled (default).

"SimPresenceDetect"

Integer, 0–1. Enable or disable SIM presence detection.

0 – Disabled.

1 – Enabled (default).

"SimPreDetectPeriod"

Integer, 1–30. SIM presence detection polling period in seconds. The default value is `29`.

"SimSlot"

Integer, 0–1. Select SIM slot. The hardware interface for the selected slot must be ready. The modem must be in power off state.

0 – Slot 0 (default).

1 – Slot 1.

"SoftSim"

Integer, 0–1. Enable or disable SoftSIM if integrated. On power on, SoftSIM or physical SIM card is used. The modem must be in power off state.

0 – Disabled (default).

1 – Enabled.

"IsimAppDisable"

Integer, 0–1. Enable or disable *IP Multimedia Services Identity Module (ISIM)* application.

0 – Enabled (default).

1 – Disabled.

"UserSetTestSim"

Integer, 0–1. User–configured test SIM.

0 – Disabled (default).

1 – Enabled.

"SimGenAccReqGetRsp"

Integer, 0–1. Generic access request and response.

0 – Disabled (default).

1 – Enabled.

"SimGenAccLargeApduRspSize"

Integer, 260–8196. Generic access large *Application Protocol Data Unit (APDU)* response size. The default is `2048`

The following command example disables the virtual SIM simulator:

```
AT%SIMCFG="SimSimulator",0
OK
```

16.3.2 Read command

The read command returns the current setting of each *SIM* configuration parameter.

One or more lines are returned with one line for each parameter.

Response syntax:

```
%SIMCFG: <param>,<value>
```

The read command parameters and their defined values are the following:

<param>

String. Name of the configuration parameter. The maximum number of parameters in a single set command is 8.

<value>

Integer or string. Value for the parameter. Supported range depends on the parameter.

The response parameters and their defined values are the following:

"SimSimulator"

Integer, 0–1. Enable or disable virtual SIM card for instrument test. Virtual SIM or physical SIM card takes effect after terminal reset.

0 – Disabled (default).

1 – Enabled.

"SimPowerSave"

Integer, 0–1. Enable or disable SIM power saving mode.

0 – Disabled.

1 – Enabled (default).

"SimPresenceDetect"

Integer, 0–1. Enable or disable SIM presence detection.

0 – Disabled.

1 – Enabled (default).

"SimPreDetectPeriod"

Integer, 1–30. SIM presence detection polling period in seconds. The default value is `29`.

"SimSlot"

Integer, 0–1. Select SIM slot. The hardware interface for the selected slot must be ready. The modem must be in power off state.

0 – Slot 0 (default).

1 – Slot 1.

"SoftSim"

Integer, 0–1. Enable or disable SoftSIM if integrated. On power on, SoftSIM or physical SIM card is used. The modem must be in power off state.

0 – Disabled (default).

1 – Enabled.

"IsimAppDisable"

Integer, 0–1. Enable or disable *ISIM* application.

0 – Enabled (default).

1 – Disabled.

"UserSetTestSim"

Integer, 0–1. User–configured test SIM.

0 – Disabled (default).

1 – Enabled.

"SimGenAccReqGetRsp"

Integer, 0–1. Generic access request and response.

0 – Disabled (default).

1 – Enabled.

"SimGenAccLargeApduRspSize"

Integer, 260–8196. Generic access large *APDU* response size. The default is `2048`

The following command example returns the current SIM configuration:

```
AT%SIMCFG?
%SIMCFG: "SimSimulator",0
%SIMCFG: "SimPowerSave",0
%SIMCFG: "SimPresenceDetect",1
%SIMCFG: "SimPreDetectPeriod",29
%SIMCFG: "SimSlot",0
%SIMCFG: "SoftSim",0
%SIMCFG: "IsimAppDisable",0
%SIMCFG: "UserSetTestSim",0
%SIMCFG: "SimGenAccReqGetRsp",0
%SIMCFG: "SimGenAccLargeApduRspSize",2048
OK
```

16.3.3 Test command

The test command returns supported values as a compound value for each parameter.

Response syntax:

```
%SIMCFG: "<param>":(range of supported <value>s)[,"<param>":(range of supported
 <value>s),...]
```

The test command parameters and their defined values are the following:

<param>

String. Name of the configuration parameter. The maximum number of parameters in a single set command is 8.

<value>

Integer or string. Value for the parameter. Supported range depends on the parameter.

The following command example returns the supported parameters and their value ranges:

```
AT%SIMCFG=?
%SIMCFG: "SimSimulator":(0-1),"SimPowerSave":(0-1),"SimPresenceDetect":
(0-1),"SimPreDetectPeriod":(1-30),"SimSlot":(0-1),
"SoftSim":(0-1),"IsimAppDisable":(0-1),"UserSetTestSim":(0-1),"SimGenAccReqGetRsp":
(0-1),"SimGenAccLargeApduRspSize":(2048-8196)
OK
```

16.4 USAT terminal profile %USATP

The proprietary **%USATP** command configures the *Universal Subscriber Identity Module Application Toolkit (USAT)* terminal profile. v1.4.x

Note: A new profile takes effect the next time the USIM is initialized.

16.4.1 Set command

The set command installs a new *USAT* terminal profile.

Syntax:

```
%USATP=<length>,"<profile>"
```

+CME ERROR code

50 – Incorrect parameters.

The set command parameters and their defined values are the following:

<length>

Integer, 64–78. Length of <profile> in hex characters. Two hex characters per byte, so the profile is 32–39 bytes.

<profile>

String. Hex-encoded ASCII representation of the terminal profile bytes with byte 1 first. The byte layout is defined in *ETSI TS 102 223, clause 5.2*.

The following command example installs a 32-byte terminal profile:

```
AT%USATP=64,"FFFFFFFF7F0DFFFF00000000000000000FFFFFFFF00000000000000000FFFFFFFF"
OK
```

16.4.2 Read command

The read command returns the currently configured *USAT* terminal profile.

Response syntax:

```
%USATP: <length>,"<profile>"
```

The response parameters and their defined values are the following:

<length>

Integer, 64–78. Length of <profile> in hex characters. Two hex characters per byte, so the profile is 32–39 bytes.

<profile>

String. Hex-encoded ASCII representation of the terminal profile bytes with byte 1 first. The byte layout is defined in *ETSI TS 102 223, clause 5.2*.

The following command example reads the current terminal profile:

```
AT%USATP?
%USATP: 64,"FFFFFFFF7F0DFFFF00000000000000000FFFFFFFF00000000000000000FFFFFFFF"
OK
```

16.4.3 Test command

The test command is not supported.

Glossary

Access Point Name (APN)

The name of a gateway between a mobile network and another computer network, usually the Internet.

Application Identifier (appId)

A JSON field used to identify the type of data in a device message. It is used to categorize and filter messages, for example, when multiple data types share the same topic.

Application Protocol Data Unit (APDU)

The communication unit between a terminal and smart card (*UICC*).

AT command

A command used to control the modem.

Basic Service Set Identifier (BSSID)

The MAC address of a Wi-Fi access point or radio interface.

Check Digit (CD)

The last one-digit number of the *International Mobile (Station) Equipment Identity (IMEI)* code used for error detection.

Discontinuous Reception (DRX)

A method in mobile communication to conserve the battery of a mobile device by turning the RF modem in a sleep state.

DER

Distinguished Encoding Rules

Dynamic Host Configuration Protocol (DHCP)

A network management protocol used for automatic and centralized management of IP addresses within a network.

EPS Mobility Management (EMM)

The *EPS* Mobility Management (EMM) sublayer in the *NAS* protocol provides mobility service to the *UE*.

E-UTRA Absolute Radio Frequency Channel Number (EARFCN)

LTE carrier channel number for unique identification of LTE band and carrier frequency.

Evolved Packet System (EPS)

A connection-oriented transmission network in LTE (Long-term Evolution) consisting of an EPC (Evolved Packet Core) and an E-UTRAN (Evolved Terrestrial Radio Access Network).

Evolved Terrestrial Radio Access Network (E-UTRAN)

The network architecture defined for the E-UTRA radio interface consisting of evolved Node Bs (eNB) and providing user plane and control plane protocol terminations towards the User Equipment (UE).

Extended Discontinuous Reception (eDRX)

A method to conserve the battery of an IoT (Internet of Things) device by allowing it to remain inactive for extended periods.

Firmware-Over-The-Air (FOTA)

A firmware update performed remotely over the air (OTA).

General Packet Radio Services (GPRS)

A packet-based mobile data service for 2G and 3G mobile networks with data rates of 56-114 kbps/second and continuous connection to the Internet.

General-Purpose Input/Output (GPIO)

A digital signal pin that can be used as input, output, or both. It is uncommitted and can be controlled by the user at runtime.

IP Multimedia Services Identity Module (ISIM)

An application which resides on a mobile device's *SIM* card (*UICC*).

Integrated Circuit Card Identifier (ICCID)

A unique serial number of a SIM card.

International Mobile (Station) Equipment Identity (IMEI)

A unique code consisting of 14 digits and a check digit for identifying 3GPP-based mobile devices.

International Mobile (Station) Equipment Identity, Software Version (IMEISV)

A unique code consisting of 16 decimal digits and two software version digits for identifying 3GPP-based mobile devices.

International Mobile Subscriber Identity (IMSI)

A unique code, usually 15 digits, used for the identification of a mobile subscriber and consisting of an *MCC*, *MNC*, and MSIN (Mobile Subscription Identification Number).

Internet Protocol (IP)

The network layer communications protocol in the Internet protocol suite for relaying datagrams across network boundaries. Its routing function enables internetworking, and essentially establishes the Internet.

International Reference Alphabet (IRA)

A seven-bit coded character set for information exchange.

JSON Web Token (JWT)

An Internet-proposed standard for creating data with optional signature and optional encryption whose payload holds JSON that asserts some number of claims.

Long-Term Evolution (LTE)

A wireless broadband communication standard for mobile devices and data terminals, based on the GSM/EDGE and UMTS/HSPA technologies.

Media Access Control (MAC) address

A unique 12–character hexadecimal hardware address that identifies a network interface.

Maximum Transmission Unit (MTU)

The largest packet or frame that can be sent in a single network layer transaction.

Mobile Country Code (MCC)

A unique three-digit part of an IMSI code identifying the country of domicile of the mobile subscriber. MCC is used together with the Mobile Network Code (MNC).

Mobile Equipment (ME)

The physical *UE* consisting of one of more *MT* and one or more *TE*.

Mobile Network Code (MNC)

A code identifying the telecommunications network. The code is defined by ITU-T Recommendation E.212, consists of two or three decimal digits, and is used together with the Mobile Country Code (MCC).

Mobile Station International Subscriber Directory Number (MSISDN)

A number consisting of a maximum of 15 digits identifying a mobile subscriber by mapping the telephone number to the *SIM* card in a phone.

Mobile Termination (MT)

A component of the Mobile Equipment (ME) performing functions specific to management of the radio interface. The R interface between *TE* and MT uses the AT command set. The *IMEI* code is attached to the MT.

Non-access Stratum (NAS)

In telecom protocol stacks, the highest stratum of the control plane between the core network and User Equipment (UE). The layer is used to manage the establishment of communication sessions and for maintaining communications with the UE as it moves.

nRF Cloud

Nordic Semiconductor's platform for connecting IoT devices to the cloud, viewing and analyzing device message data, prototyping ideas that use Nordic Semiconductor chips, and more. It includes a public REST API that can be used for building IoT solutions. See nRF Cloud portal (nrfcloud.com).

Non-access Stratum (NAS) Signalling Low Priority Indication (NSLPI)

Used by the network for NAS-level mobility management congestion control.

Non-volatile Memory (NVM)

Memory that can retrieve stored information even after having been power-cycled.

Over-the-Air (OTA)

Refers to any type of wireless transmission. For Matter, OTA is synonymous with the Over-the-Air software update mechanism.

Packet Data Network (PDN)

A network that provides data services.

Packet Data Protocol (PDP)

A packet transfer protocol in wireless GPRS (General Packet Radio Services) and HSDPA (High-speed Downlink Packet Access) networks.

Packet Data Protocol (PDP) Context

In UMTS (Universal Mobile Telecommunications System) and GPRS (General Packet Radio Service), the record that specifies *UE* access to an external packet-switched network.

Paging Time Window (PTW)

The period of time during which the User Equipment (UE) attempts to receive a paging message.

Personal Identification Number (PIN)

An optional security feature in mobile devices used for identifying a user. PIN is a numeric code which must be entered each time a mobile device is started.

Personal Unblocking Key (PUK)

A digit sequence required in 3GPP mobile phones to unlock a *SIM* that has disabled itself after an in correct personal identification number has been entered multiple times.

Power Management Unit (PMU)

A hardware block that manages power and clock resources in the system.

Power Saving Mode (PSM)

A feature introduced in 3GPP Release 12 to improve battery life of IoT (Internet of Things) devices by minimizing energy consumption. The device stays dormant during the PSM window.

Pre-shared Key (PSK)

A password authentication method, a string of text, expected before a username and password to establish a secured connection. Also known as a shared secret.

Protocol Configuration Options (PCO)

An element of *NAS* message used for transferring parameters between the *UE* and the P-GW (Packet Data Network Gateway).

Protocol Data Unit (PDU)

Information transferred as a single unit between peer entities of a computer network and containing control and address information or data. PDU mode is one of the two ways of sending and receiving SMS messages.

Public Land Mobile Network (PLMN)

A network that provides land mobile telecommunications services to the public. A PLMN is identified by the MCC and MNC.

Quality of Service (QoS)

The measured overall performance of a service, such as a telephony or computer network, a connection, or a cloud computing service.

Radio Resource Control (RRC)

A protocol used in UMTS, LTE, and 5G on the radio interface. It is an OSI model network layer 3 protocol used between the User Equipment and Base Station.

Received Signal Strength Indication (RSSI)

An indication of the power of a received radio signal.

Reference Signal Received Power (RSRP)

The average power level received from a single reference signal in a Long-Term Evolution (LTE) network.

Reference Signal Received Quality (RSRQ)

The quality of a single reference signal received in an LTE (Long-Term Evolution) network and calculated from *RSRP*.

SEGGER Embedded Studio (SES)

A cross-platform Integrated Development Environment (IDE) for embedded C/C++ programming with support for Nordic Semiconductor devices, produced by SEGGER Microcontroller.

Subscriber Identity Module (SIM)

A card used in User Equipment (UE) containing data for subscriber identification.

Serial Number (SNR)

A unique six-digit number part of the *IMEI* code identifying each equipment within each *Type Allocation Code (TAC)*.

Service Set Identifier (SSID)

The unique name of a Wi–Fi network.

Software Version Number (SVN)

Part of the *IMEI* code identifying the revision of the software installed on a mobile device.

Terminal Adapter (TA)

A device that connects a *UE* to a communications network. In mobile networks, the terminal adapter is used by the terminal equipment to access the mobile termination using AT commands.

Terminal Equipment (TE)

Communications equipment at either end of a communications link, used to permit the stations involved to accomplish the mission for which the link was established.

Tracking Area Code (TAC)

A unique code used to identify a tracking area within a particular network.

Tracking Area Update (TAU)

A procedure initiated by the User Equipment (UE) when moving to a new tracking area in the LTE (Long-term Evolution) system.

Transport Layer Security (TLS)

A cryptographic protocol that provides end-to-end security of data sent over a computer network.

Type Allocation Code (TAC)

The initial eight-digit part of an International Mobile (Station) Equipment Identity (IMEI) code used for identifying the model of a mobile phone.

Universal Asynchronous Receiver/Transmitter (UART)

A hardware device for asynchronous serial communication between devices.

Unsolicited Result Code (URC)

A result code sent by the modem independently of the direct response to an AT command.

User Equipment (UE)

Any device used by an end-user to communicate. The UE consists of the Mobile Equipment (ME) and the Universal Integrated Circuit Card (UICC).

Universal Integrated Circuit Card (UICC)

A new generation Subscriber Identity Module (SIM) used in User Equipment (UE) for ensuring the integrity and security of personal data.

Universal Subscriber Identity Module Application Toolkit (USAT)

A standard that provides mechanisms which allow applications in the *USIM* to interact and operate with *ME*.

Universal Subscriber Identity Module (USIM)

A card used in *UE* containing data for subscriber identification.

UUID

Universally Unique Identifier

Legal notices

By using this documentation you agree to our terms and conditions of use. Nordic Semiconductor may change these terms and conditions at any time without notice.

Liability disclaimer

Nordic Semiconductor ASA reserves the right to make changes without further notice to the product to improve reliability, function, or design. Nordic Semiconductor ASA does not assume any liability arising out of the application or use of any product or circuits described herein.

Nordic Semiconductor ASA does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. If there are any discrepancies, ambiguities or conflicts in Nordic Semiconductor's documentation, the Product Specification prevails.

Nordic Semiconductor ASA reserves the right to make corrections, enhancements, and other changes to this document without notice.

Life support applications

Nordic Semiconductor products are not designed for use in life support appliances, devices, or systems where malfunction of these products can reasonably be expected to result in personal injury.

Nordic Semiconductor ASA customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify Nordic Semiconductor ASA for any damages resulting from such improper use or sale.

RoHS and REACH statement

Complete hazardous substance reports, material composition reports and latest version of Nordic's REACH statement can be found on our website www.nordicsemi.com.

Trademarks

All trademarks, service marks, trade names, product names, and logos appearing in this documentation are the property of their respective owners.

Copyright notice