AirPrime UMTS MC Series

CnS Reference

2130602
Rev. 2

Important Notice

Due to the nature of wireless communications, transmission and reception of data can never be guaranteed. Data may be delayed, corrupted (i.e. have errors) or be totally lost. Although significant delays or losses of data are rare when wireless devices such as the Sierra Wireless modem are used in a normal manner with a well-constructed network, the Sierra Wireless modem should not be used in situations where failure to transmit or receive data could result in damage of any kind to the user or any other party, including but not limited to personal injury, death, or loss of property. Sierra Wireless accepts no responsibility for damages of any kind resulting from delays or errors in data transmitted or received using the Sierra Wireless modem, or for failure of the Sierra Wireless modem to transmit or receive such data.

Safety and Hazards

Do not operate the Sierra Wireless modem in areas where blasting is in progress, where explosive atmospheres may be present, near medical equipment, near life support equipment, or any equipment which may be susceptible to any form of radio interference. In such areas, the Sierra Wireless modem **MUST BE POWERED OFF**. The Sierra Wireless modem can transmit signals that could interfere with this equipment.

Do not operate the Sierra Wireless modem in any aircraft, whether the aircraft is on the ground or in flight. In aircraft, the Sierra Wireless modem **MUST BE POWERED OFF**. When operating, the Sierra Wireless modem can transmit signals that could interfere with various onboard systems.

Note: Some airlines may permit the use of cellular phones while the aircraft is on the ground and the door is open. Sierra Wireless modems may be used at this time.

The driver or operator of any vehicle should not operate the Sierra Wireless modem while in control of a vehicle. Doing so will detract from the driver or operator's control and operation of that vehicle. In some states and provinces, operating such communications devices while in control of a vehicle is an offence.

Limitation of Liability

The information in this manual is subject to change without notice and does not represent a commitment on the part of Sierra Wireless. SIERRA WIRELESS AND ITS AFFILIATES SPECIFICALLY DISCLAIM LIABILITY FOR ANY AND ALL DIRECT, INDIRECT, SPECIAL, GENERAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES INCLUDING, BUT NOT LIMITED TO, LOSS OF PROFITS OR REVENUE OR ANTICIPATED PROFITS OR REVENUE ARISING OUT OF THE USE OR INABILITY TO USE ANY SIERRA WIRELESS PRODUCT, EVEN IF SIERRA WIRELESS AND/OR ITS AFFILIATES HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES OR THEY ARE FORESEEABLE OR FOR CLAIMS BY ANY THIRD PARTY.

Notwithstanding the foregoing, in no event shall Sierra Wireless and/or its affiliates aggregate liability arising under or in connection with the Sierra Wireless product, regardless of the number of events, occurrences, or claims giving rise to liability, be in excess of the price paid by the purchaser for the Sierra Wireless product.

Patents

Portions of this product may be covered by some or all of the following US patents:

5,515,013	5,629,960	5,845,216	5,847,553	5,878,234	5,890,057	5,929,815	6,169,884
6,191,741	6,199,168	6,339,405	6,359,591	6,400,336	6,516,204	6,561,851	6,643,501
6,653,979	6,697,030	6,785,830	6,845,249	6,847,830	6,876,697	6,879,585	6,886,049
6,968,171	6,985,757	7,023,878	7,053,843	7,106,569	7,145,267	7,200,512	7,295,171
7, 287,162	D442,170	D459,303	D599,256	D560,911			

and other patents pending.

This product includes technology licensed from QUALCOMM® 3G

Manufactured or sold by Sierra Wireless or its licensees under one or more patents licensed from InterDigital Group.

Copyright

Trademarks

AirCard® and Watcher® are registered trademarks of Sierra Wireless. Sierra Wireless™, AirPrime™, AirLink™, AirVantage™ and the Sierra Wireless logo are trademarks of Sierra Wireless.

Windows® and Windows Vista® are registered trademarks of Microsoft Corporation.

Macintosh and Mac OS are registered trademarks of Apple Inc., registered in the U.S. and other countries.

QUALCOMM® is a registered trademark of QUALCOMM Incorporated. Used under license.

Other trademarks are the property of their respective owners.

Contact Information

Sales Desk:	Phone:	1-604-232-1488
	Hours:	8:00 AM to 5:00 PM Pacific Time
	E-mail:	sales@sierrawireless.com
Post:	Sierra Wireless 13811 Wireless Way Richmond, BC Canada V6V 3A4	
Fax:	1-604-231-1109	
Web:	www.sierrawireless.com	

Consult our website for up-to-date product descriptions, documentation, application notes, firmware upgrades, troubleshooting tips, and press releases:

www.sierrawireless.com

Revision History

Revision number	Release date	Changes
1.6	May 2007	• Removed unused objects (0x001A / 0x001B) • Expanded 'Phonebook Management' chapter • Added Acronyms chapter
1.7	Draft - Aug 2007	• Updated firmware download section for bootloader / self-executing application images
1.8	Unreleased	• Updated parameters in Get Modem Type (Modem type), Return Supported Object Versions (Number of entries, Object ID identifier); Report Current Band (Current band in use); Set Current Band (Device-supported band groupings); Report Available Service Details (Service type available) • Added object version 2 to Report Network Status and updated parameters (System mode, Current band) • Added new objects: Return Modem Date and Time; Return TRU-Install Version Information; Return PRI Information; Return Flash Image Information; Return RSCP and Ec/Io Measurements; Return GSM Cell Information; Manage SMS Status Reports
1.9	Jan 2008	• Added new chapter: Location Based Services
1.10	Jul 2008	• Updated Set Current Band—added new band group values • Added Report call byte counters (Rx/Tx), Reset call byte counters, Unlock protected API commands, Return SIM's ICC ID, Set RAT mode, Set PRL region • Updated modem type list for Get Modem Type and Return Modem Model • Updated Return Flash Image Information—Fixed descriptions of object values ('null padded') • Updated Return PRI Information—Corrected Get response length • Updated Report network-initiated location fix request—New object version. • Added usage note to Report System and Network Status
1.11	Sep 2008	• Added MC8791V/92V to supporting modems list • Updated modem type list for Get Modem Type and Return Modem Model • Updated Link Status Indication (additional fields) • Removed Return TRU-Install Version Information, Set PRL region

Revision number	Release date	Changes
1.12	Jul 2009	• Added MC8700 to supported products list • Deprecated 0x7000 in favor of 0x000B. • Updated Table 4-4 (HIP Link Status Indication format)—Clarified payload details • Updated Table 4-6 (UMTS PPP protocol-specific Link Status buffer) - Updated coverage codes • Updated Table 7-5 (Supported CnS operation types)—Clarified error response type • Deprecated Back Up NVRAM Data (Deprecated) and fixed Response field details • Added Back Up NVRAM Data, Manage Profile Username and Password • Updated Get Modem Type object—New object version, new modem types • Updated Return Modem Date and Time—Added notification details • Updated Return PRI Information—Clarified field descriptions • Updated Return Flash Image Information—Updated usage details • Updated Return Available PLMN List—Updated network types in response • Updated Table 15-2 (SMS envelope structure)—Added option to Packet Type field • Updated Get Unread SMS Message—Fixed initial offset value in response
2.0	July 2010	• New template/branding • Organized by chipset • Added Device Information Request and Device Information Response HIP messages • Renamed and reorganized Selected HIP messages chapter (was Unsolicited HIP messages) • Updated Get Modem Type and Return Modem Model—Added ‘other products’, suggested use of new Device Information Request HIP message • Updated Return Radio Temperature—Clarified Notification ‘trigger’; changed Current temperature parameter type • Updated Report location fix results height field description • Added Report Auto Track feature state object • Updated Report Available Service Details and Report System and Network Status—Added more service types • Removed Unlock protected API commands, Set PRL region • Deprecated Report call byte counters (Rx/Tx) and Reset call byte counters • Added Manage call byte counters (Rx/Tx) • Updated Verify CHV Code—Fixed unlock CHV code length • Updated Read Profile—Modified Configuration Buffer Label format • Updated Read Profile, Write Profile, Manage Packet Session, Return Packet Session Status—Added additional bit rate values • Updated Read Profile, Write Profile, Return IP Address—Described use of IPv6 format in several fields

Contents

1: About This Guide

Introduction

Note: Throughout this document, "MC8xxx" refers to the full suite of data-only and voice-enabled UMTS Mini Card modems.

This document provides a reference for using HIP (Host Interface Protocol) and CnS (Control and Status language) when developing applications for use on AirPrime UMTS MC-series embedded modems (MC8xxx). These are proprietary protocols for managing the control and status of the MC8xxx modem:

- **HIP** is an encapsulating protocol intended to carry a variety of other protocol packets across a single link layer
- **CnS** is one of the types of packets that can be carried inside HIP

Note: Watcher® (and other Sierra Wireless applications) implement these protocols—use these applications as functionality templates when designing your own applications to ensure proper use of HIP and CnS.
For questions or concerns relating to protocol implementation, please contact your Sierra Wireless account representative.

This document includes the following information:

- An explanation of the process used by the MC8xxx modem to transmit and receive control and status messages
- A description of the process used by the host computer to communicate with the modem
- A description of the process for performing a firmware download
- A description of the messages used to implement protocol stack logging
- Detailed descriptions of each of the supported CnS messages

The protocols as described in this document are specific to the Sierra Wireless AirPrime embedded modems listed in Table 1-1.

Table 1-1: Supported AirPrime embedded modems[a]

Chipset	Devices
MDM8200	MC8700
MSM6290	MC8790/MC8790V MC8791V MC8792V MC8795V
MSM7200 **MSM7201**	MC8780[b] MC8781[b]

a. Changes made to this document after a given device reaches end-of-life may not apply to that device. Refer to the Revision History on page 5 for a summary of changes made in recent releases of this document.
b. Legacy device. Changes made since document Rev 1.12 may not apply.

Document structure

This document includes the following chapters:

1: About This Guide—Introductory information about the MC8xxx modem and associated firmware, and typographic conventions used in the document.

2: Overview—Introduces the MC8xxx modem and provides general details explaining:

- How the modem transmits and receives data wirelessly over the GSM/UMTS network
- How the host computer communicates to the modem
- Data transfer types
- Transaction sequences

3: HIP Basics—Describes the Host Interface Protocol. This is the protocol for encapsulating other protocols to and from the modem. This chapter also includes a reference of the message types you may use with this protocol.

4: Selected HIP messages—Describes HIP requests/responses and unsolicited indications that are used to obtain device or packet session information.

5: Firmware Download—Describes the sequence of CnS and HIP messages that is required for downloading a firmware upgrade to the modem from the host.

6: Protocol Stack Logging—Describes the HIP messages used to tunnel Qualcomm DM messages via the protocol stack.

7: CnS Basics—Describes the general format and process of communicating with Wireless Interface Control Language. The specific message reference is found in the following chapters.

8: Supported CnS Messages—Provides a summary of the supported CnS messages in object number order. This allows you to locate the description of a message, given only an object ID.

9: Device Characteristics—Describes the control and status messages used to determine elements of the hardware characteristics and control fundamental hardware elements (such as resets).

10: SIM Status and Locking—Describes the CnS messages that are used to work with the SIM, including verifying and changing CHV codes, and reporting the status of SIM-related operations.

11: Profile Management—Describes the CnS messages that are used to create, maintain, and delete packet session profile IDs.

12: Status Messages—Describes the CnS messages that indicate nominal operation of the modem and notify the modem of changes in the host application status.

13: Data Services—Describes the CnS messages that are used to report network and PLMN status information, choose the band being used, and manage packet sessions.

14: Phonebook Management—Describes the CnS messages that are used to identify available phonebooks and the operations that can be performed on them.

15: SMS Messages—Describes the CnS messages that are used to handle SMS messaging.

16: Location Based Services—Describes the CnS messages that are used for LBS processing.

Document version

This document is revised periodically as new firmware features become available. This document edition is:

Rev 2, 5 August 2010

Firmware version

The protocols described in this guide apply to the modems listed in Table 1-1 on page 17. Unless otherwise noted, the following minimum firmware revisions are required to support the protocols as described:

- MDM8200—Revision M2_0_10_0ap or higher
- MSM6290—Revision K2_0_7_29ap or higher
- MSM7200/MSM7201—Changes made since Revision 1.12 of this document may not apply to devices using these chipsets. Revision U0_0_0_8mcap or higher is required for Revision 1.12 content.

To determine your firmware revision:

- Enter the Identification command: **AT+GMR**.
  The modem responds with version information for software, firmware, and hardware.
  The details following the revision number include Sierra Wireless information on the specific build followed by the date and time of the build.
  +GMR: … F/W VER: R1_4_1_…

Upgrading

If your modem firmware is an earlier version, you can acquire updated firmware by contacting your account manager.

See Downloading firmware updates on page 44.

References

For background information on GSM/UMTS technology, visit the GSM World web site at www.gsmworld.com and the UMTS World web site at www.umtsworld.com.

Conventions

Numeric values

Numeric values are shown in the following formats:

- Binary—used with a prefix of 0b (for example, 0b00111101)
- Hexadecimal—used with a prefix of 0x (for example, 0x3D)
- Decimal—used without a prefix (for example, 61)

Citations and links

Citations of command names, object values, or particular samples are shown in a bold font.

Heartbeat

If you are using the PDF version of the document, references are hyperlinked to allow you to quickly jump to target information. All page numbers are hyperlinked, and cross-referenced objects with links are shown in blue.

Heartbeat

Parameter and data element format examples

Parameter and data element samples are generally noted in a sans serif font: R*x-y-z*. In this sample, the use of italics and lowercase indicates the item(s) can vary in specifics and should not be taken literally.

Character codes

Character codes, which are described with words or standard abbreviations, are shown within angle brackets: such as <CR> for Carriage Return and <space> for a blank space character. This font is also used to highlight options, technical references such as web sites or e-mail addresses.

Reference tables

The conventions used to summarize objects and provide detailed object descriptions are defined on page 62.

Individual objects present the syntax in a table showing the object value, the supported message types and the parameters for each type.

2: Overview

Supported products

This edition of the CnS reference applies only to Sierra Wireless AirPrime UMTS MC-series embedded modems.

Some of the CnS messages described in this document belong to a core set of messages that are supported by other Sierra Wireless products. The remainder of the messages are customized to deal with specific product features.

MC8xxx embedded modem

The MC8xxx embedded modem supports these primary features:

- GPRS/EDGE/UMTS packet-switched data
- Circuit-switched data
- SMS (Short Message Service) (both transmitting and receiving)

There are other features related to modem power control, reset, and shutdown, and SIM security that are also covered by CnS.

All wireless products require some form of user activation to allow them to operate with a wireless service provider or carrier. CnS is used to manage this activation process as well.

Communication methodology

The MC8xxx embedded modem communicates with its host platform via a USB port.

Other Sierra Wireless products may use different techniques, including an NDIS interface in Windows® so that the modem appears to be a NIC (Network Interface Card) rather than a modem.

HIP and CnS

To permit easier portability of software and firmware, Sierra Wireless has created the proprietary Host Interface Protocol (HIP). It serves as a standard method to communicate over whatever physical link layer is used by the product. This allows the same applications to access devices using different connection mechanisms. Only the driver used to implement the physical link layer is different.

MC8xxx communication

The MC8xxx embedded modem communicates with its host application by using HIP to send/receive message packets over a USB port.

3: HIP Basics

HIP overview

HIP (Host Interface Protocol) is a multiplexing layer designed to allow several types of message streams to share the same physical link layer—CnS is one of the supported streams.

HIP is used to carry control and status data between the modem and the host to provide management of the modem device. This management consists of:

- Non-volatile configuration of the device
- Run-time configuration of the device
- Status reporting and monitoring of the device

Control and data messages are transferred between the modem and host using packets consisting of message frames surrounded by framing characters. Each frame includes a header (standard or extended) and optional payload, where the payload is a modified Ethernet frame carrying specific message details and, if appropriate, an IP packet. A HIP packet has the following structure:

Table 3-1: HIP packet format

Byte offset	Length	Content
0	1	Framing character (0x7E)
1		Message header (See Figure 3-1 on page 24) • Short header—Length=4 bytes • Extended header—Length=6 bytes
	2	Message payload length
	1	Short message identifier
	1	Message-specific identifier
	2	(Optional) Extended message identifier
5 (Short header) or 7 (Extended header)	Variable	(Optional) Message payload
Variable	1	Framing character (0x7E)

Note: The minimum length of a HIP packet is six bytes (short header format) or eight bytes (extended header) for a packet without the optional payload. Note that such a packet would have the Message payload length field (offset 1) set to 0x0000—no payload.

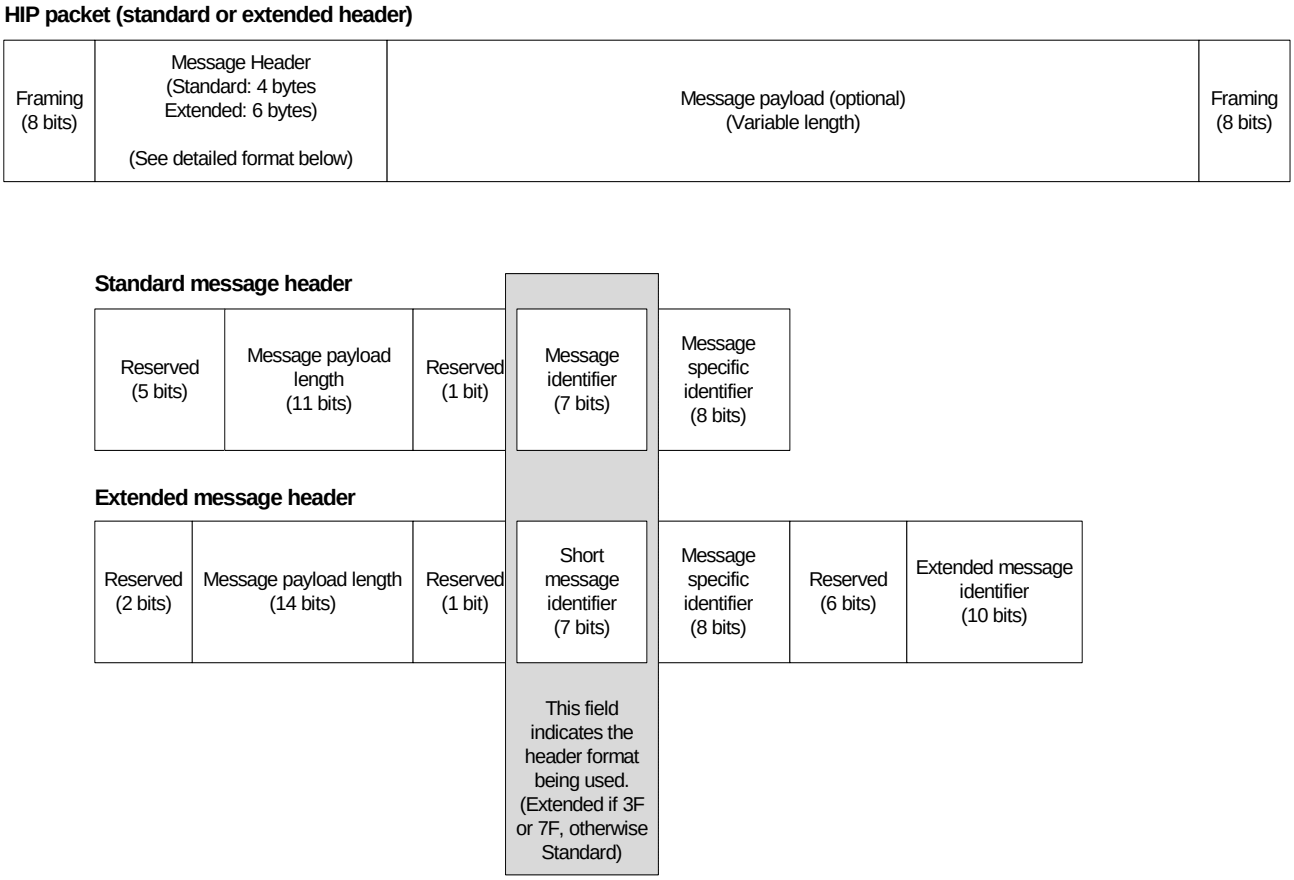

Figure 3-1: HIP packet format (with standard and extended headers)

Framing All HIP packets open and close with a framing character. This is the frame character (0x7E). Framing is discussed in more detail in Packet framing on page 25.

Length The final *unframed* message payload length. This refers only to the Message payload field of the HIP packet—it does not include the framing characters or the Message header. The maximum recommended length of an *unframed* HIP packet payload is 2000 bytes (standard header) or 4095 bytes (extended header). The size of a framed packet can be much larger, depending on the number of characters that are 'escaped' (see Using escape characters on page 25).

Note: The maximum unframed packet payload length for an extended header is 16383 bytes. The recommended length of 4095 bytes prevents excessive memory buffer allocation requirements.

Short message ID The short message ID field is composed of a 1-bit flag and 7-bit identifier. The flag bit, if set, indicates that the optional parameter field begins with a pad byte of random data (possibly added to make the data word-aligned). The pad byte is included in the length given in the first field.

The 7-bit message ID itself provides information about the type of HIP packet. Different IDs are used for messages going to, and coming from, the modem. This prevents echoing across the interface.

If the ID is 0x3F (for device bound messages) or 0x7F (for host bound messages), then the header is an extended message header where the final two bytes are the extended message ID.

Extended message ID The extended message ID field (like the short message ID field) identifies the type of HIP packet. (The extended header format was created to allow a greater number of HIP messages to be created after exhausting the message ID space available for standard message headers.)

Message specific identifier This identifier is rarely used, but some messages do carry an informative flag in this field.

Payload The payload varies by message type and purpose.

This protocol does not include checksums. It relies on a high-reliability physical interface between the host and the modem to reduce the processing burden on the microprocessor.

Packet framing

All HIP packets are variable length. The link layer over the USB port requires marking the HIP packets with start and end flags, and escaping the data within.

The HDLC mechanism is used by the MC8xxx embedded modem; the value of the flag is 0b01111110 (0x7E). This mechanism ensures that an HIP message can be detected by placing a unique character (the flag) at the beginning and end of the frame.

Note: Only one flag byte is required for back-to-back packets. Two consecutive flags constitute an empty frame and are silently discarded.

Using escape characters

As the framing character (0x7E) can occur within the frame itself, you need to use escaping/un-escaping to distinguish the data value from the frame character.

The escape character is 0x7D. Since this character is also treated as part of the protocol process, it too must be escaped if it appears in the data stream.

Note: There may be at least some escaping of characters in a message, so the actual framed packet may be larger than the size indicated in the HIP Length field.

If either of the values 0x7D or 0x7E occur in the frame, they must be replaced by two (2) bytes: 0x7D (the escape character) and then the data character exclusive-ORed with 0x20.

Therefore a binary data stream of:

0x30 0x37 0x7E 0x65 0x7D 0x66

would be sent as:

0x30 0x37 0x7D 0x5E 0x65 0x7D 0x5D 0x66

In the above example the offending frame character 0x7E is replaced by the escape character 0x7D followed by the exclusive-or result of 0x7E with 0x20 (0x5E). Since 0x7D is the escape flag and also occurs in the sample above, it too must be replaced using the same method.

HIP interface rules

- Any message containing an undefined message ID is quietly discarded when it is received by either side of the interface.
- If the maximum length of a message after deframing (replacing the ‘escaped’ character sequences) is ever exceeded, the receiver discards the entire packet and ignores the message.
- All multi-byte fields used in this protocol are in network byte order (Big-Endian). That is, the most significant byte is sent first, followed by less significant bytes. The least significant byte of a multi-byte field is sent last. Note that this differs from the way Intel processors normally store multi-byte fields; special methods must be used to write multi-byte fields.
- The host shall never send a message to the modem with the pad byte flag set (in the message ID). The modem ignores the pad byte flag.

HIP message types

Three fundamental types of HIP messages flow between the host and the modem:

- Requests
- Responses
- Indications

Requests

A request message provides a means for the host (master) to ask the modem (slave) to either perform an operation or return a value. The slave always responds to a request message with a response message. With this master/slave relationship, the modem is not allowed to make requests of the host. One, and only one, response is sent for every request.

To provide flow control to the modem, a response to a specific request must be received before a second request of the same type can be issued. This effectively provides stop-and-wait flow control on a per request basis. If the host never receives a response to a request (timeout), this is a failure case and thus the host can assume that the modem has failed and should be recovered from a restart state.

Request short message IDs are in the range of 00–31 (0x00–0x1F) or, if the short message ID is 3F, the message includes an extended message ID.

Responses

A response message is always sent from the modem, in reply to a request from the host. The format and type of data in the response varies according to the type of response. Responses can never be sent unsolicited (That is, every response is paired with a specific request).

Response short message IDs are in the range of 64–95 (0x40–0x5F) or, if the short message ID is 7F, the message includes an extended message ID.

Indications

Indication messages provide a means to send unsolicited data from either side of the interface. Indications are used when no reply or return data is required from the receiver (or conversations do not require stop-and-wait flow control).

No response is ever sent by the receiver of a indication. The encapsulated payload protocol may require responses or replies (as many CnS messages do), but this is outside of the HIP specification.

Indication message IDs sent from the host are in the range 32–63 (0x20–0x3F). Indications from the modem have a message ID in the range of 96–127 (0x60–0x7F).

Unsolicited HIP messages

At times, the module may issue unsolicited HIP messages (indications) to inform the host of changes in the current operating conditions.

MC8xxx support

The MC8xxx embedded modem supports all HIP message types (requests, responses, indications).

For example, some HIP requests and responses are described in Related HIP messages on page 49 (for firmware downloads), while all CnS messages are encapsulated within HIP indications. (CnS messages are treated as asynchronous events and can originate from either side—host or modem.)

HIP CnS message reference

To support CnS, the MC8xxx embedded modem uses only two HIP indication message IDs:

- Modem CnS indication—messages *to* the modem
- Host CnS indication—messages *from* the modem.

The CnS message payload of these packets is the subject of the following chapters.

Modem CnS indication

The modem CnS indication is an unsolicited message sent by the host to the modem for transport of CnS messages.

Table 3-2: Host–Modem indication format

Short Message ID	0x2B (43)
Message-specific parameter	None
Message payload	CnS message

Host CnS indication

The host CnS indication is an unsolicited message sent by the modem to the host for transport of CnS messages.

Table 3-3: Modem–Host indication format

Short Message ID	0x6B (107)
Message-specific parameter	None
Message payload	CnS message

4: Selected HIP messages

This chapter describes selected HIP messages that are used to obtain device or packet session information.

Device information

Device Information Request

Note: This message should be used instead of the Cns objects Get Modem Type (page 75) and Return Modem Model (page 91).

This message is sent from the host to the modem to request the modem type. The modem type is returned in a Device Information Response. The host must wait until the response is received before sending another Device Information Request.

Table 4-1: Device Information Request format

Short Message ID	**0x3F**
Extended Message ID	**0x001**
Message-specific parameter	Unused
Message payload	Message version (2 bytes) • 0x03
	Request type (1 byte) • 0x02=Product string

Device Information Response

This message is sent to the host from the modem in response to a Device Information Request.

Table 4-2: Device Information Response format

Short Message ID	**0x7F**
Extended Message ID	**0x201**
Message-specific parameter	Unused
Message payload	Message version (2 bytes) • 0x03

Table 4-2: Device Information Response format

	Request status (1 byte) • 0x01=Success • 0x02=Unknown type requested • 0x03=General failure
	Response content (TLV) • Type—0x02 (Product string) • Length—15 (bytes) • Value—ASCII string, null padded. (This is the same model string that is returned using the AT command **ATI**.)

RSSI change Indication

This unsolicited indication is sent by the module each time that the RSSI (Receive Signal Strength Indication) changes by more than 5 dB.

Table 4-3: HIP RSSI Indication format

Short Message ID	**0x75**
Message-specific parameter	Unused
Message payload	RSSI—current radio signal strength The RSSI value relative to noise level (-110 dBm for GSM, -125 dBm for WCDMA) • 4 bytes • Value is always positive

Packet session information

Link Status Indication

This unsolicited indication is sent by the module when the packet session status changes.

The host must send a Link Status Confirmation Indication when it is ready to use the new IP address identified in the Link Status Indication message payload.

Table 4-4: HIP Link Status Indication format

Short Message ID	**0x78**
Message-specific parameter	Context Index Number • 0—PDP1 • 2–3—PDP2 and PDP3 (only possible if multiple contexts are supported)

Table 4-4: HIP Link Status Indication format (Continued)

Message payload	Protocol Type (1 byte) • 0x01—UMTS (Direct IP) • 0x03—UMTS PPP
	Profile ID (1 byte)—Optional field • 0x00/0xFF—Indicates that no specific session is being referred to (for example, at startup) • 0x01–0x10—The ID of the session which has changed state. This is the ID that was used when the AT/PPP context was starting using the Dial command (ATD*98*<profileID>#)
	Length of next field—Payload space (2 bytes)
	Protocol-specific Link Status buffer • Protocol type 0x01: See Table 4-5 on page 31 for the buffer structure. • Protocol type 0x03: See Table 4-6 on page 37 for the buffer structure.

Table 4-5: UMTS (Direct IP) protocol-specific Link Status buffer

Offset (bytes)	Length (bytes)	Type	Field	Description
0	1	UINT8	Coverage Code	Type of coverage currently available: • 0x00=No coverage • 0x01=Circuit-switched available (no packet-switched coverage) • 0x02=GPRS packet coverage available • 0x03=EDGE packet coverage available • 0x04=UMTS packet coverage available • 0x05=HSDPA packet coverage available
1	1	UINT8	Network Name length	Length of the network name (next field) • Valid range: 0–20 • If 0, network name is not available.
2	40	Unicode	Network Name string	Name of network that modem is currently registered on • See previous field for string length (remaining characters are undefined). • UNICODE-format characters (up to twenty).
42	1	UINT8	Session State	Type of packet session in progress: • 0x04=NDIS Packet • 0x10=UART CSC Data %% • 0x20=UART Voice %% • 0xFF=No session

Table 4-5: UMTS (Direct IP) protocol-specific Link Status buffer (Continued)

Offset (bytes)	Length (bytes)	Type	Field	Description
43	33	CHAR	Reason string	Reason for link status change • 0–32 characters plus terminating NULL character • Any remaining characters are undefined (field is always 33 bytes long)
76	1	UINT8	Active session link type (PDP context type)	Valid values: • 0=IPv4 session • 1=PPP session • 2=IPv6 session • 255—Unknown (no session established)
77	1	UINT8	Session (PDP) address length	Length of session address (next field) • 0 (Determine the IP addresses from normal PPP negotiations) • 4 (IPv4 format) • 16 (IPv6 format)
78	16	UINT8	Session (PDP) address	IP address of the session • Length—See previous field. • Format—IPv4 or IPv6 address format. If IPv4, first four bytes contain the address and remaining bytes are set to 0. • Examples: (IPv4) 0x7F000201=127.0.2.1 (IPv6) 0x000020010DB8000000000202B3FF8329= 2001:DB8: : :0202:B3FF:8329
94	1	UINT8	Valid flag for QoS	Indicates if Quality of Service values are used. • 0x00=Invalid (QoS values in this buffer will be set to 0) • 0x01=Valid
95	1	UINT8	QoS precedence	• Valid range: 0–3 • 0=When Valid flag for QoS is 0.
96	1	UINT8	QoS delay	• Valid range: 0–4 • 0=When Valid flag for QoS is 0.
97	1	UINT8	QoS reliability	• Valid range: 0–5 • 0=When Valid flag for QoS is 0.
98	1	UINT8	QoS peak	• Valid range: 0–9 • 0=When Valid flag for QoS is 0.
99	1	UINT8	QoS mean	• Valid range: 0–31 • 0=When Valid flag for QoS is 0.

Table 4-5: UMTS (Direct IP) protocol-specific Link Status buffer (Continued)

Offset (bytes)	Length (bytes)	Type	Field	Description
100	1	UINT8	QoS Traffic Class	• Valid range: 0–4 • 0=When Valid flag for QoS is 0.
101	1	UINT8	QoS Max Bitrate UL	• 0=Based on user's subscription (or when Valid flag for QoS is 0) • 1=16 Kbps • 2=32 Kbps • 3=64 Kbps • 4=128 Kbps • 5=256 Kbps • 6=384 Kbps • 7=1.8 Mbps • 8=3.6 Mbps • 9=7.2 Mbps • 10=14.4 Mbps • 10=14.4 Mbps • 11=2.048 Mbps • 12=5.76 Mbps • 13=11.5 Mbps • 14=16 Mbps • 15=21 Mbps • 16=28 Mbps • 17=42 Mbps • 18=84 Mbps
102	1	UINT8	QoS Max Bitrate DL	Same range of values as QoS Max Bitrate UL.
103	1	UINT8	QoS Guaranteed Bitrate UL	Same range of values as QoS Max Bitrate UL.
104	1	UINT8	QoS Guaranteed Bitrate DL	Same range of values as QoS Max Bitrate UL.
105	1	UINT8	QoS Delivery Order	• Valid range: 0–2 • 0=When Valid flag for QoS is 0.
106	1	UINT8	QoS Delivery of Erroneous SDU	• Valid range: 0–3 • 0=When Valid flag for QoS is 0.
107	2	UINT16	QoS Maximum SDU size	• Valid range: 0–1520 • 0=When Valid flag for QoS is 0.
109	1	UINT8	QoS SDU error ratio	• Valid range: 0–7 • 0=When Valid flag for QoS is 0.

Table 4-5: UMTS (Direct IP) protocol-specific Link Status buffer (Continued)

Offset (bytes)	Length (bytes)	Type	Field	Description
110	1	UINT8	QoS Residual Bit error ratio	• Valid range: 0–9 • 0=When Valid flag for QoS is 0.
111	2	UINT16	QoS Transfer Delay	• Valid range: 0–4000 • 0=When Valid flag for QoS is 0.
113	1	UINT8	QoS Traffic Handling priority	• Valid range: 0–3 • 0=When Valid flag for QoS is 0.
114	1	UINT8	QoS Source statistics descriptor	• Valid range: 0–1 • 0=When Valid flag for QoS is 0.
115	1	UINT8	QoS Signaling Indication	• Valid range: 0–1 • 0=When Valid flag for QoS is 0.
116	1	UINT8	PDP Initiated Type	• 0=Primary Mobile Initiated • 1=Secondary Mobile Initiated %% • 2=Network Initiated %% • 255=No context defined
117	1	UINT8	Network-supplied Primary DNS address length	The length of the primary DNS address (next field), which also indicates address type. • 0=No gateway address is present • 4=IPv4 address • 16=IPv6 address *Note: The network cannot supply a DNS address if the activated GPRS configuration already has a user-specified DNS address.*
118	16	UINT8	Network-supplied Primary DNS address	The primary DNS address supplied by the network. • Length—See previous field. • Format—IPv4 or IPv6 address format. If IPv4, first four bytes contain the address and remaining bytes are set to 0. • Examples: (IPv4) 0x7F000201=127.0.2.1 (IPv6) 0x000020010DB8000000000202B3FF8329=2001:DB8: : :0202:B3FF:8329

Table 4-5: UMTS (Direct IP) protocol-specific Link Status buffer (Continued)

Offset (bytes)	Length (bytes)	Type	Field	Description
134	1	UINT8	Network-supplied Secondary DNS address length	The length of the secondary DNS address (next field), which also indicates address type. • 0=No gateway address is present • 4=IPv4 address • 16=IPv6 address *Note: The network cannot supply a DNS address if the activated GPRS configuration already has a user-specified DNS address.*
135	16	UINT8	Network-supplied Secondary DNS address	The secondary DNS address supplied by the network. The previous field indicates the length of the string. • Length—See previous field. • Format—IPv4 or IPv6 address format. If IPv4, first four bytes contain the address and remaining bytes are set to 0. • Examples: (IPv4) 0x7F000201=127.0.2.1 (IPv6) 0x000020010DB8000000000202B3FF8329= 2001:DB8: : :0202:B3FF:8329
151	1	UINT8	Network-supplied Primary WINS address length	The length of the primary WINS address (next field), which also indicates address type. • 0=No gateway address is present • 4=IPv4 address • 16=IPv6 address *Note: The network cannot supply a WINS address if the activated GPRS configuration already has a user-specified WINS address.*
152	16	UINT8	Network-supplied Primary WINS address	The primary WINS address supplied by the network. • Length—See previous field. • Format—IPv4 or IPv6 address format. If IPv4, first four bytes contain the address and remaining bytes are set to 0. • Examples: (IPv4) 0x7F000201=127.0.2.1 (IPv6) 0x000020010DB8000000000202B3FF8329= 2001:DB8: : :0202:B3FF:8329

Table 4-5: UMTS (Direct IP) protocol-specific Link Status buffer (Continued)

Offset (bytes)	Length (bytes)	Type	Field	Description
168	1	UINT8	Network-supplied Secondary WINS address length	The length of the secondary WINS address (next field), which also indicates address type. • 0=No gateway address is present • 4=IPv4 address • 16=IPv6 address *Note: The network cannot supply a WINS address if the activated GPRS configuration already has a user-specified DNS address.*
169	16	UINT8	Network-supplied Secondary WINS address	The secondary WINS address supplied by the network. The previous field indicates the length of the string. • Length—See previous field. • Format—IPv4 or IPv6 address format. If IPv4, first four bytes contain the address and remaining bytes are set to 0. • Examples: (IPv4) 0x7F000201=127.0.2.1 (IPv6) 0x000020010DB8000000000202B3FF8329=2001:DB8: : :0202:B3FF:8329
185	2	UINT16	Maximum D/L link speed	Maximum download link speed (kbps) • Valid range: 0–65535
187	2	UINT16	Maximum U/L link speed	Maximum upload link speed (kbps) • Valid range: 0–65535
189	1	UINT16	Network-supplied gateway address length	The length of the gateway address (next field). • Valid range: 0–16 • If length is 0, no gateway address is present.
190	16	UINT8	Network-supplied gateway address	The gateway address supplied by the network. The previous field indicates the length of the string. • Length—See previous field • Format—IPv4 or IPv6 address format. If IPv4, first four bytes contain the address and remaining bytes are set to 0. • Examples: (IPv4) 0x7F000201=127.0.2.1 (IPv6) 0x000020010DB8000000000202B3FF8329=2001:DB8: : :0202:B3FF:8329
206	8	UINT8	Reserved space	Unused space, reserved for future use. • Valid value: 0x00

Table 4-6: UMTS PPP protocol-specific Link Status buffer

Offset (bytes)	Length (bytes)	Type	Field	Description
0	1	UINT8	Coverage Code	Type of coverage currently available: • 0x00=No coverage • 0x01=Circuit-switched available (no packet-switched coverage) • 0x02=GPRS packet coverage available • 0x03=EDGE packet coverage available • 0x04=UMTS packet coverage available • 0x05=HSDPA packet coverage available • 0x06=HSUPA packet coverage available • 0x07=HSPA packet coverage available
1	1	UINT8	Network Name length	Length of the network name (next field) • Valid range: 0–20 • If 0, network name is not available.
2	40	Unicode	Network Name string	Name of network that modem is currently registered on • See previous field for string length (remaining characters are undefined). • UNICODE-format characters (up to twenty).
42	1	UINT8	Session State	Type of packet session in progress: • 0x04=NDIS Packet • 0xFF=No session
43	33	CHAR	Reason string	Reason for link status change • 0–32 characters plus terminating NULL character • Any remaining characters are undefined (field is always 33 bytes long)
76	1	UINT8	Active session link type (PDP context type)	Valid values: • 1=PPP session • 255=Unknown (no session established)
77	1	UINT8	Session (PDP) address length	Length of session address (next field) • 0 (Determine the IP addresses from normal PPP negotiations)
78	16	UINT8	Session (PDP) address	Unused—Address is determined from normal PPP negotiations.
94	1	UINT8	Valid flag for QoS	Indicates if Quality of Service values are used. • 0x00=Invalid (QoS values in this buffer will be set to 0) • 0x01=Valid
95	1	UINT8	QoS precedence	• Valid range: 0–3 • 0=When Valid flag for QoS is 0.

Table 4-6: UMTS PPP protocol-specific Link Status buffer (Continued)

Offset (bytes)	Length (bytes)	Type	Field	Description
96	1	UINT8	QoS delay	• Valid range: 0–4 • 0=When Valid flag for QoS is 0.
97	1	UINT8	QoS reliability	• Valid range: 0–5 • 0=When Valid flag for QoS is 0.
98	1	UINT8	QoS peak	• Valid range: 0–9 • 0=When Valid flag for QoS is 0.
99	1	UINT8	QoS mean	• Valid range: 0–31 • 0=When Valid flag for QoS is 0.
100	1	UINT8	QoS Traffic Class	• Valid range: 0–4 • 0=When Valid flag for QoS is 0.
101	1	UINT8	QoS Max Bitrate UL	• 0=Based on user's subscription (or when Valid flag for QoS is 0) • 1=16 Kbps • 2=32 Kbps • 3=64 Kbps • 4=128 Kbps • 5=256 Kbps • 6=384 Kbps • 7=1.8 Mbps • 8=3.6 Mbps • 9=7.2 Mbps • 10=14.4 Mbps • 11=2.048 Mbps • 12=5.76 Mbps • 13=11.5 Mbps • 14=16 Mbps • 15=21 Mbps • 16=28 Mbps • 17=42 Mbps • 18=84 Mbps
102	1	UINT8	QoS Max Bitrate DL	Same range of values as QoS Max Bitrate UL.
103	1	UINT8	QoS Guaranteed Bitrate UL	Same range of values as QoS Max Bitrate UL.
104	1	UINT8	QoS Guaranteed Bitrate DL	Same range of values as QoS Max Bitrate UL.

Table 4-6: UMTS PPP protocol-specific Link Status buffer (Continued)

Offset (bytes)	Length (bytes)	Type	Field	Description
105	1	UINT8	QoS Delivery Order	• Valid range: 0–2 • 0=When Valid flag for QoS is 0.
106	1	UINT8	QoS Delivery of Erroneous SDU	• Valid range: 0–3 • 0=When Valid flag for QoS is 0.
107	2	UINT16	QoS Maximum SDU size	• Valid range: 0–1520 • 0=When Valid flag for QoS is 0.
109	1	UINT8	QoS SDU error ratio	• Valid range: 0–7 • 0=When Valid flag for QoS is 0.
110	1	UINT8	QoS Residual Bit error ratio	• Valid range: 0–9 • 0=When Valid flag for QoS is 0.
111	2	UINT16	QoS Transfer Delay	• Valid range: 0–4000 • 0=When Valid flag for QoS is 0.
113	1	UINT8	QoS Traffic Handling priority	• Valid range: 0–3 • 0=When Valid flag for QoS is 0.
114	1	UINT8	QoS Source statistics descriptor	• Valid range: 0–1 • 0=When Valid flag for QoS is 0.
115	1	UINT8	QoS Signaling Indication	• Valid range: 0–1 • 0=When Valid flag for QoS is 0.
116	1	UINT8	PDP Initiated Type	• 0=Primary Mobile Initiated
117	1	UINT8	Network-supplied Primary DNS address length	The length of the primary DNS address (next field), which also indicates address type. • 0=No gateway address is present • 4=IPv4 address • 16=IPv6 address *Note: The network cannot supply a DNS address if the activated GPRS configuration already has a user-specified DNS address.*

Table 4-6: UMTS PPP protocol-specific Link Status buffer (Continued)

Offset (bytes)	Length (bytes)	Type	Field	Description
118	16	UINT8	Network-supplied Primary DNS address	The primary DNS address supplied by the network. • Length—See previous field. • Format—IPv4 or IPv6 address format. If IPv4, first four bytes contain the address and remaining bytes are set to 0. • Examples: (IPv4) 0x7F000201=127.0.2.1 (IPv6) 0x000020010DB8000000000202B3FF8329= 2001:DB8: : :0202:B3FF:8329
134	1	UINT8	Network-supplied Secondary DNS address length	The length of the secondary DNS address (next field), which also indicates address type. • 0=No gateway address is present • 4=IPv4 address • 16=IPv6 address *Note: The network cannot supply a DNS address if the activated GPRS configuration already has a user-specified DNS address.*
135	16	UINT8	Network-supplied Secondary DNS address	The secondary DNS address supplied by the network. The previous field indicates the length of the string. • Length—See previous field. • Format—IPv4 or IPv6 address format. If IPv4, first four bytes contain the address and remaining bytes are set to 0. • Examples: (IPv4) 0x7F000201=127.0.2.1 (IPv6) 0x000020010DB8000000000202B3FF8329= 2001:DB8: : :0202:B3FF:8329
151	73	CHAR	PPP Active User ID	The PPP active user ID currently being used by this packet session. • 0–72 characters plus terminating NULL character • Any remaining characters are undefined (field is always 73 bytes long)
224	17	CHAR	PPP Active User Password	The PPP active user password being used by this packet session. • 0–16 characters plus terminating NULL character • Any remaining characters are undefined (field is always 17 bytes long)

Table 4-6: UMTS PPP protocol-specific Link Status buffer (Continued)

Offset (bytes)	Length (bytes)	Type	Field	Description
241	1	UINT8	Connect Opcode	(For use if required by the host software driver) Indicates which action must be performed, when not using standard dial-up networking over the COM port. • 0=Do nothing • 1=Attach • 2=Authenticate • 3=Deactivate • 4=Restart
242	1	UINT8	COM port	When the modem supports multiple simultaneous PDP contexts, this value indicates which PDP context is active. • Valid range: 0–2
243	2	UINT16	Maximum D/L link speed	Maximum download link speed (kbps) • Valid range: 0–65535
245	2	UINT16	Maximum U/L link speed	Maximum upload link speed (kbps) • Valid range: 0–65535
247	1	UINT16	Network-supplied gateway address length	The length of the gateway address (next field), which also indicates address type. • 0=No gateway address is present • 4=IPv4 address • 16=IPv6 address
248	16	UINT8	Network-supplied gateway address	The gateway address supplied by the network. The previous field indicates the length of the string. • Length—See previous field • Format—IPv4 or IPv6 address format. If IPv4, first four bytes contain the address and remaining bytes are set to 0. • Examples: (IPv4) 0x7F000201=127.0.2.1 (IPv6) 0x000020010DB8000000000202B3FF8329= 2001:DB8: : :0202:B3FF:8329
264	8	UINT8	Reserved space	Unused space, reserved for future use. • Valid value: 0x00

Link Status Confirmation Indication

This indication is sent by the host in response to a Link Status Indication. It is sent when the host is ready to use the new IP address identified in the Link Status Indication message payload.

Table 4-7: HIP Link Status Confirmation Indication format

Short Message ID	**0x31**
Message-specific parameter	Context Index Number If multiple contexts are not supported, this value is 0 (PDP1).
Message payload	Unused

5: Firmware Download

Introduction

This chapter describes the process involved in downloading firmware updates to the MC8xxx embedded modem and provides a listing of the HIP messages that are used to accomplish this task. The following general update types are supported:

- Bootloaders and self-contained executables
  - QCOM—Qualcomm plus Sierra bootloaders
  - BOOT—Sierra bootloader
  - EXEC—Self-contained executable
- Applications and USB descriptors
  - APPL—Application
  - USBD—Downloadable USB descriptor

Firmware downloads can be sent from the host to the MC8xxx embedded modem using the following options:

- Via the Windows OneClick tool (an application that uses the HIP messaging protocol)
- Explicitly, via HIP messages

This document describes the second option (HIP messages).

Firmware download package header

Each downloadable firmware package includes a CWE header that identifies the included image type and other image-specific details. This header occupies the first 400 bytes of the package which is included in the payload of the Download Start Request on page 50.

Table 5-1: CWE header format

Offset	Length	Description
0 (0x0000)	256	Reserved
256 (0x0100)	4	32-bit CRC of region 0x0000–0x00FF
260 (0x0104)	4	Header revision number
264 (0x0108)	4	CRC validity indicator

268 (0x010C)	4	Image type, ASCII characters, not null-terminated
		• Valid types:
		  ◦ “QCOM”—Qualcomm plus Sierra bootloaders
		  ◦ “BOOT”—Sierra bootloader
		  ◦ “EXEC”—Self-contained executable
		  ◦ “APPL”—Application
		  ◦ “USBD”—Downloadable USB descriptor

Table 5-1: CWE header format (Continued)

Offset	Length	Description
272 (0x0110)	4	Product type, ASCII characters, not null-terminated
276 (0x0114)	4	Image size
280 (0x0118)	4	32-bit CRC of the image
284 (0x011C)	84	Version string, ASCII characters, null-terminated
368 (0x0170)	8	Release date string, ASCII characters, null-terminated
376 (0x0178)	4	Backward-compatibility field
380 (0x017C)	20	Reserved

When the host reads in the file, check the image type to determine which firmware download process to use:

- "QCOM", "BOOT", "EXEC"—uses Figure 5-1, Firmware download (bootloader/self-contained executable)—message flow, on page 45
- "APPL", "USBD"—uses Figure 5-2, Firmware download (application/USB descriptor)—message flow, on page 46

Downloading firmware updates

The process for downloading a firmware update to the modem includes the following *general* steps (for full details, see Download procedure (page 44)):

1. Make sure that the host device disables USB Advanced Power Management—the modem doesn't support USB suspend/remote wakeup during the download.
2. Back up user settings that are currently stored in non-volatile RAM.
3. Download the new firmware image to the modem—the existing firmware image may be erased during the download process, depending on the download type and size.
4. When the new image is downloaded successfully, the modem restores the user settings automatically.

Download procedure

The firmware download process (using appropriate HIP messages and CnS objects) is shown in Figure 5-1 on page 45 for bootloaders/self-contained executables, and in Figure 5-2 on page 46 for applications/USB descriptors. Refer to the steps following the figures for more details.

Note: The processes are identical until the HIP Download End Response (0x41).

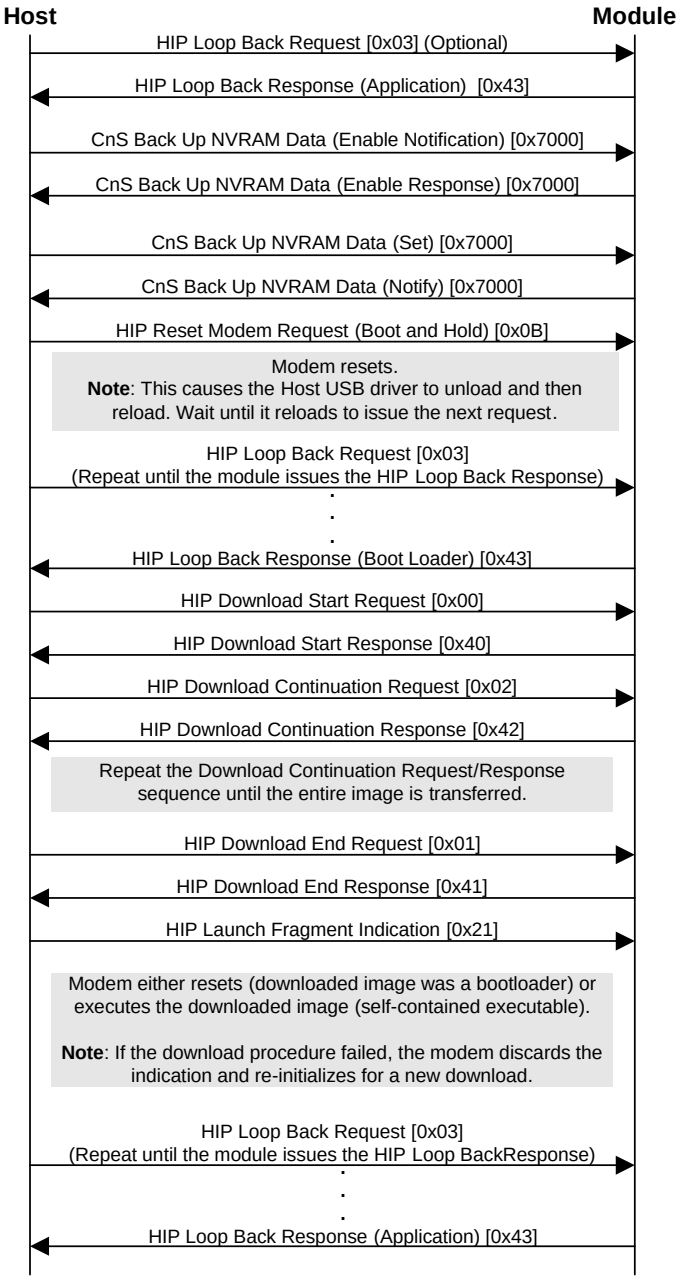

Figure 5-1: Firmware download (bootloader/self-contained executable)—message flow

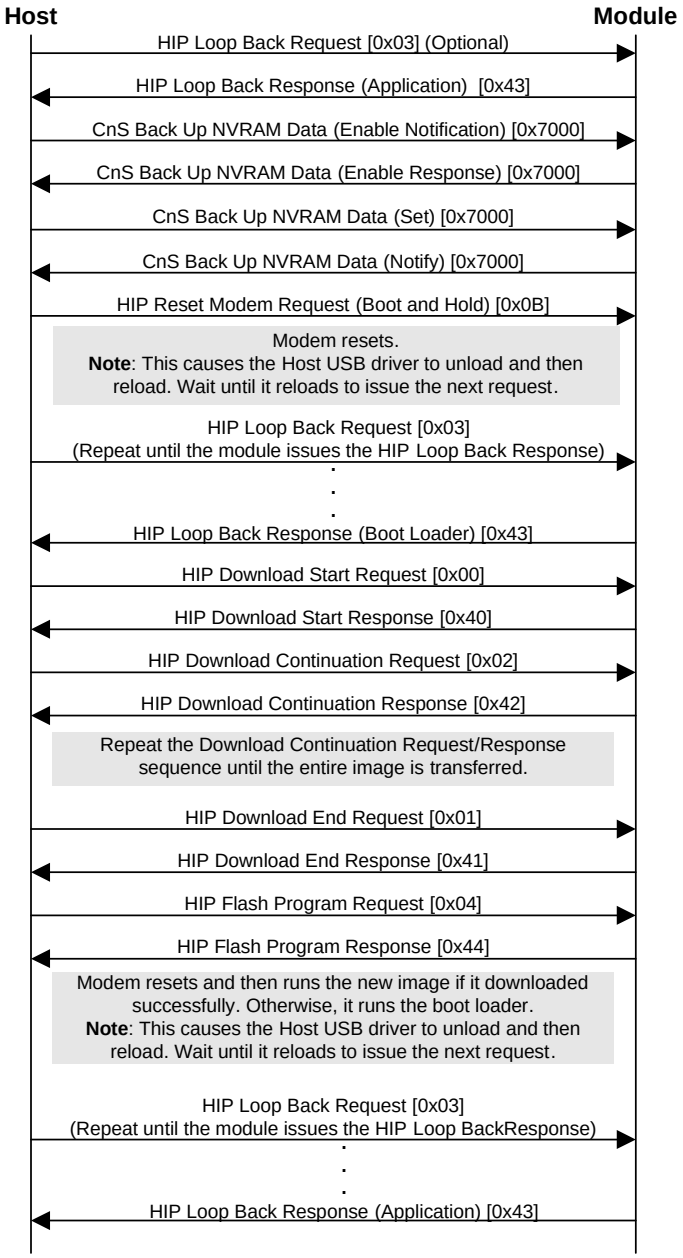

Figure 5-2: Firmware download (application/USB descriptor)—message flow

*Note: The modem **does** support USB suspend / remote wakeup under normal operating conditions.*

During the firmware update process, the modem is in 'boot and hold' mode—this mode does not support USB suspend/remote wakeup. The host device **must** disable USB advanced power management during the firmware update to avoid possible corruption of the image.

To download a firmware update to the modem (Figure 5-1 on page 45):

Note: Start the procedure at:
Step 1, if you don't know which mode the modem is running.
Step 3, if you know the modem is running the current firmware application image.
Step 10, if you know the modem is running the boot loader image.

1. Send a HIP Loop Back Request (49) (Message ID: 0x03) to determine whether the modem is running normally or running in the boot loader—the message does not require a payload.
2. If a Loop Back Request (49) was sent, the modem returns a HIP Loop Back Response (50) (Message ID: 0x43)—the response identifies the active firmware image.
   If the boot loader image is active go to Step 10.
3. Use the CnS object Back Up NVRAM Data (Deprecated) (93) to enable notification.
4. Receive the CnS object Back Up NVRAM Data (Deprecated) (93) response indicating that notification is now enabled.
5. Use the CnS object Back Up NVRAM Data (Deprecated) (93) to back up the modem's non-volatile RAM—this saves the user settings for automatic restoration after the firmware is successfully updated.
6. Receive the CnS object Back Up NVRAM Data (Deprecated) (93) notification when the operation finishes (if enabled).
7. Send a HIP Reset Modem Request (50) (Message ID: 0x0B) to tell the modem to 'boot and hold'—this causes the modem to reset, boot, and then pause without launching the current firmware application.

   The Host USB interface is stopped when the modem resets and then re-initialized after the modem boots, creating a serial channel between the modem and the host operating system.

   Wait until the USB interface has re-initialized before going to the next step.
8. Send a HIP Loop Back Request (49) (Message ID: 0x03) to test the connection—the message does not require a payload.
9. Receive a HIP Loop Back Response (50) (Message ID: 0x43) from the modem—the response identifies the active firmware image (this is the boot loader image).

Note: If a response is not received within 200 ms after issuing the Loop Back Request (49), send the request again. Repeat the step until the response is received.

Note: The host must calculate a running checksum for the downloaded image. The checksum is the sum of every byte (typecast to UINT8 format) in the firmware image.

10. Send a HIP Download Start Request (50) (Message ID: 0x00)—use *at least* the first 512 bytes of the firmware image as the message payload (you can send any amount from 512 to 2000 bytes of the image in this request).

    The modem validates the payload and prepares for the download procedure. This process may take a couple of minutes to complete.

Note: If the modem responds with an error at any time after a Download Start Request has been sent, the download sequence can be restarted at step 10.

11. Receive a HIP Download Start Response (51) (Message ID: 0x40) from the modem.

    This response indicates if the download process should continue.

12. Send a HIP Download Continuation Request (51) (Message ID: 0x02) to the modem.

    Include the next segment of the firmware image (up to 2000 bytes) in the message payload.

Note: The modem calculates a running checksum on the received segment—it is compared to the host's checksum when the download is finished.

13. Receive a HIP Download Continuation Response (51) (Message ID: 0x42) from the modem.

    This response indicates if the segment downloaded successfully.

14. Repeat steps 12–13 until the firmware image is completely transferred.
15. Send a HIP Download End Request (52) (Message ID: 0x01) to tell the modem that the download is finished.

    The modem performs a final validation of the downloaded image.

16. Receive a HIP Download End Response (52) (Message ID: 0x41) from the modem.

    This response indicates if the firmware update was downloaded successfully.

17. If the image type is a bootloader or self-contained executable, send a HIP Launch Fragment Indication (53) (Message ID: 0x21) to the modem.
    This tells the modem to launch the image (if it is a self-contained executable) or reboot (if the image is a bootloader).

    The modem does not send a response before launching an executable or rebooting.

Note: If the image is a bootloader, the Launch Fragment Indication causes the modem to reboot and then try to use the new firmware image (there may be a short delay the first time that the new image is used). If the new firmware loads correctly, the user settings are automatically copied back into NVRAM.
If the image is a self-contained executable, the indication causes the modem to launch the image, and then (depending on the executable's implementation) reboot.

During a reboot, the modem momentarily disconnects from the USB, causing it to be re-enumerated by the host. Wait for this to complete before continuing.

Go to Step 20.

18. If the image type is an application or USB descriptor, send a HIP Flash Program Request (52) (Message ID: 0x04) to the modem.
    This tells the modem to attempt to use the new firmware image, if there are no problems.
19. Receive a HIP Flash Program Response (53) (Message ID: 0x44) from the modem.

Note: The Flash Program Response causes the modem to reset, boot, and then try to use the new firmware image (there may be a short delay the first time that the new image is used). If the new firmware loads correctly, the user settings are automatically copied back into NVRAM.

The USB interface is stopped when the modem resets and then re-initialized after the modem boots, creating a serial channel between the modem and the host operating system.

Wait until the USB interface has re-initialized before going to the next step.

Note: If a response is not received within 200 ms after issuing the Loop Back Request (49), send the request again. Repeat the step until the response is received.

20. Send a HIP Loop Back Request (49) (Message ID: 0x03) to test if the application is running (download was successful) or the boot loader is running (a problem occurred during validation).

This causes the modem to return (echo) the message payload and identify the active firmware image.

21. Receive a HIP Loop Back Response (50) (Message ID: 0x43) from the modem.

This response echoes the message payload from the request and indicates which firmware image is active (this will be the application image unless there was a problem).

If the boot loader image is running, then the firmware download was unsuccessful. Repeat the download procedure.

Related HIP messages

Loop Back Request

This request is used to test hardware, host drivers and application software. The content included in the message payload is echoed back to the host in the Loop Back Response (50)—the host should wait until the response is received before continuing.

Table 5-2: HIP Loop Back Request format

Short Message ID	**0x03**
Message-specific parameter	Unused
Message payload	Optional • Any content, up to the maximum payload size

Loop Back Response

Sent by the modem in response to a Loop Back Request message. It returns the original message payload from the request, and indicates which firmware image is sending the response—the boot loader or application image.

Table 5-3: HIP Loop Back Response format

Short Message ID	**0x43**
Message-specific parameter	Firmware image that is sending the response • 0x00=Boot loader image • 0x01=Application image
Message payload	Optional • Any content, up to the maximum payload size

Reset Modem Request

This request signals the modem to 'boot and hold' prior to starting a firmware download. This causes the modem to reset and then pause without loading the current firmware application.

Table 5-4: HIP Reset Modem Request format

Short Message ID	**0x0B**
Message-specific parameter	• 01=Reset with Boot and Hold to prepare for a firmware download
Message payload	Unused

Download Start Request

This request signals the modem to prepare for a firmware download. This causes the modem to validate the file format, type, and so on, and initialize required resources.

The modem returns a Download Start Response message.

Table 5-5: HIP Download Start Request format

Short Message ID	**0x00**
Message-specific parameter	Unused
Message payload	First portion of the image file—this portion must be at least 512 bytes. See Table 3-1, HIP packet format, on page 23 for details on the maximum payload size.

Download Start Response

Sent by the modem in response to a Download Start Request message.

Table 5-6: HIP Download Start Response format

Short Message ID	**0x40**
Message-specific parameter	Return code • 0x00=Success, continue with the download • 0x01=Failure, stop the download
Message payload	Optional • If the Return code = Failure, the error reason may be returned in a null-terminated string. Example: “Invalid product ID”

Download Continuation Request

Send a portion of the firmware image to the modem.

The modem returns a Download Continuation Response message.

Table 5-7: HIP Download Continuation Request format

Short Message ID	**0x02**
Message-specific parameter	Unused
Message payload	The next portion of the firmware image (up to the maximum payload size). See Table 3-1, HIP packet format, on page 23 for details on the maximum payload size.

Download Continuation Response

Sent by the modem in response to a Download Continuation Request.

Table 5-8: HIP Download Continuation Response format

Short Message ID	**0x42**
Message-specific parameter	Return code • 0x00=Success • 0x01=Failure
Message payload	Optional • If Return code = Failure, the error reason may be returned in a null-terminated string. Example: “Program FLASH error”

Download End Request

Tell the modem that the download is complete.

The modem performs final validation of the downloaded image and returns a Download End Response message.

Table 5-9: HIP Download End Request format

Short Message ID	**0x01**
Message-specific parameter	Unused
Message payload	Downloaded firmware image checksum • 16-bit field • Calculated by the host as the sum of every byte (cast to UINT8 format) in the firmware image

Download End Response

Sent by the modem in response to a Download End Request.

The next message that the modem should receive is a Flash Program Request.

Table 5-10: HIP Download End Response format

Short Message ID	**0x41**
Message-specific parameter	Return code • 0x00=Success, complete program downloaded • 0x01=Failure, download process aborted
Message payload	Optional • If Return code = Failure, the error reason may be returned in a null-terminated string.

Flash Program Request

Tell the modem to try to use the new firmware image (application or USB descriptor).

The modem ensures that there are no other problems and then attempts to load the new firmware application. It returns a Flash Program Response message, indicating if the new firmware application is being used, and a reason if it is not.

Table 5-11: HIP Flash Program Request format

Short Message ID	**0x04**
Message-specific parameter	Unused
Message payload	Unused

Flash Program Response

Sent by the modem in response to a Flash Program Request.

The modem determines if the new firmware application can be used and then attempts to load it. If the firmware cannot be used, or fails, an appropriate error message is sent.

Table 5-12: HIP Flash Program Response format

Short Message ID	**0x44**
Message-specific parameter	Return code • 0x00=Success • 0x01=Failure, firmware has not been properly updated
Message payload	Optional • If Return code = Failure, the error reason is returned in a null-terminated string.

Launch Fragment Indication

Tells the modem to take action on the downloaded image. The message header is checked to identify the image type—the modem should execute a self-contained executable or reboot if the image is a bootloader.

The modem determines if the new firmware image downloaded successfully. If it failed, the indication is ignored and the modem re-initializes for a new download.

Table 5-13: HIP Launch Fragment Indication format

Short Message ID	**0x21**
Message-specific parameter	Reserved • One byte parameter, reserved for future use
Message payload	Unused

6: Protocol Stack Logging

Introduction

This chapter describes the HIP messages that can be used to implement protocol stack logging for the MC8xxx embedded modem.

Using the protocol stack

Two HIP indications (asynchronous messages initiated by the modem or host) are used to tunnel Qualcomm DM (Diagnostic Monitor) data between the host and modem through the protocol stack:

- Modem logging indication—Host to modem
- Host logging indication—Modem to host

Related HIP messages

Modem logging indication

This indication is used by the host to tunnel Qualcomm DM data to the modem through the protocol stack.

Table 6-1: HIP Modem Logging indication format

Short Message ID	**0x2D**
Message-specific parameter	Unused
Message payload	Variable-length binary data (raw Qualcomm DM packet)

Host logging indication

This indication is used by the modem to tunnel Qualcomm DM data to the host through the protocol stack.

Table 6-2: HIP Host Logging indication format

Short Message ID	**0x6D**
Message-specific parameter	Unused
Message payload	Variable-length binary data (raw Qualcomm DM packet)

7: CnS Basics

CnS overview

The CnS (Control and Status) language is a set of objects (and parameters) that allow an application to:

- Query the device for status
- Set parameters and configuration of the device
- Control the traffic of event notifications from the device
- Receive event notifications

CnS objects are messages sent between the modem and the host. Each object has a unique object ID. For example, the unique ID **0x0000** identifies the Heartbeat object. This object is used by the modem to tell the host that it is functioning.

Note: If both CnS messages and non-CnS interfaces are used at the same time in the same area (for example, working with SMS), either (or both) may report erroneous information.

CnS communication types

CnS communications fall into one of two general types:

- Request-and-response—the host issues a command or request (poll), and the modem replies (response)
- Notification—the modem forwards an information report without a specific request

Request and response

This allows an application to poll the modem and issue commands to control the modem. A request is sent to the modem and a response is returned by the modem to acknowledge the request. Depending on the context of the message, the response may contain data, such as status information, or it may simply be an acknowledgement that the requested command was received.

Some requests may yield an error response. If the response has set bit 7 of the Operation Type (described later), the operation failed. The parameter may include an explanation for the failure.

Responses to Set commands generally return the value of the setting. They are issued after execution. Any exceptions are indicated in the notes within each command's description.

Notification

Notifications consist of unsolicited, asynchronous, status updates that are forwarded to the host from the modem based on events. Different types of information can be automatically forwarded to the host. The criteria for an unsolicited update depends on the specific object being forwarded.

Note: Upon power-up, notifications are always disabled.
Forwarding notifications must be explicitly enabled using the request and response mechanism.

There are three general categories of notifications:

- Periodic notifications—Notifications that are sent at regular intervals (for example, Heartbeat on page 132)
- Update notifications—Notifications that are sent when they are first enabled, and when triggered by a change in state or other condition (for example, Report Network Status on page 141)
- Event notifications—Notifications that are sent only when they are triggered by a certain condition (for example, Select PLMN on page 147). No initial notification is sent.

*Note: A modem reset or power cycle disables notifications for **all** object types. Desired notifications have to be re-enabled.*

Notifications stop if the host sends the Forward Stop Request for the particular object.

Communication sequences

Communication sequences include:

- Request and response
- Request (or notification enable/disable) and error
- Notification enable/response, notification packets, notification disable/response

Some Request and response sequences initiate subsequent notification sequences, and some notification sequences require one or more subsequent Request and response sequences.

Note: The Get Request is used to request information and never has a payload; the Get Response delivers the information.
The Set Request can be used to request or set information (depends on the Object ID) and may or may not have a payload. The Set Response may carry information or may just be an acknowledgement.

Table 7-1: Request/Response sequence

Step	Host to Modem	Modem to Host
1	• Get, *or* • Set	
2		• Get Response, *or* • Set Response

Table 7-2: Request/Error sequence

Step	Host to Modem	Modem to Host
1	• Get, *or* • Set, *or* • Notification Enable, *or* • Notification Disable	
2		• Error Response

Table 7-3: Notification sequence

Step	Host to Modem	Modem to Host
1	• Notification Enable	
2		• Enable Response
3		• Notification
4		• Notification
...		…
n		• Notification
n+1	• Notification Disable	
n+2		• Disable Response

CnS message format

All CnS messages are composed of the following elements:

Table 7-4: CnS message format

Byte offset	Content
0–1	Object ID
2	Operation Type
3	Reserved
4–7	Application ID

Table 7-4: CnS message format (Continued)

Byte offset	Content
8–9	Length of payload (0–246)
[10–255]	Parameter (if needed for the object)

The maximum length of a CnS message is 256 bytes. The minimum length is 10.

Object ID This is the object that the message relates to. Object IDs are expressed as hex numbers.

Operation Type This is the type of the message. It indicates if the message is a Get (request), Set (request/command), response (to a Get), response (to a Set), error, or notification. The supported types are:

Table 7-5: Supported CnS operation types

Code	Operation Type	Direction
0x01	Get Request	Host ⇨ Modem
0x02	Get Response (with data)	Host ⇦ Modem
0x03	Set Request/Command	Host ⇨ Modem
0x04	Set Response	Host ⇦ Modem
0x05	Notification Enable	Host ⇨ Modem
0x06	Enable Response	Host ⇦ Modem
0x07	Notification	Host ⇦ Modem
0x08	Notification Disable	Host ⇨ Modem
0x09	Acknowledge to Notification Disable	Host ⇦ Modem
0x80 (0x82, 0x84, 0x86, 0x89)	Error Response If a host>modem operation is unsupported or misconfigured, the expected response is 'OR'd with 0x80. For example, If a Notification Enable (0x05) fails, then 0x86 is returned (0x05 OR 0x80).	Host ⇦ Modem

Application ID This is intended to allow the application to assign a number (perhaps indicating the process or thread that originated a request). The value in this field appears in the subsequent response from the modem. For notifications that are not in response to a specific request, the field contains zeroes (the modem does not retain the original enabler's ID).

Length Only the length of an object parameter is needed. This field indicates the length of the object's request/response/notification parameter field. If the field is zero then there is no parameter field included in the CnS message.

Parameter Any data associated with the object is carried here. Error responses may use this to convey the reason for a failure.

HIP encapsulation

All CnS messages are encapsulated in a HIP packet to be sent over the physical link. Thus the complete packet sent to the modem, including the HIP header, is in the following form:

Table 7-6: HIP packet with encapsulated CnS message

Byte offset	Content
HIP header information	
0	Frame character (0x7E)
1–2	Length of CnS format packet including parameter length (10–256)
3	Message ID: 0x2B—for host to modem 0x6B—for modem to host
4	Parameter 0x00—no HIP parameter for CnS messages
CnS message	
5–6	CnS Object ID
7	Operation Type
8	Reserved (0x00)
9–12	Application ID
13	Reserved (0x00)
14	Length of CnS parameter (0–246)
[15–260]	CnS Parameter (if needed for the object)
End of HIP header	
	Frame character (0x7E)

This reference covers only the material from offset 4 onward—the actual CnS message content. The HIP header is always constructed from the length of the CnS message, the ID type (either host to modem or modem to host), and a zero for the HIP parameter.

Reference conventions

Reference tables

Summary tables

Each reference chapter begins with a summary table, identifying the objects in the chapter and the message types they support: Set (S), Get (G), and Notification (N).

Table 7-7: Sample summary table

Object	ID	Description	S	G	N
Return Firmware Version	0x0001	Returns the version of modem firmware.	✗	✔	✗
Disable Modem	0x1011	Enable or disable the modem	✔	✔	✔

Object description tables

Note: Certain requests and responses are implied in these tables.
***Set Response**: only shown if the response includes returned data.*
***Enable / Disable Notification and associated responses**: Always implied if a Notification is expected*
***Error Response**: Always implied*

Individual object description sections follow the summary table. An object description table presents the syntax, showing the object value, supported operation types, and parameters.

The “Operations” row shows the supported operation types (for the object (✔ = supported; ✗ = unsupported).

- Get—Refers to the Get Request/Response pair.
- Set—Refers to the Set Request/Response pair.
- Notify—Refers to Notify, the Notify Enable/Response pair, and the Notify Disable/Response pair.

Each type is listed in the Operations column with their associated parameters.

The “Parameters” section provides columns for the total length of the parameter block, the offset in the block for each field of the parameter, the data type of the field, and the description of each field. Where an object has different formats based on the Object version number, the Object Ver column indicates in which versions each parameter appears.

Table 7-8: Object - Sample object name

Object ID	**0xFFFF**		Sample object name		
Operations	✗ Set ✔ Get ✔ Notify				
	Parameters (fields)				
	Len	**Offset**	**Type**	**Description**	Object Ver.
01 (Get)	0			None	n/a
02 (Get Response) **07 (Notify)**	2	0	UINT16	Object version number • **0x0002**—Revision 2 Length = 45 • **0x0003**—Revision 3 Length = 47	1–2

In the sample above:

- A host Get Request (0x01 message type) does not have parameters, so the CnS parameter length is 0.
- When the modem replies to the Get Request, it provides a parameter block with one field. The length of the parameter block is 2 bytes. Each field is listed in order, with the offset within the parameter block.
- The notification parameters appear in the **0x07** (Notification) message type.

*Note: Some 'update' notifications (such as Report Network Status) can issue a notification immediately upon being enabled. In such cases, the Notification Response may arrive **before** the Enable Response.*

Unless noted otherwise, the parameters for the notification type **0x07** are the same as those for the response (**0x02**) to a Get Request. No parameters are associated with the other notification types: **0x05**/**0x06** (enable request/response), and **0x08**/**0x09** (disable request/response).

Object version numbers

Some CnS objects include an object revision number parameter. The object revision number is used to interpret the format of the CnS object—if the field layout of the object is changed (fields added, changed, or deleted), the object number also changes. This allows CnS objects to evolve, while making sure that you can correctly parse the object format based on the object number.

8: Supported CnS Messages

Introduction

This chapter provides a cross-reference listing of all the CnS messages supported by the MC8xxx embedded modem.

Message objects are sorted object ID number.

Table 8-1: Numeric Object Listing

Object	Message	Chapter	Page
0x0000	Heartbeat	Status Messages	132
0x0001	Return Firmware Version	Device Characteristics	71
0x0002	Return Firmware Build Date	Device Characteristics	71
0x0003	Return Hardware Version	Device Characteristics	72
0x0004	Return Boot Version	Device Characteristics	72
0x0007	Manage Radio Power	Device Characteristics	73
0x0008	Return Radio Temperature	Device Characteristics	74
0x0009	Return Radio Voltage	Device Characteristics	75
0x000A	Get Modem Type	Device Characteristics	75
0x000B	Back Up NVRAM Data	Device Characteristics	77
0x000C	Return Supported Object Versions	Device Characteristics	79
0x0010	Reset Modem	Device Characteristics	81
0x0016	Report Current Band	Data Services	138
0x0017	Set Current Band	Data Services	139
0x001B	Return Modem Date and Time	Device Characteristics	81
0x0023	Return PRI Information	Device Characteristics	83
0x0024	Return Flash Image Information	Device Characteristics	84
0x0F02	Initiate location fix	Location Based Services	231
0x0F04	Start tracking session	Location Based Services	232
0x0F05	End location fix session	Location Based Services	233
0x0F0A	Report location fix results	Location Based Services	234
0x0F0B	Report location fix completed	Location Based Services	236
0x0F0C	Report location fix error	Location Based Services	236
0x0F0D	Report location fix begun	Location Based Services	238

Table 8-1: Numeric Object Listing (Continued)

Object	Message	Chapter	Page
0x0F0E	Report position determination failure	Location Based Services	238
0x0F13	Report IP address used for TCP/IP	Location Based Services	239
0x0F16	Report Port ID	Location Based Services	239
0x0F1A	Start Keep Warm process	Location Based Services	240
0x0F1B	Report Keep Warm process begun	Location Based Services	241
0x0F1C	Report Keep Warm process status	Location Based Services	241
0x0F1D	Report Keep Warm process complete	Location Based Services	242
0x0F1E	Stop Keep Warm process	Location Based Services	242
0x0F1F	Report Satellite Information	Location Based Services	242
0x0F20	Clear GPS Assistance Data	Location Based Services	243
0x0F21	Manage location fix settings	Location Based Services	244
0x0F22	Report network-initiated location fix request	Location Based Services	246
0x0F24	Report Auto Track feature state	Location Based Services	250
0x1000	Report Network Status	Data Services	141
0x1001	Report Radio Information	Data Services	143
0x1003	Return Service Provider Name	Data Services	144
0x1004	Return IMSI	Device Characteristics	87
0x1005	Return Hardware Build Date	Device Characteristics	87
0x1006	Report Available Service Details	Data Services	145
0x100B	Report Current Band (Deprecated)	Data Services	146
0x100F	Select PLMN	Data Services	147
0x1011	Disable Modem	Device Characteristics	88
0x1016	Return Factory Serial Number	Device Characteristics	89
0x1017	Return Test History	Device Characteristics	90
0x1019	Send MEP Unlock Code	SIM Status and Locking	103
0x101D	Return FPGA Version	Device Characteristics	90
0x1020	Report SMS Received Messages Status	SMS Messages	208
0x1021	Get Unread SMS Message	SMS Messages	209
0x1022	Report SMS Availability	SMS Messages	211
0x1023	Get Read SMS Message	SMS Messages	211
0x1027	Return SMS Outgoing Message Number	SMS Messages	212

Table 8-1: Numeric Object Listing (Continued)

Object	Message	Chapter	Page
0X1028	Report Outgoing SMS Message Status	SMS Messages	213
0x1029	Delete Mobile-terminated SMS Message	SMS Messages	213
0x102A	Report SIM SMS Message Storage Status	SMS Messages	214
0x102B	Return SIM SMS Configuration Details	SMS Messages	214
0x1037	Report call byte counters (Rx/Tx)	Data Services	149
0x1038	Reset call byte counters	Data Services	150
0x103C	Report Manual PLMN Selection Readiness	Data Services	150
0x103E	Verify CHV Code	SIM Status and Locking	104
0x103F	Enable / Disable CHV1 Verification	SIM Status and Locking	106
0x1040	Change CHV Codes	SIM Status and Locking	107
0x1041	Return SIM Status	SIM Status and Locking	108
0x1042	Return Available PLMN List	Data Services	151
0x1043	Report PLMN Registration Information	Data Services	154
0x1044	Delete ‘last’ numbers	Phonebook Management	174
0x1045	Read phone number	Phonebook Management	175
0x1046	Add phone number	Phonebook Management	180
0x1047	Edit Phone Numbers	Phonebook Management	182
0x1048	Return Phonebook Services Status	Phonebook Management	184
0x1049	Return Phonebook Availability	Phonebook Management	184
0x104A	Return Phonebook Size Details	Phonebook Management	186
0x1063	Return Modem Model	Device Characteristics	91
0x1067	Return IMEI	Device Characteristics	92
0x106A	Report Roaming Icon Display Method	Data Services	155
0x106B	Change Current Band (Deprecated)	Data Services	156
0x1071	Indicate Host Software State	Status Messages	132
0x1072	Return Emergency Numbers	Phonebook Management	188
0x1079	Enable/Disable FDN Mode	Phonebook Management	188
0x107A	Delete Phone Number	Phonebook Management	189
0x107B	CHV2 Status Kick	SIM Status and Locking	111
0x1083	Indicate Host Software Launched	SMS Messages	133
0x1085	Run SIM Authentication Algorithm	SIM Status and Locking	111

Table 8-1: Numeric Object Listing (Continued)

Object	Message	Chapter	Page
0x1087	Copy Mobile-originated SMS Message to SIM	SMS Messages	220
0x1088	Send Mobile-originated SMS Message To Network	SMS Messages	221
0x7000	Back Up NVRAM Data (Deprecated)	Device Characteristics	93
0x7001	Return Profile Summary	Profile Management	114
0x7002	Read Profile	Profile Management	115
0x7003	Write Profile	Profile Management	128
0x7004	Manage Packet Session	Data Services	157
0x7005	Return Packet Session Status	Data Services	164
0x7006	Return IP Address	Data Services	164
0x7007	Return SIM's ICC ID	SIM Status and Locking	112
0x700A	Report System and Network Status	Data Services	165
0x700B	Return RSCP and Ec/Io Measurements	Device Characteristics	94
0x700C	Return GSM Cell Information	Device Characteristics	95
0x700D	Manage SMS Status Reports	SMS Messages	222
0x700E	Manage Profile Username and Password	Profile Management	128
0x700F	Set RAT mode	Data Services	168
0x7012	Manage call byte counters (Rx/Tx)	Data Services	168

9: Device Characteristics

Introduction

This chapter describes device-related CnS objects that:

- List the device type, serial number, and unique IDs assigned to the modem and SIM
- List version numbers of the hardware, firmware, and software components, and supported CnS objects
- Identify available features of the modem
- Back up the modem's non-volatile memory for firmware upgrades

Device messages summary

Table 9-1: Device Messages

Object	ID	Description	S	G	N
Return Firmware Version (page 71)	0x0001	Returns the version of modem firmware	✗	✔	✗
Return Firmware Build Date (page 71)	0x0002	Returns the build date of the modem firmware	✗	✔	✗
Return Hardware Version (page 72)	0x0003	Returns the hardware version of the modem	✗	✔	✗
Return Boot Version (page 72)	0x0004	Returns the version of the modem loader firmware	✗	✔	✗
Manage Radio Power (page 73)	0x0007	Returns power information, initiates power changes, and receives power-down requests	✔	✔	✔
Return Radio Temperature (page 74)	0x0008	Returns the current temperature state of the modem	✗	✔	✔
Return Radio Voltage (page 75)	0x0009	Returns the current supply voltage state of the modem	✗	✔	✔
Get Modem Type (page 75)	0x000A	Returns the modem type and supported technology type	✗	✔	✗
Back Up NVRAM Data (page 77)	0x000B	Back up non-volatile RAM	✔	✗	✔
Return Supported Object Versions (page 79)	0x000C	Returns the version of the CnS interface supported in the device	✗	✔	✗
Reset Modem (page 81)	0x0010	Shuts down and resets the modem	✔	✗	✗
Return Modem Date and Time (page 81)	0x001B	Returns the date and time from the modem	✗	✔	✔

Table 9-1: Device Messages (Continued)

Object	ID	Description	S	G	N
Return PRI Information (page 83)	0x0023	Returns the device's current PRI information	✗	✔	✗
Return Flash Image Information (page 84)	0x0024	Returns versioning information for any CWE images in the device's flash memory	✔	✗	✗
Return IMSI (page 87)	0x1004	Returns the IMSI	✗	✔	✗
Return Hardware Build Date (page 87)	0x1005	Returns the hardware build date	✗	✔	✗
Disable Modem (page 88)	0x1011	Enable or disable the modem	✔	✔	✔
Return Factory Serial Number (page 89)	0x1016	Returns the factory serial number	✗	✔	✗
Return Test History (page 90)	0x1017	Returns test history	✗	✔	✗
Return FPGA Version (page 90)	0x101D	Returns the FPGA version	✗	✔	✗
Return Modem Model (page 91)	0x1063	Returns the model of the modem	✗	✔	✗
Return IMEI (page 92)	0x1067	Returns the IMEI	✗	✔	✗
Back Up NVRAM Data (Deprecated) (page 93)	0x7000	Back up non-volatile RAM	✔	✗	✔
Return RSCP and Ec/Io Measurements (page 94)	0x700B	Returns RSCP and ec/io measurement when in WCDMA mode	✗	✔	✔
Return GSM Cell Information (page 95)	0x700C	Returns 2G cell information, including neighbor cell measurements	✗	✔	✔

Device messages reference

Return Firmware Version

Note: Return Flash Image Information (page 84) should be used instead of this object.

Returns the version of the modem firmware application:

- On request, using Get (01)

The modem firmware includes both a bootstrap/firmware loader and an application. This version string returns the application version number.

See also Return Boot Version on page 72.

Table 9-2: Object—Return Firmware Version

Object ID	0x0001		Return Firmware Version	
Operations	✗ Set ✔ Get ✗ Notify			
	Parameters (fields)			
	Len	**Offset**	**Type**	**Description**
01 (Get)	0			None
02 (Get Response)	84	0	CHAR[84]	Application version information • 84 characters, not null-terminated • Left justified, blank filled • Format: *Rx-y-z* (x, y, and z are 1–2 digit numbers from 0–99) • Trailing space reserved for future use

Return Firmware Build Date

Note: Return Flash Image Information (page 84) should be used instead of this object.

Returns the date on which the modem's firmware application was built:

- On request, using Get (01)

Table 9-3: Object—Return Firmware Build Date

Object ID	0x0002		Return Firmware Build Date	
Operations	✗ Set ✔ Get ✗ Notify			
	Parameters (fields)			
	Len	**Offset**	**Type**	**Description**
01 (Get)	0			None
02 (Get Response)	8	0	CHAR[8]	Application build date • Format—MM/DD/YY

Return Hardware Version

Note: Return Flash Image Information (page 84) should be used instead of this object.

Returns the version of the modem hardware:

- On request, using Get (01).

Table 9-4: Object—Return Hardware Version

Object ID	0x0003		Return Hardware Version	
Operations	✗ Set ✔ Get ✗ Notify			
	Parameters (fields)			
	Len	**Offset**	**Type**	**Description**
01 (Get)	0			None
02 (Get Response)	2	0	UINT16	Hardware version number • Min—0, Max—65535

Return Boot Version

Note: Return Flash Image Information (page 84) should be used instead of this object.

Returns the version of the modem firmware bootstrap/loader:

- On request, using Get (01)

The modem firmware includes both a bootstrap/firmware loader and an application. This version string returns the bootstrap/loader version number.

See also Return Firmware Version on page 71.

Table 9-5: Object—Return Boot Version

Object ID	0x0004		Return Boot Version	
Operations	✗ Set ✔ Get ✗ Notify			
	Parameters (fields)			
	Len	**Offset**	**Type**	**Description**
01 (Get)	0			None
02 (Get Response)	84	0	CHAR[84]	Bootstrap/loader version information • 84 characters, not null-terminated • Left justified, blank filled • Version number appears at the beginning of the string in the format *Rx-y-z* (x, y, and z are 1–2 digit numbers from 0–99) • Trailing space reserved for future use

Manage Radio Power

Returns power information, initiates power changes, and receives requests from the modem to power down.

Returns power information:

- On request, using Get (01)

Initiates power changes:

- On request, using Set (03)

Receives power-down requests:

- When triggered, if Notify (07) is enabled *and* the modem requests a shutdown due to critical temperature or supply voltage levels. (This is an update-type notification.)

Table 9-6: Object—Manage Radio Power

Object ID	0x0007		Manage Radio Power	
Operations	✔ Set ✔ Get ✔ Notify			
	Parameters (fields)			
	Len	**Offset**	**Type**	**Description**
01 (Get)	0			None
02 (Get Response) **04 (Set Response)**	20	0	UINT16	Object version number • 0x0001=Revision 1
		2	UINT16	Power mode • 0x00=Low power mode • 0x01=Online mode
		4	UINT16	Low power mode reasons (bitmask) • 0x0001=W_DISABLE is asserted • 0x0002=User request • 0x0004=Out of operational temperature range • 0x0008=Out of operational supply voltage range
		6	UINT8[14]	Reserved for future use
03 (Set)	20	0	UINT16	Object version number • 0x0001=Revision 1
		2	UINT16	Requested power mode • 0x00=Low power mode • 0x01=Online mode
		4	UINT8[16]	Reserved for future use

Table 9-6: Object—Manage Radio Power (Continued)

07 (Notify)	20	0	UINT16	Object version number • 0x0001=Revision 1
		2	UINT16	Power mode • 0x00=Low power mode • 0x01=Online mode
		4	UINT16	Low power mode reasons (bitmask) • 0x0001=W_DISABLE is asserted • 0x0002=User request • 0x0004=Out of operational temperature range • 0x0008=Out of operational supply voltage range
		6	UINT8[14]	Reserved for future use

Return Radio Temperature

Returns the current temperature state of the modem:

- On request, using Get (01)
- When triggered, if Notify (07) is enabled *and* the temperature changes noticeably. (Generally, a 2°C change will trigger a notification.)

Table 9-7: Object—Return Radio Temperature

Object ID	0x0008		Return Radio Temperature	
Operations	✗ Set ✔ Get ✔ Notify			
	Parameters (fields)			
	Len	**Offset**	**Type**	**Description**
01 (Get)	0			None
02 (Get Response) **07 (Notify)**	20	0	UINT16	Object version • 0x0001=Revision 1
		2	UINT16	Temperature state • 0x0000=Normal state • 0x0001=High temperature warning state • 0x0002=High temperature critical state • 0x0003=Low temperature critical state
		4	INT16	Current temperature (°C)
		6	UINT8[14]	Reserved for future use

Return Radio Voltage

Returns the current supply voltage state of the modem:

- On request, using Get (01)
- When triggered, if Notify (07) is enabled *and* the supply voltage state changes

Table 9-8: Object—Return Radio Voltage

Object ID	0x0009		Return Radio Voltage	
Operations	✗ Set ✔ Get ✔ Notify			
	Parameters (fields)			
	Len	**Offset**	**Type**	**Description**
01 (Get)	0			None
02 (Get Response) **07 (Notify)**	20	0	UINT16	Object version • 0x0001=Revision 1
		2	UINT16	Supply voltage state • 0x0000=Normal state • 0x0001=Low battery warning state • 0x0002=Low battery critical state • 0x0003=High battery critical state
		4	UINT16	Current supply voltage (mV)
		6	UINT8[14]	Reserved for future use

Get Modem Type

Returns the modem type (model) and the wireless technology supported by the modem:

- On request, using Get (01)

Note: The HIP message—Device Information Request (page 29)—should be used instead of this object. (Newer Sierra Wireless modems return a generic product type (0xFE) with this object.)

Table 9-9: Object—Get Modem Type

Object ID	0x000A		Get Modem Type		
Operations	✗ Set ✔ Get ✗ Notify				
	Parameters (fields)				
	Len	**Offset**	**Type**	**Description**	**Object Ver**
01 (Get)	0			None	

Table 9-9: Object — Get Modem Type (Continued)

02 (Get Response)	See Object version number (Offset = 0) description for object length				
		0	UINT16	Object version • 0x0001=Revision 1 Length—24 • 0x0002=Revision 2 Length—40	1–2
		2	UINT16	Modem type (Values not listed are all reserved) • 0x0A=AC850 • 0x0B=AC860 • 0x0C=MC8755 • 0x0D=MC8765 • 0x13=AC875 • 0x14=MC8775 • 0x15=MC8775V • 0x18=AC880 • 0x19=AC881 • 0x1A=MC8780 • 0x1B=MC8781 • 0x1E=AC880E • 0x1F=AC881E • 0x24=AC595U[a] • 0x25=MC5727[a] • 0x26=C597[a] • 0x29=AC885E • 0x2A=C885 • 0x2B=MC8790 • 0x2C=MC8790V • 0x2D=AC501 • 0x2E=C888 • 0x2F=C889 • 0x32=MC8791V • 0x33=MC8792V • 0x39=AC402 • 0x3A=MC5727V[a] • 0x3B=MC5728[a] • 0x3C=MC5728V[a] • 0x45=MC8700 • 0x50=MC8795V • 0xFE=All other products • 0xFF=Invalid device	1–2

Table 9-9: Object—Get Modem Type (Continued)

		4	UINT32	Technology supported • 0x00000001=CDMA/EVDO • 0x00000002=GSM/UMTS	1–2
		8	UINT8[16]	Reserved for future use	1–2
		24	UINT8	Product string length (number of non-null ASCII characters in Product string field) • Min—0; Max—15	2
		25	UINT8[15]	Product string—Descriptive name of the modem type • ASCII format • Null padded	2

a. CDMA modem - Included for reference only

Back Up NVRAM Data

Note: This command replaces the following command: Back Up NVRAM Data (Deprecated) (page 93).

Provides the ability to back up user settings stored in the modem's non-volatile (NV) memory prior to performing a firmware download:

- On request, using Set (03) to initiate the backup. A Notify (07) occurs when the operation finishes (if notification is enabled).

The backup and restore operations may each take up to three minutes to complete. While performing a restore operation, no other CnS requests should be made.

Note: This message needs to be sent before performing a firmware download; if it is not sent, then the user settings are lost.

Restoring the NV user settings

This allows the NV user settings to be saved before downloading a firmware update. The firmware update automatically restores the NV user settings when it is finished.

Table 9-10: Object—Back Up NVRAM Data

Object ID	0x000B		Back Up NVRAM Data
Operations	✔ Set ✗ Get ✔ Notify		
	Parameters (fields)		
	Len	**Offset**	**Type**	**Description**
03 (Set)	20	0	UINT16	Object version number • 0x0001=Revision 1
		2	UINT8	Requested operation: • 0x00–0x01=Reserved • 0x02=NV backup operation • 0x03=NV restore operation • 0x04–0xFF=Reserved
		3	UINT8	Category of requested operation: • 0x00–0x01=Reserved • 0x02=User Settings • 0x03–0xFF=Reserved
		4	UINT8[16]	Reserved
04 (Set Response)	2	0	UINT16	Object version number • 0x0001=Revision 1
07 (Notify)	40	0	UINT16	Object version number • 0x0001=Revision 1
		2	UINT8	Operation status: • 0x00=Successful • 0x01=Unsuccessful
		3	UINT8	Operation performed: • 0x00–0x01=Reserved • 0x02=NV backup operation • 0x03=NV restore operation • 0x04–0xFF=Reserved
		4	UINT8	Category of performed operation: • 0x00–0x01=Reserved • 0x02=User Settings • 0x03–0xFF=Reserved
		5	UINT32	Number of items stored during the backup
		9	UINT32	Number of items skipped during the backup
		13	UINT32	Number of items with zero (0) length
		17	UINT8[23]	Reserved

Return Supported Object Versions

Returns the supported version ranges of the CnS object types recognized by the modem firmware:

- On request, using Get (01)

Table 9-11: Object—Return Supported Object Versions

Object ID	0x000C		Return Supported Object Versions	
Operations	✗ Set ✔ Get ✗ Notify			
	Parameters (fields)			
	Len	**Offset**	**Type**	**Description**
01 (Get)	0			None
02 (Get Response)	245	0	UINT16	Object version number • 0x0001=Revision 1
		2	UINT8	Number of entries in Supported Object IDs list • Maximum number of entries—35
		3	Variable length list	Supported Object IDs list and revision ranges (1–6 entries, 10 bytes/ObjectID) OIDlength = 10 bytes * NumObjects
		0	UINT8	Object ID identifier • 0x00=CWE (0x0000–0x0EFF) • 0x02=GPRS (0x1000–0x1FFF) • 0x05=UMTS (0x7000–0x70FF) • 0x06=CWE DM (0x0F80–0xFFF)
		1	UINT16	Reverse object revision number—the oldest supported version of the indicated object ID type
		3	UINT16	Forward object revision number—the latest (newest) supported version of the indicated ID type.
		5	UINT8[5]	Reserved for future use
		3+ OID length *	UINT8[...]	Remaining unused space (complete record length = 245 bytes).
		*OIDlength—see calculation in Supported Object IDs list description		

CnS object sets

All supported CnS objects belong to one of the following sets:

- CWE (Core Wireless Engine) objects
  These objects are common to all Sierra Wireless modems.
- GPRS objects
  These objects are specific to GSM-based products (including UMTS modems).
- UMTS objects
  These objects are specific to UMTS modems.

Forward and Reverse version numbers

Return Supported Object Versions returns forward and reverse object versions for CWE, GPRS, and UMTS objects. The reverse object version is the oldest supported version; the forward object revision is the latest (newest) supported version.

Example: Version numbers

Object ID range identifier = 0x00 (CWE)
Reverse Object revision = 2
Forward Object revision = 5
The modem supports CWE objects with version numbers from 2 to 5.

Object ID range identifier = 0x02 (GPRS)
Reverse Object revision = 1
Forward Object revision = 6
The modem supports GPRS objects with version numbers from 1 to 6.

Object ID range identifier = 0x05 (UMTS)
Reverse Object revision = 4
Forward Object revision = 7
The modem supports UMTS objects with version numbers from 4 to 7.

Firmware compatibility

Firmware on Sierra Wireless modems is designed to be backward compatible. This CnS message allows you to determine if the modem firmware is current with your application.

If the modem returns a forward object revision that is lower than the object IDs used in your application, you must upgrade the firmware on the modem. See Downloading firmware updates (page 44).

Reset Modem

Shuts down and resets the modem:

- On request, using Set (03) to initiate the shutdown/power-up sequence

Table 9-12: Object—Reset Modem

Object ID	0x0010		Reset Modem	
Operations	✔ Set ✗ Get ✗ Notify			
	Parameters (fields)			
	Len	**Offset**	**Type**	**Description**
03 (Set)	2	0	UINT16	Object version • 0x0001=Revision 1
04 (Set Response)	2	0	UINT16	Object version • 0x0001=Revision 1

Return Modem Date and Time

Returns the current date and time from the modem:

- On request, using Get (01)
- Via notification (if Notify (07) is enabled) when the device initially attaches, or when time zone information changes

The time in the current time zone is calculated using the following response fields: current UTC time (Hour:Minute:Second), Time zone offset from UTC, and daylight saving adjustment.

For example:

Hour = 14; Minute = 32; Second = 17;
Offset = -8; Daylight savings adjustment = 0x01

Current ‘local’ time is:

14:32:17 + (-8 * 0.25) + 1
=14:32:17 - 2 hours + 1 hour
=14:32:17 - 1
=13:32:17

Table 9-13: Object—Return modem date and time

Object ID	0x001B		Return modem date and time	
Operations	✗ Set ✔ Get ✔ Notify			
	Parameters (fields)			
	Len	**Offset**	**Type**	**Description**
01 (Get)	0			None
02 (Get Response) **07 (Notify)**	19	0	UINT16	Object version number • 0x0001=Revision 1

Table 9-13: Object — Return modem date and time (Continued)

		2	UINT16	Year • Four digit year (for example, 2007)
		4	UINT16	Month • 1–12 (January to December)
		6	UINT16	Day • 1–31
		8	UINT16	Day of week • 0x0000=Monday • 0x0001=Tuesday • 0x0002=Wednesday • 0x0003=Thursday • 0x0004=Friday • 0x0005=Saturday • 0x0006=Sunday
		10	UINT16	Hour • 0–23 (in UTC)
		12	UINT16	Minute • 0–59 (In UTC)
		14	UINT16	Second • 0–59 (In UTC)

		16	UINT16	Time zone offset from UTC
				• -48 to +48—Each step represents 15 minutes. For example:
				  ◦ -48 = UTC - (48 * 15 min) = UTC - 12 hours
				  ◦ -6 = UTC - (6 * 15 min) = UTC - 1.5 hours
				  ◦ 0 = UTC
				  ◦ 1 = UTC + (1 * 15 min) = UTC + 0.25 hours
				  ◦ 12 = UTC + (12 * 15 min) = UTC + 3 hours

		18	UINT8	Daylight Savings adjustment • 0x00=No adjustment • 0x01=Plus one hour • 0x02=Plus two hours

Return PRI Information

Returns the device's current PRI information.

- On request, using Get (01)

Note: Support for some PRI sub-version fields may be limited on some products dependent on provisioning agreements with Sierra Wireless.

Table 9-14: Object—PRI Information

Object ID	0x0023		PRI information	
Operations	✗ Set ✔ Get ✗ Notify			
	Parameters (fields)			
	Len	**Offset**	**Type**	**Description**
01 (Get)	0			None
02 (Get Response)	14	0	UINT16	Object version number • 0x0001=Revision 1
		2	UINT16	Major version • Binary-encoded
		4	UINT16	Minor version • Binary-encoded
		6	UINT16	Reserved for future use
		8	UINT16	Reserved for future use
		10	UINT32	SKU ID or part number • Binary number (theoretical range is 0x00000000–0xFFFFFFFF)

Return Flash Image Information

This object allows the host to read versioning information for any of the individual CWE images programmed into the flash memory of the device:

- On request, using Set (01)

Note: This object should be used in place of Return Firmware Version (page 71), Return Firmware Build Date (page 71), Return Hardware Version (page 72), and Return Boot Version (page 72).

Table 9-15: Object—Flash Image Information

Object ID	0x0024		Return Flash image information		
Operations	✔ Set ✗ Get ✗ Notify				
	Parameters (fields)				
	Len	**Offset**	**Type**	**Description**	**Object Ver**
03 (Set)	See Object version number (Offset = 0) description for object length				
		0	UINT16	Object version number • 0x0001=Revision 1 Length = 6 • 0x0002=Revision 2 Length = 7	1–2
		2	CHAR[4]	Image type - Indicate which image you want to read. *Note: The device does not necessarily contain all image types. The response includes a parameter indicating if the requested type is not on the device.* • Uppercase letters only ('A'–'Z'), ASCII format • Valid values: ◦ "QCOM"—Boot images, including NAND image partition table, Primary Boot Loader (PBL), Secondary Boot Loader (SBL), OEMSBL (OEM Secondary Boot Loader), and configuration data used by the PBL. ◦ "BOOT"—Sierra Wireless Boot Loader ◦ "APPL"—Application code ◦ "USBD"—USB descriptor table	1–2
		6	UINT8	CWE TLV type • 0x00=Specific • 0x01=Generic	2

Table 9-15: Object—Flash Image Information (Continued)

04 (Set Response)	Variable length depending on number of image information objects (TLVs) included in response.			
	0	UINT16	Object version number • 0x0001=Revision 1 • 0x0001=Revision 2	1–2
	2	UINT8	Action result • 0x00=Success • 0x01=Unsupported image type • 0x02=Image not found on the device	1–2
	3	CHAR[4]	Image type • As described in 03 (Set), offset 0	1–2
	7	UINT8	Number of image information objects describing this image (following three fields repeat for each object) • Each object includes a type, length, and value	1–2
	Image Information objects (Type + Length + Value) • Number of objects depends on value in previous field • Only one object of each type can be returned. For example, there will never be two release dates returned for a specific information object. • Example: · Object type = 0x05 (Release date) · Object length = 0x05 (8 bytes) · Object value = 05/15/09 (May 15, 2009)			
	+0	UINT8	Object type • 0x02=UMTS CWE image version information (includes all image types except USBD) • 0x03=UMTS USBD image version information • 0x04=Reserved • 0x05=Release date • 0x06=Product Type • 0x07=Image size • 0x08=Generic CWE image version information	
	+1	UINT8	Object length • Length depends on object type: · 0x02=84 bytes · 0x03=84 bytes · 0x04=84 bytes · 0x05=8 bytes · 0x06=4 bytes · 0x07=4 bytes · 0x08=84 bytes	

Table 9-15: Object—Flash Image Information (Continued)

		+2	UINT8 [See previous field]	Object value • Value depends on object type: ◦ Type 0x02—CWE image version information. Format: <A><W>_<X>_<Y>_<Z><xx> ASCII string, null-padded ▪ <A>—UE family type (examples: U-6275; H-6280; D7200R2.x; F-7200R3.x) ▪ <W>—Major release ▪ <X>—Minor release ▪ <Y>—Sub-release ▪ <Z>—Sub-release ▪ <xx>—Load type. Valid values are ap (application) and bt (bootloader) ◦ Type 0x03—USBD image version information Format: <A><W>_<X>ud ASCII string, null-padded ▪ <A>—UE family type (examples: U-6275; H-6280; D7200R2.x; F-7200R3.x) ▪ <W>—Major release ▪ <X>—Minor release ▪ ud—Indicates load type is USB descriptor ◦ Type 0x05—Release date Format: MM/DD/YY ASCII string, not null-terminated ▪ Image release date (month / day / year) ◦ Type 0x06—Product type ASCII string, not null-terminated ▪ Identifies the product the image is intended for. Exact values are image-specific. Examples: “6800” or “5625” for BOOT and APPL images ◦ Type 0x07—Image size Binary value ▪ Size (in bytes) of image data only. Does not include size of CWE header. ◦ Type 0x08—Generic CWE image version information. Format: ASCII string, null-padded ▪ Identical to the firmware image CWE header. ▪ Suggested use: Host software can compare the modem’s version with the firmware image version.	

Return IMSI

Note: The MCC and MNC in the IMSI identify the subscriber's home PLMN (Public Land Mobile Network)—the network on which a subscriber obtains local service.

Returns the IMSI (International Mobile Subscription Identity), a non-dialable number that uniquely identifies a SIM:

- On request, using Get (01)

The IMSI is comprised of the MCC (Mobile Country Code), MNC (Mobile Network Code), and MSIN (Mobile Station Identification Number).

Table 9-16: Object—Return IMSI

Object ID	0x1004		Return IMSI	
Operations	✗ Set ✔ Get ✗ Notify			
	Parameters (fields)			
	Len	**Offset**	**Type**	**Description**
01 (Get)	0			None
02 (Get Response)	15	0	UINT16	Home PLMN MCC (Home mobile country code)
		2	UINT16	Home PLMN MNC (Home mobile network code)
		4	UINT8	Length of MSIN (excluding null terminator) at offset 5 • Valid range: 0–10
		5	CHAR[10]	MSIN

Return Hardware Build Date

Returns the date on which the modem was manufactured and calibrated:

- On request, using Get (01)

Table 9-17: Object—Return Hardware Build Date

Object ID	0x1005		Return Hardware Build Date	
Operations	✗ Set ✔ Get ✗ Notify			
	Parameters (fields)			
	Len	**Offset**	**Type**	**Description**
01 (Get)	0			None
02 (Get Response)	6	0	CHAR[6]	Build Date • Format—MMDDYY

Disable Modem

Returns the current state of the modem, or enables/disables it.

Returns current state:

- On request, using Get (01)
- When triggered, if Notify (07) is enabled *and* the modem state changes (This is an update-type notification.)

Enables/disables:

- On request, using Set (03) and Notify (07)

This object allows the host application to put the modem's radio into low power mode, causing the modem to detach from the network. When the modem is enabled, the radio comes back online and reattaches to the network.

Note: This setting persists even if the modem is reset.

Table 9-18: Object—Disable Modem

Object ID	0x1011		Disable Modem	
Operations	✔ Set ✔ Get ✔ Notify			
	Parameters (fields)			
	Len	**Offset**	**Type**	**Description**
01 (Get)	0			None
02 (Get Response)	1	0	UINT8	Current modem state: • 0x00=Modem disabled • 0x01=Modem enabled
03 (Set)	1	0	UINT8	Set the modem state: • 0x00=Disable the modem • 0x01=Enable the modem *Note: When the state changes, a Notify (07) operation occurs.*
07 (Notify)	1	0	UINT8	Modem state has changed: • 0x00=Modem disabled • 0x01=Modem enabled

Return Factory Serial Number

Returns the FSN (Factory Serial Number) that was programmed into the modem during manufacturing:

- On request, using Get (01)

Table 9-19: Object—Return FSN

Object ID	0x1016		Return FSN	
Operations	✗ Set ✔ Get ✗ Notify			
	Parameters (fields)			
	Len	**Offset**	**Type**	**Description**
01 (Get)	0			None
02 (Get Response)	15	0	CHAR[15]	FSN (Factory Serial Number) • Format—FPPDDDYNNNNHH-BB F—Factory code PP—Product ID DDD—Day of the year Y—Year NNNN—Unique serial number for each unit produced that day HH—Hardware version BB—Batch revision

Example: FSN (Factory Serial Number)

FSN = ABC1235abcd0504

Factory: A = Acme Labs
Product: BC = MC8755
Date: 123 = 03 May (Day 123)
Year: 5 = 2005
S/N: abcd
H/W ver: 05
Batch: 04

Return Test History

Returns the results of factory tests that were performed on the modem when it was manufactured:

- On request, using Get (01)

Table 9-20: Object—Return Test History

Object ID	0x1017		Return Test History	
Operations	✗ Set ✔ Get ✗ Notify			
	Parameters (fields)			
	Len	**Offset**	**Type**	**Description**
01 (Get)	0			None
02 (Get Response)	1	0	UINT8	Test results (each bit is set if the modem passed that test) • Bit 0—Initial test • Bit 1—RF screen test • Bit 2—Calibration test • Bit 3—System test • Bit 4–Bit 7—Undefined

Return FPGA Version

Returns the FPGA (Field Programmable Gate Array) version:

- On request, using Get (01)

Table 9-21: Object—Return FPGA Version

Object ID	0x101D		Return FPGA Version	
Operations	✗ Set ✔ Get ✗ Notify			
	Parameters (fields)			
	Len	**Offset**	**Type**	**Description**
01 (Get)	0			None
02 (Get Response)	1	0	UINT8	FPGA version number • Bit 7–Bit 4—Major revision number • Bit 3–Bit 0—Minor revision number • Example: FPGA version 4.1 = 0b01000001.

Return Modem Model

Returns the model of the modem:

- On request, using Get (01)

Note: The HIP message—Device Information Request (page 29)—should be used instead of this object. (Newer Sierra Wireless modems return a generic product type (0xFE) with this object.)

Table 9-22: Object—Return Modem Model

Object ID	0x1063		Return Modem Model	
Operations	✗ Set ✔ Get ✗ Notify			
	Parameters (fields)			
	Len	**Offset**	**Type**	**Description**
01 (Get)	0			None
02 (Get Response)	1	0	UINT8	Modem type (Values not listed are all reserved) • 0x0A=AC850 • 0x0B=AC860 • 0x0C=MC8755 • 0x0D=MC8765 • 0x13=AC875 • 0x14=MC8775 • 0x15=MC8775V • 0x18=AC880 • 0x19=AC881 • 0x1A=MC8780 • 0x1B=MC8781 • 0x1E=AC880E • 0x1F=AC881E • 0x24=AC595U[a] • 0x25=MC5727[a] • 0x26=AC597[a] • 0x29=AC885E • 0x2A=C885 • 0x2B=MC8790 • 0x2C=MC8790V • 0x2D=AC501 • 0x2E=C888 • 0x2F=C889 • 0x32=MC8791V • 0x33=MC8792V Continued on next page

Table 9-22: Object—Return Modem Model (Continued)

				Modem type (continued) • 0x39=AC402 • 0x3A=MC5727V[a] • 0x3B=MC5728[a] • 0x3C=MC5728V[a] • 0x45=MC8700 • 0x50=MC8795V • 0xFE=All other products • 0xFF=Invalid device

a. CDMA modem - Included for reference only

Return IMEI

Returns the IMEI (International Mobile Equipment Identity), a number that uniquely identifies a mobile device on GSM/UMTS networks:

- On request, using Get (01)

Table 9-23: Object—Return IMEI

Object ID	0x1067		Return IMEI	
Operations	✗ Set ✔ Get ✗ Notify			
	Parameters (fields)			
	Len	**Offset**	**Type**	**Description**
01 (Get)	0			None
02 (Get Response)	15	0	CHAR[15]	IMEI (International Mobile Equipment Identity) • Format—TTTTTTFFSSC T—Type approval code (6 digits) F—Final assembly code (2 digits) S—Serial number (6 digits) C—Check digit (1 digit) See Example: IMEI format below this table.

Example: IMEI format

IMEI = 354592000007417

Type Approval Code:354592
Final Assembly Code:00
Serial Number:000741
Check Digit:7

Back Up NVRAM Data (Deprecated)

Deprecated: *This command is deprecated and should not be used in new designs. For new designs, refer to Back Up NVRAM Data on page 77.*

Provides the ability to back up user settings stored in the modem's non-volatile (NV) memory prior to performing a firmware download:

- On request, using Set (03) to initiate the backup. A Notify (07) occurs when the operation finishes (if notification is enabled)

The backup and restore operations may each take up to three minutes to complete.

Note: This message needs to be sent before performing a firmware download; if it is not sent, then the user settings are lost.

Restoring the NV user settings

This allows the NV user settings to be saved before downloading a firmware update. The firmware update automatically restores the NV user settings when it is finished.

Table 9-24: Object—Back Up NVRAM Data (Deprecated)

Object ID	0x7000		Back Up NVRAM Data (Deprecated)	
Operations	✔ Set ✗ Get ✔ Notify			
	Parameters (fields)			
	Len	**Offset**	**Type**	**Description**
03 (Set)	2	0	UINT8	Requested operation: • 0x00–0x01=Reserved • 0x02=NV backup operation • 0x03=NV restore operation • 0x04–0xFF=Reserved
		1	UINT8	Category of requested operation: • 0x00–0x01=Reserved • 0x02=User Settings • 0x03–0xFF=Reserved
04 (Set Response)	2	0	UINT16	Object version number • 0x0001=Revision 1
07 (Notify)	15	0	UINT8	Operation status: • 0x00=Successful • 0x01=Unsuccessful

Table 9-24: Object—Back Up NVRAM Data (Deprecated) (Continued)

		1	UINT8	Operation performed: • 0x00–0x01=Reserved • 0x02=NV backup operation • 0x03=NV restore operation • 0x04–0xFF=Reserved
		2	UINT8	Category of performed operation: • 0x00–0x01=Reserved • 0x02=User Settings • 0x03–0xFF=Reserved
		3	UINT32	Number of items stored during the backup
		7	UINT32	Reserved
		11	UINT32	Reserved

Return RSCP and Ec/Io Measurements

(WCDMA only) Returns the Received Signal Code Power (RSCP) and the total Energy per chip per power density (Ec/Io) value of the active set's strongest cells (to a maximum of six cells), typically for diagnostic purposes:

- On request, using Get (01)
- When triggered, if Notify (07) is enabled *and* the information changes. The minimum time between notifications is approximately three seconds.

Table 9-25: Object—RSCP and Ec/Io Measurement

Object ID	0x700B		RSCP and Ec/Io Measurement	
Operations	✗ Set ✔ Get ✔ Notify			
	Parameters (fields)			
	Len	**Offset**	**Type**	**Description**
01 (Get)	0			None
02 (Get Response) **07 (Notify)**	Variable	0	UINT16	Object version • 0x0001=Revision 1
		2	UINT16	Number of cells • WCDMA—1 to 6 • GSM—0 (This object is only meaningful for WCDMA)
		4	Cell Information (Following three fields appear for the number of cells noted in previous field.) • Cell order—Strongest to weakest (based on RSCP)	
		+0	UINT16	Primary scrambling code
		+2	UINT16	RSCP • Units in dB

Table 9-25: Object—RSCP and Ec/Io Measurement (Continued)

		+4	UINT16	Ec/Io measurement
				• Units in -dB
				• Value represents half-decibel increments. For example:
				  ◦ 0x0001: Ec/Io = -0.5 dB
				  ◦ 0x0015: Ec/Io = -21.5 dB

Return GSM Cell Information

Returns GSM (2G) cell informtion, including up to six neighbor cell measurements, typically for diagnostic purposes:

- On request, using Get (01)
- Automatically, if Notify (07) is enabled, at intervals of approximately five seconds.

Table 9-26: Object—GSM Cell Information

Object ID	0x700C		GSM cell information		
Operations	✗ Set ✔ Get ✔ Notify				
	Parameters (fields)				
	Len	**Offset**	**Type**	**Description**	**Object Ver**
01 (Get)	0			None	n/a
					n/a
02 (Get Response) **07 (Notify)**	See Object version number (Offset = 0) description for object length Offset positions that differ between revisions appear as “Off1 | Off2” where Off1 is the offset for Revision 1 and Off2 is the offset for Revision 2.				
		0	UINT16	Object version • 0x0001=Revision 1 Length—30 to variable • 0x0002=Revision 2 Length—31 to variable	1–2
		2	UINT8	Serving cell information validity flag • 0x00=No serving cell information is available • 0x01=Serving cell information is valid	1–2
		3	UINT16	MCC	1–2
		5	UINT16	MNC	1–2
		7	UINT16	LAC	1–2
		9	UINT16	Cell ID	1–2

Table 9-26: Object — GSM Cell Information (Continued)

		11	UINT8	BSIC	1–2
		12	UINT8	NCC (Network Colour Code)	1–2
		13	UINT8	BCC—Base Station Colour Code	1–2
		14	UINT16	RAC	1–2
		16	INT16	Min Rx level to register (in dBm)	1–2
		18	UINT16	Reserved for future use • Default value—0xFFFF	1–2
		20	UINT8	Service status • 0x00=Normal • 0x01=Emergency / limited service • 0x02=No service	2
		Offset values in following rows are shown as ObjVer1 | ObjVer2. For example, the Band field is at offset 20 for Object version 1, and at offset 21 for Object version 2.			
		20 | 21	UINT8	Band • 0x2B=GSM 850 • 0x2C=GSM 900 • 0x2F=GSM 1800 • 0x30=GSM 1900 • 0xFF=Invalid band	1–2
		21 | 22	UINT16	ARFCN	1–2
		23 | 24	INT16	Rx level (in dBm)	1–2
		25 | 26	UINT8	C1	1–2
		26 | 27	UINT8	C2	1–2
		27 | 28	UINT8	C31	1–2
		28 | 29	UINT8	C32	1–2
		29 | 30	UINT8	Number of valid neighbor cell measurements • Valid range: 1–6	1–2
		30 | 31	Neighbor cell information (Following fields appear for the number of cells noted in previous field.)		1–2
		+0	UINT8	Band • 0x2B=GSM 850 • 0x2C=GSM 900 • 0x2F=GSM 1800 • 0x30=GSM 1900 • 0xFF=Invalid band	1–2
		+1	UINT16	ARFCN	1–2

Table 9-26: Object — GSM Cell Information (Continued)

		+3	UINT16	RAC • 0xFFFF if invalid	1–2
		+5	INT16	Rx level (in dBm)	1–2
		+7	UINT8	C1	1–2
		+8	UINT8	C2	1–2
		+9	UINT8	C31	1–2
		+10	UINT8	C32	1–2

10: SIM Status and Locking

Introduction

This chapter describes SIM-related CnS messages that:

- Support verification and changing of CHV codes
- Report the status of SIM operations
- Provide access to special operations (Run Algorithm)

SIM messages summary

Table 10-1: SIM messages

Object	ID	Description	S	G	N
Send MEP Unlock Code (page 103)	0x1019	Submit an unlocking code to use a SIM that is blocked by Mobile Equipment Personalization	✔	✗	✗
Verify CHV Code (page 104)	0x103E	Verify a CHV code or an unblocking CHV code (and new CHV code)	✔	✗	✗
Enable / Disable CHV1 Verification (page 106)	0x103F	Enables (or disables) checking of CHV1, or returns current verification status	✔	✔	✗
Change CHV Codes (page 107)	0x1040	Change a CHV code (CHV1 or CHV2)	✔	✗	✗
Return SIM Status (page 108)	0x1041	Reports the current status of the SIM after each SIM operation	✗	✗	✔
CHV2 Status Kick (page 111)	0x107B	Forces the modem to issue a **Return SIM Status** notification for CHV2	✔	✗	✗
Run SIM Authentication Algorithm (page 111)	0x1085	Run the algorithm used to authenticate the SIM to the GSM/GSM/UMTS network.	✔	✗	✗
Return SIM's ICC ID (page 112)	0x7007	Returns the ICC ID of the installed SIM.	✗	✔	✗

SIM messages reference

SIM Codes Overview

The SIM-related objects described in this chapter deal with three types of codes:

- MEP code—stored on the modem
- CHV codes—stored on the SIM
- Unblocking CHV codes—stored on the SIM

SIM status information

The Return SIM Status notification (page 108) reports detailed SIM status information for the various codes.

This notification should be enabled before using any other SIM objects.

MEP information

The MEP (Mobile Equipment Personalization) code is used to unlock a modem that has been factory-restricted to a specific carrier's SIMs. This code must be obtained from the original carrier before the modem can be used with SIMs from other carriers.

The following objects relate to the MEP code:

- Return SIM Status (page 108) notification
  Issued by the modem when the MEP code needs to be entered.
- Send MEP Unlock Code (page 103))
  Issued by the host software in response to the Return SIM Status notification.

CHV Codes

Two CHV (Card Holder Verification) codes, CHV1 and CHV2, are used to restrict access to:

- The SIM—CHV1 is required when the modem is restarted or reset
- FDN (Fixed Dialing Number) phonebook—CHV2 is required before the phonebook is accessed.

The only difference between CHV1 and CHV2 is that CHV1 checking can be enabled/disabled (if your host software is designed to allow this); CHV2 checking is always enabled.

CHV1 Code

Access to the SIM can be restricted by requiring the user-defined CHV1 to be entered whenever the modem restarts or resets. If desired, you can design your software to allow this feature to be enabled or disabled by the user (see Enable / Disable CHV1 Verification on page 106).

CHV1 verification allows users to secure their SIMs. When verification is enabled, the correct code must be entered before the SIM can be used in *any* device.

If CHV1 is enabled, the code must be entered correctly within a certain number of attempts (usually 3), or the SIM becomes blocked. Once blocked, the SIM is unusable (except to dial emergency numbers) until the correct unblocking code is entered.

Note: When CHV1 is enabled, the modem is not operational until the CHV1 code is entered.

The following objects relate to the CHV1 code (some also relate to the CHV2 code):

- Return SIM Status (page 108) notification
  Issued by the modem when it is restarted or reset, if CHV1 is enabled, to tell the host to prompt for the CHV1. It is also issued if the wrong CHV1 code is submitted.
- Verify CHV Code (page 104)
  Issued by the host to verify the CHV1 that the user has entered against the code stored on the SIM.
- Enable / Disable CHV1 Verification (page 106)
  Issued by the host to enable or disable verification. The current setting can be checked in the Return SIM Status (page 108) notification.

Note: The number of permitted attempts at correctly entering CHV and unblocking CHV codes is SIM-dependent. The host application should rely on the Return SIM Status (page 108) notification to identify the need to prompt for the CHV and unblocking CHV codes.

Note: The Return SIM Status (page 108) notification drives the SIM interface. It is issued at startup and whenever the SIM requires an action (for CHV, unblocking, MEP) and indicates the status of the previous operation attempted by the SIM (enable/disable, change, verify, etc.).

- Change CHV Codes (page 107)
  Issued by the host to change the CHV1 code (the initial CHV1 code is provided by the carrier). The application should first ensure the new code is entered twice correctly (the modem does not perform this check).

Note: CHV1 must be enabled before the code can be changed. Return SIM Status (page 108) returns an 'Operation failed' if CHV1 verification is disabled.

CHV2 Code

Access to the FDN phonebook is prevented by requiring the user-defined CHV2 to be entered.

Like CHV1, CHV2 becomes blocked if the wrong code is entered repeatedly (usually 3 times).

Note: The FDN phonebook (if enabled) limits the user to only the numbers in the book. When FDN is active, the ADN phonebook on the SIM is not available. See Enable/Disable FDN Mode (page 188) for details.

The following objects relate to the CHV2 code (some also relate to the CHV1 code):

Note: The Return SIM Status (page 108) notification drives the SIM interface. It is issued at startup and whenever the SIM requires an action (for CHV, unblocking, MEP) and indicates the status of the previous operation attempted by the SIM (enable/disable, change, verify, etc.).

- Return SIM Status (page 108) notification
  Issued by the modem before allowing FDN phonebook access, to tell the host to prompt for the CHV2. It is also issued if the wrong CHV2 code is submitted. See FDN phonebook entries (page 173) for an explanation of how the FDN phonebook works.
- CHV2 Status Kick (page 111)
  Issued (using the parameter 'Change') by the host before changing the CHV2 code to trigger the Return SIM Status (page 108) notification.
- Verify CHV Code (page 104)
  Issued by the host to verify the CHV2 that the user has entered against the code stored on the SIM.
- Change CHV Codes (page 107)
  Issued by the host to change the CHV2 code (the initial CHV2 code is provided by the carrier). The application should first ensure the new code is entered twice correctly (the modem does not make this check).

Unblocking CHV Codes

Note: The number of permitted attempts at correctly entering CHV and unblocking CHV codes is SIM-dependent. The host application should rely on the Return SIM Status (page 108) notification to identify the need to prompt for the CHV and unblocking CHV codes.

When a CHV code has been blocked after being entered incorrectly too many times consecutively, an unblocking CHV code must be entered. Unblocking CHV codes must be obtained from the carrier.

If the unblocking code is not entered correctly after a certain number of attempts (usually 10), CHV1 or CHV2 becomes permanently blocked:

- CHV1—the SIM becomes completely unusable (except to dial emergency numbers)
- CHV2—access to special features is lost

The following objects relate to the unblocking codes:

- Verify CHV Code (page 104)
  Issued by the host to verify the unblocking CHV that the user has entered against the code stored on the SIM.
- Return SIM Status (page 108) notification
  Issued by the modem to indicate the success of the unblocking attempt.

Send MEP Unlock Code

Submit an unlocking code to use a SIM that is blocked by Mobile Equipment Personalization:

- On request, using Set (03)

Table 10-2: Object—Send MEP Unlock Code

Object ID	0x1019		Send MEP Unlock Code	
Operations	✔ Set ✗ Get ✗ Notify			
	Parameters (fields)			
	Len	**Offset**	**Type**	**Description**
03 (Set)	13	0	UINT8	Length of unlocking code • Min—1; Max—12
		1	CHAR[12]	Unlocking code • 12 bytes • 1–12 characters, see previous parameter for actual length • Left justified, ignore trailing characters

Submitting the MEP unlocking code

The MEP unlocking code is required if:

- MEP unlocking is enabled (set by the carrier), *and*
- An invalid SIM is inserted (for example, a SIM from a competitor)

To unlock the device (shown in Figure 10-1 on page 104)

1. Insert an unsupported SIM.
   The modem issues a Return SIM Status (page 108) notification requiring a MEP unlock code to be entered.
2. Enter the correct MEP unlock code:
   - If the correct code is entered, the modem resets.
   - If an incorrect code is entered, a delay occurs and another Return SIM Status (page 108) notification is issued. This delay increases after each successive incorrect unlocking code is sent.
     Repeat step 2 until the correct code is entered.

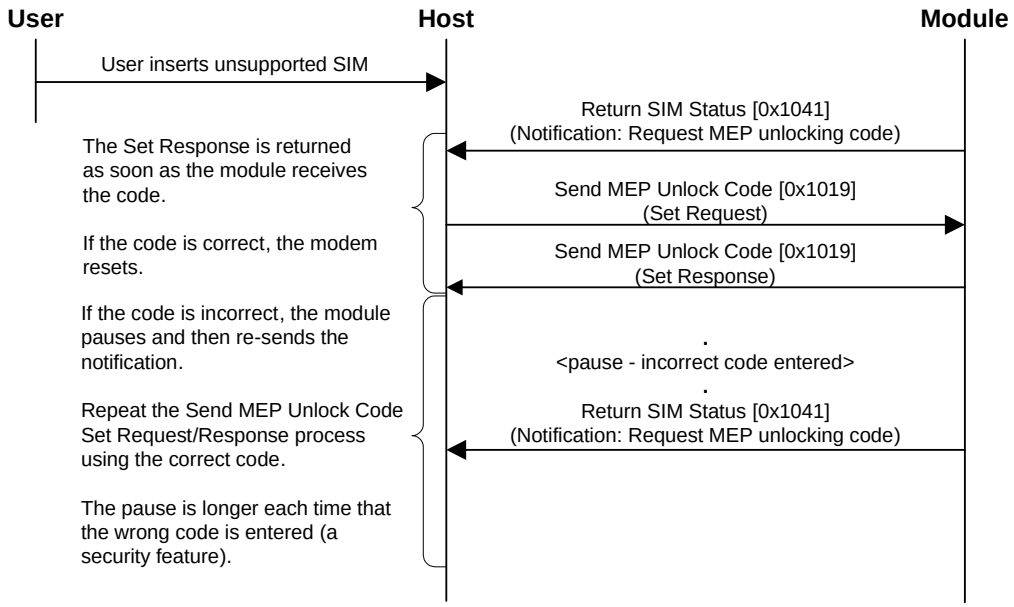

Figure 10-1: Message flow—MEP unlock

Verify CHV Code

Submit, for verification, a CHV code (CHV1 or CHV2) or CHV code and unlocking CHV code:

- When requested by the modem, using Set (03)

Table 10-3: Object—Verify CHV Code

Object ID	0x103E		Verify CHV Code	
Operations	✔ Set ✗ Get ✗ Notify			
	Parameters (fields)			
	Len	**Offset**	**Type**	**Description**
03 (Set)	20	0	UINT16	CHV type being submitted: • 0x0000=Reserved • 0x0001=CHV1 (for initial SIM access) • 0x0002=CHV2 (for FDN phonebook access) • 0x0003=Unblocking CHV1 • 0x0004=Unblocking CHV2
		2	UINT8	Length of CHV code • Min—4; Max—8
		3	CHAR[8]	CHV code (CHV1 or CHV2) • ASCII numeric characters (see previous parameter for length) • Left justified, ignore trailing characters

Table 10-3: Object — Verify CHV Code (Continued)

		11	UINT8	Length of unblocking CHV code • Valid value: 8 • Field is required
		12	CHAR[8]	Unblocking CHV code (unblocking CHV1 or unblocking CHV2) • ASCII numeric characters, no spaces (see previous parameter for length)

Submitting CHV codes

A **Verify CHV Code** object (page 104) must be issued after receiving a Return SIM Status (page 108) notification from the modem (requesting CHV1, CHV2, unblocking CHV1, or unblocking CHV2).

The modem requires either:

- A CHV code—CHV1 for initial SIM access (as shown in Figure 10-2 on page 106) or CHV2 for FDN phonebook access, *or*
- An unblocking CHV code and a new CHV code—the new CHV code (CHV1 or CHV2) is required because it is assumed that the user has forgotten the original code

The following table lists CHV request types that may be issued.

Table 10-4: CHV requests

Return SIM Status notification issued for:	When ...
CHV1	• The modem is started (or reset), *and* CHV1 checking is enabled (see Enable / Disable CHV1 Verification on page 106) • The CHV1 was entered incorrectly
Unblock CHV1	• The CHV1 was entered incorrectly too many times consecutively
CHV2	• The FDN phonebook is accessed—the host software must issue a CHV2 Status Kick (page 111) to get the CHV2 before permitting FDN access
Unblock CHV2	• The CHV2 was entered incorrectly too many times consecutively

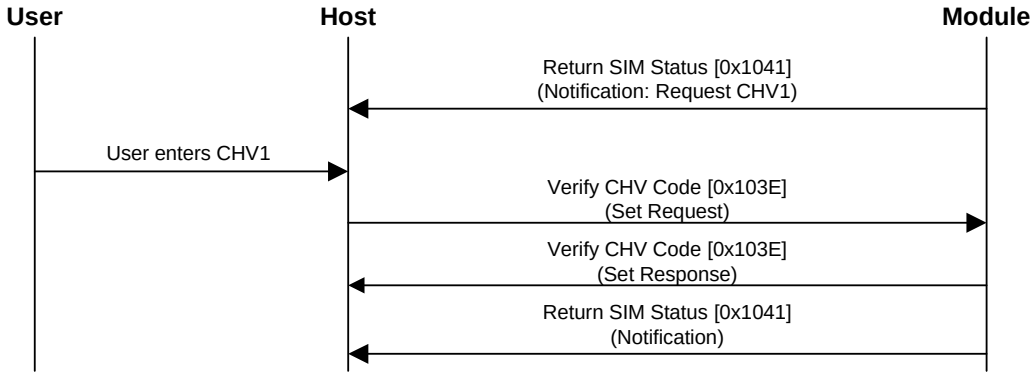

Figure 10-2: Message flow—CHV1 Verify on SIM insertion

Enable / Disable CHV1 Verification

Enables (or disables) checking of CHV1, or returns current verification status.

Enable/Disable:

- On request, using Set (03)

Return current status:

- On request, using Get (01)

Table 10-5: Object—Enable / Disable CHV1 Verification

Object ID	0x103F		Enable / Disable CHV1 Verification	
Operations	✔ Set ✔ Get ✗ Notify			
	Parameters (fields)			
	Len	**Offset**	**Type**	**Description**
01 (Get)	0			None
02 (Get Response)	1	0	UINT8	CHV1 Enable status: • 0x00=Disabled • 0x01=Enabled
03 (Set)	10	0	UINT8	Requested CHV1 Enable status: • 0x00=Disable CHV1 verification • 0x01=Enable CHV1 verification
		1	UINT8	Length of submitted CHV1 code • Min—4; Max—8
		2	CHAR[8]	Submitted CHV1 code • 4 to 8 ASCII characters (see previous parameter for length) • Left justified

Enabling / Disabling CHV verification

Use this object if your software allows users to turn CHV1 verification on or off (as shown in Figure 10-3). This change takes effect immediately.

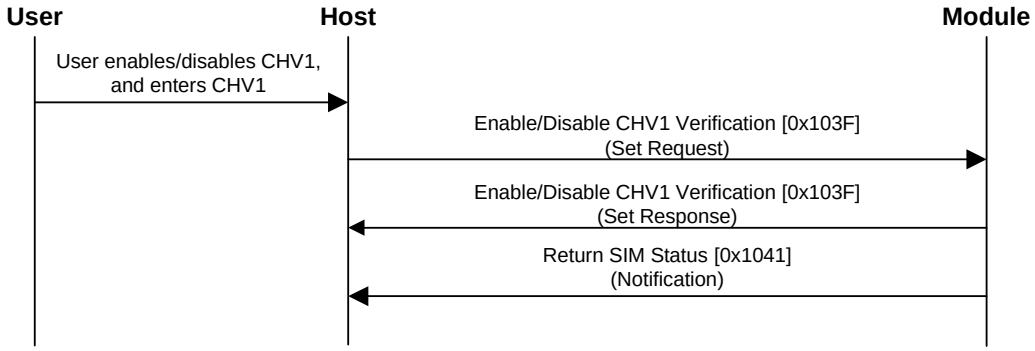

Figure 10-3: Message flow—CHV1 verification enable/disable

Note: CHV2 cannot be disabled.

Change CHV Codes

Change a CHV code (CHV1 or CHV2):

- On request, using Set (03)

Table 10-6: Object—Change CHV Codes

Object ID	0x1040		Change CHV Codes	
Operations	✔ Set ✗ Get ✗ Notify			
	Parameters (fields)			
	Len	**Offset**	**Type**	**Description**
03 (Set)	20	0	UINT16	CHV type to be changed: • 0x0000=Reserved • 0x0001=CHV1 • 0x0002=CHV2 • 0x0003=Reserved
		2	UINT8	Length of old CHV code • Min—4; Max—8
		3	CHAR[8]	Old CHV code • 4–8 ASCII characters (length from previous parameter) • Left justified, ignore unused trailing characters

Table 10-6: Object—Change CHV Codes (Continued)

		11	UINT8	Length of new CHV code • Min—4; Max—8
		12	CHAR[8]	New CHV code • 4–8 ASCII characters (length from previous parameter) • Left justified, ignore unused trailing characters

Note: CHV1 checking must be enabled (see Enable / Disable CHV1 Verification on page 106) before CNV1 or CNV2 codes can be changed.

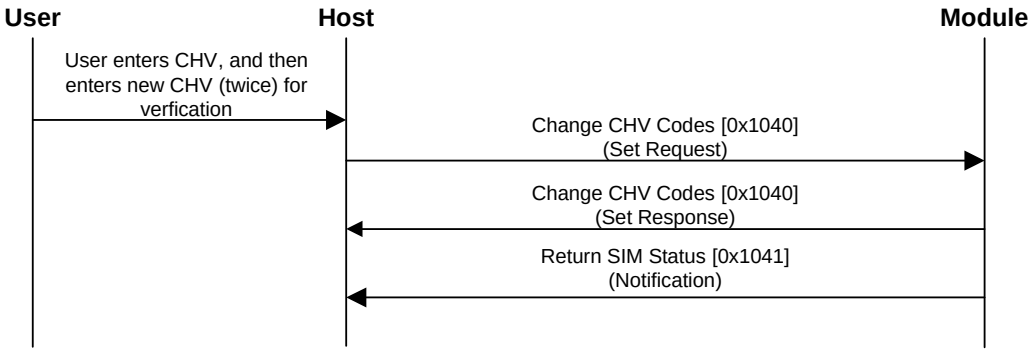

Figure 10-4: Message flow—Change a CHV code

Return SIM Status

Returns the current status of the SIM after each SIM operation:

- If Notify (07) is enabled
  (This is an update-type notification.)

Note: This notification drives the SIM interface. It is issued at startup and whenever the SIM requires an action (for CHV, unblocking, MEP) and indicates the status of the previous operation attempted by the SIM (enable/disable, change, verify, etc.)
Because the notification is issued after every attempted SIM operation, the host does not have to keep its own SIM status. Thus, the host implementation can be entirely event-driven.

Table 10-7: Object—Return SIM Status

Object ID	0x1041		Return SIM Status	
Operations	✗ Set ✗ Get ✔ Notify			
	Parameters (fields)			
	Len	**Offset**	**Type**	**Description**
07 (Notify)	7	0	UINT8	SIM status: • 0x00=SIM OK • 0x01=SIM not inserted • 0x02=SIM removed • 0x03=SIM initialize failure • 0x04=SIM general failure • 0x05=SIM locked • 0x06=SIM CHV1 blocked • 0x07=SIM CHV2 blocked • 0x08=SIM CHV1 rejected • 0x09=SIM CHV2 rejected • 0x0A=SIM MEP locked • 0x10=SIM network reject • 0x11–0xFF=Reserved *Note: The user should be made aware of the SIM status.*
		1	UINT8	SIM user operation required—Indicates if the user needs to be prompted to enter a code • 0x00=No operation required • 0x01=Enter CHV1 • 0x02=Enter CHV2 • 0x03=Enter unblocking CHV1 • 0x04=Enter unblocking CHV2 • 0x05=Enter MEP special code • 0x06–0xFF=Reserved

Table 10-7: Object—Return SIM Status (Continued)

		2	UINT8	Previous user operation Use this field and the next field to determine what message to show to the user. • 0x00=No previous operation • 0x01=Change CHV1 • 0x02=Change CHV2 • 0x03=Unused • 0x04=Verify CHV1 • 0x05=Verify CHV2 • 0x06=Verify unblocking CHV1 • 0x07=Verify unblocking CHV2 • 0x08=MEP unlock • 0x09=Enable CHV1 verification • 0x0A=Disable CHV1 verification • 0x0B–0xFF=Reserved
		3	UINT8	Result of previous user operation Use this field and the previous field to determine what message to show to the user. • 0x00=Operation succeeded • 0x01=Operation failed • 0x02=Operation failed because CHV verification is disabled. It must be enabled before this operation can be attempted (see Enable / Disable CHV1 Verification on page 106) • 0x03=Operation failed because the CHV starts with the same digits as an emergency number • 0x04–0xFF=Reserved
		4	UINT8	Information available for retry: • 0x00=No retry information is available in this message packet • 0x01=This message packet contains retry information • 0x02–0xFF=Reserved
		5	UINT8	Retry information type This field and the next field indicate the number of remaining retries that may be attempted for the specified type. • 0x00=CHV1 • 0x01=CHV2 • 0x02=Unblocking CHV1 • 0x03=Unblocking CHV2 • 0x04–0xFF=Reserved

Table 10-7: Object—Return SIM Status (Continued)

		6	UINT8	Number of retries remaining This field and the previous field indicate the number of remaining retries that may be attempted for the specified type. • Min—0; Max—10

CHV2 Status Kick

Forces the modem to issue a **Return SIM Status** notification (with current CHV2 retry details, and optionally to require entry of CHV2):

- On request, using Set (03)

Suggested usage

This object is used to force the entry of CHV2 before allowing access to the FDN phonebook.

Table 10-8: Object—CHV2 Status Kick

Object ID	0x107B		CHV2 Status Kick	
Operations	✔ Set ✗ Get ✗ Notify			
	Parameters (fields)			
	Len	**Offset**	**Type**	**Description**
03 (Set)	1	0	UINT8	CHV2 kick type • 0x00=Verify • 0x01=Change

Run SIM Authentication Algorithm

Causes the modem to run the SIM Authentication Algorithm. This is the same algorithm that is used for authenticating the SIM to the GSM/UMTS network:

- On request, using Set (03)

Table 10-9: Object—Run SIM Authentication Algorithm

Object ID	0x1085		Run SIM Authentication Algorithm	
Operations	✔ Set ✗ Get ✗ Notify			
	Parameters (fields)			
	Len	**Offset**	**Type**	**Description**
03 (Set)	16	0	UINT8[16]	A 16-byte random number used as the seed for the authentication algorithm.
04 (Set Response)	12	0	UINT32	SRES—The algorithm result.
		4	UINT8[8]	Kc—An 8-byte cipher key generated by the algorithm.

Return SIM's ICC ID

Returns the ICC ID of the installed SIM:

- On request, using Get (01)

Table 10-10: Object — Return SIM's ICC ID

Object ID	0x7007		Return SIM's ICC ID	
Operations	✗ Set ✔ Get ✗ Notify			
	Parameters (fields)			
	Len	**Offset**	**Type**	**Description**
01 (Get)	0			None
02 (Get Response)	10	0	UINT8[10]	ICC ID - Integrated Circuit Card Identifier in BCD (Binary Coded Decimal) format

11: Profile Management

Introduction

This chapter describes CnS messages that relate to creating, editing, deleting, and retrieving profiles. There is also a message that allows you to set the default profile.

Profile messages summary

Table 11-1: Profile messages

Object	ID	Description	S	G	N
Return Profile Summary (page 114)	0x7001	Returns a summary of all packet session profiles stored on the modem and sets the default user profile.	✔	✔	✔
Read Profile (page 115)	0x7002	Returns one of four different categories of information for a defined profile.	✔	✗	✗
Write Profile (page 128)	0x7003	Writes one of four different categories of information for a defined profile.	✔	✗	✔
Manage Profile Username and Password (page 128)	0x700E	Reads or writes the username and/or password for a defined profile.	✔	✔	✗

Profile messages reference

Return Profile Summary

Returns a summary of all packet session profiles stored on the modem and sets the default user profile.

Returns the summary:

- On request, using Get (01)
- When triggered, if Notify (07) is enabled *and* the default profile ID is changed from a non-CnS interface (for example, AT commands)

Sets the default:

- On request, using Set (03)

Table 11-2: Object—Return Profile Summary

Object ID	0x7001			Return Profile Summary
Operations	✔ Set ✔ Get ✔ Notify			
	Parameters (fields)			
	Len	**Offset**	**Type**	**Description**
01 (Get)	0			None
02 (Get Response)	19	0	UINT8	Maximum number of defined profiles supported—modem-dependent
		1	UINT8	Maximum number of TFTs per profile supported—modem-dependent
		2	UINT8	Default user profile ID • 0=No profiles defined • 1–16=One of the defined profiles
		3	UINT8[16]	Profile ID statuses—16 bytes where each byte represents the status of one profile ID (Bytes 1–16 correspond to profile IDs 1–16) • 0x00=Inactive • 0x01=Active • 0x02=Undefined
03 (Set)	1	0	UINT8	Default profile ID—Use this to change/set the Current default profile ID • 1–16=One of the defined profiles
07 (Notify)	1	0	UINT8	Current default profile ID • 0=No profiles defined • 1–16=One of the defined profiles

Read Profile

Reads one of four different categories of information for a defined profile:

- On request, using Set (03). The information is returned in the Set Response (04).

Note: Use Manage Profile Username and Password (page 128) to read the profile's username and password.

Note: The Write Profile (page 128) uses a similar structure:
Set Request—Same format as the Read Profile Set Response.
Set Response—Same format as the Read Profile Set Request.

Table 11-3: Object—Read Profile

Object ID	0x7002		Read Profile	
Operations	✔ Set ✗ Get ✗ Notify			
	Parameters (fields)			
	Len	**Offset**	**Type**	**Description**
03 (Set)	3	0	UINT8	Profile ID—The profile being requested • Min—1; Max—16
		1	UINT8	Profile information type—The type of profile information being requested. (The Set Response section of this table contains separate record details for each type.) • 0x00=Basic information • 0x01=QoS information • 0x02=DNS information • 0x03=TFT information
		2	UINT8	TFT filter ID • Valid only when the Profile information type is set to TFT. For all other tpes, set to 0. • Min—1; Max—8
04 (Set Response) **Profile type: Basic**	245	0	UINT8	Profile ID • Min—1; Max—16
		1	UINT8	Profile info type • 0x00=Basic info
		2	UINT8	Profile valid flag • 0x00=Profile is invalid or not yet configured • 0x01=Profile is valid

Table 11-3: Object—Read Profile (Continued)

		3	UINT8	PDP type • 0x00=IPv4 PDP type • 0x01=PPP PDP type • 0x02=IPv6 PDP type
		4	UINT8	Header compression flag • 0x00=No compression • 0x01=Compression
		5	UINT8	Data compression flag • 0x00=No compression • 0x01=Compression
		6	UINT8	Length of PDP address • 0=Dynamic addressing • 1–16=Length of next field *Note: The PDP address is not used by the modem—this field is provided only for host storage convenience.*
		7	UINT8[16]	PDP address—When using static addressing, this is the IP or PDP address. Examples: • IPv4 (4 bytes)—0x7F000201 (127.0.2.1) • IPv6 (16 bytes)—0x000020010DB8000000000202B3FF8329 (2001:DB8: : :0202:B3FF:8329) *Note: The PDP address is not used by the modem—this field is provided only for host storage convenience.*
		23	UINT8	Length of APN • Min—0; Max—100 (0 indicates no APN specified)
		24	UINT8[100]	APN—Access Point Name Example: “internet.fido.ca”
		124	UINT8	PDP Initiated Type • 0x00=Primary mobile initiated • 0x01=Secondary mobile initiated
		125	UINT8	Primary profile ID • Min—1; Max—16 *Note: This field is only valid if the PDP Initiated Type is 0x01—Secondary Mobile initiated.*

Table 11-3: Object—Read Profile (Continued)

		126	UINT8	Authentication Type • 0x00=No authentication • 0x01=PAP authentication • 0x02=CHAP authentication

		127	UINT8[33]	Username
				Note: Manage Profile Username and Password (page 128) should be used to read the username—it supports extended usernames (up to 127 bytes). This field is provided for backwards compatibility only.
				• Maximum 32-character null terminated string (total space is 33 bytes).
				• If PDP Type is:
				  ◦ PPP PDP—This field is not used by the modem. It is provided only for host storage convenience.
				  ◦ IPv4/IPv6 PDP—If the username or password (next field) is non-null, then both fields are sent to the network for authentication.

		160	UINT8[33]	Password *Note: Manage Profile Username and Password (page 128) should be used to read the password—it supports extended usernames (up to 127 bytes). This field is provided for backwards compatibility only.* • 32-character null terminated string (total space is 33 bytes) • See Username (previous field) for usage.
		193	UINT8[31]	Configuration Buffer Label • 30-character null-terminated string (total space is 31 bytes) *or* 14-UCS2 character null-terminated string (prepended by a single byte containing 0x80 to indicate that UCS2 format is used) • All strings that *can* be encoded in ASCII format *should* be stored in ASCII. *Note: This field is not used by the modem—it is provided only for host storage convenience.*

Table 11-3: Object—Read Profile (Continued)

		224	UINT8	Automatic Context Activation mode • 0x00=Deferred context activation (context only activated when requested by user) • 0x01=Immediate context activation (“always connected” configuration)
		225	UINT8	Profile Write Protect • 0x00=User can Read Profile and Write Profile • 0x01=User can only Read Profile (write-protected) *Note: All other values are rejected by the firmware and defaulted to 0 (User can read and write the profile).*
		226	UINT8	Prompt-for-password flag • 0x00=The user should NOT be prompted for a password. The Password field is valid and should be used at time of context activation. • 0x01=The user should be prompted for a password. The Password field is invalid and the host should prompt the user for a password at time of context activation. The Password is not stored in NVRAM. *Note: All other values are rejected by the firmware and defaulted to 0 (No prompt for password).*
		227	UINT8	Auto-launch application • 0x00=Don’t auto launch VPN • 0x01=Auto launch VPN
		228	UINT16	PDP Linger Timer • 0=Disables timer • 1–32767=Time in minutes
		230	UINT8	Software options • Bit 0—0=NDIS profile; 1=DUN profile • Bit 1—0=Enable data flow; 1=Disable data flow • Bit 2—0=Disable WINS; 1=Enable WINS • Bit 3—0=No prompt for DUN password; 1=Prompt for DUN password • Bit 4–Bit 7—Reserved for future use
		231	UINT8[14]	Reserved bytes • Reserved for future use • Default value—0x00

Table 11-3: Object — Read Profile (Continued)

04 (Set Response) **Profile type — QOS**	48	0	UINT8	Profile ID • Min—1; Max—16
		1	UINT8	Profile information type • 0x01=QoS information
		2	UINT8	Valid flag for request UMTS QoS • 0x00=This QoS is invalid or not yet configured • 0x01=This QoS is valid
		3	UINT8	Request UMTS QoS Traffic Class • 0=Based on user's subscription • 1=Conversational • 2=Streaming • 3=Interactive • 4=Background
		4	UINT8	Request UMTS QoS Max Bitrate UL • 0=Based on user's subscription • 1=16 Kbps • 2=32 Kbps • 3=64 Kbps • 4=128 Kbps • 5=256 Kbps • 6=384 Kbps • 7=1.8 Mbps • 8=3.6 Mbps • 9=7.2 Mbps • 10=14.4 Mbps • 11=2.048 Mbps • 12=5.76 Mbps • 13=11.5 Mbps • 14=16 Mbps • 15=21 Mbps • 16=28 Mbps • 17=42 Mbps • 18=84 Mbps
		5	UINT8	Request UMTS QoS Max Bitrate DL • Same range as Request UMTS QoS Max Bitrate UL (offset 4)
		6	UINT8	Request UMTS QoS Guaranteed Bitrate UL • Same range as Request UMTS QoS Max Bitrate UL (offset 4)

Table 11-3: Object — Read Profile (Continued)

		7	UINT8	Request UMTS QoS Guaranteed Bitrate DL • Same range as Request UMTS QoS Max Bitrate UL (offset 4)
		8	UINT8	Request UMTS QoS Delivery Order • 0=Based on user's subscription • 1=With delivery order • 2=Without delivery order
		9	UINT8	Request UMTS QoS Delivery of Erroneous SDU • 0=Based on user's subscription • 1=No detect • 2=Erroneous SDUs are delivered • 3=Erroneous SDUs are not delivered
		10	UINT16	Request UMTS QoS Maximum SDU Size • 0=Based on user's subscription • 1–1520=SDU size in bytes
		12	UINT8	Request UMTS QoS SDU error ratio • 0=Based on user's subscription • 1=1E-2 • 2=7E-3 • 3=1E-3 • 4=1E-4 • 5=1E-5 • 6=1E-6 • 7=1E-1
		13	UINT8	Request UMTS QoS Residual Bit error ratio • 0=Based on user's subscription • 1=5E-2 • 2=1E-2 • 3=5E-3 • 4=4E-3 • 5=1E-3 • 6=1E-4 • 7=1E-5 • 8=1E-6 • 9=6E-8
		14	UINT16	Request UMTS QoS Transfer Delay • 0=Based on user's subscription • 1–4000=Transfer delay in ms

Table 11-3: Object — Read Profile (Continued)

		16	UINT8	Request UMTS QoS Traffic Handling priority • 0=Based on user's subscription • 1=Priority level 1 • 2=Priority level 2 • 3=Priority level 3
		17	UINT8	Request UMTS QoS Source statistics descriptor • 0=Unknown • 1=Speech
		18	UINT8	Request UMTS QoS signalling indication • 0=Not optimized for signalling traffic • 1=Optimized for signalling traffic
		19	UINT8	Valid flag for minimum UMTS QoS • 0x00=This QoS is invalid or not yet configured • 0x01=This QoS is valid
		20	UINT8	Minimum UMTS QoS Traffic Class • 0=Based on user's subscription • 1=Conversational • 2=Streaming • 3=Interactive • 4=Background
		21	UINT8	Minimum UMTS QoS Max Bitrate UL • Same range as Request UMTS QoS Max Bitrate UL (offset 4)
		22	UINT8	Minimum UMTS QoS Max Bitrate DL • Same range as Request UMTS QoS Max Bitrate UL (offset 4)
		23	UINT8	Minimum UMTS QoS Guaranteed Bitrate UL • Same range as Request UMTS QoS Max Bitrate UL (offset 4)
		24	UINT8	Minimum UMTS QoS Guaranteed Bitrate DL • Same range as Request UMTS QoS Max Bitrate UL (offset 4)
		25	UINT8	Minimum UMTS QoS Delivery Order • 0=Based on user's subscription • 1=With delivery order • 2=Without delivery order

Table 11-3: Object — Read Profile (Continued)

		26	UINT8	Minimum UMTS QoS Delivery of Erroneous SDU • 0=-Based on user's subscription • 1=No detect • 2=Erroneous SDUs are delivered • 3=Erroneous SDUs are not delivered
		27	UINT16	Minimum UMTS QoS Maximum SDU Size • 0=Based on user's subscription • 1–1520=SDU size in bytes
		29	UINT8	Minimum UMTS QoS SDU error ratio • 0=Based on user's subscription • 1=1E-2 • 2=7E-3 • 3=1E-3 • 4=1E-4 • 5=1E-5 • 6=1E-6 • 7=1E-1
		30	UINT8	Minimum UMTS QoS Residual Bit error ratio • 0=Based on user's subscription • 1=5E-2 • 2=1E-2 • 3=5E-3 • 4=4E-3 • 5=1E-3 • 6=1E-4 • 7=1E-5 • 8=1E-6 • 9=6E-8
		31	UINT16	Minimum UMTS QoS Transfer Delay • 0=Based on user's subscription • 1–4000=Transfer delay in ms
		33	UINT8	Minimum UMTS QoS Traffic Handling priority • 0=Based on user's subscription • 1=Priority level 1 • 2=Priority level 2 • 3=Priority level 3
		34	UINT8	Minimum UMTS QoS Source statistics descriptor • 0=Unknown • 1=Speech

Table 11-3: Object — Read Profile (Continued)

		35	UINT8	Minimum UMTS QoS signalling indication • 0=Not optimized for signalling traffic • 1=Optimized for signalling traffic
		36	UINT8	Valid flag for request GPRS QoS • 0x00=This QoS is invalid or not yet configured • 0x01=This QoS is valid
		37	UINT8	Request GPRS QoS precedence • 0=Based on user's subscription • 1=High priority • 2=Normal priority • 3=Low priority
		38	UINT8	Request GPRS QoS delay • 0=Based on user's subscription • 1=Delay class 1 • 2=Delay class 2 • 3=Delay class 3 • 4=Delay class 4
		39	UINT8	Request GPRS QoS reliability • 0=Based on user's subscription • 1=Acknowledged GTP, LLC, and RLC.; Protected data • 2=Unacknowledged GTP; Acknowledged LLC and RL; Protected data • 3=Unacknowledged GTP and LLC; Acknowledged RLC; Protected data • 4=Unacknowledged GTP, LLC and RLC; Protected data • 5=Unacknowledged GTP, LLC and RLC; Unprotected data
		40	UINT8	Request GPRS QoS peak • 0=Based on user's subscription • 1=Up to 1000 octet/s • 2=Up to 2000 octet/s • 3=Up to 4000 octet/s • 4=Up to 8000 octet/s • 5=Up to 16000 octet/s • 6=Up to 32000 octet/s • 7=Up to 64000 octet/s • 8=Up to 128000 octet/s • 9=Up to 256000 octet/s

Table 11-3: Object — Read Profile (Continued)

		41	UINT8	Request GPRS QoS mean • 0=Based on user's subscription • 1=100 octet/h • 2=200 octet/h • 3=500 octet/h • 4=1000 octet/h • 5=2000 octet/h • 6=5000 octet/h • 7=10000 octet/h • 8=20000 octet/h • 9=50000 octet/h • 10=100000 octet/h • 11=200000 octet/h • 12=500000 octet/h • 13=1000000 octet/h • 14=2000000 octet/h • 15=5000000 octet/h • 16=10000000 octet/h • 17=20000000 octet/h • 18=50000000 octet/h • 31=Best effort
		42	UINT8	Valid flag for minimum GPRS QoS • 0x00=This QoS is invalid or not yet configured • 0x01=This QoS is valid
		43	UINT8	Minimum GPRS QoS precedence. • 0=Based on user's subscription • 1=High priority • 2=Normal priority • 3=Low priority
		44	UINT8	Minimum GPRS QoS delay • 0=Based on user's subscription • 1=Delay class 1 • 2=Delay class 2 • 3=Delay class 3 • 4=Delay class 4

Table 11-3: Object — Read Profile (Continued)

		45	UINT8	Minimum GPRS QoS reliability • 0=Based on user's subscription • 1=Acknowledged GPT, LLC, and RLC; Protected data • 2=Unacknowledged GTP, Acknowledged LLC and RLC; Protected data • 3=Unacknowledged GTP and LLC, Acknowledged RLC; Protected data • 4=Unacknowledged GTP, LLC and RLC; Protected data • 5=Unacknowledged GTP, LLC and RLC; Unprotected data
		46	UINT8	Minimum GPRS QoS peak • 0=Based on user's subscription • 1=Up to 1000 octet/s • 2=Up to 2000 octet/s • 3=Up to 4000 octet/s • 4=Up to 8000 octet/s • 5=Up to 16000 octet/s • 6=Up to 32000 octet/s • 7=Up to 64000 octet/s • 8=Up to 128000 octet/s • 9=Up to 256000 octet/s
		47	UINT8	Minimum GPRS QoS mean • 0=Based on user's subscription • 1=100 octet/h • 2=200 octet/h • 3=500 octet/h • 4=1000 octet/h • 5=2000 octet/h • 6=5000 octet/h • 7=10000 octet/h • 8=20000 octet/h • 9=50000 octet/h • 10=100000 octet/h • 11=200000 octet/h • 12=500000 octet/h • 13=1000000 octet/h • 14=2000000 octet/h • 15=5000000 octet/h (Continued on next page)

Table 11-3: Object—Read Profile (Continued)

				Minimum GPRS QoS mean (continued) • 16=10000000 octet/h • 17=20000000 octet/h • 18=50000000 octet/h • 31=Best Effort
04 (Set Response) **Profiletype—DNS**	36	0	UINT8	Profile ID • Min—1; Max—16
		1	UINT8	Profile information type • 0x02=DNS info
		2	UINT8	Length of primary DNS address (next field) • 0=No DNS address present. Modem solicits a network-supplied DNS address. • 4=IPv4 address • 16=IPv6 address
		3	UINT8[16]	Primary DNS address • This field is ignored if a network-supplied DNS address is being obtained (0 in previous field). Examples: • IPv4 (4 bytes)—0x7F000201 (127.0.2.1) • IPv6 (16 bytes)—0x000020010DB8000000000202B3FF8329 (2001:DB8: : :0202:B3FF:8329)
		19	UINT8	Length of secondary DNS address (next field) • 0=No DNS address present. Modem solicits a network-supplied DNS address. • 4=IPv4 address • 16=IPv6 address
		20	UINT8[16]	Secondary DNS address • This field is ignored if a network-supplied DNS address is being obtained (0 in previous field). Examples: • IPv4 (4 bytes)—0x7F000201 (127.0.2.1) • IPv6 (16 bytes)—0x000020010DB8000000000202B3FF8329 (2001:DB8: : :0202:B3FF:8329)
04 (Set Response) **Profile type—TFT**	57	0	UINT8	Profile ID • Min—1; Max—16
		1	UINT8	Profile information type • 0x03=TFT info

Table 11-3: Object—Read Profile (Continued)

		2	UINT8	TFT filter ID • Min—1; Max—8 *Note: Current firmware only supports 2 filters, so the supported TFT filter ID range is 1–2.*
		3	UINT8	Valid flag • 0x00=This TFT is invalid or not yet configured • 0x01=This TFT is valid
		4	UINT8	Evaluation precedence index (EPI) • Min—0; Max—255
		5	UINT8	Length of IP address • 4=IPv4 address (4 bytes) • 16=IPv6 address (16 bytes)
		6	UINT8[16]	Source address • 4 or 16 bytes, depending on previous field • Ignore trailing bytes for IPv4 addresses Examples: • IPv4 (4 bytes)—0x7F000201 (127.0.2.1) • IPv6 (16 bytes)—0x000020010DB8000000000202B3FF8329 (2001:DB8: : :0202:B3FF:8329)
		22	UINT8[16]	Source address mask for IPv4 addresses Example: 0xFFFFFF00 is 255.255.255.0
		38	UINT8	Protocol number (IPv4) or Next header (IPv6) • Min—0; Max—255
		39	UINT16	Destination port range ‘from’ • Min—0; Max—65535
		41	UINT16	Destination port range ‘to’ • Min—0; Max—65535
		43	UINT16	Source port range ‘from’ • Min—0; Max—65535
		45	UINT16	Source port range ‘to’ • Min—0; Max—65535
		47	UINT32	Security parameter index • Min—0x00000000; Max—0xFFFFFFFF
		51	UINT8	Type of service (IPv4) / Traffic class (IPv6) • Min—0; Max—255
		52	UINT8	Type of service mask (IPv4) / Traffic class mask (IPv6) • Min—0; Max—255

Table 11-3: Object—Read Profile (Continued)

		53	UINT32	Flow label (IPv6) • Min—0x00000; Max—0xFFFFF

Write Profile

Writes one of four different categories of information for a defined profile in the MiniCard NVRAM:

- On request, using Set (03)

Whenever the profile is changed (by this CnS object or from a non-CnS interface—for example, AT commands), a Notify (07) is issued with the updated profile ID and information type.

The profile is enabled or disabled by setting the *Profile valid flag* in the Basic profile data. If the profile is disabled, all the basic profile data is set to default values.

Note: Use Manage Profile Username and Password (page 128) to write the profile's username and password.

Table 11-4: Object—Write Profile

Object ID	0x7003		Write Profile	
Operations	✔Set ✗Get ✔Notify			
	Parameters (fields)			
	Len	**Offset**	**Type**	**Description**
03 (Set)	The Set Request uses the same format as the Set Response in Read Profile (page 115).			
07 (Notify)	The Notification uses the same format as the Set Request in Read Profile (page 115).			

Manage Profile Username and Password

This object reads or writes the username and password for a specific profile. It is used in conjunction with Read Profile (page 115) and Write Profile (page 128) to manage profiles.

Note: The Read Profile and Write Profile objects supported a maximum length of 32 bytes for usernames and passwords. This object supports 127 byte usernames and passwords.

The username and/or password are read from NVRAM:

- On request, using Get (01)

The username and/or password are written to NVRAM:

- On request, using Set (03)

Table 11-5: Object—Manage Profile Username and Password

Object ID	0x700E		Manage Profile Username and Password	
Operations	✔ Set ✔ Get ✗ Notify			
	Parameters (fields)			
	Len	**Offset**	**Type**	**Description**
01 (Get)	3	0	UINT16	Current object version number • 0x0001=Revision 1
		2	UINT8	Profile ID • 1–16=The defined profile to query
02 (Get Response) **03 (Set)**	Variable length: • Response: Max=270 • Set: Max=264			
		0	UINT16	Current object version number • 0x0001=Revision 1
		2	UINT8	Profile ID • 1–16=The defined profile to query
		3	UINT8	Number of TLV profile element objects returned (following three fields repeat for each object) • Up to two objects can be returned (password and username) • Each object includes a type, length, and value

	Profile element objects (Type + Length + Value)
	• Number of objects depends on value in previous field
	• Only one object of each type can be returned. Each profile has a specific username and password.
	• Example:
	  ◦ Object type = 0x01 (Password)
	  ◦ Object length = 0x21 (33 bytes - includes null string terminator)
	  ◦ Object value = “ThisStringHasExactly32Characters”			

		+0	UINT8	Object type • 0x00=Username (Get Response and Set Request) • 0x01=Password (Get Response and Set Request) • 0x02=Maximum username length (Get Response only) • 0x03=Maximum password length (Get Response only)

Table 11-5: Object — Manage Profile Username and Password (Continued)

		+1	UINT8	Object length • Length depends on object type: ◦ 0x00=32 to 128 bytes ◦ 0x01=32 to 128 bytes ◦ 0x02=1 byte ◦ 0x03=1 byte
		+2	UINT8 [See previous field]	Object value • Value depends on object type: ◦ Type 0x00—Username ▪ Null-terminated ASCII string ▪ PPP PDP contexts—Not used by modem; provided for host storage convenience. ▪ IPv4/IPv6 PDP context—If configured and either the Username or Password are not null, then both fields are sent to the network for authentication. ◦ Type 0x01—Password ▪ Null-terminated ASCII string ▪ PPP PDP contexts—Not used by modem; provided for host storage convenience. ▪ IPv4/IPv6 PDP context—If configured and either the Username or Password are not null, then both fields are sent to the network for authentication. ◦ Type 0x02—Maximum username string length supported by the device. ▪ Valid range: 32–128. ◦ Type 0x03—Maximum password string length supported by the device. ▪ Valid range: 32–128.
04 (Set Response)	3	0	UINT16	Current object version number • 0x0001=Revision 1
		2	UINT8	Request status • 0x00=Request successful • 0x01=UE does not support extended username/password string length • 0xFF=General failure

12: Status Messages

Introduction

This chapter describes CnS messages that:

- Detect the heartbeat of the modem
- Identify the application state
- Notify the device that the application is open

Status messages summary

Table 12-1: Status messages

Object	ID	Description	S	G	N
Heartbeat (page 132)	0x0000	Provides a 'heartbeat' notification indicating normal modem operation.	✗	✗	✔
Indicate Host Software State (page 132)	0x1071	Sends a message to the modem indicating the host software is being turned 'off' or 'on'.	✔	✗	✗
Indicate Host Software Launched (page 133)	0x1083	Sends a message to the modem indicating the host software has been launched.	✔	✗	✗

Status messages reference

Heartbeat

Provides a “heartbeat” notification indicating normal modem operation:

- Periodically, if Notify (07) is enabled

When the heartbeat notification is enabled, the modem periodically sends this object (at a maximum interval of approximately 10 seconds) to indicate that the modem is functioning.

Table 12-2: Object—Heartbeat

Object ID	0x0000		Heartbeat	
Operations	✗ Set ✗ Get ✔ Notify			
	Parameters (fields)			
	Len	**Offset**	**Type**	**Description**
07 (Notify)	0			None

Indicate Host Software State

Sends a message to the modem indicating the host software is being turned ‘off’ or ‘on’:

- On request, using Set (03)

To conserve power (and silence the radio), the modem detaches from the network whenever the host software shuts down.

Table 12-3: Object—Indicate Host Software State

Object ID	0x1071		Indicate Host Software State	
Operations	✔ Set ✗ Get ✗ Notify			
	Parameters (fields)			
	Len	**Offset**	**Type**	**Description**
03 (Set)	1	0	UINT8	Host software state • 0x00=Starting up—Application launch, PC power-up after suspend • 0x01=Shutting down—Application termination, power suspended, windows shutdown

Indicate Host Software Launched

Sends a message to the modem indicating the host software has been launched:

- On request, using Set (03)

Table 12-4: Object identification—Indicate Host Software Launched

Object ID	0x1083		Indicate Host Software Launched	
Operations	✔ Set ✗ Get ✗ Notify			
	Parameters (fields)			
	Len	**Offset**	**Type**	**Description**
03 (Set)	0			None

13: Data Services

Introduction

This chapter describes CnS messages that are related to data connection management and services, including:

- Network status, and available network service type
- Current and available bands details, and switching of current band
- PLMN availability and details, selection methods, and display issues
- Managing packet sessions

Data services messages summary

Table 13-1: Data services messages

Object	ID	Description	S	G	N
Report Current Band (page 138)	0x0016	Reports the current band being used.	✗	✔	✔
Set Current Band (page 139)	0x0017	Returns a list of supported band groupings and sets requested band grouping.	✔	✔	✗
Report Network Status (page 141)	0x1000	Reports the modem's current network status.	✗	✔	✔
Report Radio Information (page 143)	0x1001	Reports the current receive signal strength and approximate bit error rate.	✗	✔	✔
Return Service Provider Name (page 144)	0x1003	Returns the service provider name.	✗	✔	✗
Report Available Service Details (page 145)	0x1006	Reports the network service type that is available, and packet service readiness details.	✗	✗	✔
Report Current Band (Deprecated) (page 146)	0x100B	Reports the current band being used (and a list of available bands).	✗	✔	✔
Select PLMN (page 147)	0x100F	Choose a new PLMN manually or automatically, or report the current PLMN selection mode.	✔	✔	✔
Report call byte counters (Rx/Tx) (page 149)	0x1037	Reports the total number of bytes received and transmitted since the last time the totals were cleared.	✗	✔	✔
Reset call byte counters (page 150)	0x1038	Resets the counters for number of bytes received and transmitted.	✔	✗	✗
Report Manual PLMN Selection Readiness (page 150)	0x103C	Reports the current availability of manual PLMN selection service.	✗	✔	✔

Table 13-1: Data services messages (Continued)

Object	ID	Description	S	G	N
Return Available PLMN List (page 151)	0x1042	Returns a list of all currently available PLMNs.	✔	✔	✔
Report PLMN Registration Information (page 154)	0x1043	Reports the current PLMN name (if any) and service provider name (if any) that should be shown to the user.	✗	✔	✔
Report Roaming Icon Display Method (page 155)	0x106A	Indicates how the roaming icon should be displayed.	✗	✗	✔
Change Current Band (Deprecated) (page 156)	0x106B	Switch the modem to a new band.	✔	✗	✔
Manage Packet Session (page 157)	0x7004	Allows the host to initiate or terminate a connection with the MiniCard.	✔	✗	✔
Return Packet Session Status (page 164)	0x7005	Returns the current packet session session for a specific profile.	✔	✗	✗
Return IP Address (page 164)	0x7006	Returns the IP address for a specific profile.	✔	✗	✗
Report System and Network Status (page 165)	0x700A	Reports the modem's current system and network status.	✗	✔	✔
Set RAT mode (page 168)	0x700F	Reports or sets the modem's current Radio Access Technology mode.	✔	✔	✗
Manage call byte counters (Rx/Tx) (page 168)	0x7012	Reports or resets transmit/receive byte counters for a specific profile.	✔	✔	✔

Data service messages reference

Data service codes overview

The data service-related objects described in this chapter deal with:

- Controlling the modem
- Selecting a network
- Reporting additional network information

Controlling the modem

The following objects provide information about the modem's status and ability to establish a network connection, and allow selection of the band being used:

- Report Network Status (page 141)
  This indicates the current network connection status and is issued anytime the network status changes.
- Report Radio Information (page 143)
  This indicates the signal strength and is issued anytime the strength changes. The details can be used to display the RSSI in the user interface.
- Report Available Service Details (page 145)
  This indicates the availability of GPRS, EDGE, or UTMS service and is issued anytime the service status changes. The details can be used to display the current network status in the user interface.
- Report Current Band (page 138)
  This indicates the current band being used. It is issued anytime the current band changes.
- Set Current Band (page 139)
  This changes the band being used.

Selecting a network

Selection of a PLMN can be performed automatically, or manually by using a list of available PLMNs. The process of choosing and registering the modem on a PLMN requires the use of the following objects:

- Report Manual PLMN Selection Readiness (page 150)
  This indicates if PLMNs can currently be selected manually and is issued anytime the current status changes.
- Return Available PLMN List (page 151)
  When Manual PLMN Selection is available, this is used to get a complete list of the currently available local PLMNs.
- Select PLMN (page 147)
  This is used to tell the modem to automatically select an available PLMN, or to choose a specific PLMN that has been selected manually from the Available PLMN List.

Reporting additional network information

Additional information about specific PLMNs and service providers may be obtained for display in the user interface. The following objects are used to provide this information:

- Return Service Provider Name (page 144)
  This returns the name of the service provider the user subscribes to.
- Report PLMN Registration Information (page 154)
  This returns (one or both of) the service provider's name and the PLMN name for display to the user.
- Report Roaming Icon Display Method (page 155)
  This indicates the method that should be used for displaying the roaming icon indicator in the user interface.

Managing packet sessions

Packet sessions for specific profiles can be initiated/terminated by the host, and information about the sessions can be retrieved. The following objects are used to provide this information:

- Manage Packet Session (page 157)
  Allows the host to initiate or terminate the packet session for a specific profile.
- Return Packet Session Status (page 164)
  Returns details about the current packet session for a specific profile.
- Return IP Address (page 164)
  Returns the IP address associated with the current packet session for a specific profile.

Report Current Band

Reports the current band being used:

- On request, using Get (01)
- When triggered, if Notify (07) is enabled *and* the current band changes

Table 13-2: Object—Report Current Band

Object ID	0x0016		Report Current Band	
Operations	✗ Set ✔ Get ✔ Notify			
	Parameters (fields)			
	Len	**Offset**	**Type**	**Description**
01 (Get)	0			None
02 (Get Response) **07 (Notify)**	3 or 5	0	UINT16	Current object version number • 0x0001=Revision 1

Table 13-2: Object — Report Current Band (Continued)

		2	UINT8	Indicates if any band is in use • 0x0000=No band in use. • 0x0001=Band in use, identified in following fields (offsets 3 and 4). *Note: If no band is in use then the response is 3 bytes in length, otherwise the response is 4 bytes in length.*
		3	UINT8	Radio type • 0x0002=GSM/WCDMA • Other values are reserved
		4	UINT8	Current band in use. • 0x00–0x0C=Reserved • 0x28=GSM 450 • 0x29=GSM 480 • 0x2A=GSM 750 • 0x2B=GSM 850 • 0x2C=GSM E-GSM 900 • 0x2D=GSM P-GSM 900 • 0x2E=GSM R-GSM 900 • 0x2F=GSM DCS 1800 • 0x30=GSM PCS 1900 • 0x50=WCDMA I IMT 2000 • 0x51=WCDMA II PCS 1900 • 0x52=WCDMA III DCS 1800 • 0x53=WCDMA IV 1700 • 0x54=WCDMA V 850 • 0x55=WCDMA VI 800 • 0x56=WCDMA VII 2600 • 0x57=WCDMA VIII 900 • 0x58=WCDMA IX 1700 • 0xFF=Invalid band

Set Current Band

Note: This command replaces the following deprecated commands: Report Current Band (Deprecated) (page 146) and Change Current Band (Deprecated) (page 156).

Returns a list of device-supported band groupings (and the last user-selected grouping), and sets the desired band grouping:

Returns the list of band groupings (and the last user-selected grouping):

- On request, using Get (01)

Sets the desired band grouping:

- On request, using Set (03)

If the current band being used changes based on the new requested band grouping, a Report Current Band (Deprecated) (page 146) notification is issued.

Table 13-3: Object—Set Current Band

Object ID	0x0017		Set Current Band	
Operations	✔ Set ✔ Get ✗ Notify			
	Parameters (fields)			
	Len	**Offset**	**Type**	**Description**
01 (Get)	0			None
02 (Get Response)	Variable length (11+) depending on number of image information objects (TLVs) included in response.			
		0	UINT16	Current object version number • 0x0001=Revision 1
		2	UINT64	Last band grouping selected by the user. (Composite bands are not detailed. For example, GS850/PCS would be supported if the bits for GSM850 and GSM PCS 1900 are set.) • 0x0000000000010000=GSM 450 • 0x0000000000020000=GSM 480 • 0x0000000000040000=GSM 750 • 0x0000000000080000=GSM 850 • 0x0000000000100000=GSM RGSM 900 • 0x0000000000200000=GSM PCS 1900 • 0x0000000000400000=GSM DCS 1800 • 0x0000000000800000=GSM EGSM 900 • 0x0000000001000000=GSM PGSM 900 • 0x0000000100000000=WCDMA I IMT 2000 (EU) • 0x0000000200000000=WCDMA II PCS 1900 • 0x0000000400000000=WCDMA III 1700 • 0x0000000800000000=WCDMA IV 1700 • 0x0000001000000000=WCDMA V 850 • 0x0000002000000000=WCDMA VI 800 • 0x0000004000000000=WCDMA VII 2600 • 0x0000008000000000=WCDMA VIII 900 • 0x0000010000000000=WCDMA IX 1700 • 0xFFFFFFFFFFFFFFFF=Autoband

Table 13-3: Object—Set Current Band (Continued)

		10	UINT8	Number of device-supported band groupings • 0x00=No supported band groupings. • 0x01–0x19=Number of supported band groupings listed in the next field (offset 11). • Maximum number of supported band groupings is 25. *Note: If no groupings are supported (0x00), then the response is 11 bytes in length.* *If one or more groupings are supported, the response is 11 + (8 * NumberOfGroups) bytes in length.*
		11	UINT64 [#Groups]	Array of supported band groupings • Variable-length array—The number of supported groupings is in the previous field. • Valid groupings are listed at offset 2.
03 (Set)	10	0	UINT16	Object version number • 0x0001=Revision 1
		2	UINT64	Requested band grouping setting • Valid values are listed at offset=2 of the Get request.
04 (Set Response)	2	0	UINT16	Current object version number • 0x0001=Revision 1

Report Network Status

Reports the modem's current network status:

- On request, using Get (01)
- When triggered, if Notify (07) is enabled *and* the network status changes (This is an update-type notification.)

Table 13-4: Object—Report Network Status

Object ID	**0x1000**		Report Network Status	
Operations	✗ Set ✔ Get ✔ Notify			
	Parameters (fields)			
	Len	**Offset**	**Type**	**Description**
01 (Get)	0			None

Table 13-4: Object—Report Network Status (Continued)

02 (Get Response) **07 (Notify)**	27	0	UINT16	Service status: • 0x0000=Normal • 0x0001=Emergency only • 0x0002=No service • 0x0003=Access difficulty • 0x0004=Forbidden PLMN • 0x0005=Location area is forbidden • 0x0006=National roaming is forbidden • 0x0007=Illegal mobile station • 0x0008=Illegal mobile equipment • 0x0009=IMSI unknown in HLR • 0x000A=Authentication failure • 0x000B=GPRS failed • 0x000C–0xFFFF=Reserved
		2	UINT8	Service type: • 0x00=Circuit-switched service • 0x01=GPRS service • 0x02=Combined service • 0x03=Invalid service • 0x04–0xFF=Reserved
		3	UINT8	Roaming status: • 0x00=Not roaming • 0x01=Roaming
		4	UINT8	Network selection mode: • 0x00=Automatic • 0x01=Manual
		5	UINT8	Length of country string (excluding trailing spaces)
		6	CHAR[4]	Country string • Up to 4 characters, not null-terminated (see previous parameter for length) • Left justified, blank filled
		10	UINT8	Length of network string (excluding trailing spaces)
		11	CHAR[8]	Network string • Up to 8 characters, not null-terminated (see previous parameter for length) • Left justified, blank filled
		19	UINT16	MCC (Mobile Country Code)
		21	UINT16	MNC (Mobile Network Code)

Table 13-4: Object — Report Network Status (Continued)

		23	UINT16	LAC (Location Area Code)
		25	UINT16	Cell ID

Report Radio Information

Reports the current receive signal strength and approximate bit error rate:

- On request, using Get (01)
- When triggered, if Notify (07) is enabled *and* the signal strength changes (This is an update-type notification.)

Table 13-5: Object — Report Radio Information

Object ID	**0x1001**		Report Radio Information	
Operations	✗ Set ✔ Get ✔ Notify			
	Parameters (fields)			
	Len	**Offset**	**Type**	**Description**
01 (Get)	0			None
02 (Get Response) **07 (Notify)**	4	0	UINT16	Signal level value (0–63) representing dBm units (-110 to -47). Signal Level — dBm units 0 — 110 1 — 109 ... — ... 30 — 80 ... — ... 63 — 47 *Note: dBm units = -110 + Signal Level Value*
		2	UINT16	Average BER (Bit Error Rate), percentage-based: • 0x00=BER < 0.2 (assumed value) 0.14% • 0x01=0.2 < BER < 0.4 0.28% • 0x02=04. < BER < 0.8 0.57% • 0x03=0.8 < BER < 1.6 1.13% • 0x04=1.6 < BER < 3.2 2.26% • 0x05=3.2 < BER < 6.4 4.53% • 0x06=6.4 < BER < 12.8 9.05% • 0x07=12.8 < BER 18.10%

Determining RSSI levels on MMI graphical displays

The following table describes a recommended mapping of signal strength to bar graph graduations for an MMI graphical display:

Table 13-6: Recommended signal strength to bar graph mapping

# of bars	Signal level
0	0–1 (-110 dBm to -109 dBm)
1	2–8 (-108 dBm to -102 dBm)
2	9–17 (-101 dBm to -93 dBm)
3	18–23 (-92 dBm to -87 dBm)
4	24–32 (-86 dBm to -78 dBm)
5	33–63 (-77 dBm to -47 dBm)

Return Service Provider Name

Returns the SPN (Service Provider Name):

- On request, using Get (01)

Table 13-7: Object—Report Service Provider Name

Object ID	0x1003		Report Service Provider Name	
Operations	✗ Set ✔ Get ✗ Notify			
	Parameters (fields)			
	Len	**Offset**	**Type**	**Description**
01 (Get)	0			None
02 (Get Response)	17	0	UINT8	Length of service provider name (excluding trailing spaces)
		1	CHAR[16]	Service provider name • Up to 16 characters, see previous field for length • Left justified, blank filled

Report Available Service Details

Reports the network service type that is available to the modem, GPRS attached status, and packet service readiness details:

- When triggered, if Notify (07) is enabled *and* the available service details change
  (This is an update-type notification.)

Table 13-8: Object—Report Available Service Details

Object ID	0x1006		Report Available Service Details	
Operations	✗ Set ✗ Get ✔ Notify			
	Parameters (fields)			
	Len	**Offset**	**Type**	**Description**
07 (Notify)	245	0	UINT16	Object version • 0x0001=Revision 1
		2	UINT8	Display the service icon: • 0x00=No • 0x01=Yes
		3	UINT8	Service type available: • 0x00=None • 0x01=GPRS • 0x02=EDGE • 0x03=UMTS • 0x04=HSDPA • 0x05=HSUPA • 0x06=HSPA (both HSDPA and HSUPA) • 0x07=HSPA+ • 0x08=DC-HSPA+
		4	UINT8	GPRS Attached status The modem has GPRS attached to the network and is available to receive packet services: • 0x00=No • 0x01=Yes
		5	UINT8	Packet Session Active status The modem has set up a data session, received an IP address, and can transfer packet data. • 0x00=No • 0x01=Yes
		6	UINT8[239]	Reserved

Report Current Band (Deprecated)

Deprecated: *This command is deprecated and should not be used in new designs. For new designs, refer to Set Current Band on page 139.*

Reports the current band being used (and a list of available bands):

- On request, using Get (01)
- When triggered, if Notify (07) is enabled *and* the current or available bands change.

Table 13-9: Object—Report Current Band (Deprecated)

Object ID	0x100B		Report Current Band (Deprecated)	
Operations	✗ Set ✔ Get ✔ Notify			
	Parameters (fields)			
	Len	**Offset**	**Type**	**Description**
01 (Get)	0			None
02 (Get Response) **07 (Notify)**	14	0	UINT16	Current band being used: • 0x0000=None/invalid • 0x0001=PGSM • 0x0002=EGSM • 0x0003=DCS • 0x0004=PGSM DCS • 0x0005=EGSM DCS • 0x0006=PCS • 0x0007=PGSM PCS • 0x0008=EGSM PCS • 0x0009=850 MHz • 0x000A=850 MHz/PCS • 0x000B=GSM All • 0x000C=WCDMA 2100 EU • 0x000D=WCDMA 850/1900 NA • 0x000E=WCDMA 850/1900 GSM 850/1900 NA • 0x000F=WCDMA 2100 GSM 900/1800 EU • 0x0010=WCDMA All • 0x0011=WCDMA 2100/850 Australia • 0x0012=WCDMA 2100/800 Japan • 0x0013=WCDMA 2100/850 GSM 900/1800 • 0x0014=WCDMA 850 GSM 900/1800 • 0x0015=WCDMA 850 • 0x0016=WCDMA 2100 Continued on next page

Table 13-9: Object—Report Current Band (Deprecated) (Continued)

				Current band being used (continued) • 0x0017=WCDMA 2100/900 GSM 900/1800 • 0x0018=WCDMA 900 • 0x0019=WCDMA 1900 • 0x001A=WCDMA 2100/900 • 0x001B–0xFFFF=Reserved
		2	UINT16	Number of available bands (listed in next parameter): • 0=Invalid band option list • 1–5=Number of bands in list
		4	UINT16[5]	Available bands • 10 bytes (2 bytes/band) • Up to 5 bands recorded, ignore unused trailing bytes • See Current band field for valid values

Setting the current band

1. Issue a Get (01) to retrieve the list of available bands or use the list from the latest Notify (07) object).
2. Use Change Current Band (Deprecated) (page 156) to switch to the correct band.

Select PLMN

Choose a new PLMN manually or automatically, or report the current PLMN selection mode.

Report current mode:

- On request, using Get (01)

Choose new PLMN:

- On request, using Set (03). The result is returned using Notify (07), if enabled.

Table 13-10: Object—Select PLMN

Object ID	0x100F		Select PLMN	
Operations	✔ Set ✔ Get ✔ Notify			
	Parameters (fields)			
	Len	**Offset**	**Type**	**Description**
01 (Get)	0			None
02 (Get Response)	1	0	UINT8	Current PLMN selection mode: • 0x00=Automatic • 0x01=Manual

Table 13-10: Object—Select PLMN (Continued)

03 (Set)	5	0	UINT8	Selection mode for choosing new PLMN: • 0x00=Automatic • 0x01=3G Manual • 0x02=2G Manual
		1	UINT16	MCC (Mobile Country Code) • 0 if using Automatic selection mode • MCC of desired PLMN See Return Available PLMN List (page 151) to obtain MCCs for each PLMN.
		3	UINT16	MNC (Mobile Network Code) • 0 if using Automatic selection mode • MNC of desired PLMN See Return Available PLMN List (page 151) to obtain MNCs for each PLMN.
07 (Notify)	1	0	UINT8	Result of Set (03) request: • 0x00=Selection successful • 0x01=Registration attempt failed • 0x02=Request refused because PLMN registration selection is already in progress • 0x03=Request refused because external factors caused registration to fail • 0x04=Internal CnS error • 0x05=Invalid PLMN selection • 0x06–0xFF=Reserved

Automatic PLMN selection

To have the modem automatically choose a PLMN:

1. Use the Set (03) operation and choose Automatic selection mode.
2. Check the Notify (07) response to see if a PLMN was selected.

Manual PLMN selection

To manually choose a specific PLMN:

1. Use Report Manual PLMN Selection Readiness (page 150) to verify that manual selection is available.
2. Use Return Available PLMN List (page 151) to get a list of available PLMNs.
3. Use Select PLMN (page 147) ‘Set (03)’, choosing Manual selection mode with the correct MMC and MNC.
4. Check the Notify (07) response to see if the PLMN was selected.

Report call byte counters (Rx/Tx)

Note: For MSM6290-based devices, this command is deprecated. Refer to Manage call byte counters (Rx/Tx) (page 168).

Reports the total number of bytes received and transmitted since the last time the totals were cleared:

- On request, using Get (01)
- Periodically, if Notify (07) is enabled

Note: If multiple PDP contexts are in session, byte counts are reported for the first activated PDP session.

Table 13-11: Object—Report call byte counters

Object ID	0x1037		Report call byte counters	
Operations	✗ Set ✔ Get ✔ Notify			
	Parameters (fields)			
	Len	**Offset**	**Type**	**Description**
01 (Get)	0			None
02 (Get Response) 07 (Notify)	8	0	UINT32	Number of bytes transmitted
		4	UINT32	Number of bytes received

Managing call byte counters

To use the call byte counters to track the number of bytes transmitted and received during a call:

1. Enable the Report call byte counters notification.
2. When a data connection is established, use the Get(01) operation to obtain the starting receive and transmit counter values.
3. Receive notifications periodically reporting the current receive and transmit counter values. Subtract the starting values to determine the call-specific receive and transmit quantities.
4. If you are using the counters to track receive and transmit quantities for the current call only (not over a series of calls), then when the data connection ends, use Reset call byte counters on page 150 to clear the receive and transmit quantities.

Reset call byte counters

Note: For MSM6290-based devices, this command is deprecated. Refer to Manage call byte counters (Rx/Tx) (page 168).

Resets the call byte counters (Rx/Tx). See Managing call byte counters on page 149 for suggested usage.

- On request, using Set (01)

Note: If multiple PDP contexts are in session, byte counts are cleared for the PDP session (the first activated PDP session) reported in Report call byte counters (Rx/Tx) (page 149).

Table 13-12: Object—Reset call byte counters

Object ID	0x1038		Reset call byte counters	
Operations	✔ Set ✗ Get ✗ Notify			
	Parameters (fields)			
	Len	**Offset**	**Type**	**Description**
03 (Set)	0			None
04 (Response)	0			None

Report Manual PLMN Selection Readiness

Reports the current availability of manual PLMN selection service:

- On request, using Get (01)
- When triggered, if Notify (07) is enabled *and* the availability of manual PLMN selection services changes
  (This is an update-type notification.)

Table 13-13: Object—Report Manual PLMN Selection Readiness

Object ID	0x103C		Report Manual PLMN Selection Readiness	
Operations	✗ Set ✔ Get ✔ Notify			
	Parameters (fields)			
	Len	**Offset**	**Type**	**Description**
01 (Get)	0			None
02 (Get Response) **07 (Notify)**	1	0	UINT8	Manual PLMN selection service availability: • 0x00=Unavailable • 0x01=Available • 0x02–0xFF=Reserved

Using Manual PLMN selection

If manual PLMN selection is available, you can use:

- Return Available PLMN List (page 151) to obtain a list of PLMNs from which a specific PLMN will be manually selected
- Select PLMN (page 147) to attempt to register with a specific PLMN chosen from the available PLMN list

Return Available PLMN List

Returns a list of all currently available PLMNs:

- On request, using a combination of Set (03), Get (01), and Notify (07) operations

Table 13-14: Object—Return Available PLMN List

Object ID	0x1042		Return Available PLMN List	
Operations	✔ Set ✔ Get ✔ Notify			
	Parameters (fields)			
	Len	**Offset**	**Type**	**Description**
01 (Get)	0			None
02 (Get Response)	97	0	UINT8	Length of full PLMN name • Min—0; Max—31
		1	UINT8[62]	Full PLMN name • UCS2 format • 62 bytes • Up to 31 characters (2 bytes / character)—See previous parameter for actual length • Ignore unused trailing bytes
		63	UINT8	Length of abbreviated PLMN name: • Min—0; Max—13
		64	UINT8[26]	Abbreviated PLMN name • UCS2 format • 26 bytes • Up to 13 characters (2 bytes per character)—See previous parameter for actual length • Ignore unused trailing bytes
		90	UINT16	MCC (Mobile Country Code)
		92	UINT16	MNC (Mobile Network Code)

Table 13-14: Object—Return Available PLMN List (Continued)

		94	UINT8	PLMN status (when bits are set): • Bit 0—PLMN is registered • Bit 1—PLMN is forbidden • Bit 2—PLMN is the home PLMN • Bit 3—PLMN is weak • Bit 4—PLMN supports GPRS bit • Bits 5–7—Reserved
		95	UINT8	Network type: • 0x00=Unknown network • 0x01=GSM network • 0x02=DCS network • 0x03=GSM DCS network • 0x04=PCS network • 0x05=GSM PCS network • 0x06=ICO network • 0x07=UMTS network • 0x08–0xFF=Reserved
		96	UINT8	"More PLMNs" flag. Indicates if a Get (01) operation must be performed to read additional PLMN information. • 0x00=This is the last PLMN in the list. • 0x01=There are more PLMNs to be read.
03 (Set)	0			None
07 (Notify)	1	0	UINT8	Result of Set (03) operation to initiate network search for available PLMNs: • 0x00=No PLMNs are available • 0x01=PLMN list is available (use Get (01) operations to read each PLMN) • 0x02–0xFF=Reserved

Getting the list of available PLMNs

To read the list of currently available PLMNs (as shown in Figure 13-1):

1. Use Enable (05) object to support notifications.
2. Send Set (03) (and receive response).
   This initiates a network search for available PLMNs (this may take up to 15 seconds).
3. Receive Notify (07) when network search is completed.
4. If the More PLMNs flag indicates PLMNs are waiting to be read, then send Get (01).
5. Receive Response (02).
6. Repeat steps 4–5 until all PLMNs are read.

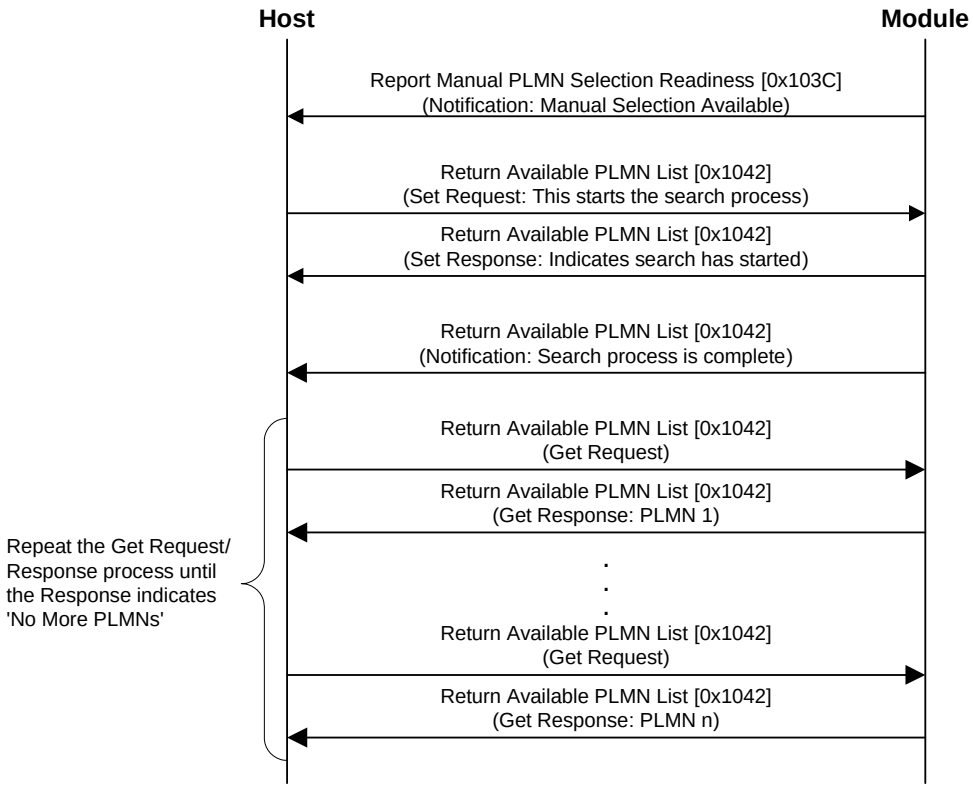

Figure 13-1: Message flow—Read PLMN list

Selecting an available PLMN

To select one of the available PLMNs, use Report Manual PLMN Selection Readiness (page 150), specifying the MCC and MNC of the selected PLMN (as shown in Figure 13-2).

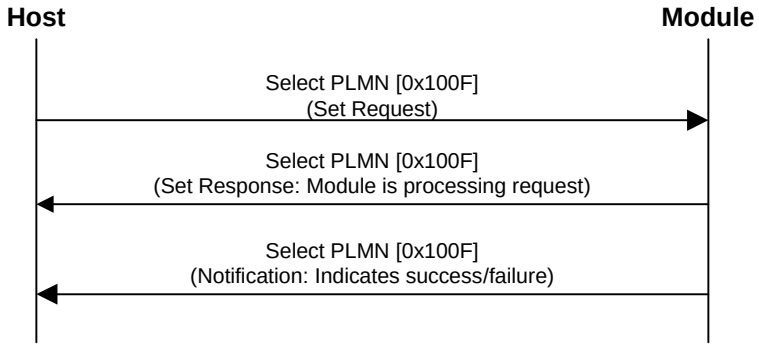

Figure 13-2: Message flow—Select a PLMN

Report PLMN Registration Information

Reports (given the present network status) the current PLMN name (if any) and service provider name (if any) that should be shown to the user:

- On request, using Get (01)
- When triggered, if Notify (07) is enabled *and* the modem registers on a new PLMN
  (This is an update-type notification.)

Note: If the SPN (Service Provider Name) is not immediately available, a second Notify (07) occurs when the SPN data is ready.

Table 13-15: Object identification—Report PLMN Registration Information

Object ID	0x1043		Report PLMN Registration Information	
Operations	✗ Set ✔ Get ✔ Notify			
	Parameters (fields)			
	Len	**Offset**	**Type**	**Description**
01 (Get)	0			None
02 (Get Response) **07 (Notify)**	125	0	UINT8	Unused
		1	UINT8	Length of service provider name • Min—0; Max—16
		2	UINT8[32]	Service provider name • UCS2 format • 32 bytes • Up to 16 characters (2 bytes per character)—See previous parameter for actual length • Ignore unused trailing bytes
		34	UINT8	Length of full PLMN name • Min—0; Max—31
		35	UINT8[62]	Full PLMN name • UCS2 format • 62 bytes • Up to 31 characters (2 bytes per character)—See previous parameter for actual number • Ignore unused trailing bytes
		97	UINT8	Length of abbreviated PLMN name • Min—0; Max—13

Table 13-15: Object identification—Report PLMN Registration Information

		98	UINT8[26]	Abbreviated PLMN name • UCS2 format • 26 bytes • Up to 13 characters (2 bytes per character)—See previous parameter for actual number • Ignore unused trailing bytes
		124	UINT8	Unused

Displaying service provider and PLMN names

Service provider and PLMN names should be displayed:

- Only when the modem is attached to a network.
  Use Report Network Status (page 141)—the modem is attached if Service Status = “normal” and Service Type is not “invalid service”.
- Using only the information received in the last Report PLMN Registration Information (page 154) object.

Missing service provider name

If the modem registers on a PLMN *before* the firmware SIM finishes initialization, then the initial notification does not include the service provider name.

When the SPN becomes available, a second Notify (07) occurs.

Report Roaming Icon Display Method

Returns a flag indicating how the roaming icon should be displayed, based on the home service provider’s requirements:

- When triggered, if Notify (07) is enabled *and* the display status changes (This is an update-type notification.)

Controlling the icon

This flag allows individual service providers to indicate if, and how, roaming icons are displayed while registered on a given PLMN.

*Note: Do **NOT** use the Report Network Status (page 141) network status flag for the roaming indicator—it only indicates if the modem has roaming service.*

Table 13-16: Object identification—Return Roaming Icon Display Method

Object ID	0x106A		Return Roaming Icon Display Method	
Operations	✗ Set ✗ Get ✔ Notify			
	Parameters (fields)			
	Len	**Offset**	**Type**	**Description**
07 (Notify)	1	0	UINT8	Roaming icon display status: • 0x00=Grey out (inactivate) the icon • 0x01=Activate the icon • 0x02=Don't display the icon • 0x03–0xFF=Undefined

Change Current Band (Deprecated)

Deprecated: *This command is deprecated and should not be used in new designs. For new designs, refer to Set Current Band on page 139.*

Switch the modem to a new band:

- On request, using Set (03) and Notify (07)

Switching bands

1. Use Report Current Band (Deprecated) (page 146) to get the list of bands supported by the modem.
2. Send Set (03) (and receive response), indicating the band that you want to use.
3. Receive Notify (07) indicating if the band was changed.

Table 13-17: Object identification—Change Current Band (Deprecated)

Object ID	0x106B		Change Current Band (Deprecated)	
Operations	✔ Set ✗ Get ✔ Notify			
	Parameters (fields)			
	Len	**Offset**	**Type**	**Description**
03 (Set)	2	0	UINT16	Requested band: • 0x0000=None / invalid • 0x0001=PGSM • 0x0002=EGSM • 0x0003=DCS Continued on next page

Table 13-17: Object identification — Change Current Band (Deprecated) (Continued)

				Requested band (continued) • 0x0004=PGSM DCS • 0x0005=EGSM DCS • 0x0006=PCS • 0x0007=PGSM PCS • 0x0008=EGSM PCS • 0x0009=850 MHz • 0x000A=850 MHz / PCS • 0x000B=GSM All • 0x000C=WCDMA 2100 EU • 0x000D=WCDMA 850/1900 NA • 0x000E=WCDMA 850/1900 GSM 850/1900 NA • 0x000F=WCDMA 2100 GSM 900/1800 EU • 0x0010=WCDMA All • 0x0011=WCDMA 2100/850 Australia • 0x0012=WCDMA 2100/800 Japan • 0x0013=WCDMA 2100/850 GSM 900/1800 • 0x0014=WCDMA 850 GSM 900/1800 • 0x0015=WCDMA 850 • 0x0016=WCDMA 2100 • 0x0017=WCDMA 2100/900 GSM 900/1800 • 0x0018=WCDMA 900 • 0x0019=WCDMA 1900 • 0x001A=WCDMA 2100/900 • 0x001B–0xFFFF=Reserved
07 (Notify)	1	0	UINT8	Changed to requested band successfully: • 0x00=No • 0x01=Yes

Manage Packet Session

Allows the host to initiate or terminate a connection with the MiniCard:

- On request, using Set (03). The Set Response (04) is returned as soon as the modem receives the request, followed by a Notify (07) when the request is completed (if notification is enabled).
- When triggered, if Notify (07) is enabled and the packet session status changes.
  (This is an update-type notification.)

The notification message includes a status code that the host can use to proceed with the connect/disconnect or perform appropriate error handling.

Table 13-18: Object—Manage Packet Session

Object ID	0x7004		Manage Packet Session	
Operations	✔ Set ✗ Get ✔ Notify			
	Parameters (fields)			
	Len	**Offset**	**Type**	**Description**
03 (Set)	2	0	UINT8	Profile ID • Min—1; Max—16
		1	UINT8	Requested action • 0x00=Deactivate the packet session • 0x01=Activate the packet session
04 (Set Response)	36	0	UINT8	Profile ID • Min—1; Max—16
		1	UINT8	Action that was requested • 0x00=Deactivate the packet session • 0x01=Activate the packet session
		2	UINT8	Status code • 0x00=Modem accepts the action • 0x01=Modem rejects the action
		3	UINT8[33]	Null-terminated error string Use these messates to aid in the process of debugging host software.
07 (Notify)	26	0	UINT8	Profile ID • Min—1; Max—16
		1	UINT8	Activation status • 0x00=Packet session is not active • 0x01=Packet session is active
		2	UINT8	Negotiated Quality of Service Valid flag • 0x00=The negotiated QoS below is not valid • 0x01=The negotiated QoS below is valid
		3	UINT8	Negotiated precedence • 0=Based on user's subscription • 1=High priority • 2=Normal priority • 3=Low priority

Table 13-18: Object — Manage Packet Session (Continued)

		4	UINT8	Negotiated delay • 0=Based on user's subscription • 1=Delay class 1 • 2=Delay class 2 • 3=Delay class 3 • 4=Delay class 4
		5	UINT8	Negotiated reliability • 0=Based on user's subscription • 1=Acknowledged GTP, LLC, and RLC; Protected data • 2=Unacknowledged GTP, Acknowledged LLC and RLC; Protected data • 3=Unacknowledged GTP and LLC, Acknowledged RLC; Protected data • 4=Unacknowledged GTP, LLC, RLC; Protected data • 5=Unacknowledged GTP, LLC, RLC; Unprotected data
		6	UINT8	Negotiated peak • 0=Based on user's subscription • 1=Up to 1000 octet/s • 2=Up to 2000 octet/s • 3=Up to 4000 octet/s • 4=Up to 8000 octet/s • 5=Up to 16000 octet/s • 6=Up to 32000 octet/s • 7=Up to 64000 octet/s • 8=Up to 128000 octet/s • 9=Up to 256000 octet/s
		7	UINT8	Negotiated mean • 0=Based on user's subscription • 1=100 octet/h • 2=200 octet/h • 3=500 octet/h • 4=1000 octet/h • 5=2000 octet/h • 6=5000 octet/h • 7=10000 octet/h • 8=20000 octet/h • 9 =50000 octet/h Continued on next page

Table 13-18: Object — Manage Packet Session (Continued)

				Negotiated mean (Continued) • 10=100000 octet/h • 11=200000 octet/h • 12=500000 octet/h • 13=1000000 octet/h • 14=2000000 octet/h • 15=5000000 octet/h • 16=10000000 octet/h • 17=20000000 octet/h • 18=50000000 octet/h • 31=Best effort
		8	UINT8	Negotiated Traffic Class • 0=Based on user's subscription • 1=Conversational • 2=Streaming • 3=Interactive • 4=Background
		9	UINT8	Negotiated Max Bitrate UL • 0=Based on user's subscription • 1=16Kbps • 2=32Kbps • 3=64Kbps • 4=128Kbps • 5=256Kbps • 6=384Kbps • 7=1.8Mbps • 8=3.6Mbps • 9=7.2Mbps • 10=14.4Mbps • 11=2.048 Mbps • 12=5.76 Mbps • 13=11.5 Mbps • 14=16 Mbps • 15=21 Mbps • 16=28 Mbps • 17=42 Mbps • 18=84 Mbps
		10	UINT8	Negotiated Max Bitrate DL • Same range as Negotiated Max Bitrate UL (offset 9)

Table 13-18: Object — Manage Packet Session (Continued)

		11	UINT8	Negotiated Guaranteed Bitrate UL • Same range as Negotiated Max Bitrate UL (offset 9)
		12	UINT8	Negotiated Guaranteed Bitrate DL • Same range as Negotiated Max Bitrate UL (offset 9)
		13	UINT8	Negotiated Delivery Order • 0=Based on user's subscription • 1=With delivery order • 2=Without delivery order
		14	UINT8	Negotiated Delivery of Erroneous SDU • 0=Based on user's subscription • 1=No detect • 2=Erroneous SDUs are delivered • 3=Erroneous SDUs are not delivered
		15	UINT16	Negotiated Maximum SDU Size • 0=Based on user's subscription • 1–1520=SDU size in bytes
		17	UINT8	Negotiated SDU error ratio • 0=Based on user's subscription • 1=1E-2 • 2=7E-3 • 3=1E-3 • 4=1E-4 • 5=1E-5 • 6=1E-6 • 7=1E-1
		18	UINT8	Negotiated Residual Bit error ratio • 0=Based on user's subscription • 1=5E-2 • 2=1E-2 • 3=5E-3 • 4=4E-3 • 5=1E-3 • 6=1E-4 • 7=1E-5 • 8=1E-6 • 9=6E-8
		19	UINT16	Negotiated Transfer Delay • 0=Based on user's subscription • 1–4000=Transfer delay in ms

Table 13-18: Object — Manage Packet Session (Continued)

		21	UINT8	Negotiated Traffic Handling priority • 0=Based on user's subscription • 1=Priority level 1 • 2=Priority level 2 • 3=Priority level 3
		22	UINT8	Negotiated Source Statistics Descriptor • 0=Unknown • 1=Speech
		23	UINT8	Negotiated Signalling Indication • 0=Not optimized for signalling traffic • 1=Optimized for signalling traffic
		24	UINT16	Inactivity Reason Code—This code is used only when the "Activation status" field above is set to 0x00. This code defines the reason why the packet session has become inactive. External cause from the network (refer to [3GPP TS 24.008 v5.12.0]): • 0x0008=Operator Determined Barring • 0x0019=LLC or SNDCP failure • 0x001A=Insufficient resources • 0x001B=Missing or unknown APN • 0x001C=Unknown PDP address or type • 0x001D=User authentication failed • 0x001E=Activation rejected by GGSN • 0x0020=Service Option not supported • 0x0021=Service Option not subscribed • x00022=Service Option temporarily out of order • 0x0024=Regular deactivation • 0x0025=QOS not accepted • 0x0026=Network failure • 0x0027=Reactivation required • 0x0028=Feature not supported • 0x0029=Semantic error in the TFT operation • 0x002A=Syntactical error in the TFT operation • 0x002B=Unknown PDP context • 0x002C=PDP context without TFT already activated • 0x002D=Semantic errors in packet filter(s) • 0x002E=Syntactical errors in packet filter(s) • 0x0051=Invalid transaction identifier value • 0x005F=Semantically incorrect message Continued on next pagej

Table 13-18: Object — Manage Packet Session (Continued)

				Inactivity reason code (continued) • 0x0060=Invalid mandatory information • 0x0061=Message type non-existent or not implemented • 0x0062=Message type not compatible with the protocol state • 0x0063=Information element non-existent or not implemented • 0x0064=Conditional IE error • 0x0065=Message not compatible with the protocol state • 0x006F=Protocol error, unspecified Internal cause from the modem: • 0x0100=Invalid Connection ID • 0x0101=Invalid NSAPI • 0x0102=Invalid PRI NSAPI • 0x0103=Invalid field • 0x0104=SNDCP failure • 0x0105=RAB Setup failure • 0x0106=No GPRS Context • 0x0107=PDP Establish Max Timeout • 0x0108=PDP Activate Max Timeout • 0x0109=PDP Modify Max Timeout • 0x010A=PDP Inactivate Max Timeout • 0x010B=PDP Lower layer Error • 0x010C=PDP Duplicate • 0x010D=UE RAT Change • 0x7000=Normal inactivate state (default value when there has never been a prior active packet session) All other values are unspecified and are reserved for future use.

Return Packet Session Status

Returns the current packet session status for a specific profile:

- On request, using Set (03). The session status information is returned in the Set Response (04).

Table 13-19: Object—Return Packet Session Status

Object ID	0x7005		Return Packet Session Status	
Operations	✔ Set ✗ Get ✗ Notify			
	Parameters (fields)			
	Len	**Offset**	**Type**	**Description**
03 (Set)	1	0	UINT8	Profile ID • Min—1; Max—16
04 (Set Response)	The Set Response uses the same format as the Notify in Manage Packet Session (page 157).			

Return IP Address

Returns the IP address for a specific profile:

- On request, using Set (03). The IP information is returned in the Set Response (04).

The MC8xxx modem supports both dynamic as well as static IP addresses. After activating a packet session, this object can be used to obtain the IP address, if desired.

Table 13-20: Object—Return IP Address

Object ID	0x7006		Return IP Address	
Operations	✔ Set ✗ Get ✗ Notify			
	Parameters (fields)			
	Len	**Offset**	**Type**	**Description**
03 (Set)	1	0	UINT8	Profile ID • Min—1; Max—16
04 (Set Response)	18	0	UINT8	Profile ID • Min—1; Max—16
		1	UINT8	Length of IP address • 0=No IP address assigned • 4=IPv4 • 16=IPv6

Table 13-20: Object—Return IP Address (Continued)

		2	UINT8[16]	IP address for the specified profile Examples: • IPv4 (4 bytes)—0x7F000201 (127.0.2.1) • IPv6 (16 bytes)—0x000020010DB8000000000202B3FF8329 (2001:DB8: : :0202:B3FF:8329)

Report System and Network Status

Reports the modem's current system and network status:

- On request, using Get (01)
- When triggered, if Notify (07) is enabled *and* the network status changes (This is an update-type notification.)

Note: Due to stack implementation requirements, this object reports WCDMA850 for both WCDMA850 and WCDMA800. Similarly, the AT!GETBAND command reports WCDMA800 for both bands.

Table 13-21: Object—Report System and Network Status

Object ID	**0x700A**		Report System and Network Status		
Operations	✗ Set ✔ Get ✔ Notify				
	Parameters (fields)				
	Len	**Offset**	**Type**	**Description**	**Object Ver.**
01 (Get)	0			None	n/a
02 (Get Response) **07 (Notify)**	See Object version number (Offset = 0) description for object length				
		0	UINT16	Object version number • 0x0001=Revision 1 Length = 52 • 0x0002=Revision 2 Length = 54	1–2
		2	UINT16	Modem status • 0x0000=Offline • 0x0001=Online • 0x0002=Low Power Mode (LPM)	1–2
		4	UINT16	Service status • 0x0000=Service available • 0x0001=Emergency / limited service • 0x0002=No service	1–2

Table 13-21: Object — Report System and Network Status (Continued)

		6	UINT16	Service error • 0x0000=No error • 0xFFFF=Unspecified protocol error	1–2
		8	UINT8	Service type • 0x00=Circuit-switched service • 0x01=GPRS service • 0x02=Combined service • 0x03=Invalid service	1–2
		9	UINT8	System mode • 0x00=None • 0x01=GPRS • 0x02=EDGE • 0x03=UMTS • 0x04=HSDPA • 0x05=HSUPA • 0x06=HSPA (HSDPA + HSUPA) • 0x07=HSPA+ • 0x08=DC-HSPA+	1–2
		10	UINT16	Current band: • 0x0000=No service • 0x0001=GSM 850 • 0x0002=GSM 900 • 0x0003=GSM 1800 • 0x0004=GSM 1900 • 0x0101=WCDMA 2100 • 0x0102=WCDMA 1900 • 0x0103=WCDMA 850 • 0x0104=WCDMA 800 • 0x0105=WCDMA 1800 • 0x0106=WCDMA 1700 (US) • 0x0107=WCDMA 2600 • 0x0108=WCDMA 900 • 0x0109=WCDMA 1700 (Japan) • 0x010A–0xFFFE=Reserved • 0xFFFF=Invalid band	1–2
		12	UINT8	Roaming status • 0x00=Not roaming • 0x01=Roaming	1–2

Table 13-21: Object—Report System and Network Status (Continued)

		13	UINT8	Manual mode: • 0x00=Automatic network selection • 0x01=Manual network selection	1–2
		14	UINT8	Length of country string (excluding trailing spaces)	1–2
		15	UINT8[6]	Country string • Up to 3 characters (2 bytes / character)—See previous parameter for actual length • Left justified, blank filled	1–2
		21	UINT8	Length of network string (excluding trailing spaces)	1–2
		22	UINT8[16]	Network string • Up to 8 characters, not null-terminated (see previous parameter for length) • Left justified, blank filled	1–2
		38	UINT16	MCC (Mobile Country Code)	1–2
		40	UINT16	MNC (Mobile Network Code)	1–2
		42	UINT16	LAC (Location Area Code)	1–2
		44	UINT16	RAC (Routing Area Code)	1–2
		46	UINT16	Cell ID • Valid values: Any 16-bit number • If object version number = 3, use the Cell ID at offset 54 (holds larger Cell ID values)	1–2
		48	UINT16	Channel number ((U)ARFCN) • Valid values are dependent on current band	1–2
		50	UINT16	Primary scrambling code • WCDMA—Valid values:1–512 • GSM—Always 0	1–2
		Remaining fields are for object version = 2.			
		52	UINT8	MNC format • 0x00=2-digit • 0x01=3-digit	2
		53	UINT8	PLMN service error bit mask • Bit 0=Manual PLMN acquisition timed out • Bits 1–7=Reserved	2

Set RAT mode

Reports or sets the device's current RAT (Radio Access Technology) mode:

- On request, using Get (01) to report the mode
- On request, using Set (03) to assign the mode

Table 13-22: Object—Set RAT mode

Object ID	0x700F		Set RAT mode	
Operations	✔ Set ✔ Get ✗ Notify			
	Parameters (fields)			
	Len	**Offset**	**Type**	**Description**
01 (Get)	2	0	UINT16	Object version number • 0x0001=Revision 1
02 (Get Response) **03 (Set)**	3	0	UINT16	Object version number • 0x0001=Revision 1
		2	UINT8	RAT mode: • 0x00=Automatic • 0x01=UMTS 3G only • 0x02=GSM 2G only • 0x03=UMTS 3G preferred • 0x04=GSM 2G preferred
04 (Set Response)	3	0	UINT16	Object version number • 0x0001=Revision 1
		2	UINT8	Action response: • 0x00=Mode changed succeeded • 0x01=General failure

Manage call byte counters (Rx/Tx)

(MSM6290-based and newer devices only)

Note: This command replaces Report call byte counters (Rx/Tx) (page 149) and Reset call byte counters (page 150).

Report or reset, for the primary data session on a specified profile, the total number of bytes received and transmitted since the last time the totals were cleared:

- On request, using Get (01) to report, or Set (03) to reset
- Receive periodic reports, if Notify (07) is enabled

Table 13-23: Object—Manage call byte counters

Object ID	0x7012		Manage call byte counters	
Operations	✔ Set ✔ Get ✔ Notify			
	Parameters (fields)			
	Len	**Offset**	**Type**	**Description**
01 (Get)	3	0	UINT16	Current object version number • 0x0001=Revision 1
		2	UINT8	Profile ID • Min—1; Max—16
02 (Get Response) **07 (Notify)**	19	0	UINT16	Current object version number • 0x0001=Revision 1
		2	UINT8	Profile ID • Min—1; Max—16
		3	UINT64	Number of bytes transmitted
		11	UINT64	Number of bytes received
03 (Set) **04 (Set Response)**	3	0	UINT16	Current object version number • 0x0001=Revision 1
		2	UINT8	Profile ID for which byte counters are being reset • Min—1; Max—16

Managing call byte counters

To use the call byte counters to track the number of bytes transmitted and received during a call:

1. Enable the Manage call byte counters notification for all profile IDs. (This is optional—notifications will only be received if this is enabled.)
2. If desired, use the Set(03) operation to reset the counters for a specific profile (Profile ‘X’). (Note: This feature is not intended to be used to track lifetime byte counts.)
3. Use the Manage Packet Session (page 157) Set(03) operation to activate a data connection for Profile ‘X’.
4. Any time after a data connection is established, use the Get(01) operation to obtain the total byte counts (transmitted and received) for the Profile ‘X’ data session, or (if enabled) wait for periodic notification reporting the values. (Notification period may vary by device. Some devices report the values on a continuous basis, while others report when an internally-defined threshold is reached.)

14: Phonebook Management

Introduction

This chapter describes CnS messages that are related to phonebooks, including:

- Indicating which phonebooks are available, and which operations are available on those books
- Indicating the current size of each phonebook
- Reading, adding, editing, and deleting phone numbers

Note: This chapter contains information that applies to both voice- and non-voice-enabled modems. Items which relate only to voice-enabled modems are identified appropriately.

Phonebook messages summary

Table 14-1: Phonebook management

Object	ID	Description	S	G	N
Delete 'last' numbers (page 174)	0x1044	Deletes all of the numbers stored in one of the 'last number' phonebooks (last number dialed / missed / received). (Voice modems only.)	✔	✗	✗
Read phone number (page 175)	0x1045	Returns the phone number entries that are stored in a supported phonebook.	✔	✗	✔
Add phone number (page 180)	0x1046	Add a phone number to a supported phonebook.	✔	✗	✗
Edit Phone Numbers (page 182)	0x1047	Edit an existing phone number in a supported phonebook.	✔	✗	✗
Return Phonebook Services Status (page 184)	0x1048	Returns the current availability of phonebook services.	✗	✔	✔
Return Phonebook Availability (page 184)	0x1049	Returns the current availability of each supported phonebook.	✗	✔	✔
Return Phonebook Size Details (page 186)	0x104A	Returns the size and total space available for each supported phonebook.	✗	✔	✗
Return Emergency Numbers (page 187)	0x1072	Returns supported emergency numbers.	✔	✗	✗
Enable/Disable FDN Mode (page 188)	0x1079	Sets the Fixed Dialing Mode.	✔	✔	✗
Delete Phone Number (page 189)	0x107A	Delete a phone number from a supported phonebook.	✔	✗	✗

Phonebook messages reference

Phonebook Management Overview

Supported phonebooks

Several phonebooks are supported, either stored on the SIM or in the modem (in NVRAM). It is important to note that, because SIMs can be moved between devices, the phonebooks that are stored in the modem will NOT move from one device to another.

The number of entries stored in each phonebook is carrier-dependent.

Table 14-2: Supported phonebooks

Abbreviation	Name	Description	Storage	Supported Actions
ADN	Abbreviated Dialing Numbers	Stores numbers for placing phone calls or sending SMS messages (this is the main phonebook). Up to 255 entries are typically stored.	SIM	Add Edit Delete
CPHS MN	Voice mailbox numbers	Stores voice mailbox numbers (typically up to four, depending on carrier provisioning) (Voice modems only)	SIM	Edit
FDN	Fixed Dialing Numbers	Stores specific phone numbers, exchanges, and so on. The user is limited to using only these numbers. Up to 100 entries are typically stored.	SIM	Add Edit Delete
LND	Last Numbers Dialed	Stores the most recent numbers dialed from the modem (typically 10). (Voice modems only)	SIM	Add Edit Delete
LNM	Last Numbers Missed	Stores the numbers of the most recent incoming calls that were missed (typically 10). (Voice modems only)	Modem	Delete (entire book)
LNR	Last Numbers Received	Stores the numbers of the most recent incoming calls that were answered (typically 10). (Voice modems only)	Modem	Delete (entire book)
MSISDN	Mobile Subscriber International Subscriber Identity Number	Stores the phone number(s) for the account. Some carriers pre-configure this on the SIM.	SIM	Edit
SDN	Service Dialing Numbers	Carrier-provisioned numbers (for example, billing enquiries, emergency numbers, etc.)	SIM	View only

Note: The phone numbers in most books are unsorted ('last number' books may be time-sorted). It is the responsibility of the host application to sort entries for display to the user (if desired).

Phone number format

Phone numbers are saved using standard and special characters:

- '0'–'9'
- '*'
- '#'
- ',' (first occurrence in phone number)
  The first comma in a telephone number is used to support overdial numbers.These are numbers that are dialed *after* a connection is established.
- ',' (subsequent appearances in phone number)
  Each comma (after the first comma in the phone number) inserts a three-second pause between overdial numbers.
- '?'
  This is a 'wildcard' character that indicates a missing digit that must be entered by the user—the host software is responsible for implementing this, if the feature is desired.

Example: Phone number strings

Phonebook entry: **6045551234,,1111,2**

When dialing this phone number, the modem will:

1. Connect to the phone number 6045551234.
2. Pause for three seconds—the first comma indicates that overdial numbers are being used; the second comma indicates the pause.
3. Dial 1111.
4. Pause again for three seconds (the third comma).
5. Dial 2.

Note: Overdial numbers are numbers that are dialed after a connection is established (for example, to access an extension number). They are separated from the main phone number by commas (three-second pauses).

FDN phonebook entries

The FDN phonebook is a feature that allows the host application to restrict the phone numbers that can be dialed and to which SMS messages can be sent.

When the FDN is enabled (see Enable/Disable FDN Mode (page 188)), only phone numbers that begin with the numbers in the FDN can be dialed.

*Note: The ADN phonebook on the SIM is **not** available when the FDN phonebook is active.*

Example: FDN phone numbers

If the FDN phonebook contains the following entries:

- **604**
- **2505558989**

then the user would only be able to dial 250-555-8989 and any phone numbers that begin with the area code 604.

Using the phonebook functions

The following objects are used to work with the supported phonebooks:

- Return Phonebook Services Status (page 184)
  This indicates that the phonebook service is available. This notification should be received before using any other phonebook objects.
- Return Phonebook Availability (page 184)
  This indicates which phonebooks are currently available.
- Delete ‘last’ numbers (page 174)
  This returns a list of all of the phone numbers in a specific phonebook. (Voice modems only)
- Return Phonebook Size Details (page 186)
  This indicates the current number of entries in all phonebooks.
- Read phone number (page 175)
  This reads a phone number tfrom a supported phonebook.
- Add phone number (page 180)
  This adds a phone number to the ADN or FDN phonebooks.
- Edit Phone Numbers (page 182)
  This edits a phone number in the ADN, FDN, or MSISDN phonebooks.
- Delete Phone Number (page 189)
  This deletes a number from the ADN or FDN phonebooks.

Delete ‘last’ numbers

(Voice modems only) Deletes all of the phone numbers stored in one of the ‘last number’ phonebooks (last number dialed / missed / received):

- On request, using Set (03).

Table 14-3: Object—Delete ‘last’ numbers

Object ID	0x1044		Delete ‘last’ numbers	
Operations	✔ Set ✗ Get ✗ Notify			
	Parameters (fields)			
	Len	**Offset**	**Type**	**Description**
03 (Set)	1	0	UINT8	Phonebook to delete all entries from • 0x02=LND (Last Number Dialed) • 0x04=LNM (Last Number Missed) • 0x05=LNR (Last Number Received)
04 (Set Response)	0			None

Read phone number

Note: Phonebooks stored in the modem are persistent—their contents are not cleared when the modem powers off / on.

Reads phone number entries from supported phonebooks stored on the module's SIM (for example, MSISDN) or in the modem (for example, LNR).

The phonebook entry is read:

- On request, using Set (03)
- Automatically (one time only), for the ADN phonebook, when Notify (07) is *first* enabled *and* the SIM has completed its initialization

Note: The LND, LNM, and LNR phonebooks depend on the network providing the dialing number. If the number is not provided, the phonebooks do not function.

Table 14-4: Object—Read phone number

Object ID	0x1045		Read phone number	
Operations	✔ Set ✗ Get ✔ Notify			
	Parameters (fields)			
	Len	**Offset**	**Type**	**Description**
03 (Set)	2	0	UINT8	Requested phonebook • 0x00=ADN (Abbreviated Dialing Numbers) • 0x01=Reserved for FDN (Fixed Dialing Numbers) • 0x02=LND (Last Numbers Dialed) (Voice modems only) • 0x03=MSISDN (Mobile Subscriber International Subscriber Identity Numbers) • 0x04=LNM (Last Numbers Missed) (Voice modems only) • 0x05=LNR (Last Numbers Received) (Voice modems only) • 0x06=SDN (Service Dialing Numbers) (Voice modems only) • 0x07=CPHS MN (CPHS Mailing Numbers) (Voice modems only)
		1	UINT8	Requested phonebook entry • 0x00=Read next entry (reads from start of phonebook if no other entries have been read) • 0x01=Read from start of phonebook *Note: If no entries have been read yet, Next entry (0x00) reads from the start of the requested phonebook.*

Table 14-4: Object—Read phone number (Continued)

04 (Set Response) **07 (Notify)**	107	0	UINT8	Selected phonebook • 0x00=ADN (Abbreviated Dialing Numbers) • 0x01=FDN (Fixed Dialing Numbers) • 0x02=LND (Last Numbers Dialed) (Voice modems only) • 0x03=MSISDN (Mobile Subscriber International Subscriber Identity Numbers) • 0x04=LNM (Last Numbers Missed) (Voice modems only) • 0x05=LNR (Last Numbers Received) (Voice modems only) • 0x06=SDN (Service Dialing Numbers) (Voice modems only) • 0x07=CPHS MN (CPHS Mailing Numbers) (Voice modems only)
		1	UINT8	Index number (position of phone number entry in the phonebook) • 1–255=Maximum number is dependent on the implementation of specific phonebooks on the module or on the SIM. *Note: Phone numbers are NOT stored in sorted order in the phonebook—the host application is responsible for sorting the numbers appropriately.*
		2	UINT8	Valid phonebook entry • 0x00=No phonebook entry present • 0x01=Valid data (phonebook entry present)
		3	UINT8	Phonebook entry name length (number of 2-byte UCS2 characters) • 0–30=Length of following field
		4	UINT8[60]	Phonebook entry name • UCS2 format • 60 bytes • Up to 30 characters (2 bytes/character)—See previous parameter for actual length • Ignore unused trailing bytes
		64	UINT8	International number flag • 0x00=Called number is national (no '+') • 0x01=Called number is international

Table 14-4: Object—Read phone number (Continued)

		65	UINT8	Phone number length • 0–40=Length of following field *Note: Each phonebook provides 20 characters for each phone number. As well, a limited number of 40 character extended phone numbers can be stored in each phonebook.*
		66	UINT8[40]	Number being called • 0–40 binary-coded decimal characters • Valid values: ◦ 0x00–0x09=0 to 9 ◦ 0x0A='*' ◦ 0x0C=',' (The first comma separates phone number digits and DTMF overdial digits. Additional commas indicate 3-second pauses before sending additional DTMF digits.) ◦ 0x0B='#' ◦ 0x0D='?' (Indicates a missing digit to be entered by the user)
		106	UINT8	More entries in phonebook • 0x00=No more entries • 0x01=More entries remaining to be read

Reading ADN phonebook using notification

Note: The ADN phonebook is not available when the FDN phonebook is enabled.

When notification is first enabled, the modem automatically issues a series of notifications to read the ADN phonebook.

1. Enable notification.
2. Receive the phonebook entries (as shown in Figure 14-1):
   a. Receive Notify (07) with the next entry from the ADN phonebook (automatically starting with the first entry).
   b. If the phonebook entry is valid, then store it.
   c. Repeat steps a–b until all entries have been read.

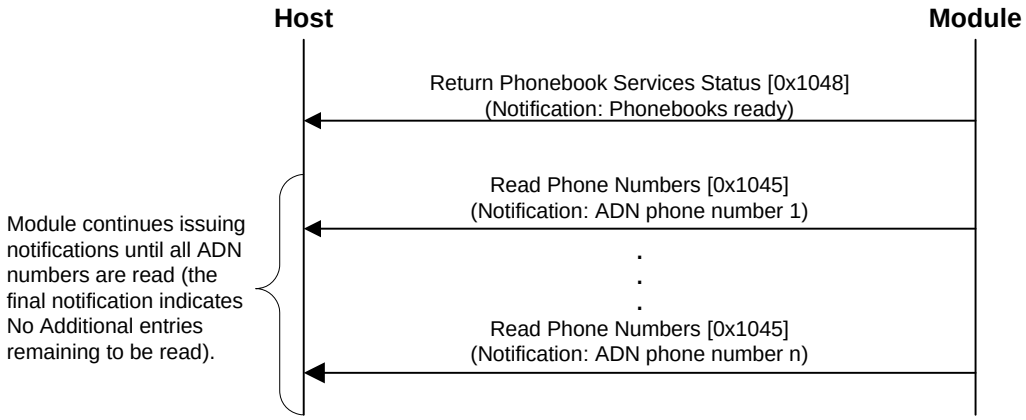

Figure 14-1: Message flow—Read ADN phonebook

Reading any phonebook using Set (03)

The contents of any of the supported phonebooks can be read at any time by using Set (03) and identifying the appropriate phonebook and entry to read (as shown in Figure 14-2 on page 179).

1. Issue a Set (03) operation, indicating the requested phonebook and the item to be read (First entry or Next entry).
2. Receive a Set Response (04) with the next entry from the requested phonebook.
3. If the phonebook entry is valid, then store it.
4. Repeat steps 1–3 until all entries have been read.

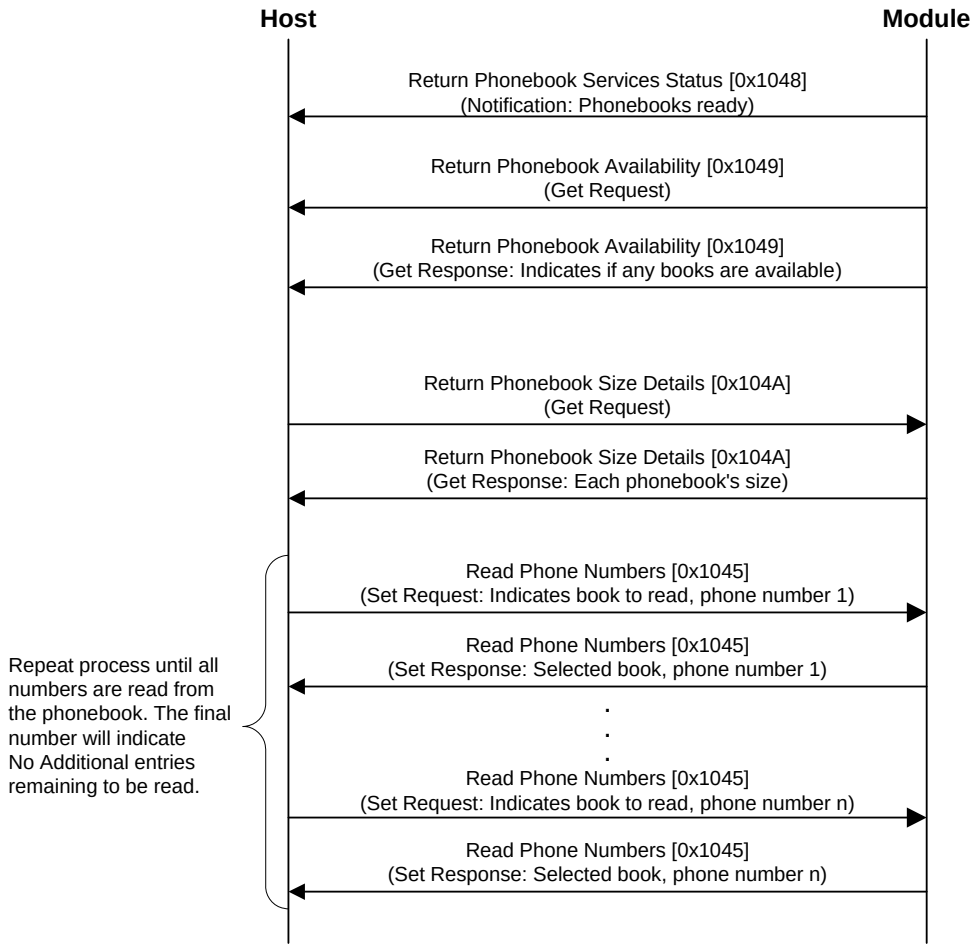

Figure 14-2: Message flow—Read ponebook

Suggested functionality:

When implementing support for phonebooks, consider providing functionality to:

- Search the phone numbers
- Sort the phone numbers
- Retrieve the contents of non-ADN phonebooks each time that they are used

Add phone number

Add a phone number to a supported phonebook stored on the module's SIM (for example, MSISDN):

- On request, using Set (03)

The number is added in the first available space within the requested phonebook.

Note: Because the SIM must be accessed to add a phone number, the Set Response may take longer to arrive at the host than other objects. The host software should take this into account when implementing a timeout period for this operation (10 seconds would be reasonable).

Note: The host device must implement a 10-second timeout period when using this object.

Note: Before a number can be added to the FDN phonebook, CHV2 must be validated. See Changing the current FDN mode (page 189) for the verification process (steps 1–5).

Table 14-5: Object—Add phone number

Object ID	0x1046		Add phone number	
Operations	✔ Set ✗ Get ✗ Notify			
	Parameters (fields)			
	Len	**Offset**	**Type**	**Description**
03 (Set)	104	0	UINT8	Requested phonebook • 0x00=ADN (Abbreviated Dialing Numbers) • 0x01=FDN (Fixed Dialing Numbers) • 0x03=MSISDN (Mobile Subscriber International Subscriber Identity Numbers) *Note: Phone numbers can only be added to ADN, FDN, and MSISDN—any other phonebook type is rejected.*
		1	UINT8	New entry name length (number of 2-byte UCS2 characters) • 0–30=Length of following field
		2	UINT8[60]	New entry name • UCS2 format • 60 bytes • Up to 30 characters (2 bytes/character)—See previous parameter for actual length • Ignore unused trailing bytes

Table 14-5: Object — Add phone number (Continued)

		62	UINT8	New entry number international flag • 0x00=New entry number is national (no '+') • 0x01=New entry number is international
		63	UINT8	New entry number length • 0–40=Length of following field *Note: Each phonebook provides 20 characters for each phone number—a limited number of 40 character extended phone numbers can be stored in each phonebook.*
		64	UINT8[40]	New entry number • 0–40 binary characters • Valid values: ◦ 0x00–0x09=0 to 9 ◦ 0x0A='*' ◦ 0x0C=',' (The first comma separates phone number digits and DTMF overdial digits. Additional commas indicate 3-second pauses before sending additional DTMF digits.) ◦ 0x0B='#' ◦ 0x0D='?' (Indicates a missing digit to be entered by the user) • Ignore unused trailing bytes
04 (Set Response)	2	0	UINT8	Index number (position of entry in the phonebook) • 0–255=Maximum number is dependent on the SIM's implementation of specific phonebooks. *Note: Phone numbers are not stored in sorted order in the phonebook—the host application is responsible for sorting the numbers appropriately.*
		1	UINT8	Return code • 0x00=Successful • 0x01=Failed: EXT1 not available • 0x02=Failed: Too many entries > 20 digits • 0x03=Failed: CHV2 validation error • 0x04=Failed: Entry does not exist • 0xFF=Failed: Other

Edit Phone Numbers

Edit a phonebook entry (name and number) in a supported phonebook stored in the module's SIM (for example, MSISDN) or in the modem (for example, LNR).

The phonebook entry is edited:

- On request, using Set (03)

Note: This object writes CPHSMN entries as absolute records on the SIM.

Note: Both the phonebook entry name and number can be edited. If only one parameter is edited, the other parameter must be included without any changes.

Note: The host device must implement a 10-second timeout period when using this object.

Note: Before a number can be edited in the FDN phonebook, CHV2 must be validated. See Changing the current FDN mode (page 189) for the verification process (steps 1–5).

Table 14-6: Object—Edit phone number

Object ID	0x1047		Edit phone number	
Operations	✔ Set ✗ Get ✗ Notify			
	Parameters (fields)			
	Len	**Offset**	**Type**	**Description**
03 (Set)	105	0	UINT8	Phonebook to update • 0x00=ADN (Abbreviated Dialing Numbers) • 0x01=Reserved for FDN (Fixed Dialing Numbers) • 0x03=MSISDN (Mobile Subscriber International Subscriber Identity Numbers) • 0x07=CPHS MN (CPHS Mailing Numbers) (Voice modems only) *Note: Phone numbers can only be edited in ADN, FDN, MSISDN, and CPHS MN—any other phonebook type is rejected.*
		1	UINT8	Index number (position of entry in the phonebook) • 0–255=Maximum number is dependent on the SIM's implementation of specific phonebooks.
		2	UINT8	Phonebook entry name length (number of 2-byte UCS2 characters) • 0–30=Length of following field

Table 14-6: Object—Edit phone number (Continued)

		3	CHAR[60]	Phonebook entry name • UCS2 format • 60 bytes • Up to 30 characters (2 bytes/character)—See previous parameter for actual length • Ignore unused trailing bytes *Note: This field is required—record the existing name if only the number is being changed.*
		63	UINT8	Phonebook entry number international flag • 0x00—Entry number is national (no '+') • 0x01—Entry number is international *Note: This field is required—record the existing flag if only the name is being changed.*
		64	UINT8	Phonebook entry number length • 0–40—Length of following field *Note: This field is required—record the length of the existing number if only the name is being changed.* *Note: Each phonebook provides 20 characters for each phone number. As well, a limited number of 40 character extended phone numbers can be stored in each phonebook.*
		65	UINT8[40]	Phonebook entry number • 0–40 binary characters • Valid values: ◦ 0x00–0x09=0 to 9 ◦ 0x0A='*' ◦ 0x0B='#' ◦ 0x0C=',' (The first comma separates phone number digits and DTMF overdial digits. Additional commas indicate 3-second pauses before sending additional DTMF digits.) ◦ 0x0D='?' (Indicates a missing digit to be entered by the user) *Note: This field is required—record the existing number if only the name is being changed.*

Table 14-6: Object—Edit phone number (Continued)

04 (Set Response)	1	0	UINT8	Return code • 0x00=Successful • 0x01=Failed—EXT1 not available • 0x02=Failed—Too many entries > 20 digits • 0x03=Failed—CHV2 validation error • 0x04=Failed—Entry does not exist • 0xFF=Failed—Other

Return Phonebook Services Status

Returns the status (availability) of phonebook services:

- On request, using Get (01)
- When triggered, if Notify (07) is enabled *and* the status changes (This is an update-type notification.)

Table 14-7: Object—Return Phonebook Services Status

Object ID	0x1048		Return Phonebook Services Status	
Operations	✗ Set ✔ Get ✔ Notify			
	Parameters (fields)			
	Len	**Offset**	**Type**	**Description**
01 (Get)	0			None
02 (Get Response) **07 (Notify)**	1	0	UINT8	Phonebook service status Indicates if services (including read, edit, and add) are available for any of the phonebooks. • 0x00=Unavailable • 0x01=Available

Return Phonebook Availability

Returns the current availability status of all supported SIM phonebooks:

- On request, using Get (01)
- When triggered, if Notify (07) is enabled *and* the status changes (This is an update-type notification.)

Table 14-8: Object—Return Phonebook Availability

Object ID	0x1049		Return Phonebook Availability	
Operations	✗ Set ✔ Get ✔ Notify			
	Parameters (fields)			
	Len	**Offset**	**Type**	**Description**
01 (Get)	0			None

Table 14-8: Object — Return Phonebook Availability (Continued)

02 (Get Response) **07 (Notify)**	10	0	UINT8	ADN phonebook availability • 0x00=Unavailable • 0x01=Available
		1	UINT8	FDN phonebook availability • 0x00=Unavailable • 0x01=Available
		2	UINT8	LND phonebook availability • 0x00=Unavailable • 0x01=Available
		3	UINT8	MSISDN phonebook availability • 0x00=Unavailable • 0x01=Available
		4	UINT8	LNM phonebook availability • 0x00=Unavailable • 0x01=Available
		5	UINT8	LNR phonebook availability • 0x00=Unavailable • 0x01=Available
		6	UINT8	SDN phonebook availability • 0x00=Unavailable • 0x01=Available
		7	UINT8	CPHS MN phonebook availability • 0x00=Unavailable • 0x01=Available
		8	UINT8[2]	Reserved • 2 bytes

Using the availability flag

These flags indicate, for each supported phonebook, if the phonebook can be used in some way:

- Read (Read phone number (page 175))
- Written (Add phone number (page 180) and Edit Phone Numbers (page 182))
- Deleted (Delete Phone Number (page 189)).

The actual actions available for each phonebook depend on the individual phonebook and must be determined by the host software.

Return Phonebook Size Details

Returns the number of free entries and the size of each supported phonebook

- On request, using Get (01)

Note: The maximum numbers of entries shown for ADN and FDN are carrier-dependent. The numbers shown are the typical maximums.

Table 14-9: Object—Return Phonebook Size Details

Object ID	0x104A		Return Phonebook Size Details	
Operations	✗ Set ✔ Get ✗ Notify			
	Parameters (fields)			
	Len	**Offset**	**Type**	**Description**
01 (Get)	0			None
02 (Get Response)	20	0	UINT8	ADN phonebook—Number of free entries • Min—0 (full); Max—255 (empty)
		1	UINT8	ADN phonebook—Total number of entries • Min—0 (empty); Max—255 (full)
		2	UINT8	FDN phonebook—Number of free entries • Min—0 (full); Max—100 (empty)
		3	UINT8	FDN phonebook—Total number of entries • Min—0 (empty); Max—100 (full)
		4	UINT8	LND phonebook—Number of free entries • Min—0 (full); Max—10 (empty)
		5	UINT8	LND phonebook—Total number of entries • Min—0 (empty); Max—10 (full)
		6	UINT8	MSISDN phonebook—Number of free entries • Min—0 (full); Max—Carrier-dependent (empty)
		7	UINT8	MSISDN phonebook—Total number of entries • Min—0 (empty); Max—Carrier-dependent (full)
		8	UINT8	LNM phonebook—Number of free entries • Min—0 (full); Max—10 (empty)
		9	UINT8	LNM phonebook—Total number of entries • Min—0 (empty); Max—10 (full)
		10	UINT8	LNR phonebook—Number of free entries • Min—0 (full); Max—10 (empty)
		11	UINT8	LNR phonebook—Total number of entries • Min—0 (empty); Max—10 (full)

Table 14-9: Object—Return Phonebook Size Details (Continued)

		12	UINT8	SDN phonebook—Number of free entries • Min—0 (full); Max—100(empty)
		13	UINT8	SDN phonebook—Total number of entries • Min—0 (empty); Max—100 (full)
		14	UINT8	CPHS MN phonebook—Number of free entries • Min—0 (full); Max—4 (empty)
		15	UINT8	CPHS MN phonebook—Total number of entries • Min—0 (empty); Max—4 (full)
		16	UINT8[4]	Reserved

Identifying unavailable phonebooks

If any phonebooks are unavailable, each of the values for those books is zero.

For example, the FDN phonebook may not be available because CHV2 has not been entered properly.

Return Emergency Numbers

Returns all supported emergency numbers (up to six digits in length) through a series of Set request / response pairs.

Returns:

- On request,the first number using Set (03), specifying ‘Read first’ (see parameters below). The response indicates if additional numbers are available.
- On request, the next number using Set (03), specifying ‘Read next’ (see parameters below). The response indicates if additional numbers are available.

Table 14-10: Object—Return Emergency Numbers

Object ID	0x1072		Return Emergency Numbers	
Operations	✔ Set ✗ Get ✗ Notify			
	Parameters (fields)			
	Len	**Offset**	**Type**	**Description**
03 (Set)	1	0	UINT8	Emergency number to read • 0x00=Read first number • 0x01=Read next number (reads first number if no other numbers have been read)
04 (Set Response)	8	0	UINT8	Emergency number length • Min—0; Max—8

Table 14-10: Object—Return Emergency Numbers (Continued)

		1	UINT8[6]	Emergency number (up to 6 digits) • One byte per digit • Binary format (0x00–0x09 corresponds to '0'–'9')
		7	UINT8	Additional numbers remaining to be read • 0x00=No more numbers remain • 0x01=More numbers remain

Enable/Disable FDN Mode

Returns or sets the Fixed Dialing Mode (if the current SIM has FDN activated).

Returns:

- On request, using Get (01)

Sets:

- On request, using Set (03)

Table 14-11: Object—Enable/Disable FDN Mode

Object ID	0x1079		Enable/Disable FDN Mode	
Operations	✔ Set ✔ Get ✗ Notify			
	Parameters (fields)			
	Len	**Offset**	**Type**	**Description**
01 (Get)	0			None
02 (Get Response)	1	0	UINT8	Current FDN mode: • 0x00=Disabled • 0x01=Enabled
03 (Set)	1	0	UINT8	Requested change to FDN mode • 0x00=Disable • 0x01=Enable
04 (Set Response)	1	0	UINT8	Result of Set (03) operation • 0x00=Successful • 0x01=Failed—CHV2 required • 0x02=Failed • 0x03–0xFF=Reserved

Changing the current FDN mode

Note: The same verification procedure is used when any attempt is made to add, edit, or delete entries from the FDN phonebook.

Before allowing access to the FDN mode, CHV2 must be successfully verified:

1. Issue a CHV2 Status Kick (page 111).
2. Receive a Return SIM Status (page 108) notification, requiring the user to enter the CHV2 code.
3. Require the user to enter the CHV2 code.
4. Issue a Verify CHV Code (page 104) with the user's input.
5. Receive a Return SIM Status (page 108) notification, indicating if the CHV2 code was entered correctly. Repeat steps 3–5 if the wrong code was entered.
6. Issue the Enable/Disable FDN Mode (page 188) after the CHV2 code is verified.
7. Receive the Enable/Disable FDN Mode (page 188) notification indicating the result of the Set (03) operation.

Delete Phone Number

Delete a phone number from a supported phonebook stored in the module's SIM or in the modem:

- On request, using Set (03).

Table 14-12: Object identification—Delete phone number

Object ID	0x107A		Delete phone number	
Operations	✔ Set ✗ Get ✗ Notify			
	Parameters (fields)			
	Len	**Offset**	**Type**	**Description**
03 (Set)	2	0	UINT8	Phonebook to update • 0x00=ADN (Abbreviated Dialing Numbers) • 0x01=FDN (Fixed Dialing Numbers) • 0x03=MSISDN (Mobile Subscriber International Subscriber Identity Numbers) • 0x07=CPHS MN (CPHS Mailing Numbers) *Note: Phone numbers can only be deleted from ADN, FDN, MSISDN, and CPHS MN—any other phonebook type is rejected.*

Table 14-12: Object identification—Delete phone number (Continued)

		1	UINT8	Index number (position of entry in the phonebook) • 0–255=Maximum number is dependent on the SIM's implementation of specific phonebooks.
04 (Set Response)	1	0	UINT8	Return code • 0x00=Successful • 0x01=Reserved • 0x02=Reserved • 0x03=Failed—CHV2 validation error • 0x04=Failed—Entry does not exist • 0x05=Failed—General error • 0xFF=Failed—Other

Note: Before a number can be deleted from the FDN phonebook, CHV2 must be validated. See Changing the current FDN mode (page 189) for the verification process (steps 1–5).

15: SMS Messages

Introduction

This chapter provides reference for CnS messages that are related to SMS messages.

Terminology

To maintain clarity, the following terminology must be precisely understood:

- SMS message—The entire SMS message composed of an SMS header and SMS body
- SMS header—The header portion of an SMS message
- SMS body—The text body portion of an SMS message
- SMS packet—The Parameter portion of a CnS message that carries some or all of an SMS message
- SMS envelope—The first six bytes of an SMS packet. It contains information about the packet type, segment management, and length of the SMS payload.
- SMS payload—The portion of an SMS packet following the SMS envelope

SMS messages summary

Table 15-1: SMS Message Handling Objects

Object	ID	Description	S	G	N
Report SMS Received Messages Status (page 208)	0x1020	Reports the number of mobile-terminated SMS messages that are stored on the SIM	✗	✔	✔
Get Unread SMS Message (page 209)	0x1021	Read an unread mobile-terminated message that has been delivered by the network to the SIM	✗	✔	✗
Report SMS Availability (page 211)	0x1022	Reports the current availability of SMS service	✗	✔	✔
Get Read SMS Message (page 211)	0x1023	Read a previously-read mobile-terminated message from the SIM	✗	✔	✗
Return SMS Outgoing Message Number (page 212)	0x1027	Returns a unique reference number that is assigned to a mobile-originated SMS message	✗	✗	✔
Report Outgoing SMS Message Status (page 213)	0x1028	Reports on the delivery status of a mobile-originated SMS message	✗	✗	✔

Table 15-1: SMS Message Handling Objects (Continued)

Object	ID	Description	S	G	N
Delete Mobile-terminated SMS Message (page 213)	0x1029	Deletes a mobile-terminated message from the SIM	✔	✗	7✗
Report SIM SMS Message Storage Status (page 214)	0x102A	Reports the current status of the SIM's message storage capacity	✗	✗	✔
Return SIM SMS Configuration Details (page 214)	0x102B	Returns the current settings for SMS-related parameters on the SIM	✔	✔	✗
Copy Mobile-originated SMS Message to SIM (page 220)	0x1087	Download a mobile-originated SMS message from the host to the SIM	✔	✗	✔
Send Mobile-originated SMS Message To Network (page 221)	0x1088	Sends the most recently downloaded mobile-originated SMS message to the network	✔	✗	✔
Manage SMS Status Reports (page 222)	0x700D	Indicate state of, and set state of (if allowed), SMS Status Reports	✔	✔	✗

SMS overview

Starting SMS

Before working with SMS messages, your software should ensure that:

- SMS service is currently available, using Report SMS Availability (page 211)
- Phonebook service is available (if phonebook lookups are to be done), using Return Phonebook Services Status (page 184) and Return Phonebook Availability (page 184)

Read SMS messages

After starting SMS, you can begin loading messages that are waiting on the SIM (as shown in Figure 15-1 on page 193):

- Report SMS Received Messages Status (page 208)
  - **a.** Use the Get operation to see if there are any messages (unread or previously read) waiting on the SIM.
  - **b.** Enable notification—the modem issues the CnS message whenever a new SMS message is received.
- Get Read SMS Message (page 211)
  Retrieve a previously-read message from the SIM.
  To read all of the previously-read messages:
  - Continue using this CnS message until an SMS message ID repeats (the SMS messages are stored in a circular list—eventually the first SMS message repeats).

- Get Unread SMS Message (page 209)
  Retrieve the first unread SMS message from the SIM (this causes the SIM to flag the message as read).
  To read all of the unread SMS messages:
  - Check for unread SMS messages using the Report SMS Received Messages Status (page 208) and then read the next SMS message, *or*
  - Continue using this CnS message until a read error is indicated in the response.

At any time after SMS has been started, as long as there is space available on the SIM, the module notifies the host when a new message has arrived. You can then load the new message (as shown in Figure 15-2 on page 194).

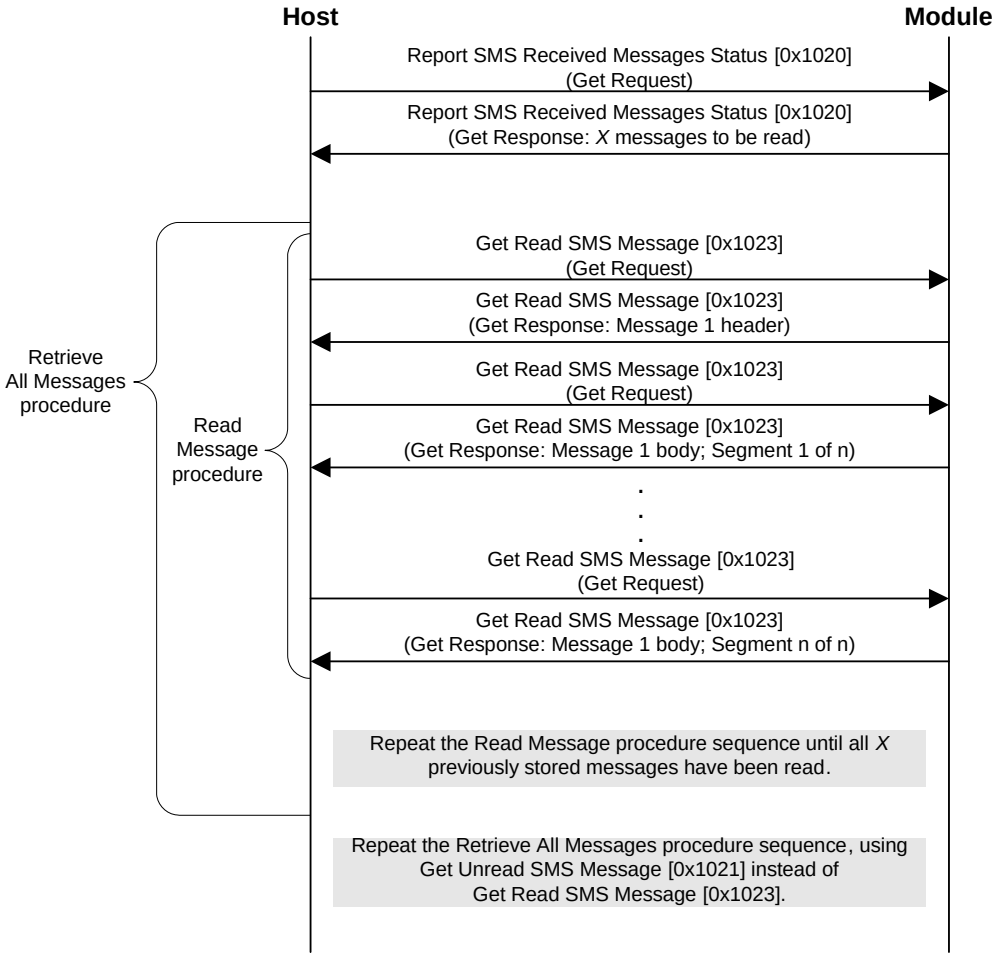

Figure 15-1: Message flow—Read SMS messages at startup

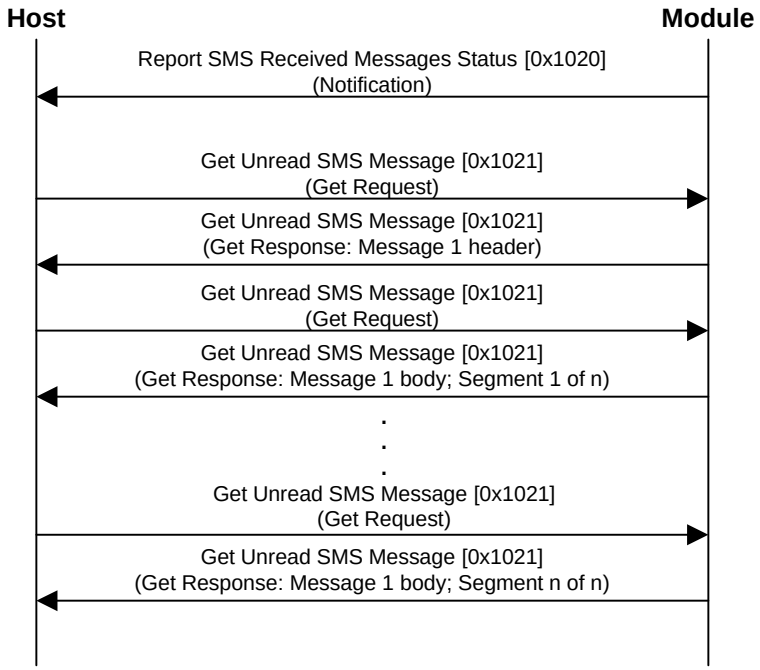

Figure 15-2: Message flow—Read SMS message when notified

Send SMS messages

To send outgoing SMS messages (as shown in Figure 15-3 on page 195), use:

1. Copy Mobile-originated SMS Message to SIM (page 220)
   Copy the new SMS message to the SIM in preparation for sending it to the network.
2. Send Mobile-originated SMS Message To Network (page 221)
   Send the stored SMS message to the network. The message is removed from the SIM (outgoing messages must be saved by the host software, if desired).
3. Return SMS Outgoing Message Number (page 212)
   This notification returns the SMS message ID number that is assigned to the outgoing message.
4. Report Outgoing SMS Message Status (page 213)
   This notification returns the current status of an outgoing message.

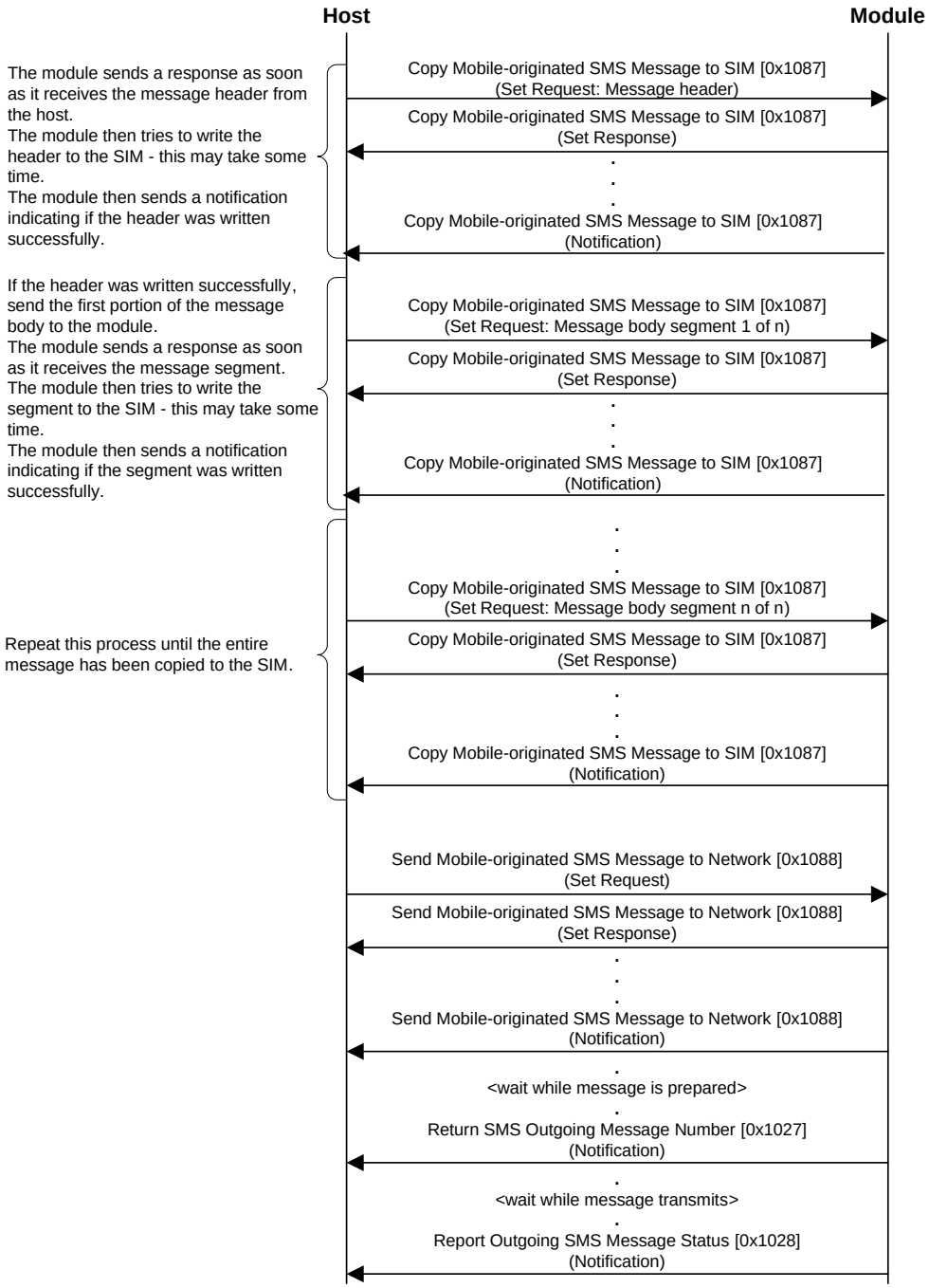

Figure 15-3: Message flow—Send SMS message

Delete SMS messages

Messages must be explicitly deleted from the SIM—there is a limited amount of space available for storage of incoming messages (read and unread).

To delete a message from the SIM (as shown in Figure 15-4), use:

- Delete Mobile-terminated SMS Message (page 213)
  Removes a specific message (using the message ID) from the SIM.

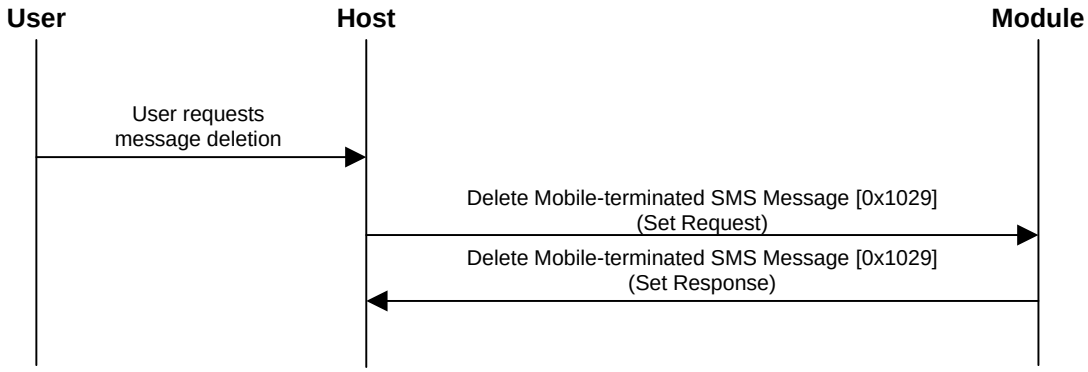

Figure 15-4: Message flow—Delete SMS message

Note: The notification message Report SIM SMS Message Storage Status (page 214) is issued when the SIM becomes full—no further mobile-terminated SMS messages will be accepted by the SIM until messages are deleted.

Configure SMS parameters

Several parameters can be used to customize the SMS service. It is the responsibility of the host software to provide support for entry of these parameters, if desired.

These parameters are stored on the SIM. To display or set the current configuration (as show in Figure 15-5), use:

- Return SIM SMS Configuration Details (page 214)
  Displays the current settings for the various SMS parameters.

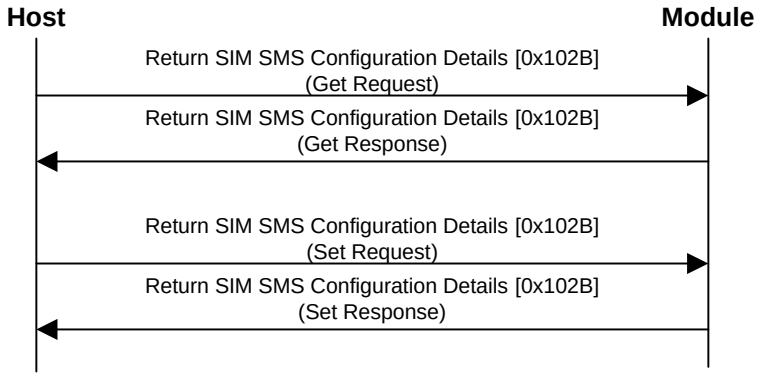

Figure 15-5: Message flow—Read and set SMS configuration

SMS Message organization

SMS message types

SMS messages are grouped into two categories:

- Mobile-terminated
  These are messages that are received by the modem. They are stored on the SIM until they are explicitly deleted by the host software.
- Mobile-originated
  These are messages that are sent by the modem to the network. They are created in the host software, temporarily stored on the SIM, sent to the network, and then automatically removed from the SIM.

  After a message has been sent to the network, it remains at the destination service center (a configurable SIM parameter) until it is downloaded to the recipient's SIM or until an optional validity period has expired (another configurable SIM parameter).

CnS / SMS message structure

Every SMS message (incoming and outgoing) is transported using the following CnS objects:

- Get Unread SMS Message (page 209)—used for incoming messages
- Get Read SMS Message (page 211)—used for incoming messages
- Copy Mobile-originated SMS Message to SIM (page 220)—used for outgoing messages

A single SMS message is received/sent using one or more CnS messages, with the body of the message being split into as many segments as necessary (as shown in Figure 15-6 on page 198):

- The first CnS message contains:
  - CnS object details (see CnS message format (page 59) for details)
    The Object ID, type, and so on
  - SMS envelope (see SMS envelope (page 198) for details)
    Message ID, number of subsequent CnS messages, and so on
  - SMS header (see SMS header (page 199) for details)
    Delivery information, protocol, compression type, and so on
  - If the SMS message is short enough to fit in the remaining CnS parameter space (149 bytes), it is included in this message.
    If the SMS message cannot fit in the remaining space, subsequent CnS messages contain the message (broken into as many segments as necessary). See SMS body (page 207) for details.
- The second (and subsequent) CnS messages contain:
  - CnS object details (see CnS message format (page 59) for details)
    Same general information as the first CnS message.
  - SMS envelope (see SMS envelope (page 198) for details)
    Same general information as the first CnS message.

- SMS body segments (see SMS body (page 207) for details)
  1–240 bytes of the SMS message. All segments have 240 bytes except for the final segment, which is variable length.

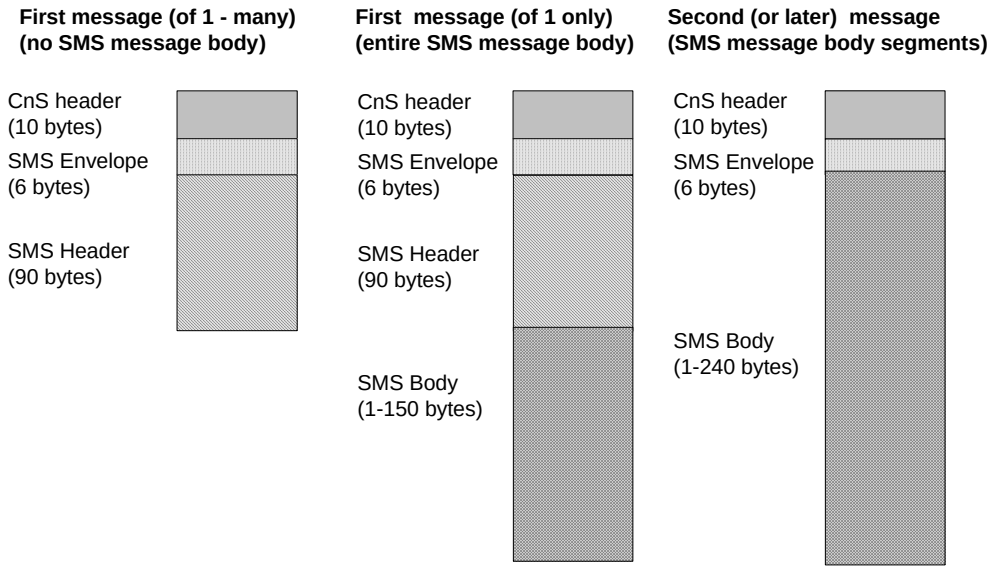

Figure 15-6: SMS message structure

Data structures

SMS envelope

When reading or writing an SMS message, each CnS message's parameter begins with an SMS envelope. This envelope conveys information about the SMS message being transferred, the type of payload (SMS header or SMS body) in this CnS message, and the presence of more SMS message body segments in the modem.

Note: Offsets in Table 15-2 on page 199 are from the beginning of the envelope. The envelope itself appears at the beginning of the CnS parameter field (offset 10 in the CnS message)

.

Table 15-2: SMS envelope structure

Byte offset	Type	Content
0–1	UINT16	Message ID—The unique reference number used to identify the message on the SIM. • Min—0; Max—255 *Note: This field is NOT used for the Copy Mobile-originated SMS Message to SIM (page 220) 'Set (03)' request.*
2	UINT8	Packet Sequence Number of the segmented message
3	UINT8	Packet Type: • 0=First segment Contains the SMS header. • 1=Intermediate segment Contains a portion of the SMS message body. • 2=Last segment1 Contains the final portion of the SMS message body. • 3=Not segmented
4	UINT8	Remaining SMS segments The number of remaining SMS message segments in the modem (excluding this one).
5	UINT8	Payload length Length of the current payload.
6	variable	Payload Based on Packet type—Contains either the SMS header or the segmented SMS message body.

SMS header

This header appears in the first CnS message for an SMS message; subsequent CnS messages are used to carry the body of the SMS message in a series of segmented sections. The CnS messages that are used to send/receive SMS messages are:

- Get Unread SMS Message (page 209)
  Get previously-unread incoming messages from the SIM
- Get Read SMS Message (page 211)
  Get previously-read incoming messages from the SIM
- Copy Mobile-originated SMS Message to SIM (page 220)
  Store outgoing messages on the SIM

The SMS headers for these CnS messages are identical in structure *with the following exceptions*:

Note: These exceptions are noted in Table 15-3 on page 200 in the appropriate field entries.

- Bytes 0–1
  - Incoming messages use one byte (UINT8) each for a result code and a record status.
  - Outgoing messages use two bytes (UINT16) for the message body length
- Destination/Source addresses
  - Incoming messages refer to Source addresses
  - Outgoing messages refer to Destination addresses

Note: Offsets in Table 15-3 on page 200 are from the beginning of the header. The header itself appears after the SMS envelope in the CnS parameter field (offset 16 in the CnS message).

Table 15-3: SMS header structure

Byte Offset	Type	Content
0–1	If the CnS message is: **Copy Mobile-originated SMS Message to SIM:**	
	UINT16	Total message body size—the sum of the bodies of each message segment.
	If the CnS message is: **Get Read SMS Message** or **Get Unread SMS Message:**	
	Byte 0: UINT8 Byte 1: UINT8	Byte 0: Result code: • 0x00=Retrieved message is OK • 0x01=Message is not available (SIM error, message not found, or message segmentation error) Byte 1: Record status—Indicates if message has been previously read • 0x00=New message that has NOT been read before • 0x01=Old message that has been read before
2	UINT8	Destination/Source service center address type: • 0x00=Unknown • 0x01=International (Typical default value) • 0x02=National • 0x03=Network-specific • 0x04=Subscriber • 0x05=Alphanumeric • 0x06=Abbreviated • 0x07–0xFF=Reserved For descriptions of these address types, see 3GPP 23.040, section 9.1.2.5 “Address fields”. *Note: For the CnS Copy Mobile-originated SMS Message to SIM (page 220) message, this is the destination service center address type. For all other cases (incoming messages), this is the source service center address type.*

Table 15-3: SMS header structure (Continued)

Byte Offset	Type	Content
3	UINT8	Destination/Source service Center address numbering plan—used only when the Service center address type is Unknown, International, or National: • 0x00=Unknown • 0x01=ISDN/telephone (E.164/E.163) (Typical default value) • 0x02=Reserved • 0x03=Data (X.121) • 0x04=Telex • 0x05–0x07=Reserved • 0x08=National • 0x09=Private • 0x0A=ERMES (ETSI DE/PS 3 01-3) • 0x0B–0xFF=Reserved For descriptions of these numbering plans, see 3GPP 23.040, section 9.12.5 “Address fields”.
4	UINT8	Length of service center address phone number • Min—0; Max—20
5–24	UINT8[20]	Destination/Source service center address phone number • Binary-coded decimal format 0x00–0x09=‘0’–’9’ Up to 20 characters—See previous parameter for actual length • Ignore trailing unused bytes
25	UINT8	Destination/Source Address type • 0x00=Unknown • 0x01=International (Typical default value) • 0x02=National • 0x03=Network-specific • 0x04=Subscriber • 0x05=Alphanumeric • 0x06=Abbreviated • 0x07–0xFF=Reserved For descriptions of these address types, see 3GPP 23.040, section 9.1.2.5 “Address fields”. *Note: For the CnS Copy Mobile-originated SMS Message to SIM (page 220) message, this is the destination address type. For all other cases (incoming messages), this is the source address type.*

Table 15-3: SMS header structure (Continued)

Byte Offset	Type	Content
26	UINT8	Destination/Source Address numbering plan* • 0x00=Unknown • 0x01=ISDN/telephone (E.164/E.163) (Typical default value) • 0x02=Reserved • 0x03=Data (X.121) • 0x04=Telex • 0x05–0x07=Reserved • 0x08=National • 0x09=Private • 0x0A=ERMES (ETSI DE/PS 3 01-3) • 0x0B–0xFF=Reserved For descriptions of these numbering plans, see 3GPP 23.040, section 9.12.5 “Address fields”.
27	UINT8	Length of destination/source address number • Min—0; Max—20
28–47	UINT8[20]	Destination/Source Address phone number • Binary-code decimal format 0x00–0x09 =‘ ‘0’–’9 • Up to 20 characters—See previous parameter for actual length • Ignore trailing unused bytes
48	UINT8	Reply allowed flag—Reserved for future use • Set to 0.
49	UINT8	Protocol type: • 0x00=Application Layer Protocol • 0x20=Telematic Interworking (Typical default value) • 0x40=SM (SIM) Functions • 0x60=ME (Mobile Equipment) Functions • 0xC0=SC (Service Center) Specific • 0xC1=Raw data • All other values are reserved See 3GPP 23.040, section 9.2.3.9 “TP-Protocol-Identifier (TP-PID)” for more details.

Table 15-3: SMS header structure (Continued)

Byte Offset	Type	Content
50	UINT8	Protocol ID—this value depends on the Protocol type (above) If Protocol type = Application Layer Protocol or Telematic Interworking ... • 0x00 = Implicit (device type is specific to this service center, or can be determined from the address) • 0x01 = Telex • 0x02 = Group 3 Fax • 0x03 = Group 4 Fax • 0x04 = Voice • 0x05 = ERMES • 0x06 = National Paging system • 0x07 = Videotex (t.100/T.101) • 0x08 = Teletex, carrier unspecified • 0x09 = Teletex, in PSPDN • 0x0A = Teletex, in CSPDN • 0x0B = Teletex, in analog PSTN • 0x0C = Teletex, in digital PSTN • 0x0D = UCI (Universal Computer Interface, ETSI DE/PS 3 01-3) • 0x0E = Message handling facility • 0x0F = Any public X.400-based message handling system • 0x10 = Internet Email • 0x11 = Value #1 specific to the SC, usage based on mutual agreement between the SME and the SC • 0x12 = Value #2 specific to the SC (see above) • 0x13 = Value #3 specific to the SC (see above) • 0x14 = Value #4 specific to the SC (see above) • 0x15 = Value #5 specific to the SC (see above) • 0x16 = Value #6 specific to the SC (see above) • 0x17 = Value #7 specific to the SC (see above) • 0x18 = GSM mobile station (Typical default value) The SC converts the SM from the received TP Data Coding Scheme to any data coding scheme supported by that MS (for example, the default).

Table 15-3: SMS header structure (Continued)

Byte Offset	Type	Content
		If Protocol type = SM Functions ... • 0x00=Type 0 message • 0x01=Type 1 message • 0x02=Type 2 message • 0x03=Type 3 message • 0x04=Type 4 message • 0x05=Type 5 message • 0x06=Type 6 message • 0x07=Type 7 message • 0X08=Type 8 message If Protocol type = ME Functions ... • 0x62=Depersonalization • 0x63=Data download If Protocol type = SC Specific or Raw Data ... • 0x00–0xFF=Determined by carrier
51	UINT8	DCS data type—Identifies the DCS data type for the payload data: • 0x00=Default (7-bit SMS default alphabet) • 0x01=8-bit • 0x02=UCS-2 • 0x03=Unknown/Reserved See 3GPP 23.038 for additional information.
52	UINT8	DCS class—Identifies the DCS message class: • 0x00=Class 0 • 0x01=Class 1 • 0x02=Class 2 • 0x03=Class 3 • 0x04=Not given (Typical default value) See 3GPP 23.038 for additional information.
53	UINT8	DCS compression—Indicates if the text is compressed: • 0x00=SMS text is uncompressed (Typical default value) • 0x01=SMS text is compressed using the compression algorithm defined in 3GPP TS 23.042 “Compression algorithm for text messaging services”. See 3GPP 23.038 for additional information.

Table 15-3: SMS header structure (Continued)

Byte Offset	Type	Content
54–56	Byte 54: UINT8 Byte 55: UINT8 Byte 56: UINT8	DCS Message Waiting Indication fields—indicates if messages (various types) are waiting in the network for the user to access
		If the CnS message is: Copy Mobile-originated SMS Message to SIM (page 220)**:**
		Set all 3 bytes to 0.
		If the CnS message is: Get Read SMS Message (page 211) or Get Unread SMS Message (page 209)**:**
		If Bytes 54–56 = 0xFFFFFF, then there are no messages waiting. Otherwise: Byte 54: Message waiting—Indicates if there are any messages waiting to be received by the modem • 0=False • 1=True Byte 55: MMI message waiting indicator—Describes how to set the indicator (type is identified in the next byte) • 0=Set indication inactive. • 1=Set indication active. Byte 56: MMI message indicator type—Identifies the type of indicator that has been activated or deactivated • 0=Voice message waiting Indicator • 1=Fax message waiting indicator • 2=Email message waiting indicator • 3=Other message waiting indicator See 3GPP 23.038 for details.
57	UINT8	More Messages flag
		If the CnS message is: Copy Mobile-originated SMS Message to SIM**:**
		Set this field to 0.
		If the CnS message is: Get Read SMS Message or Get Unread SMS Message**:**
		Ignore this field.
58–71	UINT8[14]	Message Timestamp
		If the CnS message is: Copy Mobile-originated SMS Message to SIM**:**
		Ignore this field
		If the CnS message is: Get Read SMS Message or Get Unread SMS Message**:**
		• Indicates the time when the message originator first sent the message. For more information (including field format), see 3GPP 23.040, section 9.2.3.11 “TP-Service Centre-Time-Stamp (TP-SCTS)”

Table 15-3: SMS header structure (Continued)

Byte Offset	Type	Content

72–87	UINT8[16]	Validity period—indicates (for outgoing messages) how long the message can remain at the service center without being picked up by the recipient, after which it expires and is removed
		If the CnS message is: Copy Mobile-originated SMS Message to SIM (page 220):
		Three formats can be used to record the validity period:
		• No expiry—Set all 16 bytes to 0
		• Integer format—Specify a validity period relative to the current time/date
		  ◦ Byte 0—0x02
		  ◦ Byte 1—0x00–0xFF (VP value used to calculate validity)
		  ◦ Byte 2–15—Unused
		
		Byte 1 (VP value) — Validity period
		0–143 — (VP +1) * 5min
		144–167 — 12hr + ((VP - 143) * 30min)
		168–196 — (VP - 166) * 1day
		197–255 — (VP - 192) * 1week
		
		• Timestamp (string) format—Specify an absolute expiry date/time
		  ◦ Byte 0—0x03
		  ◦ Byte 1—Unused
		  ◦ Byte 2–15—Set according to the standard GPRS/GSM timestamp formatting rules described in 3GPP TS 23.038
		If the CnS message is: Get Read SMS Message (page 211) or Get Unread SMS Message (page 209):
		Ignore this field.

88	UINT8	Reject duplicates—Reserved for future use If the CnS message is: Copy Mobile-originated SMS Message to SIM (page 220): Set to 0. If the CnS message is: Get Read SMS Message (page 211) or Get Unread SMS Message (page 209): Ignore this field.
89	UINT8	User data header indicator—Reserved for future use If the CnS message is: Copy Mobile-originated SMS Message to SIM (page 220): Set to 0. If the CnS message is: Get Read SMS Message (page 211) or Get Unread SMS Message (page 209): Ignore this field.

SMS body

The SMS body is the actual message content of an SMS message. It is transmitted using CnS messages in one of two ways, depending on the length of the message content:

- Short message (<=150 bytes based on the data coding scheme data type being used)
  Only one CnS message is used to transmit the *entire* SMS message (header and body).
- Long message (151 bytes or longer)
  Because the SMS body will not fit in a single CnS message with the SMS header, multiple CnS messages are used. Each CnS message carries a 240-byte segment of the SMS body except for the last one.

Note: Each CnS message can carry a 240-byte payload. The first CnS message always carries the SMS header—this leaves 150 bytes of space to carry a short SMS message body. If the SMS body cannot fit in this space, it is segmented and sent in the second and subsequent CnS messages.

SMS messages reference

Report SMS Received Messages Status

Reports the number of mobile-terminated SMS messages that are stored on the SIM:

- On request, using Get (01) when the application starts
- When triggered, if Notify (07) is enabled *and* a message is received during normal operation
  (This is an update-type notification.)

Table 15-4: Object—Report SMS Received Messages Status

Object ID	0x1020		Report SMS Received Messages Status	
Operations	✗ Set ✔ Get ✔ Notify			
	Parameters (fields)			
	Len	**Offset**	**Type**	**Description**
01 (Get)	0			None
02 (Get Response)	5	0	UINT8	Used record count—The number of internal records on the SIM that have been used to store all messages (read and unread) • Min—0; Max—SIM-dependent
		1	UINT8	Unread record count—The number of internal records on the SIM that have been used to store unread messages • Min—0; Max—SIM-dependent
		2	UINT8	Memory exceeded flag—If the memory exceeded flag is set, this indicates that there are messages on the network that have not been delivered to the SIM (assuming that the validity period of those messages has not expired) • 0x00=No • 0x01=Yes *Note: This serves the same purpose as the Report SIM SMS Message Storage Status (page 214) (SIM full) notification.*
		3	UINT8	SMS routing option: • 0x00=Packet-switched • 0x01=Circuit-switched • 0x02=Packet-switched preferred • 0x03=Circuit-switched preferred
		4	UINT8	Maximum record count—The total number of internal records on the SIM that can be used to store all messages (read and unread) • Max—SIM-dependent

Table 15-4: Object—Report SMS Received Messages Status (Continued)

07 (Notify)	6	0	UINT8	Used record count—The number of internal records on the SIM that have been used to store all messages (read and unread) • Min—0; Max—SIM-dependent
		1	UINT8	Unread record count—The number of internal records on the SIM that have been used to store unread messages • Min—0; Max—SIM-dependent
		2	UINT8	Memory exceeded flag—If the memory exceeded flag is set, this indicates that there are messages on the network that have not been delivered to the SIM (assuming that the validity period of those messages has not expired) • 0x00=No • 0x01=Yes *Note: This serves the same purpose as the Report SIM SMS Message Storage Status (page 214) (SIM full) notification.*
		3	UINT8	SMS routing option: • 0x00=Packet-switched • 0x01=Circuit-switched • 0x02=Packet-switched preferred • 0x03=Circuit-switched preferred
		4	UINT8	Maximum record count—The total number of internal records on the SIM that can be used to store all messages (read and unread) • Max—SIM-dependent
		5	UINT8	Received message record number

Get Unread SMS Message

Note: Get Read SMS Message is identical in structure and behavior—it is used to read previously-read messages.

Read an unread mobile-terminated message that has been delivered by the network to the SIM:

- On request, using multiple consecutive Get (01) operations. The first Get returns the message header (identifying information); subsequent Get operations return the message body, segmented into multiple CnS messages.

Note: The SMS body is included in the first CnS message only if it is short enough to fit completely.

Table 15-5: Object—Get Unread SMS Message

Object ID	0x1021		Get Unread SMS Message	
Operations	✗ Set ✔ Get ✗ Notify			
	Parameters (fields)			
	Len	**Offset**	**Type**	**Description**
01 (Get) 1st	0			None
02 (Get Response) 1st	96 or 97–246	0	Struct	SMS envelope (See SMS envelope on page 198.)
		6	Struct	SMS header (See SMS header on page 199.)
		If the SMS body can be encoded in 140 bytes or less, it is included in this CnS message—no further CnS messages are required.		
		96	Struct	SMS body (1–140 bytes) (See SMS body on page 207.)
01 (Get)	0	0		None
02 (Get Response)	7–246	0	Struct	SMS envelope (See SMS envelope on page 198.)
		6	UINT8[1..240]	SMS body segment (1–240 bytes) (See SMS body on page 207.)

Retrieving an unread message

1. Issue Get (01).
2. Receive Response (02) containing the SMS envelope, SMS header, and possibly the SMS body. The SMS envelope and SMS header include all the information needed to parse the content of the message(s) that follow.
3. If the SMS body was not included in the first CnS message:
   a. Issue Get (01).
   b. Receive Response (02) containing a packet header and a segment of the SMS message body. Each segment (except the final segment) contains 240 bytes of the message body.
   If there are segments still remaining (indicated in the packet header), repeat steps a–b.

Report SMS Availability

Report the current availability of SMS service:

- On request, using Get (01)
- When triggered, if Notify (07) is enabled *and* SMS availability changes (becomes available/unavailable)
  (This is an update-type notification.)

Table 15-6: Object—Report SMS Availability

Object ID	0x1022		Report SMS Availability
Operations	✗ Set ✔ Get ✔ Notify		
	Parameters (fields)		
	Len	**Offset**	**Type**	**Description**
01 (Get)	0	0		None
02 (Get Response) **07 (Notify)**	1	0	UINT8	Current SMS service availability • 0x00=Unavailable • 0x01=Available

Get Read SMS Message

Note: Get Unread SMS Message is identical in structure and behavior—it is used to read unread messages.

Read a previously-read mobile-terminated message that has been delivered by the network to the SIM:

- On request, using multiple consecutive Get (01) operations. The first Get returns the message header (identifying information); subsequent Get operations return the message body, segmented into multiple CnS messages.

Note: The SMS body is included in the first CnS message if and only if it is short enough to fit completely

Table 15-7: Object—Get Read SMS Message

Object ID	0x1023			Get Read SMS Message
Operations	✗ Set ✔ Get ✗ Notify			
	Parameters (fields)			
	Len	**Offset**	**Type**	**Description**
01 (Get) 1st	0			None
02 (Get Response) 1st	96 or 97–246	0	Struct	SMS envelope (See SMS envelope on page 198.)
		6	Struct	SMS header (See SMS header on page 199.)
		If the SMS body can be encoded in 140 bytes or less, it is included in this CnS message—no further CnS messages are required.		
		96	Struct	SMS body (1–140 bytes) (See SMS body on page 207.)

Table 15-7: Object—Get Read SMS Message (Continued)

01 (Get)	0	0		None
02 (Get Response)	7–246	6	Struct	SMS envelope (See SMS envelope on page 198.)
		6	UINT8	SMS body segment (1–240 bytes) (See SMS body on page 207.)

Retrieving a previously-read message

1. Issue Get (01).
2. Receive Response (02) containing the SMS envelope, SMS header, and possibly the SMS body.
   The SMS envelope and SMS header include all of the information needed to parse the content of the message(s) that follow.
3. If the SMS body was not included in the first CnS message:
   a. Issue Get (01).
   b. Receive Response (02) containing a packet header and a segment of the SMS message body. Each segment (except the final segment) contains 249 bytes of the message body.
   If there are segments still remaining (indicated in the packet header), repeat steps a–b.

Return SMS Outgoing Message Number

Returns a unique reference number that is assigned to a mobile-originated SMS message:

- Automatically, if Notify (07) is enabled, when a mobile-originated SMS message is sent to the network using Send Mobile-originated SMS Message To Network (page 221).
- This reference number can be referenced against the Report Outgoing SMS Message Status (page 213) notification that reports the delivery status of the message.

Table 15-8: Object—Return SMS Outgoing Message Number

Object ID	0x1027		Return SMS Outgoing Message Number	
Operations	✗ Set ✗ Get ✔ Notify			
	Parameters (fields)			
	Len	**Offset**	**Type**	**Description**
07 (Notify)	7	0	UINT8	Message reference number • Min—0; Max—255
		1	UINT8	Result code • 0x00=Success • 0x01=Failure (see diagnostics fields which follow)
		2	UINT8	Diagnostic code 1 (request status)

Table 15-8: Object—Return SMS Outgoing Message Number (Continued)

		3	UINT8	Diagnostic code 2 (report status)
		4	UINT8	Diagnostic code 3 (RP cause)
		5	UINT8	Diagnostic code 4 (ME problem)
		6	UINT8	Diagnostic code 5 (cipher)

Report Outgoing SMS Message Status

Reports on the delivery status of a mobile-originated SMS message that was previously sent to the network using Send Mobile-originated SMS Message To Network (page 221):

- Automatically, if Notify (07) is enabled, when the delivery status changes.

Table 15-9: Object—Report Outgoing SMS Message Status

Object ID	0x1028		Report Outgoing SMS Message Status
Operations	✗ Set ✗ Get ✔ Notify		
	Parameters (fields)		
	Len	**Offset**	**Type**	**Description**
07 (Notify)	17	0	UINT8	Message reference number • Min—0; Max—255
		1	UINT8	Result code • 0x00=OK • 0x01=General error • 0x02=FDN mismatch
		2	UINT8	Message status • 0x00=Received • 0x01–0xFF=Unknown
		3	UINT8[14]	Reception time—The time that the message arrived at the service center See 3GPP TS 23.040, section 9.2.3.11 “TP-Service Centre-Time-Stamp”.

Delete Mobile-terminated SMS Message

Deletes a mobile-terminated SMS message from the SIM:

- On request, using Set (03) (as shown in Figure 15-4 on page 196)

Note: Mobile-terminated SMS messages are NEVER automatically deleted from the SIM. It is the responsibility of the host software to explicitly remove individual messages.

Table 15-10: Object—Delete Mobile-terminated SMS Message

Object ID	0x1029		Delete Mobile-terminated SMS Message	
Operations	✔ Set ✗ Get ✗ Notify			
	Parameters (fields)			
	Len	**Offset**	**Type**	**Description**
03 (Set)	1	0	UINT8	Reference number of the message to delete • Min—0; Max—255

Report SIM SMS Message Storage Status

Reports the current status of the SIM's SMS message storage capacity (full/not full):

- When triggered, if Notify (07) is enabled *and* when the Storage condition changes
  (This is an update-type notification.)

Table 15-11: Object—Report SIM SMS Message Storage Status

Object ID	0x102A		Report SIM SMS Message Storage Status	
Operations	✗ Set ✗ Get ✔ Notify			
	Parameters (fields)			
	Len	**Offset**	**Type**	**Description**
07 (Notify)	1	0	UINT8	SIM storage status: • 0x00=SIM is not full • 0x01=SIM is full

Return SIM SMS Configuration Details

Returns the current settings for SMS-related parameters on the SIM:

- On request, using Get (01) or Set (03)

Table 15-12: Object identification—Return SIM SMS Configuration Details

Object ID	0x102B		Return SIM SMS Configuration Details	
Operations	✔ Set ✔ Get ✗ Notify			
	Parameters (fields)			
	Len	**Offset**	**Type**	**Description**
01 (Get)	0			None

Table 15-12: Object identification—Return SIM SMS Configuration Details (Continued)

02 (Get Response) **03 (Set)**	58	0	UINT8	Default destination address present • 0x00=No • 0x01=Yes
		1	UINT8	Default destination address type • 0x00=Unknown • 0x01=International (Typical default value) • 0x02=National • 0x03=Network-specific • 0x04=Subscriber • 0x05=Alphanumeric • 0x06=Abbreviated • 0x07–0xFF=Reserved For descriptions of these address types, see 3GPP 23.040, section 9.1.2.5 “Address fields”.
		2	UINT8	Default destination address numbering plan • 0x00=Unknown • 0x01=ISDN/telephone (E.164/E.163) (Typical default value) • 0x02=Reserved • 0x03=Data (X.121) • 0x04=Telex • 0x05–0x07=Reserved • 0x08=National • 0x09=Private • 0x0A=ERMES (ETSI DE/PS 3 01-3) • 0x0B–0xFF=Reserved For descriptions of these numbering plans, see 3GPP 23.040, section 9.12.5 “Address fields”.
		3	UINT8	Length of default destination phone number
		4	UINT8[20]	Default destination phone number • Binary-coded decimal format 0x00–0x09 = ‘0’–’9 • Up to 20 characters—See previous parameter for actual length • Ignore trailing unused bytes
		24	UINT8	Service Center address present • 0x00=No • 0x01=Yes

Table 15-12: Object identification — Return SIM SMS Configuration Details (Continued)

		25	UINT8	Service Center address type • 0x00 = Unknown • 0x01 = International (Typical default value) • 0x02 = National • 0x03 = Network-specific • 0x04 = Subscriber • 0x05 = Alphanumeric • 0x06 = Abbreviated • 0x07–0xFF = Reserved For descriptions of these address types, see 3GPP 23.040, section 9.1.2.5 “Address fields”.
		26	UINT8	Service Center address numbering plan • 0x00 = Unknown • 0x01 = ISDN/telephone (E.164/E.163) (Typical default value) • 0x02 = Reserved • 0x03 = Data (X.121) • 0x04 = Telex • 0x05–0x07 = Reserved • 0x08 = National • 0x09 = Private • 0x0A = ERMES (ETSI DE/PS 3 01-3) • 0x0B–0xFF = Reserved For descriptions of these numbering plans, see 3GPP 23.040, section 9.12.5 “Address fields”.
		27	UINT8	Length of service center address phone number
		28	UINT8[20]	Service Center address phone number • Binary-coded decimal format 0x00–0x09 = ‘0’–’9 • Up to 20 characters — See previous parameter for actual length • Ignore trailing unused bytes
		48	UINT8	Default Protocol ID present • 0x00 = No • 0x01 = Yes

Table 15-12: Object identification — Return SIM SMS Configuration Details (Continued)

		49	UINT8	Default Protocol Type • 0x00=Application Layer Protocol • 0x20=Telematic Interworking (Typical default value) • 0x40=SM (SIM) Functions • 0x60=ME (Mobile Equipment) Functions • 0xC0=SC (Service Center) Specific • 0xC1=Raw data • All other values are reserved See 3GPP 23.040, section 9.2.3.9 “TP-Protocol-Identifier (TP-PID)” for more details.
		50	UINT8	Protocol ID—This value depends on the Protocol type (above) If Protocol type = Application Layer Protocol or Telematic Interworking ... • 0x00=Implicit (device type is specific to this service center, or can be determined from the address) • 0x01=Telex • 0x02=Group 3 Fax • 0x03=Group 4 Fax • 0x04=Voice • 0x05=ERMES • 0x06=National Paging system • 0x07=Videotex (t.100/T.101) • 0x08=Teletex, carrier unspecified • 0x09=Teletex, in PSPDN • 0x0A=Teletex, in CSPDN • 0x0B=Teletex, in analog PSTN • 0x0C=Teletex, in digital PSTN • 0x0D=UCI (Universal Computer Interface, ETSI DE/PS 3 01-3) • 0x0E=Message handling facility • 0x0F=Any public X.400-based message handling system • 0x10=Internet Email • 0x11=Value #1 specific to the SC, usage based on mutual agreement between the SME and the SC • 0x12=Value #2 specific to the SC (see above) • 0x13=Value #3 specific to the SC (see above) • 0x14=Value #4 specific to the SC (see above) • 0x15=Value #5 specific to the SC (see above) Continued on next page

Table 15-12: Object identification — Return SIM SMS Configuration Details (Continued)

				(continued from previous page) • 0x16=Value #6 specific to the SC (see above) • 0x17=Value #7 specific to the SC (see above) • 0x18=GSM mobile station (Typical default value) The SC converts the SM from the received TP Data Coding Scheme to any data coding scheme supported by that MS (for example, the default).
				If Protocol type = SM Functions ... • 0x00=Type 0 message • 0x01=Type 1 message • 0x02=Type 2 message • 0x03=Type 3 message • 0x04=Type 4 message • 0x05=Type 5 message • 0x06=Type 6 message • 0x07=Type 7 message • 0X08=Type 8 message
				If Protocol type = ME Functions ... • 0x62=Depersonalization • 0x63=Data download
				If Protocol type = SC Specific or Raw Data ... • 0x00–0xFF=Determined by carrier
		51	UINT8	Default Data Coding Scheme (DCS) present • 0x00=No • 0x01=Yes
		52	UINT8	Default DCS Data Type—Identifies the DCS data type for the payload data • 0x00=Default (7-bit SMS default alphabet) • 0x01=8-bit • 0x02=UCS-2 • 0x03=Unknown/Reserved See 3GPP 23.038 for additional information.
		53	UINT8	Default DCS Class—Identifies the DCS message class: • 0x00=Class 0 • 0x01=Class 1 • 0x02=Class 2 • 0x03=Class 3 • 0x04=Not given (Typical default value) See 3GPP 23.038 for additional information.

Table 15-12: Object identification—Return SIM SMS Configuration Details (Continued)

		54	UINT8	Default DCS Compression—Indicates if the text is compressed: • 0x00=SMS text is uncompressed (Typical default value) • 0x01=SMS text is compressed using the compression algorithm defined in 3GPP TS 23.042 “Compression algorithm for text messaging services”. See 3GPP 23.038 for additional information.
		55	UINT8	Default validity period present • 0x00=No • 0x01=Yes
		56	UINT8	Default validity period—The amount of time that an outgoing message remains in the network before it is purged by the network (even if it is not retrieved by the destination party): Value (VP) — Validity Period 0–143 — (VP+1) * 5 minutes (5min intervals up to 12hrs) 144–167 — 12 hours + ((VP-143) * 30 minutes) 168–196 — (VP-166) * 1day 197–255 — (VP-192) * 1week See Example: Validity Period values (page 219).
		57	UINT8	SMS Routing Option • 0x00=Packet-switched • 0x01=Circuit-switched • 0x02=Packet-switched preferred • 0x03=Circuit-switched preferred

Example: Validity Period values

Table 15-13: Sample Validity Period values

If VP = ...	then validity period = ...
25	(25 + 1) * 5m = 2h 30m
150	12h + ((150 - 143) * 30m) = 15h 30m
180	(180 - 166) * 1d = 14 days
220	(220 - 192) * 1w = 28 weeks

Copy Mobile-originated SMS Message to SIM

*Note: After copying the message, you MUST send it with Send Mobile-originated SMS Message To Network (page 221) before copying another message to the SIM (the SIM only holds **one** outgoing message).*

Download a mobile-originated SMS message from the host to the SIM:

- On request, using multiple consecutive Set (03) operations; a Notify (07) indicates the copy status

Note: The SMS body is included in the first CnS message if and only if it is short enough to fit completely.

Table 15-14: Object—Copy Mobile-originated SMS Message to SIM

Object ID	0x1087		Copy Mobile-originated SMS Message to SIM	
Operations	✔ Set ✗ Get ✔ Notify			
	Parameters (fields)			
	Len	**Offset**	**Type**	**Description**
03 (Set) 1st	96 or 97–246	0	Struct	SMS envelope (See SMS envelope on page 198.)
		6	Struct	SMS header (See SMS header on page 199.)
		If the SMS body can be encoded in 140 bytes or less, it is included in this CnS message—no further CnS messages are required.		
		96	Struct	SMS body (1–140 bytes) (See SMS body on page 207.)
03 (Set) 2nd-nth	7–246	0	Struct	SMS envelope (See SMS envelope on page 198.)
		6	Struct	SMS body (See SMS body on page 207.)
07 (Notify)	1	0	UINT8	Request status: • 0x00=Success—Message segment stored okay • 0x01=SMS not ready • 0x02=SMS record not found • 0x03=SIM error • 0x04=PS error • 0x05=Missing message ID • 0x06=Invalid record • 0x07=Concatenated Link error • 0x08=Record not submitted • 0x09=Send from SIM in progress • 0x0A=NRAM error • 0x0B=FDN mismatch • 0x0C=Unknown error

Copying a message to the SIM

1. Prepare the first CnS message with the SMS header. If the SMS message body is short enough to fit in 140 bytes or less, add it to the first message (no other messages are required in this case).
2. Issue Set (03).
3. Receive Set Response (04).
   If there are any remaining SMS message body segments to send, repeat steps 2–3 (the CnS message contains only the SMS envelope and SMS message body segment).
4. After the entire message has been sent, the modem issues a Notify (07) indicating the success or failure of the operation.

Send Mobile-originated SMS Message To Network

Sends the most recently downloaded (Copy Mobile-originated SMS Message to SIM (page 220)) mobile-originated SMS message to the network:

- On request, using Set (03); a Notify (07) indicates the transmission status

Note: If more than one message is downloaded before using this object, the earlier messages are lost.

Table 15-15: Object—Send Mobile-originated SMS Message to Network

Object ID	0x1088		Send Mobile-originated SMS Message To Network	
Operations	✔ Set ✗ Get ✔ Notify			
	Parameters (fields)			
	Len	**Offset**	**Type**	**Description**
03 (Set)	0			None
07 (Notify)	1	0	UINT8	Request status: • 0x00=Success—Message sent okay • 0x01=SMS not ready • 0x02=SMS record not found • 0x03=SIM error • 0x04=PS error • 0x05=Missing message ID • 0x06=Invalid record • 0x07=Concatenated link error • 0x08=Record not submitted • 0x09=Send from SIM in progress • 0x0A=NRAM error • 0x0B=FDN mismatch • 0x0C=Unknown error

Manage SMS Status Reports

Enables, disables, or indicates the current status of the ability to receive SMS status reports—disabled, enabled, or user-definable.

Use this object to report the status:

- On request, using Get (01)

Use this object to enable or disable the feature (if the module was not factory set as read-only):

- On request, using Set (03)

Table 15-16: Object—Manage SMS Status Reports

Object ID	0x700D		Manage SMS status reports	
Operations	✔ Set ✔ Get ✗ Notify			
	Parameters (fields)			
	Len	**Offset**	**Type**	**Description**
01 (Get)	0			None
02 (Get Response)	4	0	UINT16	Object version • 0x0001=Revision 1
		2	UINT8	Status report enabled flag • 0x00=Disabled • 0x01=Enabled
		3	UINT8	Access method for status report setting—The access method is factory-configured and cannot be changed. • 0x00=Read / Write. The status report enabled flag can be modified • 0x01=Read only. The status report setting enabled flag can not be modified
03 (Set)	3	0	UINT16	Object version • 0x0001=Revision 1
		2	UINT8	Status report enabled flag—This flag can only be changed if the access method for status report seting (last field of the Get Response) is factory-configured as Read / Write. • 0x00=Disabled • 0x01=Enabled
04 (Set Response)	3	0	UINT16	Object version • 0x0001=Revision 1
		2	UINT8	Request status • 0x00=Request successful • 0x01=Status report setting can not be modified

16: Location Based Services

This chapter describes CnS messages that are related to GPS Location Based Services (LBS) commands.

Location Based Services messages summary

Table 16-1: Location Based Services messages

Object	ID	Description	S	G	N
Initiate location fix (page 231)	0x0F02	Initiates a GPS location fix	✔	✗	✗
Start tracking session (page 232)	0x0F04	Starts a GPS tracking session	✔	✗	✗
End location fix session (page 233)	0x0F05	Stops current location fix session	✔	✗	✗
Report location fix results (page 234)	0x0F0A	Reports result of a location fix	✗	✔	✔
Report location fix completed (page 236)	0x0F0B	Reports a location fix or tracking session has finished	✗	✗	✔
Report location fix error (page 236)	0x0F0C	Reports location fix or tracking session has ended prematurely	✗	✗	✔
Report location fix begun (page 238)	0x0F0D	Reports location fix has begun	✗	✗	✔
Report position determination failure (page 238)	0x0F0E	Reports modem has failed to produce a location fix within the timeout period	✗	✗	✔
Report IP address used for TCP/IP (page 239)	0x0F13	Reports or sets the IP address used for TCP/IP	✔	✔	✔
Report Port ID (page 239)	0x0F16	Reports or sets the Port ID used for TCP/IP	✔	✔	✔
Start Keep Warm process (page 240)	0x0F1A	Enables modem's keep warm functionality	✔	✗	✗
Report Keep Warm process begun (page 241)	0x0F1B	Reports keep warm processing has begun	✗	✗	✔
Report Keep Warm process status (page 241)	0x0F1C	Reports current state of keep warm process	✗	✔	✔
Report Keep Warm process complete (page 242)	0x0F1D	Reports keep warm process has finished	✗	✗	✔
Stop Keep Warm process (page 242)	0x0F1E	Stops keep warm processing	✔	✗	✗
Report Satellite Information (page 242)	0x0F1F	Reports satellite information	✗	✔	✔

Table 16-1: Location Based Services messages (Continued)

Object	ID	Description	S	G	N
Clear GPS Assistance Data (page 243)	0x0F20	Clears GPS assistance data from modem	✔	✗	✗
Manage location fix settings (page 244)	0x0F21	Stores and retrieves location fix settings	✔	✔	✗
Report network-initiated location fix request (page 246)	0x0F22	Notifies the host that a network-initiated location fix is being requested	✔	✗	✔
Report Auto Track feature state (page 250)	0x0F24	Reports or sets GPS Auto Track state and configuration.	✔	✔	✗

GPS objects overview

This section describes procedures for working with several CnS GPS objects to accomplish common tasks.

Note: LBS clients are applications, like Watcher, that use the CnS GPS objects. They may be 'registered' clients (those that are interested in receiving notifications about LBS events) or 'requesting' (active) clients (those that initate an LBS session or event). A requesting client is typically also a registered client—after it is no lo longer active, it will still receive notifications that it is registered for.

Get terminal location

Request last known modem location

To get the last location data obtained by the modem as shown in Figure 16-1:

1. Use the CnS object Report location fix results (page 234) to request the modem's last known location.
2. Receive the CnS object Report location fix results (page 234) containing the requested data, or receive a "CnS Unavailable" error message.

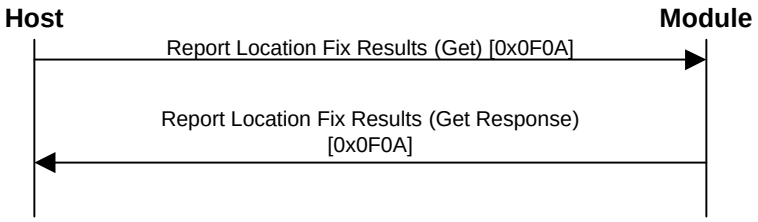

Figure 16-1: Request last known modem location

Request current modem location

To get the modem's current as shown in Figure 16-2:

1. Use the CnS object Initiate location fix (page 231) to request the modem's current location.
2. Receive the CnS object Initiate location fix (page 231). The payload indicates if the fix has been initiated or if an error occurred. If the fix was initiated, the host now needs to wait for a notification that it has begun.
3. Receive the CnS object Report location fix begun (page 238). (All registered clients receive this notification.)
4. Receive the CnS object Report location fix results (page 234) when the data becomes available.
5. Receive the CnS object Report Satellite Information (page 242) when the data becomes available.
6. The session may not complete with a successful location fix:
   - If the session encountered an error that made it stop or if the host sends the CnS object End location fix session (page 233), receive the CnS object Report location fix error (page 236).
   - If the modem cannot obtain a location fix before the specified timeout period expires, receive the CnS object Report position determination failure (page 238).
7. Receive the CnS object Report location fix completed (page 236), indicating that the session is finished. (All registered clients receive this notification.)

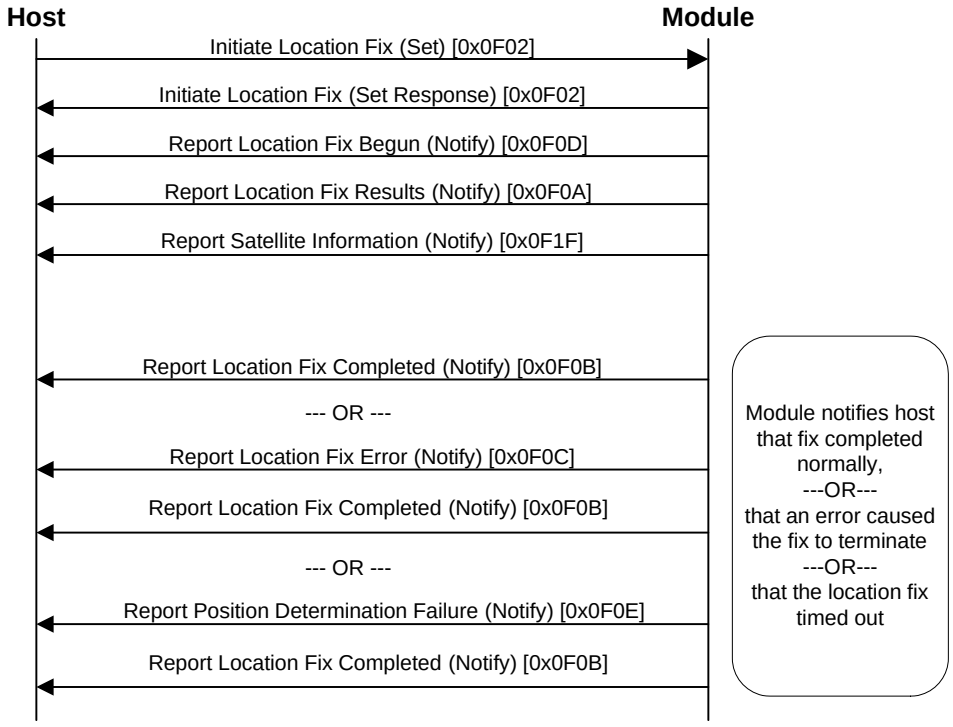

Figure 16-2: Request single location fix

Track terminal location

Frequent tracking

To track the modem's location through a series of frequent location fixes as shown in Figure 16-3 on page 227:

1. Use the CnS object Start tracking session (page 232) to request that the modem initiate the tracking session.
2. Receive the CnS object Start tracking session (page 232). The payload indicates if the session is initiated or if an error occurred. If the session was initiated, the host now needs to wait for a notification that the session has begun.
3. Receive the CnS object Report location fix begun (page 238), which indicates that the tracking session has begun. (All registered clients receive this notification.)
4. The host now waits to receive location data as location fixes occur—these fixes repeat based on the frequency parameters from the initiation request (fix rate and fix count):
   a. Receive the CnS object Report location fix results (page 234) when the data becomes available.
   b. Receive the CnS object Report Satellite Information (page 242) when the data becomes available.
5. If the session encountered an error that made it stop or if the host sends the CnS object End location fix session (page 233), receive the CnS object Report location fix error (page 236).
6. Receive the CnS object Report location fix completed (page 236) indicating that the session is finished. (All registered clients receive this notification.)

Note: If the modem times out before finishing a fix, it sends the CnS notification Report position determination failure (page 238). Go to Step 6.

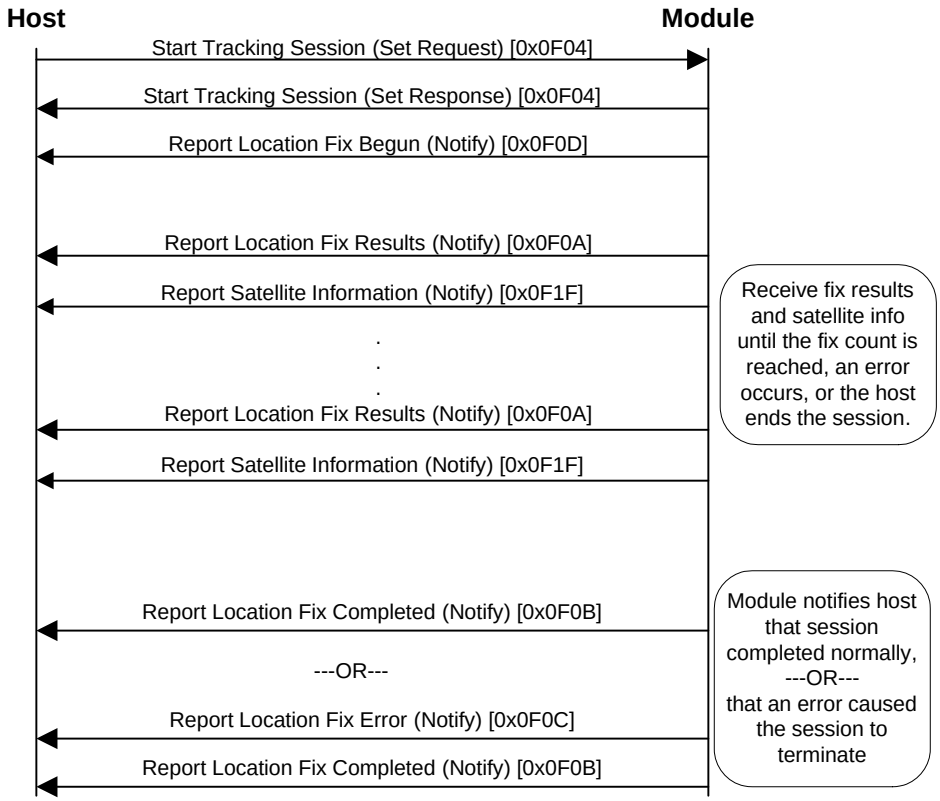

Figure 16-3: Message flow—Frequent tracking

Infrequent tracking

To track the modem's location through a series of infrequent location fixes, repeat the Request current modem location (page 225) process at the required interval.

End location fix session

To prematurely terminate an active location fix session as shown in Figure 16-4 on page 228:

1. Use the CnS object End location fix session (page 233), with Session Type = 0 (Fix), to request that the modem terminate the session.
2. Receive the CnS object End location fix session (page 233). The payload indicates if the request is initiated or if an error occurred. If the request was initiated, the host now needs to wait for a notification that the session has ended.
3. Receive the CnS object Report location fix error (page 236) with Error Code = 0x09 (Client ended). (All registered clients receive this notification.)
4. Receive the CnS object Report location fix completed (page 236) indicating that the session is finished. (All registered clients receive this notification.)

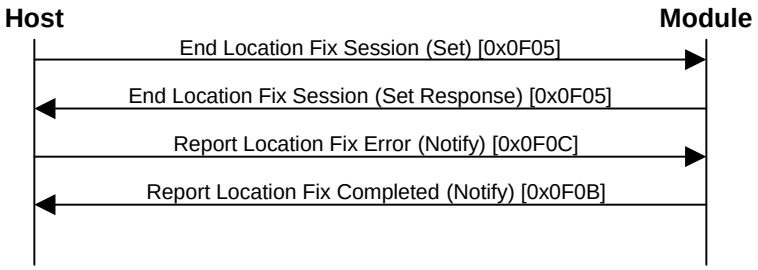

Figure 16-4: Message flow—Terminate location fix session

Download GPS assistance data

GPS assistance data (ephemeris and almanac) is downloaded as shown in Figure 16-5 on page 229 by initiating a 'keep warm' process.

Download data from GPS assistance server

To download data from a GPS assistance server by keeping the receiver in a 'warm' state:

1. Use the CnS object Start Keep Warm process (page 240) to request that the modem initiate the data download.
2. Receive the CnS object Start Keep Warm process (page 240). The payload indicates if the process is initiated or if an error occurred. If the process was initiated, the host now needs to wait for a notification that the download session has begun.
3. Receive the CnS object Report Keep Warm process begun (page 241), which indicates that the download session has begun. (All registered clients receive this notification.)
4. Receive the CnS object Report Keep Warm process status (page 241), which indicates that GPS is in a warm state (the modem has received the data).
5. Use the CnS object Stop Keep Warm process (page 242) to request that the modem stop the download process.
6. Receive the CnS object Stop Keep Warm process (page 242). The payload indicates if the request is valid or if an error occurred. If the request is valid, the host now needs to wait for a notification that the session is stopped.
7. Receive the CnS object Report Keep Warm process complete (page 242) indicating that the session is finished. (All registered clients receive this notification.)

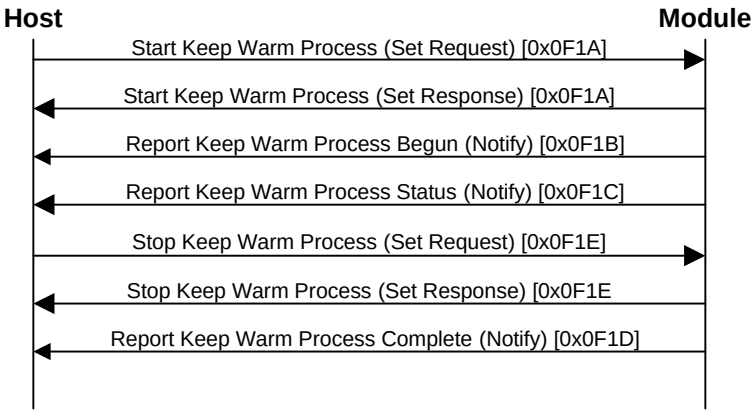

Figure 16-5: Message flow—Download GPS assistance data

Access default position parameters

Retrieve default position parameters

To retrieve default position determination parameters as shown in Figure 16-6:

1. Repeat the following steps for each required parameter:
   a. Use the appropriate CnS object—Report IP address used for TCP/IP (page 239) or Report Port ID (page 239)—to request the required parameter.
   b. Receive that object's response.

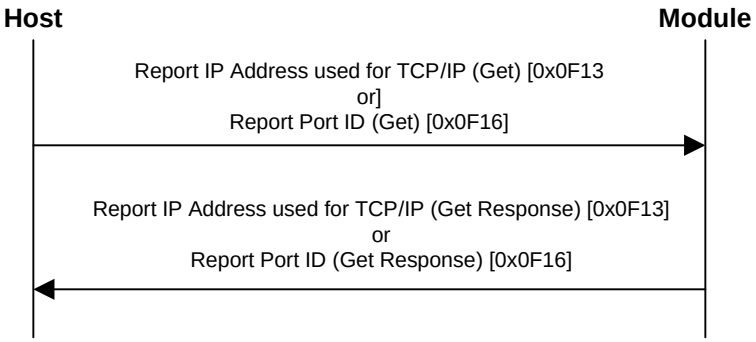

Figure 16-6: Message flow—Retrieve default position parameters

Update default position parameters

To update default position determination parameters as shown in Figure 16-7 on page 230:

1. Repeat the following steps for each required parameter:
   a. Use the appropriate CnS object—Report IP address used for TCP/IP (page 239) or Report Port ID (page 239)—to update the required parameter.
   b. Receive that object's response. The payload indicates if the request is being processed or if an error occurred. If the request is being processed,

the host now needs to wait for a notification that it has completed before proceeding to update the next parameter. Otherwise, the next update request will result in an error being returned (as shown in Figure 16-8).

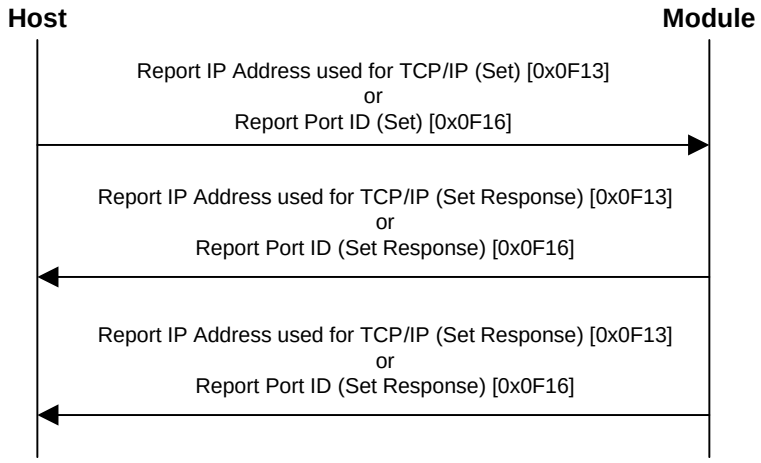

Figure 16-7: Message flow—Update default position parameters (Correct host behavior)

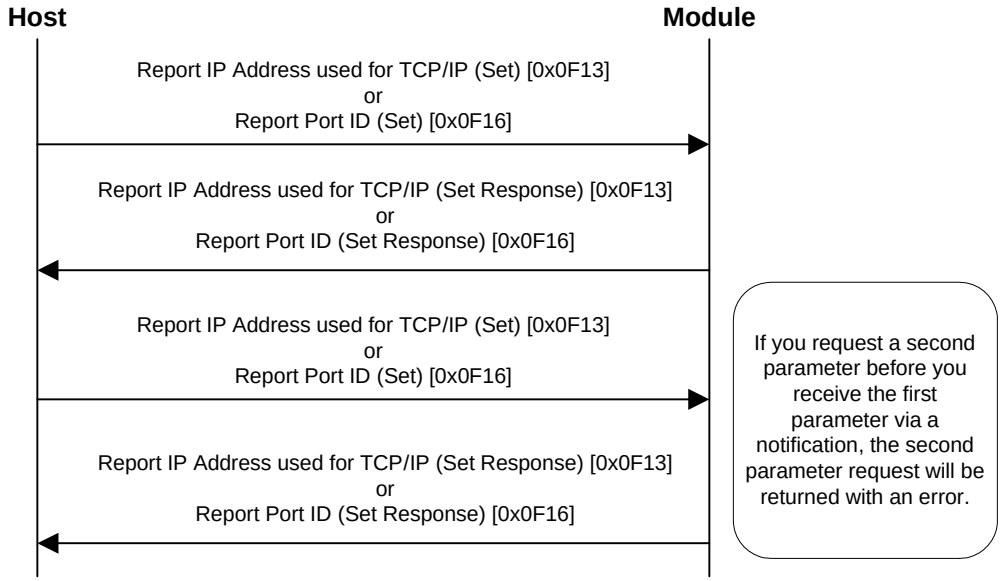

Figure 16-8: Message flow—Update default position parameters (Incorrect host behavior)

LBS Service messages reference

Initiate location fix

Initiates a GPS location fix:

- On request, using Set (03)

The modem returns a response indicating if the fix request was received successfully or if an error occurred, and then sends notifications reporting the fix's progress and termination.

Table 16-2: Object—Initiate location fix

Object ID	0x0F02		Initiate location fix	
Operations	✔ Set ✗ Get ✗ Notify			
	Parameters (fields)			
	Len	**Offset**	**Type**	**Description**
03 (Set)	10	0	UINT16	Current object version number • 0x0001=Revision 1
		2	UINT16	Requested location fix type • 1=Standalone. The mobile computes its own location; no PDE is required. • 2=MS-based only. The mobile gets assistance data from the PDE and then computes its location. • 3=MS-assisted only. The mobile provides information to the PDE and then the PDE computes the mobile's location.
		4	UINT16	Performance requirement • Time allowed for satellite acquisition (in seconds) • 1–255
		6	UINT32	Accuracy requirement • Preferred accuracy (in meters). If the preferred accuracy is not available before a timeout occurs, the best value available is returned. • Min: 1; Max: 0xFFFFFFF0 (accuracy disabled)
04 (Set Response)	4	0	UINT16	Current object version number • 0x0001=Revision 1
		2	UINT16	PD error type • See Table 16-22, PD (Position Determination) error types, on page 252 for possible error values. • 0x0000 indicates the Set request was successful (no error)

Start tracking session

Initiates a tracking session composed of multiple location fixes at frequent fixed intervals:

- On request, using Set (03)

Note: The first fix will take longer to perform than subsequent fixes if the ephemeris, almanac, or location data must be updated.

The modem returns a response indicating if the request was received successfully or if an error occurred, and then sends notifications reporting the tracking session's progress and termination.

Table 16-3: Object—Start tracking session

Object ID	0x0F04		Start tracking session	
Operations	✔ Set ✗ Get ✗ Notify			
	Parameters (fields)			
	Len	**Offset**	**Type**	**Description**
03 (Set)	16	0	UINT16	Object version number • 0x0001=Revision 1
		2	UINT16	Requested location fix type • 1=Standalone. The mobile computes its own location; no PDE is required. • 2=MS-based only. The mobile gets assistance data from the PDE and then computes its location. • 3=MS-assisted only. The mobile provides information to the PDE and then the PDE computes the mobile's location.
		4	UINT16	Performance requirement • Time allowed for satellite acquisition (in seconds) • 1–255
		6	UINT32	Accuracy requirement • Preferred accuracy (in meters). If the preferred accuracy is not available before a timeout occurs, the best value available is returned. • Min: 1; Max: 0xFFFFFFF0 (accuracy disabled)
		10	UINT16	Fix count (number of fixes to obtain) • 1–999, *or* • 1000 for continuous mode
		12	UINT32	Fix rate frequency—The number of seconds between the start of one fix and the start of a subsequent fix.

Table 16-3: Object—Start tracking session (Continued)

04 (Set Response)	4	0	UINT16	Current object version number • 0x0001=Revision 1
		2	UINT16	PD error type • See Table 16-22 on page 252 for possible error values. • 0x0000 indicates the Set request was successful (no error)

End location fix session

Instructs the modem to stop the current location fix session:

- On request, using Set (03)

Table 16-4: Object—End location fix session

Object ID	**0x0F05**		End location fix session	
Operations	✔ Set ✗ Get ✗ Notify			
	Parameters (fields)			
	Len	**Offset**	**Type**	**Description**
03 (Set)	4	0	UINT16	Object version number • 0x0001=Revision 1
		2	UINT16	Session type • 0=Location fix session • 1=Reserved
04 (Set Response)	4	0	UINT16	Object version number • 0x0001=Revision 1
		2	UINT16	PD error type • See Table 16-22, PD (Position Determination) error types, on page 252 for possible error values. • 0x0000 indicates the Set request was successful (no error)

Report location fix results

Reports the results of a location fix to registered host clients:

- On request, using Get (01)
- When triggered, if Notify (07) is enabled

Table 16-5: Object—Report location fix results

Object ID	0x0F0A	Report Location Fix Results			
Operations	✗ Set ✔ Get ✔ Notify				
	Parameters (fields)				
	Len	**Offset**	**Type**	**Description**	**Object Ver.**
01 (Get)	0			None	n/a
02 (Get Response) 07 (Notify)	See Object version number (Offset = 0) description for object length				
		0	UINT16	Object version number • 0x0001=Revision 1 Length = 36 • 0x0002=Revision 2 Length = 45	1–2
		2	INT32	Latitude • 2's complement in units of 360 / 2^25 degrees between +90 and -90 degrees	1–2
		6	INT32	Longitude • 2's complement in units of 360 / 2^26 degrees between +90 and -90 degrees	1–2
		10	UINT32	Timestamp • Timestamp of the location response (GPS system time)	1–2
		14	UINT16	Location uncertainty angle • Formatted in integer steps of 5.625 degrees (1=5.625, 2= 11.250, ...)	1–2
		16	UINT16	Location uncertainty A • Index of value in Table 16-24, Position Uncertainty lookup table, on page 253	1–2
		18	UINT16	Location uncertainty P • Index of value in Table 16-24, Position Uncertainty lookup table, on page 253	1–2
		20	UINT16	Fix type • 0=2-D fix • 1=3-D fix	1–2

Table 16-5: Object — Report location fix results (Continued)

		22	UINT16	Height information included in Height (offset 24), Location uncertainty vertical (offset 26), and Uncertainty V (offset 40) fields • 0=No • 1=Yes	1–2
		24	INT16	Height (in meters, offset by +500 meters) • Valid only if Height included field (offset 22) is 1 • Example: Reported height of 700m is actual height of 200m	1–2
		26	UINT16	Location uncertainty Vertical • Valid only if Height Included field (offset 22) is 1 • Index of value in Table 16-24, Position Uncertainty lookup table, on page 253	1–2
		28	UINT16	Velocity information included in Heading (offset 30), Velocity Horizontal (offset 32) and Velocity Vertical (offset 34) fields • 0=No • 1=Yes	1–2
		30	UINT16	Heading • Valid only if Velocity included field (offset 28) is 1 • Formatted in units of 360/2^10 degrees	1–2
		32	UINT16	Velocity Horizontal • Valid only if Velocity included field (offset 28) is 1 • Formatted in integer steps of 0.25 m/s (1=0.25, 2=0.50, ...)	1–2
		34	INT16	Velocity Vertical • Valid only if Velocity included field (offset 28) is 1, and Fix type (offset 20) is 1 • Formatted in integer steps of 0.5 m/s (1=0.5, 2=1.0, ...)	1–2
		36	UINT16	Uncertainty A • Formatted in integer steps of 0.25 meters (1=0.25, 2=0.5, ...)	2
		38	UINT16	Uncertainty P • Formatted in integer steps of 0.25 meters (1=0.25, 2=0.5, ...)	2

Table 16-5: Object—Report location fix results (Continued)

		40	UINT16	Uncertainty V • Valid only if Height included field (offset 22) is 1 • Formatted in integer steps of 0.25 meters (1=0.25, 2=0.5, ...)	2
		42	UINT16	HEPE (Horizontal Estimated Position Error) • Formatted in integer steps of 0.1 meters (1=0.1, 2=0.2, ...)	2
		44	UINT8	Number of satellites used in location fix • Maximum—12	2

Report location fix completed

Reports to registered host clients that a location fix or tracking session has completed:

- Automatically, if Notify (07) is enabled

Table 16-6: Object—Report session completed

Object ID	**0x0F0B**		Report session fix completed	
Operations	✗ Set ✗ Get ✔ Notify			
	Parameters (fields)			
	Len	**Offset**	**Type**	**Description**
07 (Notify)	0			None

Report location fix error

Reports to registered host clients that a location fix or tracking session was prematurely terminated and includes an error code to identify the cause (for example, the session could be terminated by the host application using End location fix session (page 233)):

- Automatically, if Notify (07) is enabled

Table 16-7: Object—Report location fix error

Object ID	**0x0F0C**		Report location fix error	
Operations	✗ Set ✗ Get ✔ Notify			
	Parameters (fields)			
	Len	**Offset**	**Type**	**Description**
07 (Notify)	4	0	UINT16	Object version number • 0x0001=Revision 1

Table 16-7: Object — Report location fix error (Continued)

		2	UINT16	Error code • 0x0000=Offline • 0x0001=No service • 0x0002=No connection • 0x0003=No data • 0x0004=Session busy • 0x0005=Reserved • 0x0006=GPS disabled • 0x0007=Connection failed • 0x0008=Error state • 0x0009=Client ended • 0x000A=UI ended • 0x000B=Network ended • 0x000C=Timeout • 0x000D=Privacy level • 0x000E=Network access error • 0x000F=Fix error • 0x0010=PDE rejected • 0x0011=Traffic channel exited • 0x0012=E911 • 0x0013=Server error • 0x0014=Stale BS information • 0x0015=Resource contention • 0x0016=Authentication parameter failed • 0x0017=Authentication failed - local • 0x0018=Authentication failed - network • 0x1000=VX LCS agent auth fail • 0x1001=Unknown system error • 0x1002=Unsupported service • 0x1003=Subscription violation • 0x1004=Desired fix method failed • 0x1005=Antenna switch • 0x1006=No fix reported due to no Tx confirmation received • 0x1007=Network indicated normal ending of session • 0x1008=No error specified by the network • 0x1009=No resources left on the network • 0x100A=Position server not available • 0x100B=Network reported unsupported protocol version *Continued on next page*

Table 16-7: Object—Report location fix error (Continued)

				Error code (continued) • 0x100C=SS MOLR error—System failure • 0x100D=SS MOLR error—Unexpected data value • 0x100E=SS MOLR error—Data missing • 0x100F=SS MOLR error—Facility not supported • 0x1010=SS MOLR error—SS subscription violation • 0x1011=SS MOLR error—Position method failure • 0x1012=SS MOLR error—Undefined • 0x1013=Control plane's SMLC timeout, may or may not end PD • 0x1014=Control plane's MT guard time expires • 0x1015=End waiting for additional assistance

Report location fix begun

Reports to registered host clients that a location fix session has begun:

- Automatically, if Notify (07) is enabled

Table 16-8: Object—Report location fix begun

Object ID	0x0F0D		Report location fix begun	
Operations	✗ Set ✗ Get ✔ Notify			
	Parameters (fields)			
	Len	**Offset**	**Type**	**Description**
07 (Notify)	0			None

Report position determination failure

Reports to registered host clients that the modem has failed to produce a location fix within the timeout period:

- Automatically, if Notify (07) is enabled

Table 16-9: Object—Report position determination failure

Object ID	0x0F0E		Report position determination failure	
Operations	✗ Set ✗ Get ✔ Notify			
	Parameters (fields)			
	Len	**Offset**	**Type**	**Description**
07 (Notify)	0			None

Report IP address used for TCP/IP

Reports or sets the IP address used when TCP/IP is the transport mechanism for Location Based Services (LBS).

Sets the address:

- On request, using Set (03)

Reports the address:

- On request, using Get (01)
- Automatically after the Set request has been processed, if Notify (07) is enabled. (The Set request is issued, Set response received, and then the Notify is received after the IP address is set)

Table 16-10: Object—Report IP address used for TCP/IP

Object ID	0x0F13		Report IP address used for TCP/IP	
Operations	✔ Set ✔ Get ✔ Notify			
	Parameters (fields)			
	Len	**Offset**	**Type**	**Description**
01 (Get)	0			None
02 (Get Response) **03 (Set)** **07 (Notify)**	6	0	UINT16	Object version number • 0x0001=Revision 1
		2	UINT32	IP address to use when using TCP/IP as the transport mechanism for LBS.
04 (Set Response)	4	0	UINT16	Object version number • 0x0001=Revision 1
		2	UINT16	PA error type • See Table 16-23, PA error types, on page 253 for possible error values. • 0x0000 indicates Set request succeeded (no error)

Report Port ID

Reports or sets the Port ID to use when using TCP/IP as the transport mechanism for Location Based Services.

Sets the Port ID:

- On request, using Set (03)

Reports the Port ID:

- On request, using Get (01)
- Automatically after the Set request has been processed, if Notify (07) is enabled. (The Set request is issued, the Set response is received, and then the Notify is received after the Port ID is set)

Table 16-11: Object—Report Port ID

Object ID	0x0F16		Report Port ID	
Operations	✔ Set ✔ Get ✔ Notify			
	Parameters (fields)			
	Len	**Offset**	**Type**	**Description**
01 (Get)	0			None
02 (Get Response) **03 (Set)** **07 (Notify)**	6	0	UINT16	Object version number • 0x0001=Revision 1
		2	UINT32	Port ID • Valid TCP/IP port number
04 (Set Response)	4	0	UINT16	Object version number • 0x0001=Revision 1
		2	UINT16	PA error type • See Table 16-23, PA error types, on page 253 for possible error values. • 0x0000 indicates the Set request was successful (no error)

Start Keep Warm process

Enables the modem's keep warm functionality causing the modem to begin downloading GPS assistance data from the GPS server:

- On request, using Set (03)

Table 16-12: Object—Start Keep Warm process

Object ID	0x0F1A		Start Keep Warm process	
Operations	✔ Set ✗ Get ✗ Notify			
	Parameters (fields)			
	Len	**Offset**	**Type**	**Description**
03 (Set)	2	0	UINT16	Object version number • 0x0001=Revision 1
04 (Set Response)	4	0	UINT16	Object version number • 0x0001=Revision 1
		2	UINT16	PA error type • See Table 16-23, PA error types, on page 253 for possible error values. • 0x0000 indicates the Set request was successful (no error)

Report Keep Warm process begun

Reports to registered host clients that keep warm processing has begun:

- Automatically, if Notify (07) is enabled

Table 16-13: Object—Report Keep Warm process begun

Object ID	0x0F1B		Report Keep Warm process begun	
Operations	✗ Set ✗ Get ✔ Notify			
	Parameters (fields)			
	Len	**Offset**	**Type**	**Description**
07 (Notify)	0			None

Report Keep Warm process status

Reports the current status of keep warm processing:

- On request, using Get (01)
- Automatically, if Notify (07) is enabled, to all registered host clients

Table 16-14: Object—Report Keep Warm process status

Object ID	0x0F1C		Report Keep Warm process status	
Operations	✗ Set ✔ Get ✔ Notify			
	Parameters (fields)			
	Len	**Offset**	**Type**	**Description**
01 (Get)	0			None
02 (Get Response) **07 (Notify)**	6	0	UINT16	Object version number • 0x0001=Revision 1
		2	UINT16	Keep Warm processing state • 0=Disabled • 1=Enabled
		4	UINT16	Warm status • 0=Not warm • 1=GPS is in warm state

Report Keep Warm process complete

Reports to registered host clients that keep warm processing has finished:

- Automatically, if Notify (07) is enabled

Table 16-15: Object—Report Keep Warm process complete

Object ID	0x0F1D		Report Keep Warm process complete	
Operations	✗ Set ✗ Get ✔ Notify			
	Parameters (fields)			
	Len	**Offset**	**Type**	**Description**
07 (Notify)	2	0	UINT16	Object version number • 0x0001=Revision 1

Stop Keep Warm process

Instructs the modem to stop Keep Warm processing:

- On request, using Set (03)

Table 16-16: Object—Stop Keep Warm process

Object ID	**0x0F1E**		Stop Keep Warm process	
Operations	✔ Set ✗ Get ✗ Notify			
	Parameters (fields)			
	Len	**Offset**	**Type**	**Description**
03 (Set)	2	0	UINT16	Object version number • 0x0001=Revision 1
04 (Set Response)	4	0	UINT16	Object version number • 0x0001=Revision 1
		2	UINT16	PA error type • See Table 16-23, PA error types, on page 253 for possible error values. • 0x0000 indicates the Set request was successful (no error)

Report Satellite Information

Reports information for satellites in view during the latest location fix, and identifies the satellites actually used for the fix:

- On request, using Get (01)
- Automatically, if Notify (07) is enabled, when a location fix finishes

Note: The modem returns a generic CnS 'unavailable' error message if there is no location fix to report.

Table 16-17: Object—Report Satellite Information

Object ID	0x0F1F		Report Satellite Information
Operations	✗ Set ✔ Get ✔ Notify		
	Parameters (fields)		
	Len	**Offset**	**Type**	**Description**
01 (Get)	0			None
02 (Get Response) **07 (Notify)**	63	0	UINT16	Object version number • 0x0001=Revision 1
		2	UINT8	Number of satellites in view • Maximum—12
		3	Array[12]	Satellite information • Space available for 12 satellites • If fewer than 12 satellites, content for unused array indexes is undefined
		Array[0..11]+0	UINT8	Satellite number • Bit 7—Satellite used in latest fix (0 = No, 1 = Yes) • Bit 0–6—Satellite number (0–32)
		Array[0..11]+1	UINT8	Satellite elevation (in degrees) • 0–90
		Array[0..11]+2	UINT16	Satellite azimuth (in degrees) • 0–360
		Array[0..11]+4	UINT8	Signal to Noise Ratio (in dB) • 0–99

Clear GPS Assistance Data

Clears one or more types of assistance data from the modem. This simulates a cold start state and causes the modem to search for assistance data the next time a GPS session starts.

Note: Use this object for testing purposes only.

The data is cleared:

- On request, using Set (03)

Table 16-18: Object—Clear GPS Assistance Data

Object ID	0x0F20		Clear GPS Assistance Data	
Operations	✔ Set ✗ Get ✗ Notify			
	Parameters (fields)			
	Len	**Offset**	**Type**	**Description**
03 (Set)	9	0	UINT16	Object version number • 0x0001=Revision 1
		2	UINT8	Delete ephemeris data • 0=No • 1=Yes
		3	UINT8	Delete almanac data • 0=No • 1=Yes
		4	UINT8	Delete position data • 0=No • 1=Yes
		5	UINT8	Clear standalone self-learning database • 0=No • 1=Yes
		6	UINT8	Clear standalone serving system database • 0=No • 1=Yes
		7	UINT8	Clear standalone time reference • 0=No • 1=Yes
		8	UINT8	Reset MSB throttling parameters • 0=Do nothing • 1=Reset MS-based Throttling Parameters

Manage location fix settings

Stores location fix settings from the modem allowing them to persist across sessions, and retrieves the settings to initiate a GPS session using Initiate location fix (page 231) or Start tracking session (page 232).

Stores settings in the modem:

- On request, using Set (03)

Retrieves settings from the modem:

- On request, using Get (01)

Table 16-19: Object—Manage location fix settings

Object ID	0x0F21		Manage location fix settings
Operations	✔ Set ✔ Get ✗ Notify		
	Parameters (fields)		
	Len	**Offset**	**Type**	**Description**
01 (Get)	0			None
02 (Get Response) **03 (Set)**	21	0	UINT16	Object version number • 0x0001=Revision 1
		2	UINT8	User can read/write fix type • Bit 1—Read allowed (0 = No, 1 = Yes) • Bit 2—Write allowed (0 = No, 1 = Yes)
		3	UINT16	Fix type • 1=Standalone • 2=MS-based only • 3=MS-assisted only
		5	UINT8	User can read/write performance setting • Bit 1=Read allowed (0 = No, 1 = Yes) • Bit 2=Write allowed (0 = No, 1 = Yes)
		6	UINT16	Performance setting—Time allowed for satellite acquisition, in seconds • 1–255
		8	UINT8	User can read/write accuracy setting • Bit 1—Read allowed (0 = No, 1 = Yes) • Bit 2—Write allowed (0 = No, 1 = Yes)
		9	UINT32	Accuracy preferred (in meters)
		13	UINT8	User can read/write fix count setting • Bit 1—Read allowed (0 = No, 1 = Yes) • Bit 2—Write allowed (0 = No, 1 = Yes)
		14	UINT16	Fix count (number of fixes to obtain) • 1–999, *or* • 1000 for continuous mode
		16	UINT8	User can read/write fix rate setting • Bit 1—Read allowed (0 = No, 1 = Yes) • Bit 2—Write allowed (0 = No, 1 = Yes)
		17	UINT32	Fix rate—frequency of fixes (in seconds)
04 (Set Response)	7	0	UINT16	Object version number • 0x0001=Revision 1

Table 16-19: Object—Manage location fix settings (Continued)

		2	UINT8	Fix type write status • 0=Success • 0=Failure (field not writeable)
		3	UINT8	Performance write status • 0=Success • 0=Failure (field not writeable)
		4	UINT8	Accuracy write status • 0=Success • 0=Failure (field not writeable)
		5	UINT8	Fix count write status • 0=Success • 0=Failure (field not writeable)
		6	UINT8	Fix rate write status • 0=Success • 0=Failure (field not writeable)

Report network-initiated location fix request

Notifies the host that a network-initiated location fix is being requested. If the network requires a response, the host uses a Set request to accept, reject, or cancel the location fix request.

Note: If the network requires a response and the timeout period expires before one is sent with the Set request, the modem uses the default value of 'Cancel request'.

Receives notice of a network-initiated location fix request:

- Automatically, if Notify (07) is enabled

Submits response to network:

- If required by the network, using Set (03)

Table 16-20: Object—Report network-initiated location fix request

Object ID	0x0F22		Report network-initiated location fix request		
Operations	✔ Set ✗ Get ✔ Notify				
	Parameters (fields)				
	Len	**Offset**	**Type**	**Description**	**Object Ver.**
03 (Set)	4	0	UINT16	Object version number • 0x0001=Revision 1 • 0x0002=Revision 2	1–2

Table 16-20: Object — Report network-initiated location fix request (Continued)

		2	UINT8	Request ID assigned by the modem in the notification • 0x00–0xFF	1–2
		3	UINT8	Request response • 0x00=Reject fix request • 0x01=Accept fix request • 0x02=Cancel request (Default value)	1–2
04 (Set Response)	3	0	UINT16	Object version number • 0x0001=Revision 1 • 0x0002=Revision 2	1–2
		2	UINT8	Error code (result of Set request) • 0x00=No error (Set request successful) • 0x01=Handle mismatch (incorrect Request ID) • 0x02=User response timed out • 0x03=No user response was required	1–2
07 (Notify)	See Object version number (Offset = 0) description for object length				
		0	UINT16	Object version number • 0x0001=Revision 1 Length = 138 • 0x0002=Revision 2 Length = 222	1–2
		2	UINT8	Request ID assigned by the modem. This ID must be used in the Set request to link back to this location fix request. • 0x00–0xFF	1–2
		3	UINT16	Timeout period in seconds • 0x0000–0xFFFF	1–2
		5	UINT8	Notification type • 0x00=Notify only • 0x0=User response required	1–2
		6	UINT8	Requestor Network ID type • 0x00=Unknown • 0x01=Logical name • 0x02=MSISDN • 0x03=Email address • 0x04=URL • 0x05=SIP URL • 0x06=IMS public identity • 0x07=MIN • 0x08=MDN	1–2

Table 16-20: Object—Report network-initiated location fix request (Continued)

		7	UINT8	Requestor ID string length (number of UCS2 characters in Requestor ID field) • Min—0; Max—32	1–2
		8	UINT8[64]	Requestor ID string—The name of the party requesting the position fix. • UCS2 format • 64 bytes • Up to 32 characters (2 bytes / character)—See previous parameter for actual length • Not null-terminated • Ignore unused trailing bytes	1–2
		72	UINT8	Client name type • 0x00=Unknown • 0x01=Logical name • 0x02=MSISDN • 0x03=Email address • 0x04=URL • 0x05=SIP URL • 0x06=IMS public identity • 0x07=MIN • 0x08=MDN	1–2
		73	UINT8	Client name string length (number of UCS2 characters in Client name string field) • Min—0; Max—32	1–2
		74	UINT8[64]	Client name string—The name of the local client application that should handle the request. • UCS2 format • 64 bytes • Up to 32 characters (2 bytes / character)—See previous parameter for actual length • Not null-terminated • Ignore unused trailing bytes	1–2
		138	UINT8	Default user response to use if notification type (offset 5) is 0x01 (User response required) and no response is given by the user. • 0x00=Unknown • 0x01=Accept • 0x02=Reject	2

Table 16-20: Object—Report network-initiated location fix request (Continued)

		139	UINT8	Codeword string length (number of UCS2 characters in Codeword string field) • Min—0; Max—20	2
		140	UINT8[40]	Codeword string—A string passed from the network that the host application can use to verify the network entity requesting the device's location. • UCS2 format • 40 bytes • Up to 20 characters (2 bytes / character)—See previous parameter for actual length • Not null-terminated • Ignore unused trailing bytes	2
		180	UINT8	Service type • 0x00=Emergency services • 0x01=Emergency alert services • 0x02=Person tracking • 0x03=Fleet management • 0x04=Asset management • 0x05=Traffic congestion reporting • 0x06=Roadside assistance • 0x07=Routing to nearest commercial enterprise • 0x08=Navigation • 0x09=City sightseeing • 0x0A=Localized advertising • 0x0B=Mobile Yellow Pages • 0x0C=Traffic and public transportation information • 0x0D=Weather • 0x0E=Asset and service finding • 0x0F=Gaming • 0x10=Find your friend • 0x11=Dating • 0x12=Chatting • 0x13=Route finding • 0x14=Where am I • 0xFF=Unknown	2
		181	UINT8	External client ID string length (number of UCS2 characters in External client ID field) • Min—0; Max—20	2

Table 16-20: Object—Report network-initiated location fix request (Continued)

		182	UINT8[40]	External client ID • UCS2 format • 40 bytes • Up to 20 characters (2 bytes / character)—See previous parameter for actual length • Not null-terminated • Ignore unused trailing bytes	2

Report Auto Track feature state

Reports or sets the GPS Auto Track state and configuration.

Configuration changes take effect immediately. If there is no active GPS session, a session starts automatically if all requirements are met for the new settings.

An active GPS session is automatically stopped if:

- Auto Track is disabled using this object's Set operation
- Other LBS objects or AT commands force the session to end

Auto Track is enabled/disabled or reconfigured:

- On request, using Set (03)

Auto Track state and configuration is returned:

- On request, using Get (01)

Table 16-21: Object—Report Auto Track feature state

Object ID	0x0F24		Report Auto Track feature state	
Operations	✔ Set ✔ Get ✗ Notify			
	Parameters (fields)			
	Len	**Offset**	**Type**	**Description**
01 (Get)	2	0	UINT16	Object version number • 0x0001=Revision 1
02 (Get Response) **03 (Set)**	19	0	UINT16	Object version number • 0x0001=Revision 1
		2	UINT8	Enable/disable AutoTracking • 0x00=Disable • 0x01=Enable
		3	UINT8	GPS session start control • 0x01=GPS session can start after power up
		4	UINT8	AutoTracking trigger type • 0x01=AutoTracking starts automatically (no host interaction required)

Table 16-21: Object—Report Auto Track feature state (Continued)

		5	UINT16	Requested location fix type • 1=Standalone. The mobile computes its own location; no PDE is required. • 2=MS-based only. The mobile gets assistance data from the PDE and then computes its location. • 3=MS-assisted only. The mobile provides information to the PDE and then the PDE computes the mobile's location.
		7	UINT16	Performance requirement • Time allowed for satellite acquisition (in seconds) • 1–255
		9	UINT32	Accuracy requirement • Preferred accuracy (in meters). If the preferred accuracy is not available before a timeout occurs, the best value available is returned. • Min: 1; Max: 0xFFFFFFF0 (accuracy disabled)
		13	UINT16	Fix count (number of fixes to obtain) • 1–999, *or* • 1000 for continuous mode
		15	UINT32	Fix rate frequency—The number of seconds between the start of one fix and the start of a subsequent fix.
04 (Response)	3	0	UINT16	Object version number • 0x0001=Revision 1
		2	UINT8	Response status • 0x00=Success • 0x01=Error, GPS feature is not enabled • 0x02=Error, Fix type is not enabled. Check device configuration. • 0x03=Error, Parameter option not supported by module

Reference tables

The following tables detail possible error or lookup values referenced by various commands.

Table 16-22: PD (Position Determination) error types

Error code	Description
0x0000	No error
0x0001	Internal modem error
0x0002	Bad service type
0x0003	Bad session type
0x0004	Invalid privacy
0x0005	Invalid data download
0x0006	Invalid network access
0x0007	Invalid operation mode
0x0008	Invalid number of fixes
0x0009	Invalid server info
0x000A	Invalid timeout
0x000B	Invalid QoS parameter
0x000C	No session active
0x000D	Session already active
0x000E	Session busy
0x000F	Phone is offline
0x0010	CDMA lock error
0x0011	GPS lock error
0x0012	Invalid state
0x0013	Connection failure
0x0014	No buffers available
0x0015	Searcher error
0x0016	Cannot report now
0x0017	Resource contention
0x0018	Mode not supported
0x0019	Authentication failed
0x001A	Other error
0x001B	Fix rate for tracking (time between fixes) is too large

Table 16-23: PA error types

Error code	Description
0x0000	No error
0x0001	Internal error
0x0002	Invalid client ID
0x0003	Error in parameter to set
0x0004	Error in lock type
0x0005	Phone is in offline state
0x0006	A command to set parameter is already active
0x0007	Incorrect application-specific information
0x0008	Unknown error

Table 16-24: Position Uncertainty lookup table

Position Uncertainty Index	Std. Deviation for position uncertainty (m)
0x0000	0.5
0x0001	0.75
0x0002	1
0x0003	1.5
0x0004	2
0x0005	3
0x0006	4
0x0007	6
0x0008	8
0x0009	12
0x000A	16
0x000B	24
0x000C	32
0x000D	48
0x000E	64
0x000F	96
0x0010	128
0x0011	192
0x0012	256

Table 16-24: Position Uncertainty lookup table (Continued)

Position Uncertainty Index	Std. Deviation for position uncertainty (m)
0x0013	384
0x0014	512
0x0015	768
0x0016	1024
0x0017	1536
0x0018	2048
0x0019	3072
0x001A	4096
0x001B	6144
0x001C	8192
0x001D	12288
0x001E	> 12288
0x001F	Not computable

A: Acronyms

Table A-1: Acronyms

Acronym	Definition
ADN	Abbreviated Dialing Numbers phonebook
APN	Access Point Name
ARFCN	Absolute Radio Frequency Channel Number
BER	Bit Error Rate
CDMA	Code Division Multiple Access
CHAP	Challenge Handshake Authentication Protocol
CHV (1 and 2)	Card Holder Verification
CnS	Control and Status language
CPHS	Common PCN Handset Specification
CPHS MN	CPHS voice Mailbox Numbers
CSPDN	Circuit Switched Public Data Network
CWE	Core Wireless Engine
DL	Downlink (network to mobile)
DM	Diagnostic Monitor (Qualcomm)
DNS	Domain Name Server
DTMF	Dual Tone Multiple Frequency
DUN	Dial-Up Networking
EDGE	Enhanced Data rates for GSM Evolution
EGSM	Extended GSM
EPI	Evaluation Precedence Index
ERMES	Enhanced Radio Messaging System
FDN	Fixed Dialing Number phonebook
FPGA	Field Programmable Gate Array
FSN	Factory Serial Number
GPRS	General Packet Radio Service
GSM	Global System for Mobile Communications
GTP	GPRS Tunneling Protocol
HIP	Host Interface Protocol

Table A-1: Acronyms (Continued)

Acronym	Definition
HLR	Home Location Register
HSDPA	High Speed Downlink Packet Access
HSUPA	High Speed Uplink Packet Access
IMEI	International Mobile Equipment Identity
IMSI	International Mobile Station Identity International Mobile Subscriber Identity
IPv4 / IPv6	Internet Protocol version 4 / version 6
LAC	Location Area Code
LLC	Logical Link Control
LND	Last Numbers Dialed phonebook
LNM	Last Numbers Missed phonebook
LNR	Last Numbers Received phonebook
MCC	Mobile Country Code
MEP	Mobile Equipment Personalization
MMI	Man Machine Interface
MNC	Mobile Network Code
MSIN	Mobile Station Identification Number
MSISDN	Mobile Subscriber International Subscriber Identify Number
NDIS	Network Driver Interface Specification
NSAPI	Network Service Access Point Identifier
NV	Non-Volatile memory
NVRAM	Non-Volatile RAM (memory)
PAP	Password Authentication Protocol
PDP	Packet Data Profile
PGSM	Primary GSM
PLMN	Public Land Mobile Network
PPP	Point to Point Protocol
PRI	Product Release Information
PS	Packet Switched
PSPDN	Packet Switched Public Data Network
PSTN	Public Switched Telephone Network

Table A-1: Acronyms (Continued)

Acronym	Definition
QoS	Quality of Service
RAT	Radio Access Technology
RLC	Radio Link control
RSSI	Received Signal Strength Indicator
SC	Service Center
SDN	Service Dialing Numbers phonebook
SDU	Service Data Unit
SIM	Subscriber Identity Module (GSM)
SM	SIM
SMS	Short Message Service
SNDCP	Sub-Network Dependent Convergence Protocol
SPN	Service Provider Name
SRES	Signed RESponse (authentication value returned by the SIM or USIM in 2G Authentication and Key Agreement)
TFT	Thin Film Transistor
UARFCN	UMTS Absolute Radio Frequency Channel Number
UCS2	Universal Character Set 2
UE	User equipment
UL	Uplink (mobile to network)
USB	Universal Serial Bus
UMTS	Universal Mobile Telecommunications System
VPN	Virtual Private Network
WINS	Windows Internet Name Service


SIERRA
WIRELESS