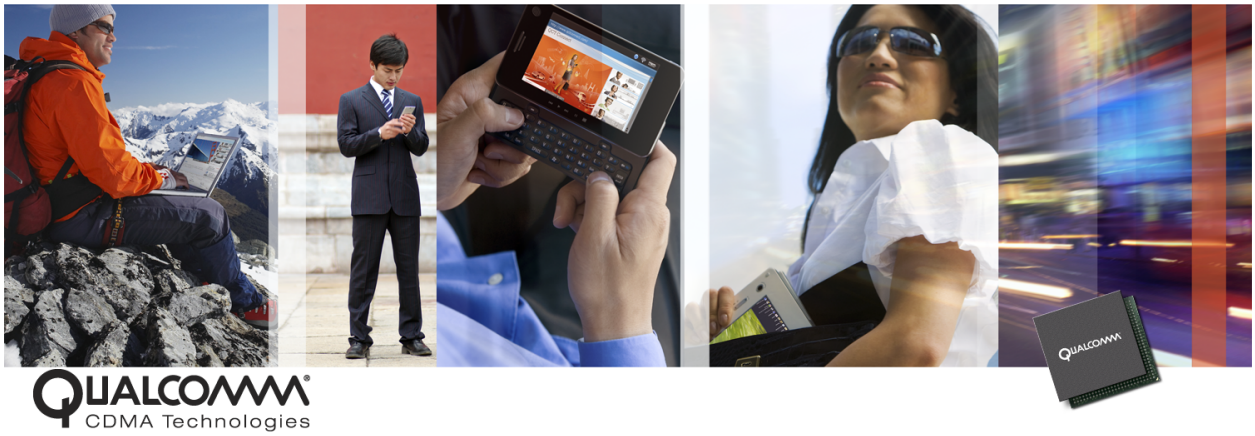

QUALCOMM CDMA Technologies

QMI Location Service (QMI_LOC)

Major Version 2, Minor Version 6

Specification

80-VB816-17 D

December 16, 2011

Submit technical questions at:

https://support.cdmatech.com

QUALCOMM Incorporated
5775 Morehouse Drive
San Diego, CA 92121-1714
U.S.A.

Contents

List of Figures

List of Tables

Revision History

Revision	Date	Description
A	Apr 2011	First draft release.
B	Jul 2011	Second draft release. Added items that were missing in the initial draft release; changed several message and symbol names for clarity.
C	Oct 2011	Initial release of the completed (nondraft) API documentation.
D	Dec 2011	Updates for this revision include minor version 3 through minor version 6 Updated: • Mandatory TLV in Section 3.32.1 • Optional TLV in Section 3.33.2 • Section 3.36.1 heading to QMI_LOC_SET_SERVER_REQ • Description of message QMI_LOC_SET_NMEA_TYPES Added new TLVs: • ID of the Application Sent with this Request • ID of the Application that Sent the Position Request • Position Source Added new Messages: • QMI_LOC_INJECT_SUPL_CERTIFICATE (Section 3.62) • QMI_LOC_DELETE_SUPL_CERTIFICATE (Section 3.63) • QMI_LOC_SET_POSITION_ENGINE_CONFIG_PARAMETERS (Section 3.64) • QMI_LOC_GET_POSITION_ENGINE_CONFIG_PARAMETERS (Section 3.65)

1 Introduction

1.1 Purpose

This specification documents Major Version 2 of the Qualcomm Messaging Interface Location Service (QMI_LOC). QMI_LOC is the second QMI interface provided for accessing Location Services on Qualcomm devices. The first QMI interface, QMI Position Determination Service (QMI_PDS), was designed specifically for MDM devices. QMI_LOC, however, is designed to meet the needs of both MSM and MDM devices of the future. QMI_LOC has addressed some significant issues present in QMI_PDS and will be used to introduce new features in the future. QMI_PDS is now a legacy API; supported only until existing customers can be moved to QMI_LOC. All new MSM and MDM customers should use QMI_LOC, since QMI_PDS is planned for deprecation as soon as existing customers are moved to QMI_LOC.

New feature enhancements to QMI_PDS v1.x will be minimal. Significant new features will be added to QMI_LOC v2.x, instead.

QMI_LOC begins with Major Version 2 for two reasons. First, it is desirable to make it clear that QMI_LOC is built upon the older QMI_PDS service. Second, it is desirable to make it clear that QMI_LOC v2.x is designed specifically to work with Qualcomm's newer High Level Operating System (HLOS) C API: Loc API v2.0.

QMI_PDS v1.0 is not supported on MSM and APQ platforms. QMI_PDS v1.0 and QMI_LOC v2.x may both be available on early MDM9x15 platforms to provide existing QMI_PDS v1.0 customers with sufficient time to move to QMI_LOC v2.x, but QMI_PDS v1.0 will be deprecated on all platforms following MDM9x15 (and possibly later releases of MDM9x15 as well). No customers should begin using QMI_PDS v1.0 from now on. Any existing QMI_PDS v1.0 customers should immediately begin moving to QMI_LOC v2.x.

QMI_LOC provides applications running on a tethered device or on the HLOS's side of a dual processor MSM with commands related to location and position determination, including commands to do the following:

- Determine current position
- Manage configurations for the MSM GPS service
- Inject external assistance data to improve performance (coarse position, time, etc.)
- Respond to network-initiated requests for position fixes or measurements

1.2 Scope

This document is intended for software developers who are developing code to interact with the positioning engine in Qualcomm MSM and MDM devices from a host or application processor.

This document provides the following details about QMI_LOC:

- Theory of operation
- Message formats, syntax, and semantics
- Error messages

1.3 Conventions

Function declarations, function names, type declarations, and code samples appear in a different font. For example, `#include`.

Parameter types are indicated by arrows:

→ Designates an input parameter

← Designates an output parameter

↔ Designates a parameter used for both input and output

1.4 References

Table 1-2 lists reference documents, which may include Qualcomm documents and non-Qualcomm standards and resources. Reference documents that are no longer applicable are deleted from this table; therefore, reference numbers might not be sequential.

Table 1-2 Reference documents and standards

Ref.	Document	
Qualcomm		
Q1	*Application Note: Software Glossary for Customers*	CL93-V3077-1
Q2	*Qualcomm MSM™ Interface (QMI) Architecture*	80-VB816-1
Q3	*gpsOne® Interface Specification and Operational Description*	80-V1960-2
Q4	*QMI Position Determination Service (QMI_PDS)*	80-VB816-8
Standards		
S1	*The NMEA 0183 Protocol*	NMEA-0183
S2	*IEEE Standard for Binary Floating-Point Arithmetic*	IEEE Std 754-1985
S3	*World Geodetic System* (http://earth-info.nga.mil/GandG/wgs84)	1984 (updated 2004)
S4	*ETSI TS 101 109 Ver. 7.2.0: Digital cellular telecommunications system (Phase2+); Universal Geographical Area Description (GAD)*	3GPP TS 03.32 version 7.2.0 Release 1998
Resources		
R1	*Understanding GPS: Principles and Applications, Second Edition*	ISBN-10: 1-58053-894-0

1.5 Technical Assistance

For assistance or clarification on information in this guide, submit a case to Qualcomm CDMA Technologies at https://support.cdmatech.com.

If you do not have access to the CDMATech Support Services website, register for access or send email to support.cdmatech@qualcomm.com.

1.6 Acronyms

For definitions of terms and abbreviations, see [Q1]. Table 1-3 lists terms that are specific to this document.

Table 1-3 Acronyms

Acronym	Definition
A-GPS	assisted global positioning system
AFLT	advanced forward link trilateration
AGNSS	assisted GNSS
AP	access point
APN	access point name
CP	control point
DOP	dilution of precision
ECID	exclusive chip ID
EOTD	enhanced observed time difference
ETSI	European Telecommunications Standards Institute
GGA	NMEA string containing position information
GLONASS	Global Navigation Satellite System (Russian version of GPS)
GNSS	global navigation satellite services
GPS	global positioning system
GSA	NMEA string containing active satellite vehicle and DOP information
GSV	NMEA string containing satellite vehicle information
HDOP	horizontal dilution of precision
HEPE	horizontal estimated position error (geocaching)
HLOS	high level operating system
IDL	interface description language
LOC	location
LPM	low power mode
MAC	message authentication code
MDN	mobile directory number
MI	mobile-initiated
MIN	mobile identification number
MO	mobile-originated
MPC	Mobile Positioning Center
MS	mobile station
MT	mobile-terminated
NI	network initiated
NMEA	National Marine Electronics Association
OTDOA	observed time delay of arrival

Table 1-3 Acronyms (cont.)

Acronym	Definition
PDE	position determination entity
PDOP	position dilution of precision
PDS	position determination service
PQXFI	proprietary Qualcomm extended fix information
PRN	pseudorandom noise
PSTIS	proprietary SnapTrack, Inc. session
QMI	Qualcomm messaging interface
QoP	quality of position
QoS	quality of service
RMC	recommended minimum specific GPS/transit data
SBAS	satellite-based augmentation system
SFT	solve for time
SLP	SUPL location platform
SNR	signal-to-noise ratio
SPI	stationary position indicator
SS	supplementary service
SSID	service set identifier
SUPL	secure user plane location
SV	satellite vehicle
TLV	type-length-value
TTFF	time to first fix
UMTS	universal mobile telecommunications system
UMTS-CP	UMTS control plane
URL	universal resource locator
UTC	universal time coordinated
VDOP	vertical dilution of precision
VTG	NMEA string containing velocity information
VX	Version x (User Plane Protocol versions 1 and 2)
WGS	world geodetic system
XTRA	extended receiver assistance (gpsOne®)
XTRA-T	XTRA terrestrial

2 Theory of Operation

2.1 Generalized QMI Service Compliance

The QMI_LOC service complies with the generalized QMI service specification, including the rules for messages, indications and responses, byte ordering, arbitration, constants, results, and error code values, as described in the QMI Generalized Message Protocol section of [Q2].

As with other QMI services, the data types of values defined and used in QMI_LOC are assumed to be unsigned integers, unless explicitly stated otherwise. Also, values defined as strings do not include NULL terminating characters unless explicitly stated.

Any extensions to the generalized QMI service theory of operation are noted in the subsequent subsections of this chapter.

QMI_LOC messages are defined with an Interface Description Language (IDL) file. An IDL compiler generates the interface code, which marshals messages to/from wire formats.

2.2 LOC Service Type

The LOC service is assigned QMI service type 16.

2.3 Message Definition Template

2.3.1 Mandatory Result Type-Length-Value

All response messages returned by the QMI_LOC service are identical. All of them contain a mandatory TLV 0x02, which contains a QMI result code and a QMI error code. In the case of an error (i.e., QMI_RESULT_FAILURE), the QMI error code field will contain a specific error code that can be used to accurately pinpoint the source of the error. When the result code does not contain an error (i.e QMI_RESULT_SUCCESS), the QMI error code field can be ignored.

The format of a QMI_LOC response message (including the optional TLV that will only be present if qmi_result equals QMI_RESULT_FAILURE) is shown in the following tables.

Name	Version last modified
Result Code	Corresponding messages "Version Introduced"

Field	Field value	Parameter	Size (byte)	Description
Type	0x02		1	Result code
Length	4		2	
Value	→	qmi_result	2	Result code: • QMI_RESULT_SUCCESS • QMI_RESULT_FAILURE
		qmi_error	2	Error code; possible error code values are described in the error code section of each message definition

2.4 Backward Compatibility and Version Negotiation

QMI services have a major version and a minor version. Minor version upgrades must be backward compatible with previous minor versions of the same major version. Major version upgrades are not backward compatible. QMI_LOC will maintain backward compatibility for the foreseeable future. Only minor version upgrades are planned at this time.

Backward compatibility of QMI_LOC means:

- Existing Type-Length-Value (TLV) items cannot be modified
- TLVs cannot be deleted
- Mandatory TLVs must remain mandatory and optional TLVs must remain optional
- New optional TLVs may be added to existing messages, but new mandatory TLVs are not allowed
- Messages cannot be deprecated
- API behavior cannot be modified in non-backward compatible ways
- Unrecognized messages (which occur when one side of the interface is upgraded before the other) shall be ignored.
- Unrecognized TLVs within a message shall also be ignored.
- Optional TLVs shall not be treated as mandatory.
- Clients and services shall not misbehave (i.e., cause crashes or unexpected behavior) when sent messages that contain a subset of optional TLVs that the client or service does not expect.

2.5 Asynchronous Messaging Paradigm

QMI_LOC is specifically designed for asynchronous messaging. This is the reason why Response (RESP) messages only contain ACK/NAK status. All asynchronous events or location engine status information are provided to clients in Indication (IND) messages.

The QMI interface does not guarantee delivery of every message. Because of this, it is possible that messages may occasionally be dropped. Clients must be designed in such a way that they can recover from this rare event. This is one of the advantages of an asynchronous interface: it allows for a more robust design, especially when the interface is not 100% reliable.

2.6 Input Message Queuing

The QMI infrastructure allows multiple QMI messages to be sent to a service in a burst without waiting for a response. This feature can present issues for the QMI_LOC service, since the QMI_LOC service is built on top of a Global Navigation Satellite Services (GNSS) subsystem that has limited ability to process commands in parallel. QMI_LOC can respond to requests (both at the time of the RESP and at the time of the IND) with an error. This error can include a busy indication, which means that QMI_LOC is in a state where this request cannot be serviced. This is most likely to occur when a client sends multiple commands of a similar type (e.g., commands that SET or GET the GNSS system state) back-to-back. Clients must handle this case carefully.

The easiest or safest way for clients to handle this is to send only one command (which returns an IND message) at a time, waiting for the IND before sending another. Another option is for clients to send bursts of messages as long as the client does not care whether they are executed in order, since one or more messages in the burst may be rejected and thus return an error while the others are executed. The decision as to whether to send one command at a time or send in bursts is up to the client, but clients must be prepared to handle the case of one or more commands in a burst being rejected if the client chooses to send in bursts.

The gpsOne® drivers delivered for current HLOSs send only one command at a time. The gpsOne team strongly recommends this model until such time that the limitations on parallel command execution in the GNSS subsystem are removed.

These restrictions may be relaxed in the future (when limitations in the underlying GNSS engine are removed), but are necessary at this time to avoid system failure.

2.7 Error Messages

The RESP message contains a result and an error code. If the result indicates a failure, the error code will contain the specific reason for the failure. If the RESP message for a particular request indicates a failure, the QMI_LOC client should not expect any further indications corresponding to that request. If the RESP message indicates, success it implies that the QMI_LOC service accepted the request for processing. The QMI_LOC service sends the actual result of processing the request in an IND message. The IND message corresponding to the request will contain a mandatory status field. The status field indicates if the request was successful, or in the case of a failure, specifies the cause of the failure.

2.8 QMI_LOC Design Fundamentals

QMI_LOC v2.2 is designed a bit differently than QMI_PDS. The major design changes were needed to overcome deficiencies and robustness issues found in QMI_PDS, as it was commercialized across several targets.

The primary changes in overall QMI_LOC API philosophy vs. QMI_PDS are:

- In QMI_PDS, results from Request (REQ) messages were returned in a RESP message. In QMI_LOC, only acknowledgment of receipt of the REQ message is returned in a RESP message. The actual results are returned in an IND message.
- QMI_LOC has a concept of solicited/unsolicited IND messages. Solicited IND messages are those that are output in response to a REQ message. Unsolicited IND messages are also known as asynchronous events. Solicited IND messages are sent only to the client that sent the REQ message that triggered the IND message. Unsolicited IND messages are multicast to all clients that have registered to receive the particular IND message. Clients need not register for receipt of solicited IND messages; only for unsolicited IND messages.
- The QMI_PDS_EVENT_REPORT_IND message was overloaded in QMI_PDS, with each TLV related to a different event. In QMI_LOC, each different asynchronous event is output with its own QMI message to allow for future extensibility.
- In QMI_PDS, it is often not obvious what to expect in response to a REQ message. QMI_LOC makes things more explicit with an over-arching rule that there is an IND message for (almost) every REQ message. Every REQ message has a corresponding RESP and IND (with a few minor exceptions), even if the IND contains nothing more than a success/error indicator. The exceptions are: QMI_LOC_INFORM_CLIENT_REVISION, QMI_LOC_REG_EVENTS, QMI_LOC_START, and QMI_LOC_STOP.
- Fix criteria has been changed in QMI_LOC to allow more freedom for QMI_LOC to choose the optimal positioning method "under the hood."
- Fix criteria is global and applies to all clients in QMI_PDS. Fix criteria is local to a client in QMI_LOC (with limitations to be described later in this document).
- Fix criteria and position fix START requests are independent of each other in QMI_PDS. Fix criteria is (an optional) part of the START request in QMI_LOC.
- Positioning mode has been made into a separate message in QMI_LOC so that it can be limited to only privileged clients. Allowing clients to select the positioning mode severely impairs QMI_LOC's ability to choose the best positioning method and to support simultaneous positioning by multiple clients, since positioning modes are typically mutually exclusive. The Positioning Mode message should not be used except for testing purposes.
- QMI_LOC takes advantage of the Optional TLV concept in QMI to allow clients to omit parameters (such as fix criteria) and thus allow QMI_LOC to use defaults for anything that the client does not specify.

2.9 QMI_LOC Fundamental Positioning Concepts

2.9.1 GNSS

GNSS uses a network of orbiting satellites to provide the MSM with accurate location measurements. See [R1] for a detailed explanation.

A wireless MSM device supporting QMI_LOC provides control and accessibility to the GNSS functionality of the device.

2.9.2 LOC Methods

Various methods exist for determining position. For the definition of each of the methods supported by the Mobile Station (MS), see Section 2.3 of [Q3].

2.9.3 Multiple-Client Support

QMI_LOC supports multiple controlling clients, each connected as a separate QMI control point. Different position fix criteria may be specified by each client. Some clients may not even be positioning clients. Some clients may exist solely for the purpose of providing assistance data to the GNSS subsystem, such as an eXTended Receiver Assistance (XTRA) client.

Each QMI_LOC positioning client may specify a different desired fix criteria. QMI_LOC will attempt to satisfy all of the fix criteria of each client, as long as the fix criteria are not mutually exclusive. When mutually exclusive requests are made, QMI_LOC is forced to select the closest set of criteria that meet the client's requirements and still allow all clients to be serviced. Because of this, many fix criteria items are considered optional or desired parameters. QMI_LOC cannot guarantee that all criteria will be honored when there are multiple clients attempting to use QMI_LOC simultaneously.

When multiple clients request position fixes, QMI_LOC services those requests via an internal deadline-first scheduling algorithm. This algorithm works well for single-shot position fix clients or position fix clients with large TBFs (Time-Between-Fixes), but does not work so well for servicing multiple, periodic positioning clients with similar (or identical) TBFs. The deadline-first scheduling algorithm can cause the actual TBF for each client to be greater than what the client requested, especially when multiple clients are attempting to obtain 1-Hz position fixes. For example, if two clients request a TBF of 1 Hz, the deadline-first scheduling algorithm will service the clients in an alternating fashion such that each client gets a fix every two seconds.

2.9.4 Single-Shot Position Fix Sessions

A QMI control point may request a single fix (i.e., single-shot fix) from the GPS service (as opposed to a tracking session). The control point uses the command QMI_LOC_START_REQ to initiate this request and set the desired timeout, accuracy, etc. The GPS service attempts to compute a fix until the accuracy threshold is reached or until the number of seconds specified in the timeout parameter has elapsed. During the session, all control points receive NMEA sentences, satellite information reports, and parsed position reports, if registered for them.

2.9.5 Auto-Tracking Sessions

Tracking sessions are used to request and obtain a continuous stream of fixes from the MSM GNSS subsystem. A tracking session can be requested with the same QMI_LOC_START_REQ message as that used for single-shot fixes, but with a different value supplied for fix_recurrence in the (optional) fix criteria that can be provided with this message.

Auto-tracking continues to run the GNSS subsystem until all clients requesting auto-tracking request that auto-tracking be stopped. The periodic rate of position outputs as well as a number of other parameters can be set by the client requesting a position fix.

Auto-tracking for a particular client is stopped under the following circumstances:

- The client QMI_LOC control point explicitly disables it through a QMI command
- The client QMI_LOC control point resets
- The client QMI_LOC control point is closed

2.9.6 NMEA Sentence Data

QMI_LOC supports the output of NMEA sentences. Control points that have optionally registered for NMEA event reports will receive NMEA sentences sent as QMI indications.

QMI_LOC provides messages to configure various NMEA settings:

- Which NMEA sentence types are to be generated
- How often NMEA sentences are generated (while the fix is being performed)

NMEA configuration is global, and any change made by a QMI_LOC control point affects the generation of NMEA sentence data for all clients.

2.9.7 External Information Injection

2.9.7.1 External Time Injection

See [Q3] for a definition and use of this value.

2.9.7.2 Coarse Position Injection

A QMI control point can inject a coarse position to the GPS service. This can help to obtain faster times to the first fix, depending on the accuracy and uncertainty values of the desired position.

2.9.7.3 WiFi Position Injection

WiFi position injection, similar to coarse position injection, is essentially a coarse position obtained from WiFi measurements that also contains information about WiFi access points. A control point may obtain WiFi positions from a third party (e.g., Skyhook Wireless™) and inject them to the GPS service.

2.9.8 gpsOneXTRA™ Satellite Database Information

QMI_LOC can support an extension of the standalone position determination method XTRA. A database of satellite information is downloaded from a server and is used to improve the performance of standalone fixes.

The satellite database is only valid for a fixed amount of time after it is generated. After this time has elapsed, if XTRA data is to be used, a new database must be downloaded.

QMI_LOC can be configured to send download requests to an external XTRA client sitting above the QMI. The XTRA client can download the XTRA data from whatever network interface it can and then inject it to the GPS service using the QMI_LOC_INJECT_PREDICTED_ORBITS_DATA command.

The validity information of the current database can be queried by the control point and a download may be forced to start regardless of the valid time remaining for the database.

QMI_LOC can be set to automatically request an update of the XTRA database at a specified interval. When enabled, the GPS engine must send a request to the preferred XTRA client to download and inject a fresh XTRA data file from an XTRA server.

Use the following steps to inject an XTRA file from an external XTRA client:

1. Register for the external XTRA database request via the QMI_LOC_REG_EVENTS_REQ message (this step only needs to be done once at power-up).
2. Wait for at least one external XTRA database request indication (this indication may be sent to the external XTRA client as a result of the external XTRA calling QMI_LOC_FORCE_XTRA_DOWNLOAD, or if the GPS service needs a fresh download at the beginning of a positioning session).
3. Download the XTRA database file from the server using at least one of the URLs in the indication.
4. Inject the downloaded XTRA database file using QMI_LOC_INJECT_PREDICTED_ORBITS_DATA.

Subsequent XTRA file injections during a power cycle only require steps 4 and 5 to be done. If the device is power cycled, the XTRA client must start from step 1.

2.9.9 Satellite-Based Augmentation System Configuration

Satellite-Based Augmentation System (SBAS) is a system that supports wide-area or regional augmentation through the use of additional satellite broadcast messages. Such systems are commonly composed of multiple ground stations, located at accurately surveyed points. The ground stations take measurements of one or more of the GNSS satellites, the satellite signals, or other environmental factors that may impact the signal received by users. Using these measurements, information messages are created and sent to one or more satellites for broadcast to the end users.

When enabled, the GPS service attempts to make use of the SBAS.

2.9.10 External Sensor Data Input

The GPS service has the ability to use various types of sensor data injected by a control point. A control point must register for sensor data requests from the GPS service using the QMI_LOC_REG_EVENTS_REQ message. The GPS service indicates when it is ready/not ready to receive sensor data inputs from the control point by sending a QMI_LOC_EVENT_SENSOR_STREAMING_READY_STATUS_IND message to the control point. A separate TLV is used for each sensor type that is supported (e.g., 3-axis accelerometer or 3-axis gyro). The control point must inject sensor data using the QMI_LOC_INJECT_SENSOR_DATA message.

Time synchronization between the GPS processor and the external sensor processor is crucial for the GPS service to be able to use the sensor data input from the control point. The control point must register for the time sync request message using the QMI_LOC_REG_EVENTS_REQ message.

The GPS service periodically sends a time sync request via the QMI_LOC_INJECT_TIME_SYNC_DATA message with a reference counter. The control point is then expected to return this counter along with the sensor processor's time to the GPS service using the QMI_LOC_INJECT_TIME_SYNC_DATA message. This handshake allows the GPS service to maintain a time correlation between the processors and use the sensor data effectively.

QMI_LOC defines sensor time or sensor processor time as a monotonically increasing counter with a jitter value $\leq$ 1 ms. This counter must never be stopped until the processor is rebooted. This time source must be used in the QMI_LOC_INJECT_SENSOR_DATA and QMI_LOC_INJECT_TIME_SYNC_DATA messages.

Figure 2-1 illustrates the call flow sequence for configuration, request, and injection of sensor and time sync data.

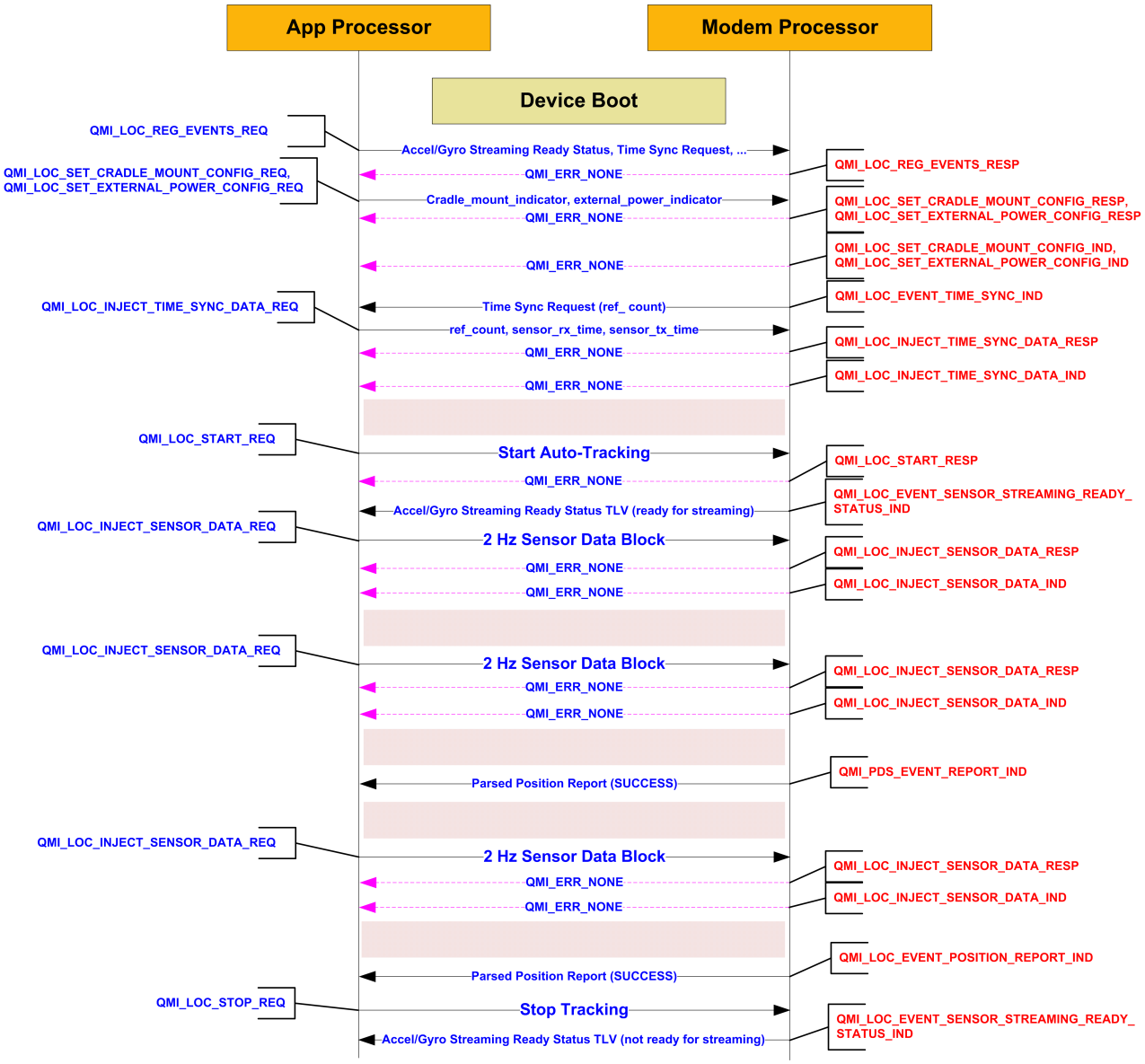

Figure 2-1 Call flow for configuration, request, and injection of sensor and time sync data

Figure 2-2 illustrates the coordinate axes orientation for acceleration measurements.

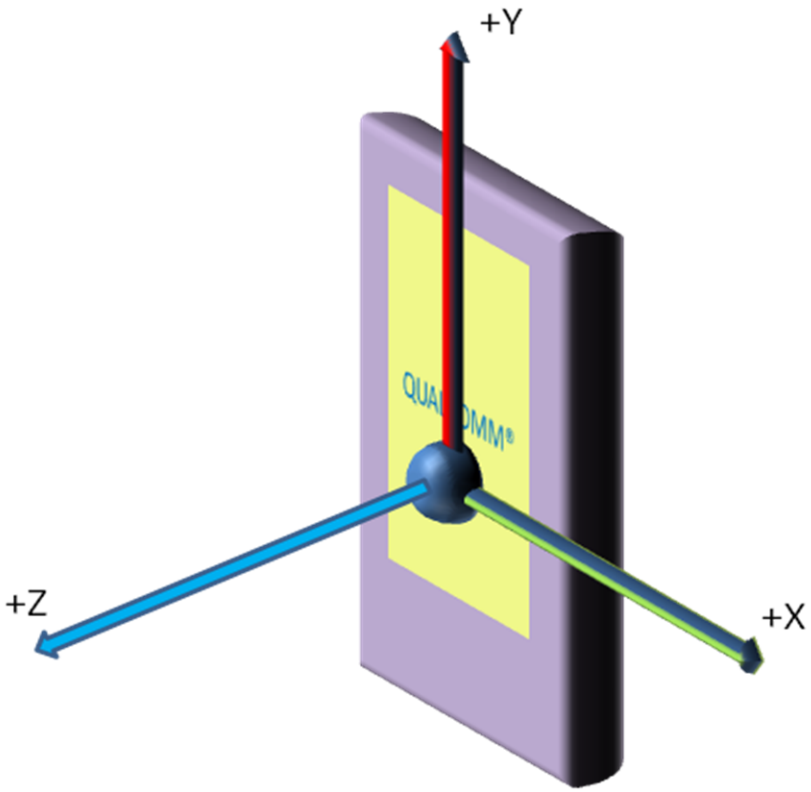

Figure 2-2 Orientation of coordinate axes for acceleration measurements

The accelerometer reading is specific force (proper force), i.e., the acceleration of the device in the inertial coordinate frame minus gravity:

$$\vec{a}_{\text{measured}} = \ddot{\vec{r}} - \vec{g}$$

The physical reason for this is that the accelerometer does not measure gravity; the gravitational acceleration (-9.81 m/s^2) must be subtracted from the actual kinematic acceleration of the device in order to get the reading on the accelerometer output.

For example, the reading on the accelerometer y-axis of the device shown in Figure 2-2 is approximately +9.81 m/s^2 when the device is stationary on a stable surface, such as a desk.

When the accelerometer is free falling, the acceleration equals the gravitational acceleration and therefore, according to the equation above, the accelerometer output is zero.

If the accelerometer measurements do not comply with the above description, an additional bit for accelerometer data sign reversal must be set in the flag field of the QMI_LOC_INJECT_SENSOR_DATA_REQ message.

2.9.11 Gyroscope Measurements

The gyroscope reading of a specified channel is positive when the rotation around the corresponding axis happens in the counterclockwise direction in a mathematical sense.

When looking at the axis such that the arrow points toward you, positive rotation is counterclockwise.

Figure 2-3 illustrates positive rotation the right-hand rule.

Figure 2-3 Right-hand rule

If the gyro measurements do not comply with the above description, an additional bit for gyro data sign reversal must be set in the flag field of the QMI_LOC_INJECT_SENSOR_DATA_REQ message.

3 QMI_LOC Messages

Table 3-1 lists the QMI_LOC messages.

Note : The floating point variables specified in this document are IEEE-754 compliant.

Table 3-1 QMI_LOC messages

Command	ID	Description
QMI_LOC_GEN_RESP		Generic response definition. This message is used to tell clients whether their message was accepted for further processing or rejected.
QMI_LOC_INFORM_CLIENT_REVISION	0x0020	Informs the service of the minor revision of the interface definition that the control point implements.
QMI_LOC_REG_EVENTS	0x0021	Used by the control point to register for events from the location subsystem.
QMI_LOC_START	0x0022	The control point sends this message when it wants to initiate a GPS session.
QMI_LOC_STOP	0x0023	The control point sends this message when it wants to stop a GPS session.
QMI_LOC_EVENT_POSITION_REPORT	0x0024	This message is used to send the position report to the control point.
QMI_LOC_EVENT_GNSS_SV_INFO	0x0025	Used to send a satellite report to the control point.
QMI_LOC_EVENT_NMEA	0x0026	Used to send NMEA sentences to the control point.
QMI_LOC_EVENT_NI_NOTIFY_VERIFY_ REQ	0x0027	Indicates an NI notify/verify request to the control point.
QMI_LOC_EVENT_INJECT_TIME_REQ	0x0028	Requests the control point to inject time information.
QMI_LOC_EVENT_INJECT_PREDICTED_ ORBITS_REQ	0x0029	Requests the control point to inject predicted orbits data.
QMI_LOC_EVENT_INJECT_POSITION_REQ	0x002A	Requests the control point to inject a position.
QMI_LOC_EVENT_ENGINE_STATE	0x002B	Sends the engine state to the control point.
QMI_LOC_EVENT_FIX_SESSION_STATE	0x002C	Sends the fix session state to the control point.
QMI_LOC_EVENT_WIFI_REQ	0x002D	Sends a WiFi request to the control point.

Table 3-1 QMI_LOC messages (cont.)

Command	ID	Description
QMI_LOC_EVENT_SENSOR_STREAMING_READY_STATUS	0x002E	Notifies the control point if the GNSS location engine is ready to accept sensor data.
QMI_LOC_EVENT_TIME_SYNC_REQ	0x002F	Notifies the control point to inject time synchronization data.
QMI_LOC_EVENT_SET_SPI_STREAMING_REPORT	0x0030	Requests the control point to enable Stationary Position Indicator (SPI) streaming reports.
QMI_LOC_EVENT_LOCATION_SERVER_CONNECTION_REQ	0x0031	Requests the client to open or close a connection to the assisted GPS location server.
QMI_LOC_GET_SERVICE_REVISION	0x0032	Client can query the service revision using this message.
QMI_LOC_GET_FIX_CRITERIA	0x0033	Gets the fix criteria from the location engine.
QMI_LOC_INFORM_NI_USER_RESPONSE	0x0034	Sends the NI user response back to the engine; success or failure is reported in a separate indication.
QMI_LOC_INJECT_PREDICTED_ORBITS_DATA	0x0035	Injects predicted orbits data.
QMI_LOC_GET_PREDICTED_ORBITS_DATA_SOURCE	0x0036	Gets the predicted orbits data source.
QMI_LOC_GET_PREDICTED_ORBITS_DATA_VALIDITY	0x0037	Gets the predicted orbits data validity.
QMI_LOC_INJECT_UTC_TIME	0x0038	Injects UTC time in the location engine.
QMI_LOC_INJECT_POSITION	0x0039	Injects a position to the location engine.
QMI_LOC_SET_ENGINE_LOCK	0x003A	Sets the location engine lock.
QMI_LOC_GET_ENGINE_LOCK	0x003B	Gets the location engine lock.
QMI_LOC_SET_SBAS_CONFIG	0x003C	Sets the SBAS configuration.
QMI_LOC_GET_SBAS_CONFIG	0x003D	Gets the SBAS configuration from the location engine.
QMI_LOC_SET_NMEA_TYPES	0x003E	Sets the NMEA types.
QMI_LOC_GET_NMEA_TYPES	0x003F	Gets the NMEA types from the location engine.
QMI_LOC_SET_LOW_POWER_MODE	0x0040	Enables/disables Low Power Mode (LPM) configuration.
QMI_LOC_GET_LOW_POWER_MODE	0x0041	Gets the LPM status from the location engine.
QMI_LOC_SET_SERVER	0x0042	Specifies the A-GPS server type and address.

Table 3-1 QMI_LOC messages (cont.)

Command	ID	Description
QMI_LOC_GET_SERVER	0x0043	Gets the location server from the location engine.
QMI_LOC_DELETE_ASSIST_DATA	0x0044	This command is used to delete the location engine assistance data
QMI_LOC_SET_XTRA_T_SESSION_ CONTROL	0x0045	Enables/disables XTRA-T session control.
QMI_LOC_GET_XTRA_T_SESSION_ CONTROL	0x0046	Gets the XTRA-T session control value from the location engine.
QMI_LOC_INJECT_WIFI_POSITION	0x0047	Injects the WiFi position.
QMI_LOC_NOTIFY_WIFI_STATUS	0x0048	Notifies the location engine of the WiFi status.
QMI_LOC_GET_REGISTERED_EVENTS	0x0049	Gets the mask of the events for which a client has registered.
QMI_LOC_SET_OPERATION_MODE	0x004A	Tells the engine to use the specified operation mode while making the position fixes. This command is not to be used by multiple clients concurrently.
QMI_LOC_GET_OPERATION_MODE	0x004B	Gets the current operation mode from the engine.
QMI_LOC_SET_SPI_STATUS	0x004C	Used by the control point to set the SPI status, which indicates whether the device is stationary.
QMI_LOC_INJECT_SENSOR_DATA	0x004D	Used by the control point to inject sensor data into the GNSS location engine.
QMI_LOC_INJECT_TIME_SYNC_DATA	0x004E	Used by the control point to inject time sync data.
QMI_LOC_GET_CRADLE_MOUNT_CONFIG	0x0050	Used by the control point to get the current cradle mount configuration.
QMI_LOC_SET_CRADLE_MOUNT_CONFIG	0x004F	Used by the control point to set the current cradle mount configuration.
QMI_LOC_GET_EXTERNAL_POWER_ CONFIG	0x0052	Used by the control point to get the current external power configuration.
QMI_LOC_SET_EXTERNAL_POWER_ CONFIG	0x0051	Used by the control point to set the current external power configuration.
QMI_LOC_INFORM_LOCATION_SERVER_ CONN_STATUS	0x0053	Used by the control point to inform the service about the status of the location server connection request that the service may have sent via the QMI_LOC_EVENT_LOCATION_ SERVER_REQ_IND event.
QMI_LOC_SET_PROTOCOL_CONFIG_ PARAMETERS	0x0054	Used by the control point to configure parameters stored in the nonvolatile memory.

Table 3-1 QMI_LOC messages (cont.)

Command	ID	Description
QMI_LOC_GET_PROTOCOL_CONFIG_ PARAMETERS	0x0055	Used by the control point to get the configuration parameters stored in the nonvolatile memory.
QMI_LOC_SET_SENSOR_CONTROL_ CONFIG	0x0056	Sets the sensor control configuration.
QMI_LOC_GET_SENSOR_CONTROL_ CONFIG	0x0057	Retrieves the current sensor control configuration.
QMI_LOC_SET_SENSOR_PROPERTIES	0x0058	Sets the properties specific to the type of sensor used. The control point must set sensor properties before they can be used to aid in heading and positioning performance improvement.
QMI_LOC_GET_SENSOR_PROPERTIES	0x0059	Retrieves the current sensor properties.
QMI_LOC_SET_SENSOR_PERFORMANCE_ CONTROL_CONFIGURATION	0x005A	Provides fine-grained control of sensor based positioning performance
QMI_LOC_GET_SENSOR_PERFORMANCE_ CONTROL_CONFIGURATION	0x005B	Retrieves the current sensor performance control configuration.
QMI_LOC_INJECT_SUPL_CERTIFICATE	0x005C	Injects a SUPL certificate to be used in AGNSS sessions.
QMI_LOC_DELETE_SUPL_CERTIFICATE	0x005D	Deletes a SUPL certificate.
QMI_LOC_SET_POSITION_ENGINE_ CONFIG_PARAMETERS	0x005E	Used by the control point to configure position engine functionality.
QMI_LOC_GET_POSITION_ENGINE_ CONFIG_PARAMETERS	0x005F	Used by the control point to get the position engine configuration parameters.

3.1 QMI_LOC_GEN_RESP

Generic response definition. This message is used to tell clients whether their message was accepted for further processing or rejected.

LOC message ID

0x0000

Version introduced

Major - 2, Minor - 2

3.1.1 Response - QMI_LOC_GEN_RESP

Message type

Response

Sender

Service

Mandatory TLVs

The Result Code TLV (defined in Section 2.3.1) is always present in the response.

Optional TLVs

None

Error codes

QMI_ERR_NONE	No error in the request
QMI_ERR_INTERNAL	Unexpected error occurred during processing
QMI_ERR_MALFORMED_MSG	Message was not formulated correctly by the control point or the message was corrupted during transmission
QMI_ERR_NO_MEMORY	Device could not allocate memory to formulate a response
QMI_ERR_INVALID_HANDLE	Invalid client handle was received

3.1.2 Description of QMI_LOC_GEN_RESP REQ/RESP

This is a generic response message returned to the client in response to a request message. All response messages are identical in format. Each response indicates whether the request message was accepted for further processing or was rejected (typically due to some sort of message format or invalid argument error). The actual results of the processing that is triggered by a valid request are returned asynchronously via an indicator message. The indicator message indicates successful completion or failure to complete the command.

3.2 QMI_LOC_INFORM_CLIENT_REVISION

Informs the service of the minor revision of the interface definition that the control point implements.

LOC message ID

0x0020

Version introduced

Major - 2, Minor - 2

3.2.1 Request - QMI_LOC_INFORM_CLIENT_REVISION_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

Name	Version last modified
Revision	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x01		1	Revision
Length	4		2	
Value	→	revision	4	Revision that the control point is using. • Type: Unsigned integer

Optional TLVs

None

Error codes

QMI_ERR_NONE	No error in the request
QMI_ERR_INTERNAL	Unexpected error occurred during processing
QMI_ERR_MALFORMED_MSG	Message was not formulated correctly by the control point or the message was corrupted during transmission
QMI_ERR_NO_MEMORY	Device could not allocate memory to formulate a response
QMI_ERR_INVALID_HANDLE	Invalid client handle was received

3.2.2 Description of QMI_LOC_INFORM_CLIENT_REVISION REQ/RESP

This message is sent from the control point to the service indicating the revision of the interface definition implemented by the client. If the control point‘s revision is greater than that supported by the service itself, the messages sent by the control point may not be interpreted properly. The control point can query the service revision using the QMI_LOC_GET_SERVICE_REVISION_REQ message to identify the revision of the service‘s interface definition.

3.3 QMI_LOC_REG_EVENTS

Used by the control point to register for events from the location subsystem.

LOC message ID

0x0021

Version introduced

Major - 2, Minor - 2

3.3.1 Request - QMI_LOC_REG_EVENTS_REQ

Message type

Request

Sender

Control Point

Mandatory TLVs

Name	Version last modified
Event Registration Mask	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x01		1	Event Registration Mask
Length	8		2	

Field	Field value	Parameter	Size (byte)	Description
Value	→	eventRegMask	8	Specifies the events that the control point is interested in receiving. Refer to the definition of the following bitmasks: • 0x00000001 – POSITION_REPORT • 0x00000002 – GNSS_SV_INFO • 0x00000004 – NMEA • 0x00000008 – NI_NOTIFY_VERIFY_REQ • 0x00000010 – INJECT_TIME_REQ • 0x00000020 – INJECT_PREDICTED_ ORBITS_REQ • 0x00000040 – INJECT_POSITION_REQ • 0x00000080 – ENGINE_STATE • 0x00000100 – FIX_SESSION_STATE • 0x00000200 – WIFI_REQ • 0x00000400 – SENSOR_STREAMING_ READY_STATUS • 0x00000800 – TIME_SYNC_REQ • 0x00001000 – SET_SPI_STREAMING_ REPORT • 0x00002000 – LOCATION_SERVER_ CONNECTION_REQ Multiple events can be registered by ORing the individual masks and sending them in this TLV. All unused bits in this mask must be set to 0.

Optional TLVs

None

Error codes

QMI_ERR_NONE	No error in the request
QMI_ERR_INTERNAL	Unexpected error occurred during processing
QMI_ERR_MALFORMED_MSG	Message was not formulated correctly by the control point or the message was corrupted during transmission
QMI_ERR_NO_MEMORY	Device could not allocate memory to formulate a response
QMI_ERR_INVALID_HANDLE	Invalid client handle was received

3.3.2 Description of QMI_LOC_REG_EVENTS REQ/RESP

This command informs the service about the asynchronous events that the control point is interested in receiving. A client receives the events for which it has registered through the indication messages (QMI_LOC_EVENT_*_IND).

3.4 QMI_LOC_START

The control point sends this message when it wants to initiate a GPS session.

LOC message ID

0x0022

Version introduced

Major - 2, Minor - 2

3.4.1 Request - QMI_LOC_START_REQ

Message type

Request

Sender

Control Point

Mandatory TLVs

Name	Version last modified
Session ID	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x01		1	Session ID
Length	1		2	
Value	→	sessionId	1	ID of the session as identified by the control point. The session ID is reported back in the position reports. The control point must specify the same session ID in the QMI_LOC_STOP_REQ message. • Type: Unsigned integer • Range: 0 to 255

Optional TLVs

Name	Version last modified
Recurrence Type	2.2
Horizontal Accuracy	2.2
Enable/Disable Intermediate Reports	2.2
Minimum Interval Between Position Reports	2.2
ID of the Application that Sent this Request	2.6

Field	Field value	Parameter	Size (byte)	Description
Type	0x10		1	Recurrence Type
Length	4		2	
Value	→	fixRecurrence	4	Specifies the type of session in which the control point is interested. If this TLV is not specified, recurrence defaults to SINGLE. Valid values: • 0x00000001 – Request periodic fixes • 0x00000002 – Request a single fix
Type	0x11		1	Horizontal Accuracy
Length	4		2	
Value	→	horizontalAccuracyLevel	4	Specifies the horizontal accuracy level required by the control point. If not specified, accuracy defaults to LOW. Valid values: • 0x00000001 – LOW: Client requires low horizontal accuracy. • 0x00000002 – MED: Client requires medium horizontal accuracy. • 0x00000003 – HIGH: Client requires high horizontal accuracy.
Type	0x12		1	Enable/Disable Intermediate Reports
Length	4		2	

Field	Field value	Parameter	Size (byte)	Description
Value	→	intermediateReportState	4	Specifies if the control point is interested in receiving intermediate reports. The control point must explicitly set this field to OFF if it does not wish to receive intermediate position reports. Intermediate position reports are generated at 1 Hz and are ON by default. If intermediate reports are turned ON, the client receives position reports even if the accuracy criteria are not met. The status in such a position report is set to IN_PROGRESS in order for the control point to identify intermediate reports. Valid values: • 0x00000001 – ON: Client is interested in receiving intermediate reports • 0x00000002 – OFF: Client is not interested in receiving intermediate reports
Type	0x13		1	Minimum Interval Between Position Reports
Length	4		2	
Value	→	minInterval	4	Minimum time interval, specified by the control point, that must elapse between position reports. • Type: Unsigned integer • Units: Milliseconds • Default: 1000 ms
Type	0x14		1	ID of the Application that Sent this Request Application provider, name, and version.
Length	Var		2	
Value	→	applicationProvider_len	1	Number of sets of the following elements: • applicationProvider
		applicationProvider	Var	Application provider.
		applicationName_len	1	Number of sets of the following elements: • applicationName
		applicationName	Var	Application name.
		applicationVersion_valid	1	Specifies whether the application version string contains a valid value: • 0x00 (FALSE) – Application version string is invalid • 0x01 (TRUE) – Application version string is valid
		applicationVersion_len	1	Number of sets of the following elements: • applicationVersion
		applicationVersion	Var	Application version.

Error codes

QMI_ERR_NONE	No error in the request
QMI_ERR_INTERNAL	Unexpected error occurred during processing
QMI_ERR_MALFORMED_MSG	Message was not formulated correctly by the control point or the message was corrupted during transmission
QMI_ERR_NO_MEMORY	Device could not allocate memory to formulate a response
QMI_ERR_INVALID_HANDLE	Invalid client handle was received

3.4.2 Description of QMI_LOC_START REQ/RESP

This message starts a positioning session with the specified configuration.

3.5 QMI_LOC_STOP

The control point sends this message when it wants to stop a GPS session.

LOC message ID

0x0023

Version introduced

Major - 2, Minor - 2

3.5.1 Request - QMI_LOC_STOP_REQ

Message type

Request

Sender

Control Point

Mandatory TLVs

Name	Version last modified
Session ID	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x01		1	Session ID
Length	1		2	
Value	→	sessionId	1	ID of the session that was specified in the Start request (QMI_LOC_START_REQ). • Type: Unsigned integer • Range: 0 to 255

Optional TLVs

None

Error codes

QMI_ERR_NONE	No error in the request
QMI_ERR_INTERNAL	Unexpected error occurred during processing
QMI_ERR_MALFORMED_MSG	Message was not formulated correctly by the control point or the message was corrupted during transmission
QMI_ERR_NO_MEMORY	Device could not allocate memory to formulate a response
QMI_ERR_INVALID_HANDLE	Invalid client handle was received

3.5.2 Description of QMI_LOC_STOP REQ/RESP

This command stops a client‘s positioning session. If any other client is requesting a position, this client will continue to receive all events registered for except the position, satellite, and NMEA report events.

3.6 QMI_LOC_EVENT_POSITION_REPORT

This message is used to send the position report to the control point.

LOC message ID

0x0024

Version introduced

Major - 2, Minor - 2

3.6.1 Indication - QMI_LOC_EVENT_POSITION_REPORT_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version last modified
Session Status	2.2
Session ID	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x01		1	Session Status
Length	4		2	

Field	Field value	Parameter	Size (byte)	Description
Value	→	sessionStatus	4	Session status. Valid values: • 0x00000000 – SESS_STATUS_SUCCESS • 0x00000001 – SESS_STATUS_IN_PROGRESS • 0x00000002 – SESS_STATUS_GENERAL_FAILURE • 0x00000003 – SESS_STATUS_TIMEOUT • 0x00000004 – SESS_STATUS_USER_END • 0x00000005 – SESS_STATUS_BAD_PARAMETER • 0x00000006 – SESS_STATUS_PHONE_OFFLINE • 0x00000007 – SESS_STATUS_ENGINE_LOCKED
Type	0x02		1	Session ID
Length	1		2	
Value	→	sessionId	1	ID of the session that was specified in the Start request QMI_LOC_START_REQ. • Type: Unsigned integer • Range: 0 to 255

Optional TLVs

Name	Version last modified
Latitude	2.2
Longitude	2.2
Circular Horizontal Position Uncertainty	2.2
Horizontal Elliptical Uncertainty	2.2
Horizontal Elliptical Uncertainty	2.2
Elliptical Horizontal Uncertainty Azimuth	2.2
Horizontal Confidence	2.2
Horizontal Reliability	2.2
Horizontal Speed	2.2
Speed Uncertainty	2.2
Altitude With Respect to Ellipsoid	2.2
Altitude With Respect to Sea Level	2.2
Vertical Uncertainty	2.2
Vertical Confidence	2.2
Vertical Reliability	2.2
Vertical Speed	2.2
Heading	2.2
Heading Uncertainty	2.2
Magnetic Deviation	2.2
Technology Used	2.2
Dilution of Precision	2.2

Name	Version last modified
UTC Timestamp	2.2
Leap Seconds	2.2
GPS Time	2.2
Time Uncertainty	2.2
Time Source	2.2
Sensor Data Usage	2.2
Fix Count for This Session	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x10		1	Latitude
Length	8		2	
Value	→	latitude	8	Latitude (specified in WGS84 datum). • Type: Floating point • Units: Degrees • Range: -90.0 to 90.0 • Positive values indicate northern latitude • Negative values indicate southern latitude
Type	0x11		1	Longitude
Length	8		2	
Value	→	longitude	8	Longitude (specified in WGS84 datum). • Type: Floating point • Units: Degrees • Range: -180.0 to 180.0 • Positive values indicate eastern longitude • Negative values indicate western longitude
Type	0x12		1	Circular Horizontal Position Uncertainty
Length	4		2	
Value	→	horUncCircular	4	Horizontal position uncertainty (circular). • Type: Floating point • Units: Meters
Type	0x13		1	Horizontal Elliptical Uncertainty (Semi-Minor Axis)
Length	4		2	
Value	→	horUncEllipseSemiMinor	4	Semi-minor axis of horizontal elliptical uncertainty. • Type: Floating point • Units: Meters
Type	0x14		1	Horizontal Elliptical Uncertainty (Semi-Major Axis)
Length	4		2	
Value	→	horUncEllipseSemiMajor	4	Semi-major axis of horizontal elliptical uncertainty. • Type: Floating point • Units: Meters
Type	0x15		1	Elliptical Horizontal Uncertainty Azimuth
Length	4		2	

Field	Field value	Parameter	Size (byte)	Description
Value	→	horUncEllipseOrientAzimuth	4	Elliptical horizontal uncertainty azimuth of orientation. • Type: Floating point • Units: Decimal degrees • Range: 0 to 180
Type	0x16		1	Horizontal Confidence
Length	1		2	
Value	→	horConfidence	1	Horizontal uncertainty confidence. • Type: Unsigned integer • Units: Percent • Range: 0 to 99
Type	0x17		1	Horizontal Reliability
Length	4		2	
Value	→	horReliability	4	Specifies the reliability of the horizontal position. Valid values: • 0x00000000 – RELIABILITY_NOT_SET • 0x00000001 – RELIABILITY_VERY_LOW • 0x00000002 – RELIABILITY_LOW • 0x00000003 – RELIABILITY_MEDIUM • 0x00000004 – RELIABILITY_HIGH
Type	0x18		1	Horizontal Speed
Length	4		2	
Value	→	speedHorizontal	4	Horizontal speed. • Type: Floating point • Units: Meters/second
Type	0x19		1	Speed Uncertainty
Length	4		2	
Value	→	speedUnc	4	Speed uncertainty. • Type: Floating point • Units: Meters/second
Type	0x1A		1	Altitude With Respect to Ellipsoid
Length	4		2	
Value	→	altitudeWrtEllipsoid	4	Altitude with respect to the WGS84 ellipsoid. • Type: Floating point • Units: Meters • Range: -500 to 15883
Type	0x1B		1	Altitude With Respect to Sea Level
Length	4		2	
Value	→	altitudeWrtMeanSeaLevel	4	Altitude with respect to mean sea level. • Type: Floating point • Units: Meters
Type	0x1C		1	Vertical Uncertainty
Length	4		2	
Value	→	vertUnc	4	Vertical uncertainty. • Type: Floating point • Units: Meters

Field	Field value	Parameter	Size (byte)	Description
Type	0x1D		1	Vertical Confidence
Length	1		2	
Value	→	vertConfidence	1	Vertical uncertainty confidence. • Type: Unsigned integer • Units: Percent • Range: 0 to 99
Type	0x1E		1	Vertical Reliability
Length	4		2	
Value	→	vertReliability	4	Specifies the reliability of the vertical position. Valid values: • 0x00000000 – RELIABILITY_NOT_SET • 0x00000001 – RELIABILITY_VERY_LOW • 0x00000002 – RELIABILITY_LOW • 0x00000003 – RELIABILITY_MEDIUM • 0x00000004 – RELIABILITY_HIGH
Type	0x1F		1	Vertical Speed
Length	4		2	
Value	→	speedVertical	4	Vertical speed. • Type: Floating point • Units: Meters/second
Type	0x20		1	Heading
Length	4		2	
Value	→	heading	4	Heading. • Type: Floating point • Units: Degrees • Range: 0 to 359.999
Type	0x21		1	Heading Uncertainty
Length	4		2	
Value	→	headingUnc	4	Heading uncertainty. • Type: Floating point • Units: Degrees • Range: 0 to 359.999
Type	0x22		1	Magnetic Deviation
Length	4		2	
Value	→	magneticDeviation	4	Difference between the bearing to true north and the bearing shown on a magnetic compass. The deviation is positive when the magnetic north is east of true north. • Type: Floating point
Type	0x23		1	Technology Used
Length	4		2	
Value	→	technologyMask	4	Technology used in computing this fix. Valid bitmasks: • 0x00000001 – SATELLITE • 0x00000002 – CELLID • 0x00000004 – WIFI

Field	Field value	Parameter	Size (byte)	Description
Type	0x24		1	Dilution of Precision Dilution of precision associated with this position.
Length	12		2	
Value	→	PDOP	4	Position dilution of precision. • Type: Floating point • Range: 1 (highest accuracy) to 50 (lowest accuracy) • PDOP = square root of (HDOP$\hat{2}$ + VDOP$\hat{2}$)
		HDOP	4	Horizontal dilution of precision. • Type: Floating point • Range: 1 (highest accuracy) to 50 (lowest accuracy)
		VDOP	4	Vertical dilution of precision. • Type: Floating point. • Range: 1 (highest accuracy) to 50 (lowest accuracy)
Type	0x25		1	UTC Timestamp
Length	8		2	
Value	→	timestampUtc	8	UTC timestamp. • Type: Unsigned integer • Units: Milliseconds since Jan. 1, 1970
Type	0x26		1	Leap Seconds
Length	1		2	
Value	→	leapSeconds	1	Leap second information. If leapSeconds is not available, timestampUtc is calculated based on a hard-coded value for leap seconds. • Type: Unsigned integer • Units: Seconds
Type	0x27		1	GPS Time The number of weeks since Jan. 5, 1980, and milliseconds into the current week.
Length	6		2	
Value	→	gpsWeek	2	Current GPS week as calculated from midnight, Jan. 6, 1980. • Type: Unsigned integer • Units: Weeks
		gpsTimeOfWeekMs	4	Amount of time into the current GPS week. • Type: Unsigned integer • Units: Milliseconds
Type	0x28		1	Time Uncertainty
Length	4		2	
Value	→	timeUnc	4	Time uncertainty. • Type: Floating point • Units: Milliseconds
Type	0x29		1	Time Source
Length	4		2	

Field	Field value	Parameter	Size (byte)	Description
Value	→	timeSrc	4	Time source. Valid values: • 0x00000000 – TIME_SRC_INVALID • 0x00000001 – TIME_SRC_NETWORK_ TIME_TRANSFER • 0x00000002 – TIME_SRC_NETWORK_ TIME_TAGGING • 0x00000003 – TIME_SRC_EXTERNAL_ INPUT • 0x00000004 – TIME_SRC_TOW_DECODE • 0x00000005 – TIME_SRC_TOW_ CONFIRMED • 0x00000006 – TIME_SRC_TOW_AND_WEEK_ CONFIRMED • 0x00000007 – TIME_SRC_NAV_ SOLUTION • 0x00000008 – TIME_SRC_SOLVE_FOR_ TIME
Type	0x2A		1	Sensor Data Usage Whether sensor data was used in computing the position in this position report.
Length	8		2	
Value	→	usageMask	4	Specifies which sensors are used. Valid bitmasks are specified by the following constants: • 0x00000001 – SENSOR_USED_ACCEL • 0x00000002 – SENSOR_USED_GYRO
		aidingIndicatorMask	4	Specifies which results are aided by sensors. Valid bitmasks are specified by the following constants: • 0x00000001 – AIDED_HEADING • 0x00000002 – AIDED_SPEED • 0x00000004 – AIDED_POSITION • 0x00000008 – AIDED_VELOCITY
Type	0x2B		1	Fix Count for This Session
Length	4		2	
Value	→	fixId	4	Fix count for the session. Starts with 0 and increments by one for each successive position report for a particular session.

Error codes

QMI_ERR_NONE	No error in the request
QMI_ERR_INTERNAL	Unexpected error occurred during processing
QMI_ERR_MALFORMED_MSG	Message was not formulated correctly by the control point or the message was corrupted during transmission
QMI_ERR_NO_MEMORY	Device could not allocate memory to formulate a response
QMI_ERR_INVALID_HANDLE	Invalid client handle was received

3.6.2 Description of QMI_LOC_EVENT_POSITION_REPORT

This event is used to send the position report to the control point. The position report is relayed to the control point only if the control point has indicated its interest by setting the appropriate mask in QMI_LOC_REG_EVENTS_REQ.

3.7 QMI_LOC_EVENT_GNSS_SV_INFO

Used to send a satellite report to the control point.

LOC message ID

0x0025

Version introduced

Major - 2, Minor - 2

3.7.1 Indication - QMI_LOC_EVENT_GNSS_SV_INFO_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version last modified
Altitude Source	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x01		1	Altitude Source
Length	1		2	
Value	→	altitudeAssumed	1	Altitude assumed or calculated: • 0x00 (FALSE) – Valid altitude is calculated • 0x01 (TRUE) – Valid altitude is assumed; there may not be enough satellites to determine precise altitude

Optional TLVs

Name	Version last modified
Satellite Info	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x10		1	Satellite Info SV information list.
Length	Var		2	
Value	→	svList_len	1	Number of sets of the following elements: • validMask • system • gnssSvId • healthStatus • svStatus • svInfoMask • elevation • azimuth • snr
		validMask	4	Bitmask indicating which of the fields in this TLV are valid. Valid bitmasks: • 0x00000001 – VALID_SYSTEM • 0x00000002 – VALID_GNSS_SVID • 0x00000004 – VALID_HEALTH_STATUS • 0x00000008 – VALID_PROCESS_STATUS • 0x00000010 – VALID_SVINFO_MASK • 0x00000020 – VALID_ELEVATION • 0x00000040 – VALID_AZIMUTH • 0x00000080 – VALID_SNR
		system	4	Indicates to which constellation this SV belongs. Valid values: • 0x00000001 – eQMI_LOC_SV_SYSTEM_GPS • 0x00000002 – eQMI_LOC_SV_SYSTEM_GALILEO • 0x00000003 – eQMI_LOC_SV_SYSTEM_SBAS • 0x00000004 – eQMI_LOC_SV_SYSTEM_COMPASS • 0x00000005 – eQMI_LOC_SV_SYSTEM_GLONASS

Field	Field value	Parameter	Size (byte)	Description
		gnssSvId	2	GNSS SV ID. • Type: Unsigned integer • Range: • - For GPS: 1 to 32 • - For SBAS: 33 to 64 • - For GLONASS: 65 to 96
		healthStatus	1	Health status. • Type: Unsigned integer • Range: 0 = unhealthy; 1 = healthy
		svStatus	4	SV processing status. Valid values: • 0x00000001 – SV_STATUS_IDLE • 0x00000002 – SV_STATUS_SEARCH • 0x00000003 – SV_STATUS_TRACK
		svInfoMask	1	Whether almanac and ephemeris information is available. Valid bitmasks: • 0x01 – SVINFO_HAS_EPHEMERIS • 0x02 – SVINFO_HAS_ALMANAC
		elevation	4	SV elevation angle. • Type: Floating point • Units: Degrees • Range: 0 to 90
		azimuth	4	SV azimuth angle. • Type: Floating point • Units: Degrees • Range: 0 to 360
		snr	4	SV signal-to-noise ratio. • Type: Floating point • Units: dB-Hz

Error codes

QMI_ERR_NONE	No error in the request
QMI_ERR_INTERNAL	Unexpected error occurred during processing
QMI_ERR_MALFORMED_MSG	Message was not formulated correctly by the control point or the message was corrupted during transmission
QMI_ERR_NO_MEMORY	Device could not allocate memory to formulate a response
QMI_ERR_INVALID_HANDLE	Invalid client handle was received

3.7.2 Description of QMI_LOC_EVENT_GNSS_SV_INFO

This event is used to send the satellite report to the control point. The satellite report is relayed to the control point only if the control point has indicated its interest by setting the appropriate mask in QMI_LOC_REG_EVENTS_REQ.

3.8 QMI_LOC_EVENT_NMEA

Used to send NMEA sentences to the control point.

LOC message ID

0x0026

Version introduced

Major - 2, Minor - 2

3.8.1 Indication - QMI_LOC_EVENT_NMEA_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version last modified
NMEA String	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x01		1	NMEA String
Length	Var		2	
Value	→	nmea	Var	NMEA string. • Type: NULL-terminated string • Maximum string length (including NULL terminator): 201

Optional TLVs

None

Error codes

QMI_ERR_NONE	No error in the request
QMI_ERR_INTERNAL	Unexpected error occurred during processing
QMI_ERR_MALFORMED_MSG	Message was not formulated correctly by the control point or the message was corrupted during transmission
QMI_ERR_NO_MEMORY	Device could not allocate memory to formulate a response
QMI_ERR_INVALID_HANDLE	Invalid client handle was received

3.8.2 Description of QMI_LOC_EVENT_NMEA

This event is used to send the NMEA report to the control point. The NMEA report is generated at a 1 Hz rate and consists of multiple NMEA sentences. Each event contains one NMEA sentence and hence multiple events are sent for the entire NMEA report. The report is relayed to the control point only if the control point has indicated its interest by setting the appropriate mask in QMI_LOC_REG_EVENTS_REQ.

3.9 QMI_LOC_EVENT_NI_NOTIFY_VERIFY_REQ

Indicates an NI notify/verify request to the control point.

LOC message ID

0x0027

Version introduced

Major - 2, Minor - 2

3.9.1 Indication - QMI_LOC_EVENT_NI_NOTIFY_VERIFY_REQ_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version last modified
Notification Type	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x01		1	Notification Type
Length	4		2	
Value	→	notificationType	4	Type of notification/verification performed. Valid values: • 0x00000001 – NO_NOTIFY_NO_VERIFY • 0x00000002 – NOTIFY_ONLY • 0x00000003 – ALLOW_NO_RESP • 0x00000004 – NOT_ALLOW_NO_RESP • 0x00000005 – PRIVACY_OVERRIDE

Optional TLVs

Name	Version last modified
Network Initiated Vx Request	2.2
Network Initiated SUPL Request	2.2
Network Initiated UMTS Control Plane Request	2.2
Network Initiated Service Interaction Request	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x10		1	Network Initiated Vx Request Optional NI Vx request payload.
Length	Var		2	
Value	→	posQosIncl	1	Whether quality of service is included: • 0x01 (TRUE) – QoS is included • 0x00 (FALSE) – QoS is not included
		posQos	1	Position QoS timeout. • Type: Unsigned integer • Units: Seconds • Range: 0 to 255
		numFixes	4	Number of fixes allowed. • Type: Unsigned integer
		timeBetweenFixes	4	Time between fixes. • Type: Unsigned integer • Units: Seconds
		posMode	4	Position mode. Valid values: • 0x00000001 – NI_VX_MS_ASSISTED_ONLY • 0x00000002 – NI_VX_MS_BASED_ONLY • 0x00000003 – NI_VX_MS_ASSISTED_PREFERRED_MS_BASED_ALLOWED • 0x00000004 – NI_VX_MS_BASED_PREFERRED_MS_ASSISTED_ALLOWED
		encodingScheme	4	VX encoding scheme. Valid values: • 0x00000000 – NI_VX_OCTET • 0x00000001 – NI_VX_EXN_PROTOCOL_MSG • 0x00000002 – NI_VX_ASCII • 0x00000003 – NI_VX_IA5 • 0x00000004 – NI_VX_UNICODE • 0x00000005 – NI_VX_SHIFT_JIS • 0x00000006 – NI_VX_KOREAN • 0x00000007 – NI_VX_LATIN_HEBREW • 0x00000008 – NI_VX_LATIN • 0x00000009 – NI_VX_GSM
		requestorId_len	1	Number of sets of the following elements: • requestorId

Field	Field value	Parameter	Size (byte)	Description
		requestorId	Var	Requestor ID. • Type: Array of bytes • Maximum array length: 200
		userRespTimerInSeconds	2	Time to wait for the user to respond. • Type: Unsigned integer • Units: Seconds
Type	0x11		1	Network Initiated SUPL Request Optional NI SUPL request payload.
Length	Var		2	
Value	→	valid_flags	4	Indicates which of the following fields are present in this value. Valid bitmasks: • 0x00000001 – SUPL_SERVER_INFO • 0x00000002 – SUPL_SESSION_ID • 0x00000004 – SUPL_HASH • 0x00000008 – SUPL_POS_METHOD • 0x00000010 – SUPL_DATA_CODING_SCHEME • 0x00000020 – SUPL_REQUESTOR_ID • 0x00000040 – SUPL_CLIENT_NAME • 0x00000080 – SUPL_QOP • 0x00000100 – SUPL_USER_RESP_TIMER
		suplServerAddrTypeMask	1	Mask specifying the valid fields in this value. Valid bitmasks: • 0x01 – IPV4 • 0x02 – IPV6 • 0x04 – URL
		addr	4	IPV4 address. • Type: Unsigned integer
		port	2	IPV4 port. • Type: Unsigned integer
		addr	16	IPV6 address. • Type: Array of unsigned integers • Maximum length of the array: 8
		port	4	IPV6 port. • Type: Unsigned integer
		urlAddr_len	1	Number of sets of the following elements: • urlAddr
		urlAddr	Var	URL. • Type: NULL-terminated string • Maximum string length (including NULL terminator): 256
		suplSessionId	4	SUPL session ID. • Type: Array of unsigned integers • Maximum length of the array: 4

Field	Field value	Parameter	Size (byte)	Description
		suplHash	8	Hash for SUPL_INIT; used to validate that the message was not corrupted. • Type: Array of unsigned integers • Length of the array: 8
		posMethod	4	GPS mode to be used for the fix. Valid values: • 0x00000001 – AGPS_SETASSISTED • 0x00000002 – AGPS_SETBASED • 0x00000003 – AGPS_SETASSISTED_PREF • 0x00000004 – AGPS_SETBASED_PREF • 0x00000005 – AUTONOMOUS_GPS • 0x00000006 – AFLT • 0x00000007 – ECID • 0x00000008 – EOTD • 0x00000009 – OTDOA • 0x0000000A – NO_POSITION
		dataCodingScheme	4	Data coding scheme applies to both the requestor ID and the client name. Valid values: • 0x0000000C – NI_SS_GERMAN • 0x0000000D – NI_SS_ENGLISH • 0x0000000E – NI_SS_ITALIAN • 0x0000000F – NI_SS_FRENCH • 0x00000010 – NI_SS_SPANISH • 0x00000011 – NI_SS_DUTCH • 0x00000012 – NI_SS_SWEDISH • 0x00000013 – NI_SS_DANISH • 0x00000014 – NI_SS_PORTUGUESE • 0x00000015 – NI_SS_FINNISH • 0x00000016 – NI_SS_NORWEGIAN • 0x00000017 – NI_SS_GREEK • 0x00000018 – NI_SS_TURKISH • 0x00000019 – NI_SS_HUNGARIAN • 0x0000001A – NI_SS_POLISH • 0x0000001B – NI_SS_LANGUAGE_ UNSPEC • 0x0000001C – NI_SUPL_UTF8 • 0x0000001D – NI_SUPL_UCS2 • 0x0000001E – NI_SUPL_GSM_DEFAULT

Field	Field value	Parameter	Size (byte)	Description
		formatType	4	Format of the formatted string. Valid values: • 0x00000000 – FORMAT_LOGICAL_NAME • 0x00000001 – FORMAT_EMAIL_ADDRESS • 0x00000002 – FORMAT_MSISDN • 0x00000003 – FORMAT_URL • 0x00000004 – FORMAT_SIP_URL • 0x00000005 – FORMAT_MIN • 0x00000006 – FORMAT_MDN • 0x00000007 – FORMAT_IMSPUBLIC_IDENTITY • 0x7FFFFFFF – FORMAT_OSS_UNKNOWN
		formattedString_len	1	Number of sets of the following elements: • formattedString
		formattedString	Var	Formatted string. • Type: Byte array • Maximum string length: 64
		formatType	4	Format of the formatted string. Valid values: • 0x00000000 – FORMAT_LOGICAL_NAME • 0x00000001 – FORMAT_EMAIL_ADDRESS • 0x00000002 – FORMAT_MSISDN • 0x00000003 – FORMAT_URL • 0x00000004 – FORMAT_SIP_URL • 0x00000005 – FORMAT_MIN • 0x00000006 – FORMAT_MDN • 0x00000007 – FORMAT_IMSPUBLIC_IDENTITY • 0x7FFFFFFF – FORMAT_OSS_UNKNOWN
		formattedString_len	1	Number of sets of the following elements: • formattedString
		formattedString	Var	Formatted string. • Type: Byte array • Maximum string length: 64
		validMask	1	Bit field indicating which fields are valid in this value. Valid bitmasks: • 0x01 – QOP_HORZ_ACC_VALID • 0x02 – QOP_VER_ACC_VALID • 0x04 – QOP_MAXAGE_VALID • 0x08 – QOP_DELAY_VALID
		horizontalAccuracy	1	Horizontal accuracy. • Type: Unsigned integer • Units: Meters

Field	Field value	Parameter	Size (byte)	Description
		verticalAccuracy	1	Vertical accuracy. • Type: Unsigned integer • Units: Meters
		maxLocAge	2	Maximum age of the location if the engine sends a previously computed position. • Type: Unsigned integer • Units: Seconds
		delay	1	Delay the server is willing to tolerate for the fix. • Type: Unsigned integer • Units: Seconds
		userResponseTimer	2	Time to wait for the user to respond. • Type: Unsigned integer • Units: Seconds
Type	0x12		1	Network Initiated UMTS Control Plane Request Optional NI UMTS-CP request payload.
Length	Var		2	
Value	→	valid_flags	2	Fields that are valid in this value. Valid bitmasks: • 0x0001 – INVOKE_ID_MASK • 0x0002 – DATA_CODING_SCHEME_ MASK • 0x0004 – NOTIFICATION_TEXT_MASK • 0x0008 – CLIENT_ADDRESS_MASK • 0x0010 – LOCATION_TYPE_MASK • 0x0020 – REQUESTOR_ID_MASK • 0x0040 – CODEWORD_STRING_MASK • 0x0080 – SERVICE_TYPE_MASK • 0x0100 – USER_RESP_TIMER_MASK
		invokeId	1	Supplementary Services invoke ID. • Type: Unsigned integer

Field	Field value	Parameter	Size (byte)	Description
		dataCodingScheme	4	Type of data encoding scheme for the text. Applies to both the notification text and the client address. Valid values: • 0x0000000C – NI_SS_GERMAN • 0x0000000D – NI_SS_ENGLISH • 0x0000000E – NI_SS_ITALIAN • 0x0000000F – NI_SS_FRENCH • 0x00000010 – NI_SS_SPANISH • 0x00000011 – NI_SS_DUTCH • 0x00000012 – NI_SS_SWEDISH • 0x00000013 – NI_SS_DANISH • 0x00000014 – NI_SS_PORTUGUESE • 0x00000015 – NI_SS_FINNISH • 0x00000016 – NI_SS_NORWEGIAN • 0x00000017 – NI_SS_GREEK • 0x00000018 – NI_SS_TURKISH • 0x00000019 – NI_SS_HUNGARIAN • 0x0000001A – NI_SS_POLISH • 0x0000001B – NI_SS_LANGUAGE_UNSPEC • 0x0000001C – NI_SUPL_UTF8 • 0x0000001D – NI_SUPL_UCS2 • 0x0000001E – NI_SUPL_GSM_DEFAULT
		notificationText_len	1	Number of sets of the following elements: • notificationText
		notificationText	Var	Notification text; the encoding method is specified in dataCodingScheme. • Type: Array of bytes • Maximum array length: 64
		clientAddress_len	1	Number of sets of the following elements: • clientAddress
		clientAddress	Var	Client address; the encoding method is specified in dataCodingScheme. • Maximum array length: 20
		locationType	4	Location type. Valid values: • 0x00000001 – CURRENT_LOCATION • 0x00000002 – CURRENT_OR_LAST_KNOWN_LOCATION • 0x00000004 – INITIAL_LOCATION

Field	Field value	Parameter	Size (byte)	Description
		dataCodingScheme	4	Identifies the coding scheme of the coded string. Valid values: • 0x0000000C – NI_SS_GERMAN • 0x0000000D – NI_SS_ENGLISH • 0x0000000E – NI_SS_ITALIAN • 0x0000000F – NI_SS_FRENCH • 0x00000010 – NI_SS_SPANISH • 0x00000011 – NI_SS_DUTCH • 0x00000012 – NI_SS_SWEDISH • 0x00000013 – NI_SS_DANISH • 0x00000014 – NI_SS_PORTUGUESE • 0x00000015 – NI_SS_FINNISH • 0x00000016 – NI_SS_NORWEGIAN • 0x00000017 – NI_SS_GREEK • 0x00000018 – NI_SS_TURKISH • 0x00000019 – NI_SS_HUNGARIAN • 0x0000001A – NI_SS_POLISH • 0x0000001B – NI_SS_LANGUAGE_ UNSPEC • 0x0000001C – NI_SUPL_UTF8 • 0x0000001D – NI_SUPL_UCS2 • 0x0000001E – NI_SUPL_GSM_DEFAULT
		codedString_len	1	Number of sets of the following elements: • codedString
		codedString	Var	Coded string. • Type: Array of bytes • Maximum string length: 20

Field	Field value	Parameter	Size (byte)	Description
		dataCodingScheme	4	Identifies the coding scheme of the coded string. Valid values: • 0x0000000C – NI_SS_GERMAN • 0x0000000D – NI_SS_ENGLISH • 0x0000000E – NI_SS_ITALIAN • 0x0000000F – NI_SS_FRENCH • 0x00000010 – NI_SS_SPANISH • 0x00000011 – NI_SS_DUTCH • 0x00000012 – NI_SS_SWEDISH • 0x00000013 – NI_SS_DANISH • 0x00000014 – NI_SS_PORTUGUESE • 0x00000015 – NI_SS_FINNISH • 0x00000016 – NI_SS_NORWEGIAN • 0x00000017 – NI_SS_GREEK • 0x00000018 – NI_SS_TURKISH • 0x00000019 – NI_SS_HUNGARIAN • 0x0000001A – NI_SS_POLISH • 0x0000001B – NI_SS_LANGUAGE_ UNSPEC • 0x0000001C – NI_SUPL_UTF8 • 0x0000001D – NI_SUPL_UCS2 • 0x0000001E – NI_SUPL_GSM_DEFAULT
		codedString_len	1	Number of sets of the following elements: • codedString
		codedString	Var	Coded string. • Type: Array of bytes • Maximum string length: 20
		lcsServiceTypeId	1	Service type ID. • Type: Unsigned integer
		userResponseTimer	2	Time to wait for the user to respond. • Type: Unsigned integer • Units: Seconds
Type	0x13		1	Network Initiated Service Interaction Request Optional NI service interaction payload.
Length	Var		2	
Value	→	posQosIncl	1	Whether quality of service is included: • 0x01 (TRUE) – QoS is included • 0x00 (FALSE) – QoS is not included
		posQos	1	Position QoS timeout. • Type: Unsigned integer • Units: Seconds • Range: 0 to 255
		numFixes	4	Number of fixes allowed. • Type: Unsigned integer
		timeBetweenFixes	4	Time between fixes. • Type: Unsigned integer • Units: Seconds

Field	Field value	Parameter	Size (byte)	Description
		posMode	4	Position mode. Valid values: • 0x00000001 – NI_VX_MS_ASSISTED_ONLY • 0x00000002 – NI_VX_MS_BASED_ONLY • 0x00000003 – NI_VX_MS_ASSISTED_PREFERRED_MS_BASED_ALLOWED • 0x00000004 – NI_VX_MS_BASED_PREFERRED_MS_ASSISTED_ALLOWED
		encodingScheme	4	VX encoding scheme. Valid values: • 0x00000000 – NI_VX_OCTET • 0x00000001 – NI_VX_EXN_PROTOCOL_MSG • 0x00000002 – NI_VX_ASCII • 0x00000003 – NI_VX_IA5 • 0x00000004 – NI_VX_UNICODE • 0x00000005 – NI_VX_SHIFT_JIS • 0x00000006 – NI_VX_KOREAN • 0x00000007 – NI_VX_LATIN_HEBREW • 0x00000008 – NI_VX_LATIN • 0x00000009 – NI_VX_GSM
		requestorId_len	1	Number of sets of the following elements: • requestorId
		requestorId	Var	Requestor ID. • Type: Array of bytes • Maximum array length: 200
		userRespTimerInSeconds	2	Time to wait for the user to respond. • Type: Unsigned integer • Units: Seconds
		serviceInteractionType	4	Service interaction type specified in qmiLocNiServiceInteractionEnumT. Valid values: • 0x00000001 – ONGOING_NI_INCOMING_MO

Error codes

QMI_ERR_NONE	No error in the request
QMI_ERR_INTERNAL	Unexpected error occurred during processing
QMI_ERR_MALFORMED_MSG	Message was not formulated correctly by the control point or the message was corrupted during transmission
QMI_ERR_NO_MEMORY	Device could not allocate memory to formulate a response
QMI_ERR_INVALID_HANDLE	Invalid client handle was received

3.9.2 Description of QMI_LOC_EVENT_NI_NOTIFY_VERIFY_REQ

This event is used to send the Notify Verify request to the control point. The Notify Verify request is relayed to the control point only if the control point has indicated its interest by setting the appropriate mask in QMI_LOC_REG_EVENTS_REQ.

3.10 QMI_LOC_EVENT_INJECT_TIME_REQ

Requests the control point to inject time information.

LOC message ID

0x0028

Version introduced

Major - 2, Minor - 2

3.10.1 Indication - QMI_LOC_EVENT_INJECT_TIME_REQ_IND

Message type

Indication

Sender

Service

Mandatory TLVs

None

Optional TLVs

Name	Version last modified
Time Server Info	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x10		1	Time Server Info Contains information about the time servers recommended by the location service for NTP time.
Length	Var		2	
Value	→	delayThreshold	4	The time server is to be skipped if a one-way delay to the server exceeds this threshold. • Type: Unsigned integer • Units: Milliseconds

Field	Field value	Parameter	Size (byte)	Description
		timeServerList_len	1	Number of sets of the following elements: • serverUrl_len • serverUrl
		serverUrl_len	1	Number of sets of the following elements: • serverUrl
		serverUrl	Var	Assistance server URL. • Type: NULL-terminated string • Maximum string length (including NULL terminator): 256

Error codes

QMI_ERR_NONE	No error in the request
QMI_ERR_INTERNAL	Unexpected error occurred during processing
QMI_ERR_MALFORMED_MSG	Message was not formulated correctly by the control point or the message was corrupted during transmission
QMI_ERR_NO_MEMORY	Device could not allocate memory to formulate a response
QMI_ERR_INVALID_HANDLE	Invalid client handle was received

3.10.2 Description of QMI_LOC_EVENT_INJECT_TIME_REQ

This event is used to send the Time Injection request to the control point. The Time Injection request is relayed to the control point only if the control point has indicated its interest by setting the appropriate mask in QMI_LOC_REG_EVENTS_REQ.

3.11 QMI_LOC_EVENT_INJECT_PREDICTED_ORBITS_REQ

Requests the control point to inject predicted orbits data.

LOC message ID

0x0029

Version introduced

Major - 2, Minor - 2

3.11.1 Indication - QMI_LOC_EVENT_INJECT_PREDICTED_ORBITS_REQ_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version last modified
Allowed Sizes	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x01		1	Allowed Sizes Maximum part and file size allowed to be injected in the engine.
Length	8		2	
Value	→	maxFileSizeInBytes	4	Maximum allowable predicted orbits file size (in bytes). • Type: Unsigned integer
		maxPartSize	4	Maximum allowable predicted orbits file chunk size (in bytes). • Type: Unsigned integer

Optional TLVs

Name	Version last modified
Server List	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x10		1	Server List List of servers that can be used by the client to download predicted orbits data.
Length	Var		2	
Value	→	serverList_len	1	Number of sets of the following elements: • serverUrl_len • serverUrl
		serverUrl_len	1	Number of sets of the following elements: • serverUrl
		serverUrl	Var	Assistance server URL. • Type: NULL-terminated string • Maximum string length (including NULL terminator): 256

Error codes

QMI_ERR_NONE	No error in the request
QMI_ERR_INTERNAL	Unexpected error occurred during processing
QMI_ERR_MALFORMED_MSG	Message was not formulated correctly by the control point or the message was corrupted during transmission
QMI_ERR_NO_MEMORY	Device could not allocate memory to formulate a response
QMI_ERR_INVALID_HANDLE	Invalid client handle was received

3.11.2 Description of QMI_LOC_EVENT_INJECT_PREDICTED_ORBITS_REQ

This event is used to request the control point to inject predicted orbits data information. This request is generated only if the control point has indicated its interest by setting the appropriate mask in QMI_LOC_REG_EVENTS_REQ.

3.12 QMI_LOC_EVENT_INJECT_POSITION_REQ

Requests the control point to inject a position.

LOC message ID

0x002A

Version introduced

Major - 2, Minor - 2

3.12.1 Indication - QMI_LOC_EVENT_INJECT_POSITION_REQ_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version last modified
Latitude	2.2
Longitude	2.2
Circular Horizontal Uncertainty	2.2
UTC Timestamp	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x01		1	Latitude
Length	8		2	
Value	→	latitude	8	Latitude (specified in WGS84 datum). • Type: Floating point • Units: Degrees • Range: -90.0 to 90.0 • Positive values indicate northern latitude • Negative values indicate southern latitude
Type	0x02		1	Longitude
Length	8		2	

Field	Field value	Parameter	Size (byte)	Description
Value	→	longitude	8	Longitude (specified in WGS84 datum). • Type: Floating point • Units: Degrees • Range: -180.0 to 180.0 • Positive values indicate eastern longitude • Negative values indicate western longitude
Type	0x03		1	Circular Horizontal Uncertainty
Length	4		2	
Value	→	horUncCircular	4	Horizontal position uncertainty (circular). • Type: Floating point • Units: Meters
Type	0x04		1	UTC Timestamp
Length	8		2	
Value	→	timestampUtc	8	UTC timestamp. • Type: Unsigned integer • Units: Milliseconds since Jan. 1, 1970

Optional TLVs

None

Error codes

QMI_ERR_NONE	No error in the request
QMI_ERR_INTERNAL	Unexpected error occurred during processing
QMI_ERR_MALFORMED_MSG	Message was not formulated correctly by the control point or the message was corrupted during transmission
QMI_ERR_NO_MEMORY	Device could not allocate memory to formulate a response
QMI_ERR_INVALID_HANDLE	Invalid client handle was received

3.12.2 Description of QMI_LOC_EVENT_INJECT_POSITION_REQ

This command sends a Position Injection request to the control point. It contains the current position estimate that the location engine has. If the position estimate of the client is worse than (greater uncertainty, older timestamp, etc.) the position estimate, the client does not need to inject it. This request is generated only if the control point has indicated its interest by setting the appropriate mask in QMI_LOC_REG_EVENTS_REQ.

3.13 QMI_LOC_EVENT_ENGINE_STATE

Sends the engine state to the control point.

LOC message ID

0x002B

Version introduced

Major - 2, Minor - 2

3.13.1 Indication - QMI_LOC_EVENT_ENGINE_STATE_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version last modified
Engine State	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x01		1	Engine State
Length	4		2	
Value	→	engineState	4	Location engine state. Valid values: • 0x00000001 – ON • 0x00000002 – OFF

Optional TLVs

None

Error codes

QMI_ERR_NONE	No error in the request
QMI_ERR_INTERNAL	Unexpected error occurred during processing
QMI_ERR_MALFORMED_MSG	Message was not formulated correctly by the control point or the message was corrupted during transmission
QMI_ERR_NO_MEMORY	Device could not allocate memory to formulate a response
QMI_ERR_INVALID_HANDLE	Invalid client handle was received

3.13.2 Description of QMI_LOC_EVENT_ENGINE_STATE

This command sends the GPS State Information event to the control point. This request is generated only if the control point has indicated its interest by setting the appropriate mask in QMI_LOC_REG_EVENTS_REQ

3.14 QMI_LOC_EVENT_FIX_SESSION_STATE

Sends the fix session state to the control point.

LOC message ID

0x002C

Version introduced

Major - 2, Minor - 2

3.14.1 Indication - QMI_LOC_EVENT_FIX_SESSION_STATE_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version last modified
Session State	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x01		1	Session State
Length	4		2	
Value	→	sessionState	4	LOC fix session state. Valid values: • 0x00000001 – STARTED • 0x00000002 – FINISHED

Optional TLVs

Name	Version last modified
Session ID	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x10		1	Session ID
Length	1		2	
Value	→	sessionId	1	ID of the session that was specified in the Start request. This may not be specified for a fix session corresponding to a network-initiated request. • Type: Unsigned integer • Range: 0 to 255

Error codes

QMI_ERR_NONE	No error in the request
QMI_ERR_INTERNAL	Unexpected error occurred during processing
QMI_ERR_MALFORMED_MSG	Message was not formulated correctly by the control point or the message was corrupted during transmission
QMI_ERR_NO_MEMORY	Device could not allocate memory to formulate a response
QMI_ERR_INVALID_HANDLE	Invalid client handle was received

3.14.2 Description of QMI_LOC_EVENT_FIX_SESSION_STATE

This command sends the Fix Session State Information event to the control point. This request is generated only if the control point has indicated its interest by setting the appropriate mask in QMI_LOC_REG_EVENTS_REQ

3.15 QMI_LOC_EVENT_WIFI_REQ

Sends a WiFi request to the control point.

LOC message ID

0x002D

Version introduced

Major - 2, Minor - 2

3.15.1 Indication - QMI_LOC_EVENT_WIFI_REQ_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version last modified
Request Type	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x01		1	Request Type
Length	4		2	
Value	→	requestType	4	Request type as specified in qmiWifiRequestEnumT. Valid values: • 0x00000000 – START_PERIODIC_HI_FREQ_FIXES • 0x00000001 – START_PERIODIC_KEEP_WARM • 0x00000002 – STOP_PERIODIC_FIXES

Optional TLVs

Name	Version last modified
Time Between Fixes	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x10		1	Time Between Fixes
Length	2		2	
Value	→	tbfInMs	2	Time between fixes for a periodic request. • Type: Unsigned integer • Units: Milliseconds

Error codes

QMI_ERR_NONE	No error in the request
QMI_ERR_INTERNAL	Unexpected error occurred during processing
QMI_ERR_MALFORMED_MSG	Message was not formulated correctly by the control point or the message was corrupted during transmission
QMI_ERR_NO_MEMORY	Device could not allocate memory to formulate a response
QMI_ERR_INVALID_HANDLE	Invalid client handle was received

3.15.2 Description of QMI_LOC_EVENT_WIFI_REQ

This command sends a WiFi Position Request event to the control point. This request is generated only if the control point has indicated its interest by setting the appropriate mask in QMI_LOC_REG_EVENTS_REQ.

3.16 QMI_LOC_EVENT_SENSOR_STREAMING_READY_STATUS

Notifies the control point if the GNSS location engine is ready to accept sensor data.

LOC message ID

0x002E

Version introduced

Major - 2, Minor - 2

3.16.1 Indication - QMI_LOC_EVENT_SENSOR_STREAMING_READY_STATUS_IND

Message type

Indication

Sender

Service

Mandatory TLVs

None

Optional TLVs

Name	Version last modified
Accelerometer Accept Ready	2.2
Gyrometer Accept Ready	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x10		1	Accelerometer Accept Ready Whether the GNSS location engine is ready to accept accelerometer sensor data.
Length	5		2	

Field	Field value	Parameter	Size (byte)	Description
Value	→	injectEnable	1	Whether the GNSS location engine is ready to accept data from this sensor. Valid values: • 0x01 (TRUE) – GNSS location engine is ready to accept sensor data • 0x00 (FALSE) – GNSS location engine is not ready to accept sensor data
		samplesPerBatch	2	Specifies the number of samples per batch the GNSS location engine is to receive. The sensor sampling frequency can be computed as follows: samplingFrequency = samplesPerBatch * batchesPerSecond samplesPerBatch must be a non-zero positive value.
		batchesPerSecond	2	Number of sensor-data batches the GNSS location engine is to receive per second. The rate is specified in integral number of batches per second (Hz). batchesPerSecond must be a non-zero positive value.
Type	0x11		1	Gyrometer Accept Ready Whether the GNSS location engine is ready to accept gyrometer sensor data.
Length	5		2	
Value	→	injectEnable	1	Whether the GNSS location engine is ready to accept data from this sensor. Valid values: • 0x01 (TRUE) – GNSS location engine is ready to accept sensor data • 0x00 (FALSE) – GNSS location engine is not ready to accept sensor data
		samplesPerBatch	2	Specifies the number of samples per batch the GNSS location engine is to receive. The sensor sampling frequency can be computed as follows: samplingFrequency = samplesPerBatch * batchesPerSecond samplesPerBatch must be a non-zero positive value.

Field	Field value	Parameter	Size (byte)	Description
		batchesPerSecond	2	Number of sensor-data batches the GNSS location engine is to receive per second. The rate is specified in integral number of batches per second (Hz). batchesPerSecond must be a non-zero positive value.

Error codes

QMI_ERR_NONE	No error in the request
QMI_ERR_INTERNAL	Unexpected error occurred during processing
QMI_ERR_MALFORMED_MSG	Message was not formulated correctly by the control point or the message was corrupted during transmission
QMI_ERR_NO_MEMORY	Device could not allocate memory to formulate a response
QMI_ERR_INVALID_HANDLE	Invalid client handle was received

3.16.2 Description of QMI_LOC_EVENT_SENSOR_STREAMING_READY_STATUS

This command sends a Sensor Streaming Ready status event to the control point. This request is generated only if the control point has indicated its interest by setting the appropriate mask in QMI_LOC_REG_EVENTS_REQ.

3.17 QMI_LOC_EVENT_TIME_SYNC_REQ

Notifies the control point to inject time synchronization data.

LOC message ID

0x002F

Version introduced

Major - 2, Minor - 2

3.17.1 Indication - QMI_LOC_EVENT_TIME_SYNC_REQ_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version last modified
Opaque Time Sync Reference Counter	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x01		1	Opaque Time Sync Reference Counter
Length	4		2	
Value	→	refCounter	4	This TLV is sent to registered control points. It is sent by the location engine when it needs to synchronize location engine and control point (sensor processor) times. This TLV must be echoed back in the Time Sync Inject request. • Type: Unsigned integer

Optional TLVs

None

Error codes

QMI_ERR_NONE	No error in the request
QMI_ERR_INTERNAL	Unexpected error occurred during processing
QMI_ERR_MALFORMED_MSG	Message was not formulated correctly by the control point or the message was corrupted during transmission
QMI_ERR_NO_MEMORY	Device could not allocate memory to formulate a response
QMI_ERR_INVALID_HANDLE	Invalid client handle was received

3.17.2 Description of QMI_LOC_EVENT_TIME_SYNC_REQ

This command sends a Time Synchronization Request event to the control point. This request is generated only if the control point has indicated its interest by setting the appropriate mask in QMI_LOC_REG_EVENTS_REQ.

3.18 QMI_LOC_EVENT_SET_SPI_STREAMING_REPORT

Requests the control point to enable Stationary Position Indicator (SPI) streaming reports.

LOC message ID

0x0030

Version introduced

Major - 2, Minor - 2

3.18.1 Indication - QMI_LOC_EVENT_SET_SPI_STREAMING_REPORT_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version last modified
Enable/Disable SPI Requests	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x01		1	Enable/Disable SPI Requests
Length	1		2	
Value	→	enable	1	Whether the client is to start or stop sending an SPI status stream. • 0x01 (TRUE) – Client is to start sending an SPI status stream • 0x00 (FALSE) – Client is to stop sending an SPI status stream

Optional TLVs

None

Error codes

QMI_ERR_NONE	No error in the request
QMI_ERR_INTERNAL	Unexpected error occurred during processing
QMI_ERR_MALFORMED_MSG	Message was not formulated correctly by the control point or the message was corrupted during transmission
QMI_ERR_NO_MEMORY	Device could not allocate memory to formulate a response
QMI_ERR_INVALID_HANDLE	Invalid client handle was received

3.18.2 Description of QMI_LOC_EVENT_SET_SPI_STREAMING_REPORT

This command notifies the client to start/stop injections of SPI reports. This request is generated only if the control point has indicated its interest by setting the appropriate mask in QMI_LOC_REG_EVENTS_REQ.

3.19 QMI_LOC_EVENT_LOCATION_SERVER_CONNECTION_REQ

Requests the client to open or close a connection to the assisted GPS location server.

LOC message ID

0x0031

Version introduced

Major - 2, Minor - 2

3.19.1 Indication - QMI_LOC_EVENT_LOCATION_SERVER_CONNECTION_REQ_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version last modified
Connection Handle	2.2
Request Type	2.2
WWAN Type	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x01		1	Connection Handle
Length	4		2	
Value	→	connHandle	4	Identifies a connection across Open and Close request events. • Type: Unsigned integer
Type	0x02		1	Request Type
Length	4		2	
Value	→	requestType	4	Open or close a connection to the location server. Valid values: • 0x00000001 – OPEN • 0x00000002 – CLOSE

Field	Field value	Parameter	Size (byte)	Description
Type	0x03		1	WWAN Type
Length	4		2	
Value	→	wwanType	4	Identifies the WWAN type for this request. Valid values: • 0x00000000 – WWAN_TYPE_INTERNET • 0x00000001 – WWAN_TYPE_AGNSS

Optional TLVs

None

Error codes

QMI_ERR_NONE	No error in the request
QMI_ERR_INTERNAL	Unexpected error occurred during processing
QMI_ERR_MALFORMED_MSG	Message was not formulated correctly by the control point or the message was corrupted during transmission
QMI_ERR_NO_MEMORY	Device could not allocate memory to formulate a response
QMI_ERR_INVALID_HANDLE	Invalid client handle was received

3.19.2 Description of QMI_LOC_EVENT_LOCATION_SERVER_CONNECTION_REQ

This command notifies the client to open/close a connection to the server. This request is generated only if the control point has indicated its interest by setting the appropriate mask in QMI_LOC_REG_EVENTS_REQ.

3.20 QMI_LOC_GET_SERVICE_REVISION

Client can query the service revision using this message.

LOC message ID

0x0032

Version introduced

Major - 2, Minor - 2

3.20.1 Request - QMI_LOC_GET_SERVICE_REVISION_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

None

Optional TLVs

None

3.20.2 Indication - QMI_LOC_GET_SERVICE_REVISION_IND

Message type

Indication

Sender

Service

Mandatory TLVs

At least one of the following optional TLVs is present if the status is eQMI_LOC_SUCCESS.

Name	Version last modified
Get Revision Status	2.2
Interface Definition Minor Revision	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x01		1	Get Revision Status
Length	4		2	
Value	→	status	4	Status of the Get Revision request. Valid values: • 0x00000000 – SUCCESS • 0x00000001 – GENERAL_FAILURE • 0x00000002 – UNSUPPORTED • 0x00000003 – INVALID_PARAMETER • 0x00000004 – ENGINE_BUSY • 0x00000005 – PHONE_OFFLINE • 0x00000006 – LOC_TIMEOUT
Type	0x02		1	Interface Definition Minor Revision
Length	4		2	
Value	→	revision	4	Revision of the service. This is the minor revision of the interface that the service implements. Minor revision updates of the service are always backward compatible. • Type: Unsigned integer

Optional TLVs

None

Error codes

QMI_ERR_NONE	No error in the request
QMI_ERR_INTERNAL	Unexpected error occurred during processing
QMI_ERR_MALFORMED_MSG	Message was not formulated correctly by the control point or the message was corrupted during transmission
QMI_ERR_NO_MEMORY	Device could not allocate memory to formulate a response
QMI_ERR_INVALID_HANDLE	Invalid client handle was received

3.20.3 Description of QMI_LOC_GET_SERVICE_REVISION

This command is used by the control point to query the service for its minor revision. The client can use the service revision of the QMI_LOC service to decide if it is compatible with the service.

3.21 QMI_LOC_GET_FIX_CRITERIA

Gets the fix criteria from the location engine.

LOC message ID

0x0033

Version introduced

Major - 2, Minor - 2

3.21.1 Request - QMI_LOC_GET_FIX_CRITERIA_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

None

Optional TLVs

None

3.21.2 Indication - QMI_LOC_GET_FIX_CRITERIA_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version last modified
Get Fix Criteria Status	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x01		1	Get Fix Criteria Status
Length	4		2	
Value	→	status	4	Status of the Get Fix Criteria request. Valid values: • 0x00000000 – SUCCESS • 0x00000001 – GENERAL_FAILURE • 0x00000002 – UNSUPPORTED • 0x00000003 – INVALID_PARAMETER • 0x00000004 – ENGINE_BUSY • 0x00000005 – PHONE_OFFLINE • 0x00000006 – LOC_TIMEOUT

Optional TLVs

At least one of the following optional TLVs is present if the status is eQMI_LOC_SUCCESS.

Name	Version last modified
Horizontal Accuracy	2.2
Enable/Disable Intermediate Fixes	2.2
Minimum Interval Between Fixes	2.2
ID of the Application that Sent the Position Request	2.6

Field	Field value	Parameter	Size (byte)	Description
Type	0x10		1	Horizontal Accuracy
Length	4		2	
Value	→	horizontalAccuracyLevel	4	Horizontal accuracy level. Valid values: • 0x00000001 – LOW: Client requires low horizontal accuracy. • 0x00000002 – MED: Client requires medium horizontal accuracy. • 0x00000003 – HIGH: Client requires high horizontal accuracy.
Type	0x11		1	Enable/Disable Intermediate Fixes
Length	4		2	

Field	Field value	Parameter	Size (byte)	Description
Value	→	intermediateReportState	4	Intermediate Report state (ON, OFF). The client must explicitly set this field to OFF to stop receiving intermediate position reports. Intermediate position reports are generated at 1 Hz and are ON by default. If intermediate reports are turned ON, the client receives position reports even if the accuracy criteria is not met. The status in the position report is set to IN_PROGRESS for intermediate reports. Valid values: • 0x00000001 – ON: Client is interested in receiving intermediate reports • 0x00000002 – OFF: Client is not interested in receiving intermediate reports
Type	0x12		1	Mimimum Interval Between Fixes
Length	4		2	
Value	→	minInterval	4	Time that must elapse before alerting the client. • Type: Unsigned integer • Units: Milliseconds
Type	0x13		1	ID of the Application that Sent the Position Request Application provider, name, and version.
Length	Var		2	
Value	→	applicationProvider_len	1	Number of sets of the following elements: • applicationProvider
		applicationProvider	Var	Application provider.
		applicationName_len	1	Number of sets of the following elements: • applicationName
		applicationName	Var	Application name.
		applicationVersion_valid	1	Specifies whether the application version string contains a valid value: • 0x00 (FALSE) – Application version string is invalid • 0x01 (TRUE) – Application version string is valid
		applicationVersion_len	1	Number of sets of the following elements: • applicationVersion
		applicationVersion	Var	Application version.

Error codes

QMI_ERR_NONE	No error in the request
QMI_ERR_INTERNAL	Unexpected error occurred during processing
QMI_ERR_MALFORMED_MSG	Message was not formulated correctly by the control point or the message was corrupted during transmission
QMI_ERR_NO_MEMORY	Device could not allocate memory to formulate a response
QMI_ERR_INVALID_HANDLE	Invalid client handle was received

3.21.3 Description of QMI_LOC_GET_FIX_CRITERIA

This command is used to get the fix criteria under which the service is currently operating for this control point. If the fix criteria was not set previously, it returns the default fix criteria. The request is acknowledged through the response. If successful, the indication also contains the fix criteria for this control point.

3.22 QMI_LOC_INFORM_NI_USER_RESPONSE

Sends the NI user response back to the engine; success or failure is reported in a separate indication.

LOC message ID

0x0034

Version introduced

Major - 2, Minor - 2

3.22.1 Request - QMI_LOC_NI_USER_RESPONSE_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

Name	Version last modified
User Response	2.2
Notification Type	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x01		1	User Response
Length	4		2	
Value	→	userResp	4	User accepted or denied. Valid values: • 0x00000001 – NOTIFY_VERIFY_ACCEPT • 0x00000002 – NOTIFY_VERIFY_DENY • 0x00000003 – NOTIFY_VERIFY_NORESP
Type	0x02		1	Notification Type
Length	4		2	

Field	Field value	Parameter	Size (byte)	Description
Value	→	notificationType	4	Type of notification/verification performed. Valid values: • 0x00000001 – NO_NOTIFY_NO_VERIFY • 0x00000002 – NOTIFY_ONLY • 0x00000003 – ALLOW_NO_RESP • 0x00000004 – NOT_ALLOW_NO_RESP • 0x00000005 – PRIVACY_OVERRIDE

Optional TLVs

Name	Version last modified
Network Initiated Vx Request	2.2
Network Initiated SUPL Request	2.2
Network Initiated UMTS Control Plane Request	2.2
Network Initiated Service Interaction Request	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x10		1	Network Initiated Vx Request Optional NI VX request payload.
Length	Var		2	
Value	→	posQosIncl	1	Whether quality of service is included: • 0x01 (TRUE) – QoS is included • 0x00 (FALSE) – QoS is not included
		posQos	1	Position QoS timeout. • Type: Unsigned integer • Units: Seconds • Range: 0 to 255
		numFixes	4	Number of fixes allowed. • Type: Unsigned integer
		timeBetweenFixes	4	Time between fixes. • Type: Unsigned integer • Units: Seconds
		posMode	4	Position mode. Valid values: • 0x00000001 – NI_VX_MS_ASSISTED_ONLY • 0x00000002 – NI_VX_MS_BASED_ONLY • 0x00000003 – NI_VX_MS_ASSISTED_PREFERRED_MS_BASED_ALLOWED • 0x00000004 – NI_VX_MS_BASED_PREFERRED_MS_ASSISTED_ALLOWED

Field	Field value	Parameter	Size (byte)	Description
		encodingScheme	4	VX encoding scheme. Valid values: • 0x00000000 – NI_VX_OCTET • 0x00000001 – NI_VX_EXN_PROTOCOL_MSG • 0x00000002 – NI_VX_ASCII • 0x00000003 – NI_VX_IA5 • 0x00000004 – NI_VX_UNICODE • 0x00000005 – NI_VX_SHIFT_JIS • 0x00000006 – NI_VX_KOREAN • 0x00000007 – NI_VX_LATIN_HEBREW • 0x00000008 – NI_VX_LATIN • 0x00000009 – NI_VX_GSM
		requestorId_len	1	Number of sets of the following elements: • requestorId
		requestorId	Var	Requestor ID. • Type: Array of bytes • Maximum array length: 200
		userRespTimerInSeconds	2	Time to wait for the user to respond. • Type: Unsigned integer • Units: Seconds
Type	0x11		1	Network Initiated SUPL Request Optional NI SUPL request payload.
Length	Var		2	
Value	→	valid_flags	4	Indicates which of the following fields are present in this value. Valid bitmasks: • 0x00000001 – SUPL_SERVER_INFO • 0x00000002 – SUPL_SESSION_ID • 0x00000004 – SUPL_HASH • 0x00000008 – SUPL_POS_METHOD • 0x00000010 – SUPL_DATA_CODING_SCHEME • 0x00000020 – SUPL_REQUESTOR_ID • 0x00000040 – SUPL_CLIENT_NAME • 0x00000080 – SUPL_QOP • 0x00000100 – SUPL_USER_RESP_TIMER
		suplServerAddrTypeMask	1	Mask specifying the valid fields in this value. Valid bitmasks: • 0x01 – IPV4 • 0x02 – IPV6 • 0x04 – URL
		addr	4	IPV4 address. • Type: Unsigned integer
		port	2	IPV4 port. • Type: Unsigned integer

Field	Field value	Parameter	Size (byte)	Description
		addr	16	IPV6 address. • Type: Array of unsigned integers • Maximum length of the array: 8
		port	4	IPV6 port. • Type: Unsigned integer
		urlAddr_len	1	Number of sets of the following elements: • urlAddr
		urlAddr	Var	URL. • Type: NULL-terminated string • Maximum string length (including NULL terminator): 256
		suplSessionId	4	SUPL session ID. • Type: Array of unsigned integers • Maximum length of the array: 4
		suplHash	8	Hash for SUPL_INIT; used to validate that the message was not corrupted. • Type: Array of unsigned integers • Length of the array: 8
		posMethod	4	GPS mode to be used for the fix. Valid values: • 0x00000001 – AGPS_SETASSISTED • 0x00000002 – AGPS_SETBASED • 0x00000003 – AGPS_SETASSISTED_PREF • 0x00000004 – AGPS_SETBASED_PREF • 0x00000005 – AUTONOMOUS_GPS • 0x00000006 – AFLT • 0x00000007 – ECID • 0x00000008 – EOTD • 0x00000009 – OTDOA • 0x0000000A – NO_POSITION

Field	Field value	Parameter	Size (byte)	Description
		dataCodingScheme	4	Data coding scheme applies to both the requestor ID and the client name. Valid values: • 0x0000000C – NI_SS_GERMAN • 0x0000000D – NI_SS_ENGLISH • 0x0000000E – NI_SS_ITALIAN • 0x0000000F – NI_SS_FRENCH • 0x00000010 – NI_SS_SPANISH • 0x00000011 – NI_SS_DUTCH • 0x00000012 – NI_SS_SWEDISH • 0x00000013 – NI_SS_DANISH • 0x00000014 – NI_SS_PORTUGUESE • 0x00000015 – NI_SS_FINNISH • 0x00000016 – NI_SS_NORWEGIAN • 0x00000017 – NI_SS_GREEK • 0x00000018 – NI_SS_TURKISH • 0x00000019 – NI_SS_HUNGARIAN • 0x0000001A – NI_SS_POLISH • 0x0000001B – NI_SS_LANGUAGE_ UNSPEC • 0x0000001C – NI_SUPL_UTF8 • 0x0000001D – NI_SUPL_UCS2 • 0x0000001E – NI_SUPL_GSM_DEFAULT
		formatType	4	Format of the formatted string. Valid values: • 0x00000000 – FORMAT_LOGICAL_NAME • 0x00000001 – FORMAT_EMAIL_ ADDRESS • 0x00000002 – FORMAT_MSISDN • 0x00000003 – FORMAT_URL • 0x00000004 – FORMAT_SIP_URL • 0x00000005 – FORMAT_MIN • 0x00000006 – FORMAT_MDN • 0x00000007 – FORMAT_IMSPUBLIC_ IDENTITY • 0x7FFFFFFF – FORMAT_OSS_UNKNOWN
		formattedString_len	1	Number of sets of the following elements: • formattedString
		formattedString	Var	Formatted string. • Type: Byte array • Maximum string length: 64

Field	Field value	Parameter	Size (byte)	Description
		formatType	4	Format of the formatted string. Valid values: • 0x00000000 – FORMAT_LOGICAL_NAME • 0x00000001 – FORMAT_EMAIL_ADDRESS • 0x00000002 – FORMAT_MSISDN • 0x00000003 – FORMAT_URL • 0x00000004 – FORMAT_SIP_URL • 0x00000005 – FORMAT_MIN • 0x00000006 – FORMAT_MDN • 0x00000007 – FORMAT_IMSPUBLIC_IDENTITY • 0x7FFFFFFF – FORMAT_OSS_UNKNOWN
		formattedString_len	1	Number of sets of the following elements: • formattedString
		formattedString	Var	Formatted string. • Type: Byte array • Maximum string length: 64
		validMask	1	Bit field indicating which fields are valid in this value. Valid bitmasks: • 0x01 – QOP_HORZ_ACC_VALID • 0x02 – QOP_VER_ACC_VALID • 0x04 – QOP_MAXAGE_VALID • 0x08 – QOP_DELAY_VALID
		horizontalAccuracy	1	Horizontal accuracy. • Type: Unsigned integer • Units: Meters
		verticalAccuracy	1	Vertical accuracy. • Type: Unsigned integer • Units: Meters
		maxLocAge	2	Maximum age of the location if the engine sends a previously computed position. • Type: Unsigned integer • Units: Seconds
		delay	1	Delay the server is willing to tolerate for the fix. • Type: Unsigned integer • Units: Seconds
		userResponseTimer	2	Time to wait for the user to respond. • Type: Unsigned integer • Units: Seconds
Type	0x12		1	Network Initiated UMTS Control Plane Request Optional NI UMTS-CP request payload.
Length	Var		2	

Field	Field value	Parameter	Size (byte)	Description
Value	→	valid_flags	2	Fields that are valid in this value. Valid bitmasks: • 0x0001 – INVOKE_ID_MASK • 0x0002 – DATA_CODING_SCHEME_ MASK • 0x0004 – NOTIFICATION_TEXT_MASK • 0x0008 – CLIENT_ADDRESS_MASK • 0x0010 – LOCATION_TYPE_MASK • 0x0020 – REQUESTOR_ID_MASK • 0x0040 – CODEWORD_STRING_MASK • 0x0080 – SERVICE_TYPE_MASK • 0x0100 – USER_RESP_TIMER_MASK
		invokeId	1	Supplementary Services invoke ID. • Type: Unsigned integer
		dataCodingScheme	4	Type of data encoding scheme for the text. Applies to both the notification text and the client address. Valid values: • 0x0000000C – NI_SS_GERMAN • 0x0000000D – NI_SS_ENGLISH • 0x0000000E – NI_SS_ITALIAN • 0x0000000F – NI_SS_FRENCH • 0x00000010 – NI_SS_SPANISH • 0x00000011 – NI_SS_DUTCH • 0x00000012 – NI_SS_SWEDISH • 0x00000013 – NI_SS_DANISH • 0x00000014 – NI_SS_PORTUGUESE • 0x00000015 – NI_SS_FINNISH • 0x00000016 – NI_SS_NORWEGIAN • 0x00000017 – NI_SS_GREEK • 0x00000018 – NI_SS_TURKISH • 0x00000019 – NI_SS_HUNGARIAN • 0x0000001A – NI_SS_POLISH • 0x0000001B – NI_SS_LANGUAGE_ UNSPEC • 0x0000001C – NI_SUPL_UTF8 • 0x0000001D – NI_SUPL_UCS2 • 0x0000001E – NI_SUPL_GSM_DEFAULT
		notificationText_len	1	Number of sets of the following elements: • notificationText
		notificationText	Var	Notification text; the encoding method is specified in dataCodingScheme. • Type: Array of bytes • Maximum array length: 64
		clientAddress_len	1	Number of sets of the following elements: • clientAddress

Field	Field value	Parameter	Size (byte)	Description
		clientAddress	Var	Client address; the encoding method is specified in dataCodingScheme. • Maximum array length: 20
		locationType	4	Location type. Valid values: • 0x00000001 – CURRENT_LOCATION • 0x00000002 – CURRENT_OR_LAST_KNOWN_LOCATION • 0x00000004 – INITIAL_LOCATION
		dataCodingScheme	4	Identifies the coding scheme of the coded string. Valid values: • 0x0000000C – NI_SS_GERMAN • 0x0000000D – NI_SS_ENGLISH • 0x0000000E – NI_SS_ITALIAN • 0x0000000F – NI_SS_FRENCH • 0x00000010 – NI_SS_SPANISH • 0x00000011 – NI_SS_DUTCH • 0x00000012 – NI_SS_SWEDISH • 0x00000013 – NI_SS_DANISH • 0x00000014 – NI_SS_PORTUGUESE • 0x00000015 – NI_SS_FINNISH • 0x00000016 – NI_SS_NORWEGIAN • 0x00000017 – NI_SS_GREEK • 0x00000018 – NI_SS_TURKISH • 0x00000019 – NI_SS_HUNGARIAN • 0x0000001A – NI_SS_POLISH • 0x0000001B – NI_SS_LANGUAGE_UNSPEC • 0x0000001C – NI_SUPL_UTF8 • 0x0000001D – NI_SUPL_UCS2 • 0x0000001E – NI_SUPL_GSM_DEFAULT
		codedString_len	1	Number of sets of the following elements: • codedString
		codedString	Var	Coded string. • Type: Array of bytes • Maximum string length: 20

Field	Field value	Parameter	Size (byte)	Description
		dataCodingScheme	4	Identifies the coding scheme of the coded string. Valid values: • 0x0000000C – NI_SS_GERMAN • 0x0000000D – NI_SS_ENGLISH • 0x0000000E – NI_SS_ITALIAN • 0x0000000F – NI_SS_FRENCH • 0x00000010 – NI_SS_SPANISH • 0x00000011 – NI_SS_DUTCH • 0x00000012 – NI_SS_SWEDISH • 0x00000013 – NI_SS_DANISH • 0x00000014 – NI_SS_PORTUGUESE • 0x00000015 – NI_SS_FINNISH • 0x00000016 – NI_SS_NORWEGIAN • 0x00000017 – NI_SS_GREEK • 0x00000018 – NI_SS_TURKISH • 0x00000019 – NI_SS_HUNGARIAN • 0x0000001A – NI_SS_POLISH • 0x0000001B – NI_SS_LANGUAGE_ UNSPEC • 0x0000001C – NI_SUPL_UTF8 • 0x0000001D – NI_SUPL_UCS2 • 0x0000001E – NI_SUPL_GSM_DEFAULT
		codedString_len	1	Number of sets of the following elements: • codedString
		codedString	Var	Coded string. • Type: Array of bytes • Maximum string length: 20
		lcsServiceTypeId	1	Service type ID. • Type: Unsigned integer
		userResponseTimer	2	Time to wait for the user to respond. • Type: Unsigned integer • Units: Seconds
Type	0x13		1	Network Initiated Service Interaction Request Optional NI service interaction payload.
Length	Var		2	
Value	→	posQosIncl	1	Whether quality of service is included: • 0x01 (TRUE) – QoS is included • 0x00 (FALSE) – QoS is not included
		posQos	1	Position QoS timeout. • Type: Unsigned integer • Units: Seconds • Range: 0 to 255
		numFixes	4	Number of fixes allowed. • Type: Unsigned integer
		timeBetweenFixes	4	Time between fixes. • Type: Unsigned integer • Units: Seconds

Field	Field value	Parameter	Size (byte)	Description
		posMode	4	Position mode. Valid values: • 0x00000001 – NI_VX_MS_ASSISTED_ONLY • 0x00000002 – NI_VX_MS_BASED_ONLY • 0x00000003 – NI_VX_MS_ASSISTED_PREFERRED_MS_BASED_ALLOWED • 0x00000004 – NI_VX_MS_BASED_PREFERRED_MS_ASSISTED_ALLOWED
		encodingScheme	4	VX encoding scheme. Valid values: • 0x00000000 – NI_VX_OCTET • 0x00000001 – NI_VX_EXN_PROTOCOL_MSG • 0x00000002 – NI_VX_ASCII • 0x00000003 – NI_VX_IA5 • 0x00000004 – NI_VX_UNICODE • 0x00000005 – NI_VX_SHIFT_JIS • 0x00000006 – NI_VX_KOREAN • 0x00000007 – NI_VX_LATIN_HEBREW • 0x00000008 – NI_VX_LATIN • 0x00000009 – NI_VX_GSM
		requestorId_len	1	Number of sets of the following elements: • requestorId
		requestorId	Var	Requestor ID. • Type: Array of bytes • Maximum array length: 200
		userRespTimerInSeconds	2	Time to wait for the user to respond. • Type: Unsigned integer • Units: Seconds
		serviceInteractionType	4	Service interaction type specified in qmiLocNiServiceInteractionEnumT. Valid values: • 0x00000001 – ONGOING_NI_INCOMING_MO

3.22.2 Indication - QMI_LOC_NI_USER_RESPONSE_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version last modified
NI User Response Status	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x01		1	NI User Response Status
Length	4		2	
Value	→	status	4	Status of the NI User Response request. Valid values: • 0x00000000 – SUCCESS • 0x00000001 – GENERAL_FAILURE • 0x00000002 – UNSUPPORTED • 0x00000003 – INVALID_PARAMETER • 0x00000004 – ENGINE_BUSY • 0x00000005 – PHONE_OFFLINE • 0x00000006 – LOC_TIMEOUT

Optional TLVs

None

Error codes

QMI_ERR_NONE	No error in the request
QMI_ERR_INTERNAL	Unexpected error occurred during processing
QMI_ERR_MALFORMED_MSG	Message was not formulated correctly by the control point or the message was corrupted during transmission
QMI_ERR_NO_MEMORY	Device could not allocate memory to formulate a response
QMI_ERR_INVALID_HANDLE	Invalid client handle was received

3.22.3 Description of QMI_LOC_INFORM_NI_USER_RESPONSE

This command sends the user response to the engine corresponding to a prior NI request. The indication contains the status of the User Response request.

3.23 QMI_LOC_INJECT_PREDICTED_ORBITS_DATA

Injects predicted orbits data.

LOC message ID

0x0035

Version introduced

Major - 2, Minor - 2

3.23.1 Request - QMI_LOC_INJECT_PREDICTED_ORBITS_DATA_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

Name	Version last modified
Total Size	2.2
Total Parts	2.2
Part Number	2.2
Data	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x01		1	Total Size
Length	4		2	
Value	→	totalSize	4	Total size of the predicted orbits data to be injected. • Type: Unsigned integer • Units: Bytes
Type	0x02		1	Total Parts
Length	2		2	
Value	→	totalParts	2	Total number of parts into which the predicted orbits data is divided. • Type: Unsigned integer

Field	Field value	Parameter	Size (byte)	Description
Type	0x03		1	Part Number
Length	2		2	
Value	→	partNum	2	Number of the current predicted orbits data part; starts at 1. • Type: Unsigned integer
Type	0x04		1	Data
Length	Var		2	
Value	→	partData_len	2	Number of sets of the following elements: • partData
		partData	Var	Predicted orbits data. • Type: Array of bytes • Maximum length of the array: 1024

Optional TLVs

Name	Version last modified
Format Type	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x10		1	Format Type
Length	4		2	
Value	→	formatType	4	Predicted orbits data format. Valid values: • 0x00000000 – PREDICTED_ORBITS_XTRA

3.23.2 Indication - QMI_LOC_INJECT_PREDICTED_ORBITS_DATA_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version last modified
Data Injection Status	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x01		1	Data Injection Status
Length	4		2	
Value	→	status	4	Status of the Data Injection request. Valid values: • 0x00000000 – SUCCESS • 0x00000001 – GENERAL_FAILURE • 0x00000002 – UNSUPPORTED • 0x00000003 – INVALID_PARAMETER • 0x00000004 – ENGINE_BUSY • 0x00000005 – PHONE_OFFLINE • 0x00000006 – LOC_TIMEOUT

Optional TLVs

At least one of the following optional TLVs is present if the status is eQMI_LOC_SUCCESS.

Name	Version last modified
Part Number	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x10		1	Part Number
Length	2		2	
Value	→	partNum	2	Number of the predicted orbits data part for which this indication is sent; starts at 1. • Type: Unsigned integer

Error codes

QMI_ERR_NONE	No error in the request
QMI_ERR_INTERNAL	Unexpected error occurred during processing
QMI_ERR_MALFORMED_MSG	Message was not formulated correctly by the control point or the message was corrupted during transmission
QMI_ERR_NO_MEMORY	Device could not allocate memory to formulate a response
QMI_ERR_INVALID_HANDLE	Invalid client handle was received

3.23.3 Description of QMI_LOC_INJECT_PREDICTED_ORBITS_DATA

This command is called to inject predicted orbits data parts. Each data part is acknowledged through the general response. The indication QMI_LOC_INJECT_PREDICTED_ORBITS_DATA_IND is sent after each part to denote whether the injection of that data part succeeded.

3.24 QMI_LOC_GET_PREDICTED_ORBITS_DATA_SOURCE

Gets the predicted orbits data source.

LOC message ID

0x0036

Version introduced

Major - 2, Minor - 2

3.24.1 Request - QMI_LOC_GET_PREDICTED_ORBITS_DATA_SOURCE_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

None

Optional TLVs

None

3.24.2 Indication - QMI_LOC_GET_PREDICTED_ORBITS_DATA_SOURCE_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version last modified
Predicted Oribits Data Source Status	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x01		1	Predicted Oribits Data Source Status
Length	4		2	
Value	→	status	4	Status of the query request for a predicted orbits data source. Valid values: • 0x00000000 – SUCCESS • 0x00000001 – GENERAL_FAILURE • 0x00000002 – UNSUPPORTED • 0x00000003 – INVALID_PARAMETER • 0x00000004 – ENGINE_BUSY • 0x00000005 – PHONE_OFFLINE • 0x00000006 – LOC_TIMEOUT

Optional TLVs

At least one of the following optional TLVs is present if the status is eQMI_LOC_SUCCESS.

Name	Version last modified
Allowed Sizes	2.2
Server List	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x10		1	Allowed Sizes Maximum part and file size allowed to be injected in the engine.
Length	8		2	
Value	→	maxFileSizeInBytes	4	Maximum allowable predicted orbits file size (in bytes). • Type: Unsigned integer
		maxPartSize	4	Maximum allowable predicted orbits file chunk size (in bytes). • Type: Unsigned integer
Type	0x11		1	Server List List of servers that can be used by the client to download predicted orbits data.
Length	Var		2	

Field	Field value	Parameter	Size (byte)	Description
Value	→	serverList_len	1	Number of sets of the following elements: • serverUrl_len • serverUrl
		serverUrl_len	1	Number of sets of the following elements: • serverUrl
		serverUrl	Var	Assistance server URL. • Type: NULL-terminated string • Maximum string length (including NULL terminator): 256

Error codes

QMI_ERR_NONE	No error in the request
QMI_ERR_INTERNAL	Unexpected error occurred during processing
QMI_ERR_MALFORMED_MSG	Message was not formulated correctly by the control point or the message was corrupted during transmission
QMI_ERR_NO_MEMORY	Device could not allocate memory to formulate a response
QMI_ERR_INVALID_HANDLE	Invalid client handle was received

3.24.3 Description of QMI_LOC_GET_PREDICTED_ORBITS_DATA_SOURCE

This command is used to get the URL of the server(s) that host the predicted orbits data, its file size, etc. The request is acknowledged through the general response, but the SUCCESS/FAILURE status is sent through the indication QMI_LOC_QUERY_PREDICTED_ORBITS_DATA_SOURCE_IND. If the query is successful, the indication will also contain the source information.

3.25 QMI_LOC_GET_PREDICTED_ORBITS_DATA_VALIDITY

Gets the predicted orbits data validity.

LOC message ID

0x0037

Version introduced

Major - 2, Minor - 2

3.25.1 Request - QMI_LOC_GET_PREDICTED_ORBITS_DATA_VALIDITY_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

None

Optional TLVs

None

3.25.2 Indication - QMI_LOC_GET_PREDICTED_ORBITS_DATA_VALIDITY_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version last modified
Predicted Orbits Data Validity Status	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x01		1	Predicted Orbits Data Validity Status
Length	4		2	
Value	→	status	4	Status of the query request for predicted orbits data validity. Valid values: • 0x00000000 – SUCCESS • 0x00000001 – GENERAL_FAILURE • 0x00000002 – UNSUPPORTED • 0x00000003 – INVALID_PARAMETER • 0x00000004 – ENGINE_BUSY • 0x00000005 – PHONE_OFFLINE • 0x00000006 – LOC_TIMEOUT

Optional TLVs

At least one of the following optional TLVs is present if the status is eQMI_LOC_SUCCESS.

Name	Version last modified
Validity Info	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x10		1	Validity Info
Length	10		2	
Value	→	startTimeInUTC	8	Predicted orbits data is valid starting from this time. • Type: Unsigned integer • Units: Seconds (since Jan. 1, 1970)
		durationHours	2	Duration from the start time for which the data is valid. • Type: Unsigned integer • Units: Hours

Error codes

QMI_ERR_NONE	No error in the request
QMI_ERR_INTERNAL	Unexpected error occurred during processing
QMI_ERR_MALFORMED_MSG	Message was not formulated correctly by the control point or the message was corrupted during transmission
QMI_ERR_NO_MEMORY	Device could not allocate memory to formulate a response
QMI_ERR_INVALID_HANDLE	Invalid client handle was received

3.25.3 Description of QMI_LOC_GET_PREDICTED_ORBITS_DATA_VALIDITY

This command is used to query the predicted orbits data validity. The request is acknowledged through the response, but the SUCCESS/FAILURE status is sent through the indication QMI_LOC_QUERY_PREDICTED_ORBITS_DATA_VALIDITY_IND. If the query is successful, the indication will contain the data validity information.

3.26 QMI_LOC_INJECT_UTC_TIME

Injects UTC time in the location engine.

LOC message ID

0x0038

Version introduced

Major - 2, Minor - 2

3.26.1 Request - QMI_LOC_INJECT_UTC_TIME_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

Name	Version last modified
UTC Time	2.2
Time Uncertainty	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x01		1	UTC Time
Length	8		2	
Value	→	timeUtc	8	UTC time since Jan. 1, 1970. • Type: Unsigned integer • Units: Milliseconds
Type	0x02		1	Time Uncertainty
Length	4		2	
Value	→	timeUnc	4	Time uncertainty. • Type: Unsigned integer • Units: Milliseconds

Optional TLVs

None

3.26.2 Indication - QMI_LOC_INJECT_UTC_TIME_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version last modified
UTC Time Injection Status	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x01		1	UTC Time Injection Status
Length	4		2	
Value	→	status	4	Status of the UTC Time Injection request. Valid values: • 0x00000000 – SUCCESS • 0x00000001 – GENERAL_FAILURE • 0x00000002 – UNSUPPORTED • 0x00000003 – INVALID_PARAMETER • 0x00000004 – ENGINE_BUSY • 0x00000005 – PHONE_OFFLINE • 0x00000006 – LOC_TIMEOUT

Optional TLVs

None

Error codes

QMI_ERR_NONE	No error in the request
QMI_ERR_INTERNAL	Unexpected error occurred during processing
QMI_ERR_MALFORMED_MSG	Message was not formulated correctly by the control point or the message was corrupted during transmission
QMI_ERR_NO_MEMORY	Device could not allocate memory to formulate a response
QMI_ERR_INVALID_HANDLE	Invalid client handle was received

3.26.3 Description of QMI_LOC_INJECT_UTC_TIME

This command is used to inject UTC time into the location engine. The request is acknowledged through the response, but the SUCCESS/FAILURE status is sent through the indication QMI_LOC_INJECT_UTC_TIME_IND.

3.27 QMI_LOC_INJECT_POSITION

Injects a position to the location engine.

LOC message ID

0x0039

Version introduced

Major - 2, Minor - 2

3.27.1 Request - QMI_LOC_INJECT_POSITION_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

None

Optional TLVs

Name	Version last modified
Latitude	2.2
Longitude	2.2
Circular Horizontal Uncertainty	2.2
Horizontal Confidence	2.2
Horizontal Reliability	2.2
Altitude With Respect to Ellipsoid	2.2
Altitude With Respect to Sea Level	2.2
Vertical Uncertainty	2.2
Vertical Confidence	2.2
Vertical Reliability	2.2
Altitude Source Info	2.2
UTC Timestamp	2.2
Position Age	2.2
Position Source	2.4

Field	Field value	Parameter	Size (byte)	Description
Type	0x10		1	Latitude
Length	8		2	
Value	→	latitude	8	Latitude (specified in WGS84 datum). • Type: Floating point • Units: Degrees • Range: -90.0 to 90.0 • Positive values indicate northern latitude • Negative values indicate southern latitude
Type	0x11		1	Longitude
Length	8		2	
Value	→	longitude	8	Longitude (specified in WGS84 datum). • Type: Floating point • Units: Degrees • Range: -180.0 to 180.0 • Positive values indicate eastern longitude • Negative values indicate western longitude
Type	0x12		1	Circular Horizontal Uncertainty
Length	4		2	
Value	→	horUncCircular	4	Horizontal position uncertainty (circular). • Type: Floating point • Units: Meters
Type	0x13		1	Horizontal Confidence
Length	1		2	
Value	→	horConfidence	1	Horizontal confidence, as defined by ETSI TS 101 109 (see [S4]). • Type: Unsigned integer • Units: Percent (0 to 99) • 0 – invalid value • 100 to 256 – not used • If 100 is received, reinterpret to 99 This field must be specified together with horizontal uncertainty. If not specified, the default value will be 50.
Type	0x14		1	Horizontal Reliability
Length	4		2	

Field	Field value	Parameter	Size (byte)	Description
Value	→	horReliability	4	Specifies the reliability of the horizontal position. Valid values: • 0x00000000 – eQMI_LOC_RELIABILITY_NOT_SET • 0x00000001 – eQMI_LOC_RELIABILITY_VERY_LOW • 0x00000002 – eQMI_LOC_RELIABILITY_LOW • 0x00000003 – eQMI_LOC_RELIABILITY_MEDIUM • 0x00000004 – eQMI_LOC_RELIABILITY_HIGH
Type	0x15		1	Altitude With Respect to Ellipsoid
Length	4		2	
Value	→	altitudeWrtEllipsoid	4	Altitude with respect to the WGS84 ellipsoid. • Type: Floating point • Units: Meters; positive = height, negative = depth
Type	0x16		1	Altitude With Respect to Sea Level
Length	4		2	
Value	→	altitudeWrtMeanSeaLevel	4	Altitude with respect to mean sea level. • Type: Floating point • Units: Meters
Type	0x17		1	Vertical Uncertainty
Length	4		2	
Value	→	vertUnc	4	Vertical uncertainty. This is mandatory if either altitudeWrtEllipsoid or altitudeWrtMeanSeaLevel is specified. • Type: Floating point • Units: Meters
Type	0x18		1	Vertical Confidence
Length	1		2	
Value	→	vertConfidence	1	Vertical confidence, as defined by ETSI TS 101 109 (see [S4]). • Type: Unsigned integer • Units: Percent (0-99) • 0 – invalid value • 100 to 256 – not used • If 100 is received, reinterpret to 99 This field must be specified together with the vertical uncertainty. If not specified, the default value will be 50.
Type	0x19		1	Vertical Reliability
Length	4		2	

Field	Field value	Parameter	Size (byte)	Description
Value	→	vertReliability	4	Specifies the reliability of the vertical position. Valid values: • 0x00000000 – eQMI_LOC_RELIABILITY_NOT_SET • 0x00000001 – eQMI_LOC_RELIABILITY_VERY_LOW • 0x00000002 – eQMI_LOC_RELIABILITY_LOW • 0x00000003 – eQMI_LOC_RELIABILITY_MEDIUM • 0x00000004 – eQMI_LOC_RELIABILITY_HIGH
Type	0x1A		1	Altitude Source Info Specifies information regarding the altitude source.
Length	12		2	
Value	→	source	4	Specifies the source of the altitude. Valid values: • 0x00000000 – ALT_SRC_UNKNOWN • 0x00000001 – ALT_SRC_GPS • 0x00000002 – ALT_SRC_CELL_ID • 0x00000003 – ALT_SRC_ENHANCED_CELL_ID • 0x00000004 – ALT_SRC_WIFI • 0x00000005 – ALT_SRC_TERRESTRIAL • 0x00000006 – ALT_SRC_TERRESTRIAL_HYBRID • 0x00000007 – ALT_SRC_ALTITUDE_DATABASE • 0x00000008 – ALT_SRC_BAROMETRIC_ALTIMETER • 0x00000009 – ALT_SRC_OTHER
		linkage	4	Specifies the dependency between the horizontal and altitude position components. Valid values: • 0x00000000 – SRC_LINKAGE_NOT_SPECIFIED • 0x00000001 – SRC_LINKAGE_FULLY_INTERDEPENDENT • 0x00000002 – SRC_LINKAGE_DEPENDS_ON_LAT_LONG • 0x00000003 – SRC_LINKAGE_FULLY_INDEPENDENT

Field	Field value	Parameter	Size (byte)	Description
		coverage	4	Specifies the region of uncertainty. Valid values: • 0x00000000 – UNCERTAINTY_NOT_ SPECIFIED • 0x00000001 – UNCERTAINTY_POINT: Altitude uncertainty is valid at the injected horizontal position coordinates only. • 0x00000002 – UNCERTAINTY_FULL: Altitude uncertainty applies to the position of the device regardless of horizontal position (within the horizontal uncertainty region, if provided).
Type	0x1B		1	UTC Timestamp
Length	8		2	
Value	→	timestampUtc	8	UTC timestamp. • Type: Unsigned integer • Units: Milliseconds (since Jan. 1, 1970)
Type	0x1C		1	Position Age
Length	4		2	
Value	→	timestampAge	4	Position age, which is an estimate of how long ago this fix was made. • Type: Signed integer • Units: Milliseconds
Type	0x1D		1	Position Source
Length	4		2	

Field	Field value	Parameter	Size (byte)	Description
Value	→	positionSrc	4	Source from which this position was obtained. Valid values: • 0x00000000 – eQMI_LOC_POSITION_SRC_GNSS • 0x00000001 – eQMI_LOC_POSITION_SRC_CELLID • 0x00000002 – eQMI_LOC_POSITION_SRC_ENH_CELLID • 0x00000003 – eQMI_LOC_POSITION_SRC_WIFI • 0x00000004 – eQMI_LOC_POSITION_SRC_TERRESTRIAL • 0x00000005 – eQMI_LOC_POSITION_SRC_GNSS_TERRESTRIAL_HYBRID • 0x00000006 – eQMI_LOC_POSITION_SRC_OTHER If altitude is specified and the altitude source is not specified, the engine assumes that the altitude was obtained using the specified position source. If both altitude and altitude source are specified, the engine assumes that only latitude and longitude were obtained using the specified position source.

3.27.2 Indication - QMI_LOC_INJECT_POSITION_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version last modified
UTC Position Injection Status	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x01		1	UTC Position Injection Status

Field	Field value	Parameter	Size (byte)	Description
Length	4		2	
Value	→	status	4	Status of the UTC Position Injection request. Valid values: • 0x00000000 – SUCCESS • 0x00000001 – GENERAL_FAILURE • 0x00000002 – UNSUPPORTED • 0x00000003 – INVALID_PARAMETER • 0x00000004 – ENGINE_BUSY • 0x00000005 – PHONE_OFFLINE • 0x00000006 – LOC_TIMEOUT

Optional TLVs

None

Error codes

QMI_ERR_NONE	No error in the request
QMI_ERR_INTERNAL	Unexpected error occurred during processing
QMI_ERR_MALFORMED_MSG	Message was not formulated correctly by the control point or the message was corrupted during transmission
QMI_ERR_NO_MEMORY	Device could not allocate memory to formulate a response
QMI_ERR_INVALID_HANDLE	Invalid client handle was received

3.27.3 Description of QMI_LOC_INJECT_POSITION

This command is used to inject a position to the location engine. The engine can use this information to better estimate the TTFF. The request is acknowledged through the response, but the SUCCESS/FAILURE status is sent through the indication QMI_LOC_INJECT_POSITION_IND.

3.28 QMI_LOC_SET_ENGINE_LOCK

Sets the location engine lock.

LOC message ID

0x003A

Version introduced

Major - 2, Minor - 2

3.28.1 Request - QMI_LOC_SET_ENGINE_LOCK_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

Name	Version last modified
Lock Type	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x01		1	Lock Type
Length	4		2	
Value	→	lockType	4	Type of lock. Valid values: • 0x00000001 – LOCK_NONE • 0x00000002 – LOCK_MI • 0x00000003 – LOCK_MT • 0x00000004 – LOCK_ALL

Optional TLVs

None

3.28.2 Indication - QMI_LOC_SET_ENGINE_LOCK_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version last modified
Set Engine Lock Status	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x01		1	Set Engine Lock Status
Length	4		2	
Value	→	status	4	Status of the Set Engine Lock request. Valid values: • 0x00000000 – SUCCESS • 0x00000001 – GENERAL_FAILURE • 0x00000002 – UNSUPPORTED • 0x00000003 – INVALID_PARAMETER • 0x00000004 – ENGINE_BUSY • 0x00000005 – PHONE_OFFLINE • 0x00000006 – LOC_TIMEOUT

Optional TLVs

None

Error codes

QMI_ERR_NONE	No error in the request
QMI_ERR_INTERNAL	Unexpected error occurred during processing
QMI_ERR_MALFORMED_MSG	Message was not formulated correctly by the control point or the message was corrupted during transmission
QMI_ERR_NO_MEMORY	Device could not allocate memory to formulate a response
QMI_ERR_INVALID_HANDLE	Invalid client handle was received

3.28.3 Description of QMI_LOC_SET_ENGINE_LOCK

This command is used to lock the location engine. The request is acknowledged through the response, but the SUCCESS/FAILURE status is sent through the indication QMI_LOC_SET_ENGINE_LOCK_IND.

3.29 QMI_LOC_GET_ENGINE_LOCK

Gets the location engine lock.

LOC message ID

0x003B

Version introduced

Major - 2, Minor - 2

3.29.1 Request - QMI_LOC_GET_ENGINE_LOCK_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

None

Optional TLVs

None

3.29.2 Indication - QMI_LOC_GET_ENGINE_LOCK_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version last modified
Get Engine Lock Status	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x01		1	Get Engine Lock Status
Length	4		2	
Value	→	status	4	Status of the Get Engine Lock request. Valid values: • 0x00000000 – SUCCESS • 0x00000001 – GENERAL_FAILURE • 0x00000002 – UNSUPPORTED • 0x00000003 – INVALID_PARAMETER • 0x00000004 – ENGINE_BUSY • 0x00000005 – PHONE_OFFLINE • 0x00000006 – LOC_TIMEOUT

Optional TLVs

At least one of the following optional TLVs is present if the status is eQMI_LOC_SUCCESS.

Name	Version last modified
Lock Type	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x10		1	Lock Type
Length	4		2	
Value	→	lockType	4	Type of lock. Valid values: • 0x00000001 – LOCK_NONE • 0x00000002 – LOCK_MI • 0x00000003 – LOCK_MT • 0x00000004 – LOCK_ALL

Error codes

QMI_ERR_NONE	No error in the request
QMI_ERR_INTERNAL	Unexpected error occurred during processing
QMI_ERR_MALFORMED_MSG	Message was not formulated correctly by the control point or the message was corrupted during transmission
QMI_ERR_NO_MEMORY	Device could not allocate memory to formulate a response
QMI_ERR_INVALID_HANDLE	Invalid client handle was received

3.29.3 Description of QMI_LOC_GET_ENGINE_LOCK

This command is used to get location engine lock information. The request is acknowledged through the response, but the SUCCESS/FAILURE status is sent through the indication QMI_LOC_GET_ENGINE_LOCK_IND. If successful, the indication also contains the current engine lock type.

3.30 QMI_LOC_SET_SBAS_CONFIG

Sets the SBAS configuration.

LOC message ID

0x003C

Version introduced

Major - 2, Minor - 2

3.30.1 Request - QMI_LOC_SET_SBAS_CONFIG_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

Name	Version last modified
SBAS Config	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x01		1	SBAS Config
Length	1		2	
Value	→	sbasConfig	1	Whether SBAS configuration is enabled. • 0x01 (TRUE) – SBAS configuration is enabled • 0x00 (FALSE) – SBAS configuration is disabled

Optional TLVs

None

3.30.2 Indication - QMI_LOC_SET_SBAS_CONFIG_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version last modified
Set SBAS Config Status	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x01		1	Set SBAS Config Status
Length	4		2	
Value	→	status	4	Status of the Set SBAS Configuration request. Valid values: • 0x00000000 – SUCCESS • 0x00000001 – GENERAL_FAILURE • 0x00000002 – UNSUPPORTED • 0x00000003 – INVALID_PARAMETER • 0x00000004 – ENGINE_BUSY • 0x00000005 – PHONE_OFFLINE • 0x00000006 – LOC_TIMEOUT

Optional TLVs

None

Error codes

QMI_ERR_NONE	No error in the request
QMI_ERR_INTERNAL	Unexpected error occurred during processing
QMI_ERR_MALFORMED_MSG	Message was not formulated correctly by the control point or the message was corrupted during transmission
QMI_ERR_NO_MEMORY	Device could not allocate memory to formulate a response
QMI_ERR_INVALID_HANDLE	Invalid client handle was received

3.30.3 Description of QMI_LOC_SET_SBAS_CONFIG

This command is used to set the SBAS configuration. The request is acknowledged through the response, but the SUCCESS/FAILURE status is sent through the indication QMI_LOC_SET_SBAS_CONFIG_IND.

3.31 QMI_LOC_GET_SBAS_CONFIG

Gets the SBAS configuration from the location engine.

LOC message ID

0x003D

Version introduced

Major - 2, Minor - 2

3.31.1 Request - QMI_LOC_GET_SBAS_CONFIG_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

None

Optional TLVs

None

3.31.2 Indication - QMI_LOC_GET_SBAS_CONFIG_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version last modified
Get SBAS Config Status	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x01		1	Get SBAS Config Status
Length	4		2	
Value	→	status	4	Status of the Get SBAS Configuration request. Valid values: • 0x00000000 – SUCCESS • 0x00000001 – GENERAL_FAILURE • 0x00000002 – UNSUPPORTED • 0x00000003 – INVALID_PARAMETER • 0x00000004 – ENGINE_BUSY • 0x00000005 – PHONE_OFFLINE • 0x00000006 – LOC_TIMEOUT

Optional TLVs

At least one of the following optional TLVs is present if the status is eQMI_LOC_SUCCESS.

Name	Version last modified
SBAS Config	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x10		1	SBAS Config
Length	1		2	
Value	→	sbasConfig	1	Whether SBAS configuration is enabled. • 0x01 (TRUE) – SBAS configuration is enabled • 0x00 (FALSE) – SBAS configuration is disabled

Error codes

QMI_ERR_NONE	No error in the request
QMI_ERR_INTERNAL	Unexpected error occurred during processing
QMI_ERR_MALFORMED_MSG	Message was not formulated correctly by the control point or the message was corrupted during transmission
QMI_ERR_NO_MEMORY	Device could not allocate memory to formulate a response
QMI_ERR_INVALID_HANDLE	Invalid client handle was received

3.31.3 Description of QMI_LOC_GET_SBAS_CONFIG

This command is used to get the SBAS configuration. The request is acknowledged through the response, but the SUCCESS/FAILURE status is sent through the indication QMI_LOC_GET_SBAS_CONFIG_IND. If successful, the indication also contains the SBAS configuration.

3.32 QMI_LOC_SET_NMEA_TYPES

Sets the NMEA types.

LOC message ID

0x003E

Version introduced

Major - 2, Minor - 2

3.32.1 Request - QMI_LOC_SET_NMEA_TYPES_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

Name	Version last modified
NMEA Sentence Types	2.5

Field	Field value	Parameter	Size (byte)	Description
Type	0x01		1	NMEA Sentence Types
Length	4		2	
Value	→	nmeaSentenceType	4	Bitmasks of NMEA types to enable. Valid bitmasks: • 0x00000001 – NMEA_MASK_GGA • 0x00000002 – NMEA_MASK_RMC • 0x00000004 – NMEA_MASK_GSV • 0x00000008 – NMEA_MASK_GSA • 0x00000010 – NMEA_MASK_VTG • 0x00000020 – NMEA_MASK_PQXFI • 0x00000040 – NMEA_MASK_PSTIS

Optional TLVs

None

3.32.2 Indication - QMI_LOC_SET_NMEA_TYPES_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version last modified
Set NMEA Types Status	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x01		1	Set NMEA Types Status
Length	4		2	
Value	→	status	4	Status of Set NMEA Types request. Valid values: • 0x00000000 – SUCCESS • 0x00000001 – GENERAL_FAILURE • 0x00000002 – UNSUPPORTED • 0x00000003 – INVALID_PARAMETER • 0x00000004 – ENGINE_BUSY • 0x00000005 – PHONE_OFFLINE • 0x00000006 – LOC_TIMEOUT

Optional TLVs

None

Error codes

QMI_ERR_NONE	No error in the request
QMI_ERR_INTERNAL	Unexpected error occurred during processing
QMI_ERR_MALFORMED_MSG	Message was not formulated correctly by the control point or the message was corrupted during transmission
QMI_ERR_NO_MEMORY	Device could not allocate memory to formulate a response
QMI_ERR_INVALID_HANDLE	Invalid client handle was received

3.32.3 Description of QMI_LOC_SET_NMEA_TYPES

This command is used to set the NMEA types. The request is acknowledged through the response, but the SUCCESS/FAILURE status is sent through the indication QMI_LOC_SET_NMEA_TYPES_IND.

Proprietary Qualcomm eXtended Fix Information (PQXFI) is a proprietary NMEA sentence containing position and error information. It is tied to the GPGGA sentence configuration.

```
    1      2      3  4       5         6  7   8   9   10  11
    |      |      |  |       |         |  |   |   |   |   |

    $PQXFI,hhmmss.ss,llll.ll,a,yyyyy.yy,a,x.x,x.x,x.x,x.x,*hh<CR><LF>

    Field number:

    1.  Proprietary Qualcomm eXtended Fix information
    2.  UTC of the position
    3.  Latitude (DDmm.mm); DD = two digits of degrees, mm.mm = minutes
    4.  N or S (North or South)
    5.  Longitude (DDDmm.mm)
    6.  E or W (East or West)
    7.  Geoidal separation -- the difference between the WGS-84 earth ellipsoid
        and the mean-sea-level (geoid); a negative value means a below mean-sea-level
        ellipsoid
    8.  HEPE
    9.  Vertical error of fix
    10. Velocity error
    11. Checksum

    Example

:    $PQXFI,212122.7,3622.482568,N,12551.046925,W,20.8,2.94,3.57,0.03,*42
```

Proprietary SnapTrack, Inc. Session (PSTIS) control message is a proprietary NMEA sentence transmitting the GPS session end indications.

```
    12      3
    ||      |

    $PSTIS,*hh<CR><LF>

    Field Number
:    1. Talker ID
    2. Proprietary SnapTrack Inc. session control message
    3. Checksum

    Example

:    $PSTIS,*61
```

3.33 QMI_LOC_GET_NMEA_TYPES

Gets the NMEA types from the location engine.

LOC message ID

0x003F

Version introduced

Major - 2, Minor - 2

3.33.1 Request - QMI_LOC_GET_NMEA_TYPES_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

None

Optional TLVs

None

3.33.2 Indication - QMI_LOC_GET_NMEA_TYPES_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version last modified
Get NMEA Types Status	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x01		1	Get NMEA Types Status
Length	4		2	
Value	→	status	4	Status of the Get NMEA Types request. Valid values: • 0x00000000 – SUCCESS • 0x00000001 – GENERAL_FAILURE • 0x00000002 – UNSUPPORTED • 0x00000003 – INVALID_PARAMETER • 0x00000004 – ENGINE_BUSY • 0x00000005 – PHONE_OFFLINE • 0x00000006 – LOC_TIMEOUT

Optional TLVs

At least one of the following optional TLVs is present if the status is eQMI_LOC_SUCCESS.

Name	Version last modified
NMEA Sentence Types	2.5

Field	Field value	Parameter	Size (byte)	Description
Type	0x10		1	NMEA Sentence Types
Length	4		2	
Value	→	nmeaSentenceType	4	NMEA types to enable. Valid bitmasks: • 0x0000ffff – NMEA_MASK_ALL • 0x00000001 – NMEA_MASK_GGA • 0x00000002 – NMEA_MASK_RMC • 0x00000004 – NMEA_MASK_GSV • 0x00000008 – NMEA_MASK_GSA • 0x00000010 – NMEA_MASK_VTG • 0x00000020 – NMEA_MASK_PQXFI • 0x00000040 – NMEA_MASK_PSTIS

Error codes

QMI_ERR_NONE	No error in the request
QMI_ERR_INTERNAL	Unexpected error occurred during processing
QMI_ERR_MALFORMED_MSG	Message was not formulated correctly by the control point or the message was corrupted during transmission
QMI_ERR_NO_MEMORY	Device could not allocate memory to formulate a response
QMI_ERR_INVALID_HANDLE	Invalid client handle was received

3.33.3 Description of QMI_LOC_GET_NMEA_TYPES

This command is used to get the NMEA types. The request is acknowledged through the response, but the SUCCESS/FAILURE status is sent through the indication QMI_LOC_GET_NMEA_TYPES_IND. If the call is successful, the NMEA types will be available in the indication.

3.34 QMI_LOC_SET_LOW_POWER_MODE

Enables/disables Low Power Mode (LPM) configuration.

LOC message ID

0x0040

Version introduced

Major - 2, Minor - 2

3.34.1 Request - QMI_LOC_SET_LOW_POWER_MODE_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

Name	Version last modified
Enable Low Power Mode	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x01		1	Enable Low Power Mode
Length	1		2	
Value	→	lowPowerMode	1	Whether to enable Low Power mode: • 0x01 (TRUE) – Enable LPM • 0x00 (FALSE) – Disable LPM

Optional TLVs

None

3.34.2 Indication - QMI_LOC_SET_LOW_POWER_MODE_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version last modified
Set LPM Status	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x01		1	Set LPM Status
Length	4		2	
Value	→	status	4	Status of the Set Low Power Mode request. Valid values: • 0x00000000 – SUCCESS • 0x00000001 – GENERAL_FAILURE • 0x00000002 – UNSUPPORTED • 0x00000003 – INVALID_PARAMETER • 0x00000004 – ENGINE_BUSY • 0x00000005 – PHONE_OFFLINE • 0x00000006 – LOC_TIMEOUT

Optional TLVs

None

Error codes

QMI_ERR_NONE	No error in the request
QMI_ERR_INTERNAL	Unexpected error occurred during processing
QMI_ERR_MALFORMED_MSG	Message was not formulated correctly by the control point or the message was corrupted during transmission
QMI_ERR_NO_MEMORY	Device could not allocate memory to formulate a response
QMI_ERR_INVALID_HANDLE	Invalid client handle was received

3.34.3 Description of QMI_LOC_SET_LOW_POWER_MODE

This command is used to enable/disable the lower power mode. The request is acknowledged through the response, but the SUCCESS/FAILURE status is sent through the indication QMI_LOC_SET_LOW_POWER_MODE_IND.

3.35 QMI_LOC_GET_LOW_POWER_MODE

Gets the LPM status from the location engine.

LOC message ID

0x0041

Version introduced

Major - 2, Minor - 2

3.35.1 Request - QMI_LOC_GET_LOW_POWER_MODE_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

None

Optional TLVs

None

3.35.2 Indication - QMI_LOC_GET_LOW_POWER_MODE_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version last modified
Get LPM Status	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x01		1	Get LPM Status
Length	4		2	
Value	→	status	4	Status of the Get LPM request. Valid values: • 0x00000000 – SUCCESS • 0x00000001 – GENERAL_FAILURE • 0x00000002 – UNSUPPORTED • 0x00000003 – INVALID_PARAMETER • 0x00000004 – ENGINE_BUSY • 0x00000005 – PHONE_OFFLINE • 0x00000006 – LOC_TIMEOUT

Optional TLVs

At least one of the following optional TLVs is present if the status is eQMI_LOC_SUCCESS.

Name	Version last modified
Enable/Disable LPM	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x10		1	Enable/Disable LPM
Length	1		2	
Value	→	lowPowerMode	1	Whether to enable Low Power mode: • 0x01 (TRUE) – Enable LPM • 0x00 (FALSE) – Disable LPM

Error codes

QMI_ERR_NONE	No error in the request
QMI_ERR_INTERNAL	Unexpected error occurred during processing
QMI_ERR_MALFORMED_MSG	Message was not formulated correctly by the control point or the message was corrupted during transmission
QMI_ERR_NO_MEMORY	Device could not allocate memory to formulate a response
QMI_ERR_INVALID_HANDLE	Invalid client handle was received

3.35.3 Description of QMI_LOC_GET_LOW_POWER_MODE

This command is used to get the low power mode. The request is acknowledged through the response, but the SUCCESS/FAILURE status is sent through the indication QMI_LOC_GET_LOW_POWER_MODE_IND. If successful, the indication also contains the current low power mode state (enabled/disabled).

3.36 QMI_LOC_SET_SERVER

Specifies the A-GPS server type and address.

LOC message ID

0x0042

Version introduced

Major - 2, Minor - 2

3.36.1 Request - QMI_LOC_SET_SERVER_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

Name	Version last modified
Server Type	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x01		1	Server Type
Length	4		2	
Value	→	serverType	4	Type of server. Valid values: • 0x00000001 – CDMA_PDE • 0x00000002 – CDMA_MPC • 0x00000003 – UMTS_SLP • 0x00000004 – CUSTOM_PDE

Optional TLVs

Name	Version last modified
IPV4 Address	2.2
IPV6 Address	2.2
Uniform Resource Locator	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x10		1	IPV4 Address IPV4 address and port.
Length	6		2	
Value	→	addr	4	IPV4 address. • Type: Unsigned integer
		port	2	IPV4 port. • Type: Unsigned integer
Type	0x11		1	IPV6 Address IPV6 address and port.
Length	20		2	
Value	→	addr	16	IPV6 address. • Type: Array of unsigned integers • Maximum length of the array: 8
		port	4	IPV6 port. • Type: Unsigned integer
Type	0x12		1	Uniform Resource Locator
Length	Var		2	
Value	→	urlAddr	Var	URL address. • Type: NULL-terminated string • Maximum string length (including NULL terminator): 256

3.36.2 Indication - QMI_LOC_SET_SERVER_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version last modified
Set Server Status	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x01		1	Set Server Status
Length	4		2	
Value	→	status	4	Status of the Set Server request. Valid values: • 0x00000000 – SUCCESS • 0x00000001 – GENERAL_FAILURE • 0x00000002 – UNSUPPORTED • 0x00000003 – INVALID_PARAMETER • 0x00000004 – ENGINE_BUSY • 0x00000005 – PHONE_OFFLINE • 0x00000006 – LOC_TIMEOUT

Optional TLVs

None

Error codes

QMI_ERR_NONE	No error in the request
QMI_ERR_INTERNAL	Unexpected error occurred during processing
QMI_ERR_MALFORMED_MSG	Message was not formulated correctly by the control point or the message was corrupted during transmission
QMI_ERR_NO_MEMORY	Device could not allocate memory to formulate a response
QMI_ERR_INVALID_HANDLE	Invalid client handle was received

3.36.3 Description of QMI_LOC_SET_SERVER

This command is used to set the A-GPS server address. If multiple types of addresses are specified in the request, the IPV4 address takes precedence over the IPV6 address and the IPV6 address takes precedence over the URL address. The request is acknowledged through the response, but the SUCCESS/FAILURE status is sent through indication QMI_LOC_SET_SERVER_IND.

3.37 QMI_LOC_GET_SERVER

Gets the location server from the location engine.

LOC message ID

0x0043

Version introduced

Major - 2, Minor - 2

3.37.1 Request - QMI_LOC_GET_SERVER_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

Name	Version last modified
Server Type	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x01		1	Server Type
Length	4		2	
Value	→	serverType	4	Type of server, as defined in qmiLocServerTypeEnumT. Valid values: • 0x00000001 – CDMA_PDE • 0x00000002 – CDMA_MPC • 0x00000003 – UMTS_SLP • 0x00000004 – CUSTOM_PDE

Optional TLVs

Name	Version last modified
Server Address Type	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x10		1	Server Address Type
Length	1		2	
Value	→	serverAddrTypeMask	1	Type of address the client wants. If unspecified, the indication will contain all the types of addresses it has for the specified server type. Valid bitmasks: • 0x01 – IPV4 • 0x02 – IPV6 • 0x04 – URL

3.37.2 Indication - QMI_LOC_GET_SERVER_IND

Message type

Indication

Sender

Service

Mandatory TLVs

At least one of the following optional TLVs is present if the status is eQMI_LOC_SUCCESS.

Name	Version last modified
Get Server Status	2.2
Server Type	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x01		1	Get Server Status
Length	4		2	

Field	Field value	Parameter	Size (byte)	Description
Value	→	status	4	Status of the Get Server request. Valid values: • 0x00000000 – SUCCESS • 0x00000001 – GENERAL_FAILURE • 0x00000002 – UNSUPPORTED • 0x00000003 – INVALID_PARAMETER • 0x00000004 – ENGINE_BUSY • 0x00000005 – PHONE_OFFLINE • 0x00000006 – LOC_TIMEOUT
Type	0x02		1	Server Type
Length	4		2	
Value	→	serverType	4	Type of server, as defined in qmiLocServerTypeEnumT. Valid values: • 0x00000001 – CDMA_PDE • 0x00000002 – CDMA_MPC • 0x00000003 – UMTS_SLP • 0x00000004 – CUSTOM_PDE

Optional TLVs

Name	Version last modified
IPV4 Address	2.2
IPV6 Address	2.2
Uniform Resource Locator	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x10		1	IPV4 Address IPV4 address and port.
Length	6		2	
Value	→	addr	4	IPV4 address. • Type: Unsigned integer
		port	2	IPV4 port. • Type: Unsigned integer
Type	0x11		1	IPV6 Address IPV6 address and port.
Length	20		2	
Value	→	addr	16	IPV6 address. • Type: Array of unsigned integers • Maximum length of the array: 8
		port	4	IPV6 port. • Type: Unsigned integer
Type	0x12		1	Uniform Resource Locator
Length	Var		2	

Field	Field value	Parameter	Size (byte)	Description
Value	→	urlAddr	Var	URL. • Type: NULL-terminated string • Maximum string length (including NULL terminator): 256

Error codes

QMI_ERR_NONE	No error in the request
QMI_ERR_INTERNAL	Unexpected error occurred during processing
QMI_ERR_MALFORMED_MSG	Message was not formulated correctly by the control point or the message was corrupted during transmission
QMI_ERR_NO_MEMORY	Device could not allocate memory to formulate a response
QMI_ERR_INVALID_HANDLE	Invalid client handle was received

3.37.3 Description of QMI_LOC_GET_SERVER

This command is used to get the A-GPS server address from the engine. The request is acknowledged through the response, but the SUCCESS/FAILURE status is sent through the indication QMI_LOC_GET_SERVER_IND. If successful, the indication also contains the A-GPS server address.

3.38 QMI_LOC_DELETE_ASSIST_DATA

This command is used to delete the location engine assistance data

LOC message ID

0x0044

Version introduced

Major - 2, Minor - 2

3.38.1 Request - QMI_LOC_DELETE_ASSIST_DATA_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

Name	Version last modified
Delete All	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x01		1	Delete All
Length	1		2	
Value	→	deleteAllFlag	1	Whether all assistance data is to be deleted. Valid values: • 0x01 (TRUE) – All assistance data is to be deleted; if this flag is set, all the other information contained in the optional fields for this message are ignored • 0x00 (FALSE) – The optional fields in the message are to be used to determine which data is to be deleted

Optional TLVs

Name	Version last modified
Delete SV Info	2.2
Delete GNSS Data	2.2
Delete Cell Database	2.2
Delete Clock Info	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x10		1	Delete SV Info List of satellites for which the assitance data is to be deleted.
Length	Var		2	
Value	→	deleteSvInfoList_len	1	Number of sets of the following elements: • gnssSvId • system • deleteSvInfoMask
		gnssSvId	2	SV ID of the satellite whose data is to be deleted. • Type: Unsigned integer • Range: • For GPS: 1 to 32 • For SBAS: 33 to 64 • For GLONASS: 65 to 96
		system	4	Indicates to which constellation this SV belongs. Valid values: • 0x00000001 – eQMI_LOC_SV_SYSTEM_GPS • 0x00000002 – eQMI_LOC_SV_SYSTEM_GALILEO • 0x00000003 – eQMI_LOC_SV_SYSTEM_SBAS • 0x00000004 – eQMI_LOC_SV_SYSTEM_COMPASS • 0x00000005 – eQMI_LOC_SV_SYSTEM_GLONASS
		deleteSvInfoMask	1	Indicates if the ephemeris or almanac for a satellite is to be deleted. Valid values: • 0x01 – DELETE_EPHEMERIS • 0x02 – DELETE_ALMANAC
Type	0x11		1	Delete GNSS Data
Length	8		2	

Field	Field value	Parameter	Size (byte)	Description
Value	→	deleteGnssDataMask	8	Mask for the GNSS data that is to be deleted. Valid values: • 0x00000001 – DELETE_GPS_SVDIR • 0x00000002 – DELETE_GPS_SVSTEER • 0x00000004 – DELETE_GPS_TIME • 0x00000008 – DELETE_GPS_ALM_CORR • 0x00000010 – DELETE_GLO_SVDIR • 0x00000020 – DELETE_GLO_SVSTEER • 0x00000040 – DELETE_GLO_TIME • 0x00000080 – DELETE_GLO_ALM_CORR • 0x00000100 – DELETE_SBAS_SVDIR • 0x00000200 – DELETE_SBAS_SVSTEER • 0x00000400 – DELETE_POSITION • 0x00000800 – DELETE_TIME • 0x00001000 – DELETE_IONO • 0x00002000 – DELETE_UTC • 0x00004000 – DELETE_HEALTH • 0x00008000 – DELETE_SADATA • 0x00010000 – DELETE_RTI • 0x00020000 – DELETE_SV_NO_EXIST • 0x00040000 – DELETE_FREQ_BIAS_EST
Type	0x12		1	Delete Cell Database
Length	4		2	
Value	→	deleteCellDbDataMask	4	Mask for the cell database assistance data that is to be deleted. Valid values: • 0x00000001 – DELETE_CELLDB_POS • 0x00000002 – DELETE_CELLDB_ LATEST_GPS_POS • 0x00000004 – DELETE_CELLDB_ OTA_POS • 0x00000008 – DELETE_CELLDB_EXT_ REF_POS • 0x00000010 – DELETE_CELLDB_ TIMETAG • 0x00000020 – DELETE_CELLDB_ CELLID • 0x00000040 – DELETE_CELLDB_ CACHED_CELLID • 0x00000080 – DELETE_CELLDB_ LAST_SRV_CELL • 0x00000100 – DELETE_CELLDB_CUR_ SRV_CELL • 0x00000200 – DELETE_CELLDB_ NEIGHBOR_INFO
Type	0x13		1	Delete Clock Info
Length	4		2	

Field	Field value	Parameter	Size (byte)	Description
Value	→	deleteClockInfoMask	4	Mask for the clock information assistance data that is to be deleted. Valid values: • 0x00000001 – DELETE_CLOCK_INFO_TIME_EST • 0x00000002 – DELETE_CLOCK_INFO_FREQ_EST • 0x00000004 – DELETE_CLOCK_INFO_WEEK_NUMBER • 0x00000008 – DELETE_CLOCK_INFO_RTC_TIME • 0x00000010 – DELETE_CLOCK_INFO_TIME_TRANSFER • 0x00000020 – DELETE_CLOCK_INFO_GPSTIME_EST • 0x00000040 – DELETE_CLOCK_INFO_GLOTIME_EST • 0x00000080 – DELETE_CLOCK_INFO_GLODAY_NUMBER • 0x00000100 – DELETE_CLOCK_INFO_GLO4YEAR_NUMBER • 0x00000200 – DELETE_CLOCK_INFO_GLO_RF_GRP_DELAY • 0x00000400 – DELETE_CLOCK_INFO_DISABLE_TT

3.38.2 Indication - QMI_LOC_DELETE_ASSIST_DATA_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version last modified
Delete Assist Data Status	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x01		1	Delete Assist Data Status

Field	Field value	Parameter	Size (byte)	Description
Length	4		2	
Value	→	status	4	Status of the Delete Assist Data request. Valid values: • 0x00000000 – SUCCESS • 0x00000001 – GENERAL_FAILURE • 0x00000002 – UNSUPPORTED • 0x00000003 – INVALID_PARAMETER • 0x00000004 – ENGINE_BUSY • 0x00000005 – PHONE_OFFLINE • 0x00000006 – LOC_TIMEOUT

Optional TLVs

None

Error codes

QMI_ERR_NONE	No error in the request
QMI_ERR_INTERNAL	Unexpected error occurred during processing
QMI_ERR_MALFORMED_MSG	Message was not formulated correctly by the control point or the message was corrupted during transmission
QMI_ERR_NO_MEMORY	Device could not allocate memory to formulate a response
QMI_ERR_INVALID_HANDLE	Invalid client handle was received

3.38.3 Description of QMI_LOC_DELETE_ASSIST_DATA

This command is used to delete location engine assistance data. The request is acknowledged through the response, but the SUCCESS/FAILURE status is sent through the indication QMI_LOC_DELETE_ASSIST_DATA_IND.

3.39 QMI_LOC_SET_XTRA_T_SESSION_CONTROL

Enables/disables XTRA-T session control.

LOC message ID

0x0045

Version introduced

Major - 2, Minor - 2

3.39.1 Request - QMI_LOC_SET_XTRA_T_SESSION_CONTROL_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

Name	Version last modified
Enable XTRA-T	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x01		1	Enable XTRA-T
Length	1		2	
Value	→	xtraTSessionControl	1	Whether to enable XTRA-T: • 0x01 (TRUE) – Enable XTRA-T • 0x00 (FALSE) – Disable XTRA-T

Optional TLVs

None

3.39.2 Indication - QMI_LOC_SET_XTRA_T_SESSION_CONTROL_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version last modified
Set XTRA-T Session Control Status	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x01		1	Set XTRA-T Session Control Status
Length	4		2	
Value	→	status	4	Status of the Set XTRA-T Session Control request. Valid values: • 0x00000000 – SUCCESS • 0x00000001 – GENERAL_FAILURE • 0x00000002 – UNSUPPORTED • 0x00000003 – INVALID_PARAMETER • 0x00000004 – ENGINE_BUSY • 0x00000005 – PHONE_OFFLINE • 0x00000006 – LOC_TIMEOUT

Optional TLVs

None

Error codes

QMI_ERR_NONE	No error in the request
QMI_ERR_INTERNAL	Unexpected error occurred during processing
QMI_ERR_MALFORMED_MSG	Message was not formulated correctly by the control point or the message was corrupted during transmission
QMI_ERR_NO_MEMORY	Device could not allocate memory to formulate a response
QMI_ERR_INVALID_HANDLE	Invalid client handle was received

3.39.3 Description of QMI_LOC_SET_XTRA_T_SESSION_CONTROL

This command is used to enable/disable XTRA-T user session control. The request is acknowledged through the response, but the SUCCESS/FAILURE status is sent through the indication QMI_LOC_SET_XTRA_T_SESSION_CONTROL_IND.

3.40 QMI_LOC_GET_XTRA_T_SESSION_CONTROL

Gets the XTRA-T session control value from the location engine.

LOC message ID

0x0046

Version introduced

Major - 2, Minor - 2

3.40.1 Request - QMI_LOC_GET_XTRA_T_SESSION_CONTROL_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

None

Optional TLVs

None

3.40.2 Indication - QMI_LOC_GET_XTRA_T_SESSION_CONTROL_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version last modified
Get XTRA-T Session Control Status	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x01		1	Get XTRA-T Session Control Status
Length	4		2	
Value	→	status	4	Status of the Get XTRA-T Session Control request. Valid values: • 0x00000000 – SUCCESS • 0x00000001 – GENERAL_FAILURE • 0x00000002 – UNSUPPORTED • 0x00000003 – INVALID_PARAMETER • 0x00000004 – ENGINE_BUSY • 0x00000005 – PHONE_OFFLINE • 0x00000006 – LOC_TIMEOUT

Optional TLVs

At least one of the following optional TLVs is present if the status is eQMI_LOC_SUCCESS.

Name	Version last modified
Enable/Disable XTRA-T	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x10		1	Enable/Disable XTRA-T
Length	1		2	
Value	→	xtraTSessionControl	1	Whether to enable XTRA-T: • 0x01 (TRUE) – Enable XTRA-T • 0x00 (FALSE) – Disable XTRA-T

Error codes

QMI_ERR_NONE	No error in the request
QMI_ERR_INTERNAL	Unexpected error occurred during processing
QMI_ERR_MALFORMED_MSG	Message was not formulated correctly by the control point or the message was corrupted during transmission
QMI_ERR_NO_MEMORY	Device could not allocate memory to formulate a response
QMI_ERR_INVALID_HANDLE	Invalid client handle was received

3.40.3 Description of QMI_LOC_GET_XTRA_T_SESSION_CONTROL

This command is used to get XTRA-T session control. The request is acknowledged through the response, but the SUCCESS/FAILURE status is sent through the indication QMI_LOC_GET_XTRA_T_SESSION_CONTROL_IND. If successful, the indication also contains the current XTRA-T session control state (enabled/disabled).

3.41 QMI_LOC_INJECT_WIFI_POSITION

Injects the WiFi position.

LOC message ID

0x0047

Version introduced

Major - 2, Minor - 2

3.41.1 Request - QMI_LOC_INJECT_WIFI_POSITION_REQ

Message type

Request

Sender

Control Point

Mandatory TLVs

None

Optional TLVs

Name	Version last modified
WiFi Fix Time	2.2
WiFi Position	2.2
WiFi Access Point Information	2.2
Horizontal Reliability	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x10		1	WiFi Fix Time Time of WiFi position fix.
Length	4		2	
Value	→	wifiPositionTime	4	Common counter (typically, the number of milliseconds since bootup). This field is only to be provided if the modem and host processors are synchronized. • Type: Unsigned integer

Field	Field value	Parameter	Size (byte)	Description
Type	0x11		1	WiFi Position WiFi position fix.
Length	23		2	
Value	→	lat	8	WiFi position latitude. • Type: Floating point • Units: Degrees
		lon	8	WiFi position longitude. • Type: Floating point • Units: Degrees
		hepe	2	WiFi position HEPE. • Type: Unsigned integer • Units: Meters
		numApsUsed	1	Number of Access Points (AP) used to generate a fix. • Type: Unsigned integer
		fixErrorCode	4	WiFi position error code; set to 0 if the fix succeeds. This position is only used by a module if the value is 0. If there was a failure, the error code provided by the WiFi positioning system can be provided here. Valid values: • 0x00000000 – ERROR_SUCCESS • 0x00000001 – ERROR_WIFI_NOT_AVAILABLE • 0x00000002 – ERROR_NO_AP_FOUND • 0x00000003 – ERROR_UNAUTHORIZED • 0x00000004 – ERROR_SERVER_UNAVAILABLE • 0x00000005 – ERROR_LOCATION_CANNOT_BE_DETERMINED • 0x00000006 – ERROR_UNKNOWN
Type	0x12		1	WiFi Access Point Information AP scan list.
Length	Var		2	
Value	→	apInfo_len	1	Number of sets of the following elements: • macAddr • rssi • channel • apQualifier
		macAddr	6	Associated MAC address of the AP. • Type: Array of unsigned integers • Address length: 6
		rssi	4	Receive signal strength indicator. • Type: Signed integer • Units: dBm (offset with +100 dB)
		channel	2	WiFi channel on which a beacon was received. • Type: Unsigned integer

Field	Field value	Parameter	Size (byte)	Description
		apQualifier	1	A bitmask of Boolean qualifiers for APs. All unused bits in this mask must be set to 0. Valid values: • 0x01 – BEING_USED • 0x02 – HIDDEN_SSID • 0x04 – PRIVATE • 0x08 – INFRASTRUCTURE_MODE
Type	0x13		1	Horizontal Reliability
Length	4		2	
Value	→	horizontalReliability	4	Specifies the reliability of the horizontal position. Valid values: • 0x00000000 – eQMI_LOC_RELIABILITY_NOT_SET • 0x00000001 – eQMI_LOC_RELIABILITY_VERY_LOW • 0x00000002 – eQMI_LOC_RELIABILITY_LOW • 0x00000003 – eQMI_LOC_RELIABILITY_MEDIUM • 0x00000004 – eQMI_LOC_RELIABILITY_HIGH

3.41.2 Indication - QMI_LOC_INJECT_WIFI_POSITION_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version last modified
Inject WiFi Position Status	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x01		1	Inject WiFi Position Status
Length	4		2	

Field	Field value	Parameter	Size (byte)	Description
Value	→	status	4	Status of the Inject WiFi Position request. Valid values: • 0x00000000 – SUCCESS • 0x00000001 – GENERAL_FAILURE • 0x00000002 – UNSUPPORTED • 0x00000003 – INVALID_PARAMETER • 0x00000004 – ENGINE_BUSY • 0x00000005 – PHONE_OFFLINE • 0x00000006 – LOC_TIMEOUT

Optional TLVs

None

Error codes

QMI_ERR_NONE	No error in the request
QMI_ERR_INTERNAL	Unexpected error occurred during processing
QMI_ERR_MALFORMED_MSG	Message was not formulated correctly by the control point or the message was corrupted during transmission
QMI_ERR_NO_MEMORY	Device could not allocate memory to formulate a response
QMI_ERR_INVALID_HANDLE	Invalid client handle was received

3.41.3 Description of QMI_LOC_INJECT_WIFI_POSITION

This message injects coarse position information into the QMI_LOC service. The purpose of this message is to improve performance of the service; specifically, to shorten the time to first fix.

How to set the reliability indicator: It is suggested that this TLV not be used by customers unless they have an advanced usage plan for this input or output. Pre-existing GPS system performance will remain the same if this feature is not used. If customers want to use this feature, it is recommended that the reliability level input is set in a way that is consistent with the metric definitions (defined in the TLV above).

3.42 QMI_LOC_NOTIFY_WIFI_STATUS

Notifies the location engine of the WiFi status.

LOC message ID

0x0048

Version introduced

Major - 2, Minor - 2

3.42.1 Request - QMI_LOC_NOTIFY_WIFI_STATUS_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

Name	Version last modified
Availablility of WiFi	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x01		1	Availablility of WiFi
Length	4		2	
Value	→	wifiStatus	4	WiFi status information. Valid values: • 0x00000001 – WIFI_STATUS_ AVAILABLE • 0x00000002 – WIFI_STATUS_ UNAVAILABLE

Optional TLVs

None

3.42.2 Indication - QMI_LOC_NOTIFY_WIFI_STATUS_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version last modified
Status of Notify WiFi Status Request	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x01		1	Status of Notify WiFi Status Request
Length	4		2	
Value	→	status	4	Status of the Notify WiFi Status request. Valid values: • 0x00000000 – SUCCESS • 0x00000001 – GENERAL_FAILURE • 0x00000002 – UNSUPPORTED • 0x00000003 – INVALID_PARAMETER • 0x00000004 – ENGINE_BUSY • 0x00000005 – PHONE_OFFLINE • 0x00000006 – LOC_TIMEOUT

Optional TLVs

None

Error codes

QMI_ERR_NONE	No error in the request
QMI_ERR_INTERNAL	Unexpected error occurred during processing
QMI_ERR_MALFORMED_MSG	Message was not formulated correctly by the control point or the message was corrupted during transmission
QMI_ERR_NO_MEMORY	Device could not allocate memory to formulate a response
QMI_ERR_INVALID_HANDLE	Invalid client handle was received

3.42.3 Description of QMI_LOC_NOTIFY_WIFI_STATUS

This command is used to notify the location engine of the WiFi status. The request is acknowledged through the response, but the SUCCESS/FAILURE status is sent through the indication QMI_LOC_NOTIFY_WIFI_STATUS_IND.

3.43 QMI_LOC_GET_REGISTERED_EVENTS

Gets the mask of the events for which a client has registered.

LOC message ID

0x0049

Version introduced

Major - 2, Minor - 2

3.43.1 Request - QMI_LOC_GET_REGISTERED_EVENTS_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

None

Optional TLVs

None

3.43.2 Indication - QMI_LOC_GET_REGISTERED_EVENTS_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version last modified
Get Registered Events Status	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x01		1	Get Registered Events Status
Length	4		2	
Value	→	status	4	Status of the Get Registered Events request. Valid values: • 0x00000000 – SUCCESS • 0x00000001 – GENERAL_FAILURE • 0x00000002 – UNSUPPORTED • 0x00000003 – INVALID_PARAMETER • 0x00000004 – ENGINE_BUSY • 0x00000005 – PHONE_OFFLINE • 0x00000006 – LOC_TIMEOUT

Optional TLVs

At least one of the following optional TLVs is present if the status is eQMI_LOC_SUCCESS.

Name	Version last modified
Event Registration Mask	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x10		1	Event Registration Mask
Length	8		2	

Field	Field value	Parameter	Size (byte)	Description
Value	→	eventRegMask	8	Event registration mask. Valid bitmasks: • 0x00000001 – POSITION_REPORT • 0x00000002 – GNSS_SV_INFO • 0x00000004 – NMEA • 0x00000008 – NI_NOTIFY_VERIFY_REQ • 0x00000010 – INJECT_TIME_REQ • 0x00000020 – INJECT_PREDICTED_ ORBITS_REQ • 0x00000040 – INJECT_POSITION_REQ • 0x00000080 – ENGINE_STATE • 0x00000100 – FIX_SESSION_STATE • 0x00000200 – WIFI_REQ • 0x00000400 – SENSOR_STREAMING_ READY_STATUS • 0x00000800 – TIME_SYNC_REQ • 0x00001000 – SET_SPI_STREAMING_ REPORT • 0x00002000 – LOCATION_SERVER_ CONNECTION_REQ

Error codes

QMI_ERR_NONE	No error in the request
QMI_ERR_INTERNAL	Unexpected error occurred during processing
QMI_ERR_MALFORMED_MSG	Message was not formulated correctly by the control point or the message was corrupted during transmission
QMI_ERR_NO_MEMORY	Device could not allocate memory to formulate a response
QMI_ERR_INVALID_HANDLE	Invalid client handle was received

3.43.3 Description of QMI_LOC_GET_REGISTERED_EVENTS

This command is used to get the events for which a client has registered. The request is acknowledged through the response, but the SUCCESS/FAILURE status is sent through the indication QMI_LOC_GET_REGISTERED_EVENTS_IND. If successful, the indication also contains the event mask for the client that made the request.

3.44 QMI_LOC_SET_OPERATION_MODE

Tells the engine to use the specified operation mode while making the position fixes. This command is not to be used by multiple clients concurrently.

LOC message ID

0x004A

Version introduced

Major - 2, Minor - 2

3.44.1 Request - QMI_LOC_SET_OPERATION_MODE_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

Name	Version last modified
Operation Mode	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x01		1	Operation Mode
Length	4		2	
Value	→	operationMode	4	Preferred operation mode. Valid values: • 0x00000001 – OPER_MODE_DEFAULT • 0x00000002 – OPER_MODE_MSB • 0x00000003 – OPER_MODE_MSA • 0x00000004 – OPER_MODE_ STANDALONE • 0x00000005 – OPER_MODE_CELL_ID

Optional TLVs

None

3.44.2 Indication - QMI_LOC_SET_OPERATION_MODE_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version last modified
Set Operation Mode Status	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x01		1	Set Operation Mode Status
Length	4		2	
Value	→	status	4	Status of the Set Operation Mode request. Valid values: • 0x00000000 – SUCCESS • 0x00000001 – GENERAL_FAILURE • 0x00000002 – UNSUPPORTED • 0x00000003 – INVALID_PARAMETER • 0x00000004 – ENGINE_BUSY • 0x00000005 – PHONE_OFFLINE • 0x00000006 – LOC_TIMEOUT

Optional TLVs

None

Error codes

QMI_ERR_NONE	No error in the request
QMI_ERR_INTERNAL	Unexpected error occurred during processing
QMI_ERR_MALFORMED_MSG	Message was not formulated correctly by the control point or the message was corrupted during transmission
QMI_ERR_NO_MEMORY	Device could not allocate memory to formulate a response
QMI_ERR_INVALID_HANDLE	Invalid client handle was received

3.44.3 Description of QMI_LOC_SET_OPERATION_MODE

This command is used to ask the location engine to use the specified operation mode. Only privileged clients should be allowed to set position mode, as it may affect the clients who are making a fix request. The request is acknowledged through the response, but the SUCCESS/FAILURE status is sent through the indication QMI_LOC_SET_OPERATION_MODE_IND.

3.45 QMI_LOC_GET_OPERATION_MODE

Gets the current operation mode from the engine.

LOC message ID

0x004B

Version introduced

Major - 2, Minor - 2

3.45.1 Request - QMI_LOC_GET_OPERATION_MODE_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

None

Optional TLVs

None

3.45.2 Indication - QMI_LOC_GET_OPERATION_MODE_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version last modified
Get Operation Mode Status	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x01		1	Get Operation Mode Status
Length	4		2	
Value	→	status	4	Status of the Get Operation Mode request. Valid values: • 0x00000000 – SUCCESS • 0x00000001 – GENERAL_FAILURE • 0x00000002 – UNSUPPORTED • 0x00000003 – INVALID_PARAMETER • 0x00000004 – ENGINE_BUSY • 0x00000005 – PHONE_OFFLINE • 0x00000006 – LOC_TIMEOUT

Optional TLVs

At least one of the following optional TLVs is present if the status is eQMI_LOC_SUCCESS.

Name	Version last modified
Operation Mode	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x10		1	Operation Mode
Length	4		2	
Value	→	operationMode	4	Current operation mode. Valid values: • 0x00000001 – OPER_MODE_DEFAULT • 0x00000002 – OPER_MODE_MSB • 0x00000003 – OPER_MODE_MSA • 0x00000004 – OPER_MODE_ STANDALONE • 0x00000005 – OPER_MODE_CELL_ID

Error codes

QMI_ERR_NONE	No error in the request
QMI_ERR_INTERNAL	Unexpected error occurred during processing
QMI_ERR_MALFORMED_MSG	Message was not formulated correctly by the control point or the message was corrupted during transmission
QMI_ERR_NO_MEMORY	Device could not allocate memory to formulate a response
QMI_ERR_INVALID_HANDLE	Invalid client handle was received

3.45.3 Description of QMI_LOC_GET_OPERATION_MODE

This command is used to get the operation mode that the location engine is using. The request is acknowledged through the response, but the SUCCESS/FAILURE status is sent through the indication QMI_LOC_GET_OPERATION_MODE_IND.

3.46 QMI_LOC_SET_SPI_STATUS

Used by the control point to set the SPI status, which indicates whether the device is stationary.

LOC message ID

0x004C

Version introduced

Major - 2, Minor - 2

3.46.1 Request - QMI_LOC_SET_SPI_STATUS_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

Name	Version last modified
Stationary Status	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x01		1	Stationary Status
Length	1		2	
Value	→	stationary	1	Whether the device is stationary: • 0x00 (FALSE) – Device is not stationary • 0x01 (TRUE) – Device is stationary

Optional TLVs

Name	Version last modified
Confidence	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x10		1	Confidence
Length	1		2	
Value	→	confidenceStationary	1	Confidence in the Stationary state expressed as a percentage. • Type: Unsigned integer • Range: 0 to 100

3.46.2 Indication - QMI_LOC_SET_SPI_STATUS_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version last modified
Status of SPI Status Request	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x01		1	Status of SPI Status Request
Length	4		2	
Value	→	status	4	Status of the SPI Status request. Valid values: • 0x00000000 – SUCCESS • 0x00000001 – GENERAL_FAILURE • 0x00000002 – UNSUPPORTED • 0x00000003 – INVALID_PARAMETER • 0x00000004 – ENGINE_BUSY • 0x00000005 – PHONE_OFFLINE • 0x00000006 – LOC_TIMEOUT

Optional TLVs

None

Error codes

QMI_ERR_NONE	No error in the request
QMI_ERR_INTERNAL	Unexpected error occurred during processing
QMI_ERR_MALFORMED_MSG	Message was not formulated correctly by the control point or the message was corrupted during transmission
QMI_ERR_NO_MEMORY	Device could not allocate memory to formulate a response
QMI_ERR_INVALID_HANDLE	Invalid client handle was received

3.46.3 Description of QMI_LOC_SET_SPI_STATUS

This command is used by the control point to inject the current SPI status.

3.47 QMI_LOC_INJECT_SENSOR_DATA

Used by the control point to inject sensor data into the GNSS location engine.

LOC message ID

0x004D

Version introduced

Major - 2, Minor - 2

3.47.1 Request - QMI_LOC_INJECT_SENSOR_DATA_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

None

Optional TLVs

Name	Version last modified
Opaque Identifier	2.2
3-Axis Accelerometer Data	2.2
3-Axis Gyrometer Data	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x10		1	Opaque Identifier
Length	4		2	
Value	→	opaqueIdentifier	4	An opaque identifier that is sent in by the client that will be echoed in the indication so the client can relate the indication to the request. • Type: Unsigned integer
Type	0x11		1	3-Axis Accelerometer Data Accelerometer sensor samples.

Field	Field value	Parameter	Size (byte)	Description
Length	Var		2	
Value	→	timeOfFirstSample	4	Denotes a full 32-bit time tag of the first (oldest) sample in this message. • Type: Unsigned integer • Units: Milliseconds
		flags	1	Flags to indicate any deviation from the default measurement assumptions. All unused bits in this field must be set to 0. Valid bitmasks: • 0x01 – SIGN REVERSAL
		sensorData_len	1	Number of sets of the following elements: • timeOffset • xAxis • yAxis • zAxis
		timeOffset	2	Sample time offset. This time offset must be relative to the sensor time of the first sample. • Type: Unsigned integer • Units: Milliseconds
		xAxis	4	Sensor x-axis sample. • Type: Floating point • Units Accelerometer: ((meters)/(seconds$\hat{2}$)) • Units Gyrometer: ((rads)/(seconds$\hat{2}$))
		yAxis	4	Sensor y-axis sample. • Type: Floating point • Units Accelerometer: ((meters)/(seconds$\hat{2}$)) • Units Gyrometer: ((rads)/(seconds$\hat{2}$))
		zAxis	4	Sensor z-axis sample. • Type: Floating point • Units Accelerometer: ((meters)/(seconds$\hat{2}$)) • Units Gyrometer: ((rads)/(seconds$\hat{2}$))
Type	0x12		1	3-Axis Gyrometer Data Gyrometer sensor samples.
Length	Var		2	
Value	→	timeOfFirstSample	4	Denotes a full 32-bit time tag of the first (oldest) sample in this message. • Type: Unsigned integer • Units: Milliseconds
		flags	1	Flags to indicate any deviation from the default measurement assumptions. All unused bits in this field must be set to 0. Valid bitmasks: • 0x01 – SIGN REVERSAL
		sensorData_len	1	Number of sets of the following elements: • timeOffset • xAxis • yAxis • zAxis

Field	Field value	Parameter	Size (byte)	Description
		timeOffset	2	Sample time offset. This time offset must be relative to the sensor time of the first sample. • Type: Unsigned integer • Units: Milliseconds
		xAxis	4	Sensor x-axis sample. • Type: Floating point • Units Accelerometer: ((meters)/(seconds$\hat{2}$)) • Units Gyrometer: ((rads)/(seconds$\hat{2}$))
		yAxis	4	Sensor y-axis sample. • Type: Floating point • Units Accelerometer: ((meters)/(seconds$\hat{2}$)) • Units Gyrometer: ((rads)/(seconds$\hat{2}$))
		zAxis	4	Sensor z-axis sample. • Type: Floating point • Units Accelerometer: ((meters)/(seconds$\hat{2}$)) • Units Gyrometer: ((rads)/(seconds$\hat{2}$))

3.47.2 Indication - QMI_LOC_INJECT_SENSOR_DATA_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version last modified
Inject Sensor Data Status	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x01		1	Inject Sensor Data Status
Length	4		2	

Field	Field value	Parameter	Size (byte)	Description
Value	→	status	4	Status of the Inject Sensor Data request. Valid values: • 0x00000000 – SUCCESS • 0x00000001 – GENERAL_FAILURE • 0x00000002 – UNSUPPORTED • 0x00000003 – INVALID_PARAMETER • 0x00000004 – ENGINE_BUSY • 0x00000005 – PHONE_OFFLINE • 0x00000006 – LOC_TIMEOUT

Optional TLVs

At least one of the following optional TLVs is present if the status is eQMI_LOC_SUCCESS.

Name	Version last modified
Opaque Identifier	2.2
Accelerometer Samples Accepted	2.2
Gyrometer Samples Accepted	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x10		1	Opaque Identifier
Length	4		2	
Value	→	opaqueIdentifier	4	An opaque identifier that was sent in by the client echoed so the client can relate the indication to the request. • Type: Unsigned integer
Type	0x11		1	Accelerometer Samples Accepted
Length	1		2	
Value	→	threeAxisAccelSamples Accepted	1	This field lets the client know how many 3-axis accelerometer samples were accepted.
Type	0x12		1	Gyrometer Samples Accepted
Length	1		2	
Value	→	threeAxisGyroSamples Accepted	1	This field lets the client know how many 3-axis gyrometer samples were accepted.

Error codes

QMI_ERR_NONE	No error in the request
QMI_ERR_INTERNAL	Unexpected error occurred during processing
QMI_ERR_MALFORMED_MSG	Message was not formulated correctly by the control point or the message was corrupted during transmission
QMI_ERR_NO_MEMORY	Device could not allocate memory to formulate a response
QMI_ERR_INVALID_HANDLE	Invalid client handle was received

3.47.3 Description of QMI_LOC_INJECT_SENSOR_DATA

This command is used by the control point to inject sensor data into the GPS engine.

3.48 QMI_LOC_INJECT_TIME_SYNC_DATA

Used by the control point to inject time sync data.

LOC message ID

0x004E

Version introduced

Major - 2, Minor - 2

3.48.1 Request - QMI_LOC_INJECT_TIME_SYNC_DATA_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

Name	Version last modified
Reference Time Sync Counter	2.2
Sensor Receive Time	2.2
Sensor Transmit Time	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x01		1	Reference Time Sync Counter
Length	4		2	
Value	→	refCounter	4	Must be set to the value that was sent to the control point when the GNSS location engine requested time sync injection. • Type: Unsigned integer
Type	0x02		1	Sensor Receive Time
Length	4		2	

Field	Field value	Parameter	Size (byte)	Description
Value	→	sensorProcRxTime	4	The value of the sensor time when the control point received the Time Sync Inject request from the GNSS location engine. Must be monotonically increasing, jitter ≤ 1 millisecond, never stopping until the process is rebooted. • Type: Unsigned integer • Units: Milliseconds
Type	0x03		1	Sensor Transmit Time
Length	4		2	
Value	→	sensorProcTxTime	4	The value of the sensor time when the control point injects this message for use by the GNSS location engine. Must be monotonically increasing, jitter ≤ 1 millisecond, never stopping until the process is rebooted. • Type: Unsigned integer • Units: Milliseconds

Optional TLVs

None

3.48.2 Indication - QMI_LOC_INJECT_TIME_SYNC_DATA_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version last modified
Inject Time Sync Data Status	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x01		1	Inject Time Sync Data Status
Length	4		2	

Field	Field value	Parameter	Size (byte)	Description
Value	→	status	4	Status of the Inject Time Sync Data request. Valid values: • 0x00000000 – SUCCESS • 0x00000001 – GENERAL_FAILURE • 0x00000002 – UNSUPPORTED • 0x00000003 – INVALID_PARAMETER • 0x00000004 – ENGINE_BUSY • 0x00000005 – PHONE_OFFLINE • 0x00000006 – LOC_TIMEOUT

Optional TLVs

None

Error codes

QMI_ERR_NONE	No error in the request
QMI_ERR_INTERNAL	Unexpected error occurred during processing
QMI_ERR_MALFORMED_MSG	Message was not formulated correctly by the control point or the message was corrupted during transmission
QMI_ERR_NO_MEMORY	Device could not allocate memory to formulate a response
QMI_ERR_INVALID_HANDLE	Invalid client handle was received

3.48.3 Description of QMI_LOC_INJECT_TIME_SYNC_DATA

This command is used by the control point to inject time sync data.

3.49 QMI_LOC_GET_CRADLE_MOUNT_CONFIG

Used by the control point to get the current cradle mount configuration.

LOC message ID

0x0050

Version introduced

Major - 2, Minor - 2

3.49.1 Request - QMI_LOC_GET_CRADLE_MOUNT_CONFIG_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

None

Optional TLVs

None

3.49.2 Indication - QMI_LOC_GET_CRADLE_MOUNT_CONFIG_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version last modified
Get Cradle Mount Config Status	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x01		1	Get Cradle Mount Config Status
Length	4		2	
Value	→	status	4	Status of the Get Cradle Mount Configuration request. Valid values: • 0x00000000 – SUCCESS • 0x00000001 – GENERAL_FAILURE • 0x00000002 – UNSUPPORTED • 0x00000003 – INVALID_PARAMETER • 0x00000004 – ENGINE_BUSY • 0x00000005 – PHONE_OFFLINE • 0x00000006 – LOC_TIMEOUT

Optional TLVs

At least one of the following optional TLVs is present if the status is eQMI_LOC_SUCCESS.

Name	Version last modified
Cradle Mount State	2.2
Cradle Mount Confidence	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x10		1	Cradle Mount State
Length	4		2	
Value	→	cradleMountState	4	Cradle Mount state set by the control point. Valid values: • 0x00000000 – CRADLE_STATE_ NOT_MOUNTED • 0x00000001 – CRADLE_STATE_ MOUNTED • 0x00000002 – CRADLE_STATE_ UNKNOWN
Type	0x11		1	Cradle Mount Confidence
Length	1		2	
Value	→	confidenceCradleMount State	1	Confidence of the Cradle Mount state expressed as a percentage. • Type: Unsigned integer • Range: 0 to 100

Error codes

QMI_ERR_NONE	No error in the request
QMI_ERR_INTERNAL	Unexpected error occurred during processing
QMI_ERR_MALFORMED_MSG	Message was not formulated correctly by the control point or the message was corrupted during transmission
QMI_ERR_NO_MEMORY	Device could not allocate memory to formulate a response
QMI_ERR_INVALID_HANDLE	Invalid client handle was received

3.49.3 Description of QMI_LOC_GET_CRADLE_MOUNT_CONFIG

This command is used by the control point to get the current cradle mount configuration.

3.50 QMI_LOC_SET_CRADLE_MOUNT_CONFIG

Used by the control point to set the current cradle mount configuration.

LOC message ID

0x004F

Version introduced

Major - 2, Minor - 2

3.50.1 Request - QMI_LOC_SET_CRADLE_MOUNT_CONFIG_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

Name	Version last modified
Cradle Mount State	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x01		1	Cradle Mount State
Length	4		2	
Value	→	cradleMountState	4	Cradle Mount state set by the control point. Valid values: • 0x00000000 – CRADLE_STATE_ NOT_MOUNTED • 0x00000001 – CRADLE_STATE_ MOUNTED • 0x00000002 – CRADLE_STATE_ UNKNOWN

Optional TLVs

Name	Version last modified
Cradle Mount Confidence	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x10		1	Cradle Mount Confidence
Length	1		2	
Value	→	confidenceCradleMount State	1	Confidence in the Cradle Mount state expressed as a percentage. • Type: Unsigned integer • Range: 0 to 100

3.50.2 Indication - QMI_LOC_SET_CRADLE_MOUNT_CONFIG_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version last modified
Set Cradle Mount Config Status	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x01		1	Set Cradle Mount Config Status
Length	4		2	
Value	→	status	4	Status of the Set Cradle Mount Configuration request. Valid values: • 0x00000000 – SUCCESS • 0x00000001 – GENERAL_FAILURE • 0x00000002 – UNSUPPORTED • 0x00000003 – INVALID_PARAMETER • 0x00000004 – ENGINE_BUSY • 0x00000005 – PHONE_OFFLINE • 0x00000006 – LOC_TIMEOUT

Optional TLVs

None

Error codes

QMI_ERR_NONE	No error in the request
QMI_ERR_INTERNAL	Unexpected error occurred during processing
QMI_ERR_MALFORMED_MSG	Message was not formulated correctly by the control point or the message was corrupted during transmission
QMI_ERR_NO_MEMORY	Device could not allocate memory to formulate a response
QMI_ERR_INVALID_HANDLE	Invalid client handle was received

3.50.3 Description of QMI_LOC_SET_CRADLE_MOUNT_CONFIG

This command is used by the control point to set the current cradle mount configuration.

3.51 QMI_LOC_GET_EXTERNAL_POWER_CONFIG

Used by the control point to get the current external power configuration.

LOC message ID

0x0052

Version introduced

Major - 2, Minor - 2

3.51.1 Request - QMI_LOC_GET_EXTERNAL_POWER_CONFIG_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

None

Optional TLVs

None

3.51.2 Indication - QMI_LOC_GET_EXTERNAL_POWER_CONFIG_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version last modified
Get Ext Power Config Status	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x01		1	Get Ext Power Config Status
Length	4		2	
Value	→	status	4	Status of the Get External Power Configuration request. Valid values: • 0x00000000 – SUCCESS • 0x00000001 – GENERAL_FAILURE • 0x00000002 – UNSUPPORTED • 0x00000003 – INVALID_PARAMETER • 0x00000004 – ENGINE_BUSY • 0x00000005 – PHONE_OFFLINE • 0x00000006 – LOC_TIMEOUT

Optional TLVs

At least one of the following optional TLVs is present if the status is eQMI_LOC_SUCCESS.

Name	Version last modified
External Power State	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x10		1	External Power State
Length	4		2	
Value	→	externalPowerState	4	Power state; injected by the control point. Valid values: • 0x00000000 – EXTERNAL_POWER_NOT_CONNECTED • 0x00000001 – EXTERNAL_POWER_CONNECTED • 0x00000002 – EXTERNAL_POWER_UNKNOWN

Error codes

QMI_ERR_NONE	No error in the request
QMI_ERR_INTERNAL	Unexpected error occurred during processing
QMI_ERR_MALFORMED_MSG	Message was not formulated correctly by the control point or the message was corrupted during transmission
QMI_ERR_NO_MEMORY	Device could not allocate memory to formulate a response
QMI_ERR_INVALID_HANDLE	Invalid client handle was received

3.51.3 Description of QMI_LOC_GET_EXTERNAL_POWER_CONFIG

This command is used by the control point to get the current external power configuration.

3.52 QMI_LOC_SET_EXTERNAL_POWER_CONFIG

Used by the control point to set the current external power configuration.

LOC message ID

0x0051

Version introduced

Major - 2, Minor - 2

3.52.1 Request - QMI_LOC_SET_EXTERNAL_POWER_CONFIG_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

Name	Version last modified
External Power State	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x01		1	External Power State
Length	4		2	
Value	→	externalPowerState	4	Power state; injected by the control point. Valid values: • 0x00000000 – EXTERNAL_POWER_NOT_CONNECTED • 0x00000001 – EXTERNAL_POWER_CONNECTED • 0x00000002 – EXTERNAL_POWER_UNKNOWN

Optional TLVs

None

3.52.2 Indication - QMI_LOC_SET_EXTERNAL_POWER_CONFIG_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version last modified
Set Ext Power Config Status	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x01		1	Set Ext Power Config Status
Length	4		2	
Value	→	status	4	Status of the Set External Power Configuration request. Valid values: • 0x00000000 – SUCCESS • 0x00000001 – GENERAL_FAILURE • 0x00000002 – UNSUPPORTED • 0x00000003 – INVALID_PARAMETER • 0x00000004 – ENGINE_BUSY • 0x00000005 – PHONE_OFFLINE • 0x00000006 – LOC_TIMEOUT

Optional TLVs

None

Error codes

QMI_ERR_NONE	No error in the request
QMI_ERR_INTERNAL	Unexpected error occurred during processing
QMI_ERR_MALFORMED_MSG	Message was not formulated correctly by the control point or the message was corrupted during transmission
QMI_ERR_NO_MEMORY	Device could not allocate memory to formulate a response
QMI_ERR_INVALID_HANDLE	Invalid client handle was received

3.52.3 Description of QMI_LOC_SET_EXTERNAL_POWER_CONFIG

This command is used by the control point to set the current external power configuration.

3.53 QMI_LOC_INFORM_LOCATION_SERVER_CONN_STATUS

Used by the control point to inform the service about the status of the location server connection request that the service may have sent via the QMI_LOC_EVENT_LOCATION_SERVER_REQ_IND event.

LOC message ID

0x0053

Version introduced

Major - 2, Minor - 2

3.53.1 Request - QMI_LOC_INFORM_LOCATION_SERVER_CONN_STATUS_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

Name	Version last modified
Connection Handle	2.2
Request Type	2.2
Connection Status	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x01		1	Connection Handle
Length	4		2	
Value	→	connHandle	4	Connection handle that the service specified in the Location Server Connection request event. • Type: Unsigned integer
Type	0x02		1	Request Type
Length	4		2	

Field	Field value	Parameter	Size (byte)	Description
Value	→	requestType	4	Type of connection request service that was specified in the Location Server Connection Request event. Valid values: • 0x00000001 – OPEN • 0x00000002 – CLOSE
Type	0x03		1	Connection Status
Length	4		2	
Value	→	statusType	4	Status of the Connection request. Valid values: • 0x00000001 – STATUS_SUCCESS = 1 • 0x00000002 – STATUS_FAILURE = 2

Optional TLVs

Name	Version last modified
APN Profile	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x10		1	APN Profile APN profile information is present only when requestType is OPEN and statusType is SUCCESS.
Length	Var		2	
Value	→	pdnType	4	PDN type of the Access Point Name (APN) profile. Valid values: • 0x00000001 – PDN_TYPE_IPV4 • 0x00000002 – PDN_TYPE_IPV6 • 0x00000003 – PDN_TYPE_IPV4V6 • 0x00000004 – PDN_TYPE_PPP
		apnName_len	1	Number of sets of the following elements: • apnName
		apnName	Var	APN name. • Type: NULL-terminated string • Maximum string length (including NULL terminator): 101

3.53.2 Indication - QMI_LOC_INFORM_LOCATION_SERVER_CONN_STATUS_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version last modified
Status of Inform Loc Server Conn Status	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x01		1	Status of Inform Loc Server Conn Status
Length	4		2	
Value	→	status	4	Status of the Inform Location Server Connection Status request. Valid values: • 0x00000000 – SUCCESS • 0x00000001 – GENERAL_FAILURE • 0x00000002 – UNSUPPORTED • 0x00000003 – INVALID_PARAMETER • 0x00000004 – ENGINE_BUSY • 0x00000005 – PHONE_OFFLINE • 0x00000006 – LOC_TIMEOUT

Optional TLVs

None

Error codes

QMI_ERR_NONE	No error in the request
QMI_ERR_INTERNAL	Unexpected error occurred during processing
QMI_ERR_MALFORMED_MSG	Message was not formulated correctly by the control point or the message was corrupted during transmission
QMI_ERR_NO_MEMORY	Device could not allocate memory to formulate a response
QMI_ERR_INVALID_HANDLE	Invalid client handle was received

3.53.3 Description of QMI_LOC_INFORM_LOCATION_SERVER_CONN_STATUS

This command is used by the control point to inform the service about the status of the Location Server Connection request sent by the engine via the event QMI_LOC_EVENT_LOCATION_SERVER_REQ_IND.

3.54 QMI_LOC_SET_PROTOCOL_CONFIG_PARAMETERS

Used by the control point to configure parameters stored in the nonvolatile memory.

LOC message ID

0x0054

Version introduced

Major - 2, Minor - 2

3.54.1 Request - QMI_LOC_SET_PROTOCOL_CONFIG_PARAMETERS_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

None

Optional TLVs

Name	Version last modified
SUPL Security	2.2
VX Version	2.2
SUPL Version	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x10		1	SUPL Security
Length	1		2	
Value	→	suplSecurity	1	Indicates whether SUPL security is enabled. • 0x01 (TRUE) – SUPL security is enabled • 0x00 (FALSE) – SUPL security is disabled
Type	0x11		1	VX Version
Length	4		2	

Field	Field value	Parameter	Size (byte)	Description
Value	→	vxVersion	4	VX version. Valid values: • 0x00000001 – VX_VERSION_V1_ONLY • 0x00000002 – VX_VERSION_V2_ONLY
Type	0x12		1	SUPL Version
Length	4		2	
Value	→	suplVersion	4	SUPL version. Valid values: • 0x00000001 – SUPL_VERSION_1_0 • 0x00000002 – SUPL_VERSION_2_0

3.54.2 Indication - QMI_LOC_SET_PROTOCOL_CONFIG_PARAMETERS_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version last modified
Set Config Params Status	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x01		1	Set Config Params Status
Length	4		2	
Value	→	status	4	Status of the Set Configuration Parameters request. Valid values: • 0x00000000 – SUCCESS • 0x00000001 – GENERAL_FAILURE • 0x00000002 – UNSUPPORTED • 0x00000003 – INVALID_PARAMETER • 0x00000004 – ENGINE_BUSY • 0x00000005 – PHONE_OFFLINE • 0x00000006 – LOC_TIMEOUT

Optional TLVs

At least one of the following optional TLVs is present if the status is not eQMI_LOC_SUCCESS.

Name	Version last modified
Failed Parameters	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x10		1	Failed Parameters
Length	8		2	
Value	→	failedProtocolConfig ParamMask	8	This field is sent only if the status is not a success. And if it is not successful, this field will identify the parameters that were not set successfully. Valid bitmasks: • 0x0000000000000001 – CONFIG_PARAM_MASK_SUPL_SECURITY • 0x0000000000000002 – CONFIG_PARAM_MASK_VX_VERSION • 0x0000000000000004 – CONFIG_PARAM_MASK_SUPL_VERSION

Error codes

QMI_ERR_NONE	No error in the request
QMI_ERR_INTERNAL	Unexpected error occurred during processing
QMI_ERR_MALFORMED_MSG	Message was not formulated correctly by the control point or the message was corrupted during transmission
QMI_ERR_NO_MEMORY	Device could not allocate memory to formulate a response
QMI_ERR_INVALID_HANDLE	Invalid client handle was received

3.54.3 Description of QMI_LOC_SET_PROTOCOL_CONFIG_PARAMETERS

This command is used by the control point to set the configuration parameters that are stored in the nonvolatile memory. The command can be used to set one or more configuration parameters at a time.

If the implementation does not support multiple optional TLVs, then eQMI_LOC_UNSUPPORTED error will be returned and no action will be taken.

3.55 QMI_LOC_GET_PROTOCOL_CONFIG_PARAMETERS

Used by the control point to get the configuration parameters stored in the nonvolatile memory.

LOC message ID

0x0055

Version introduced

Major - 2, Minor - 2

3.55.1 Request - QMI_LOC_GET_PROTOCOL_CONFIG_PARAMETERS_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

Name	Version last modified
Config Parameters	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x01		1	Config Parameters
Length	8		2	
Value	→	getProtocolConfigParam Mask	8	Mask denoting the configuration parameters to be retrieved. Valid bitmasks: • 0x0000000000000001 – CONFIG_PARAM_MASK_SUPL_SECURITY • 0x0000000000000002 – CONFIG_PARAM_MASK_VX_VERSION • 0x0000000000000004 – CONFIG_PARAM_MASK_SUPL_VERSION

Optional TLVs

None

3.55.2 Indication - QMI_LOC_GET_PROTOCOL_CONFIG_PARAMETERS_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version last modified
Get Config Params Status	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x01		1	Get Config Params Status
Length	4		2	
Value	→	status	4	Status of the Get Configuration Parameters request. Valid values: • 0x00000000 – SUCCESS • 0x00000001 – GENERAL_FAILURE • 0x00000002 – UNSUPPORTED • 0x00000003 – INVALID_PARAMETER • 0x00000004 – ENGINE_BUSY • 0x00000005 – PHONE_OFFLINE • 0x00000006 – LOC_TIMEOUT

Optional TLVs

At least one of the following optional TLVs is present if the status is eQMI_LOC_SUCCESS.

Name	Version last modified
SUPL Security	2.2
VX Version	2.2
SUPL Version	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x10		1	SUPL Security
Length	1		2	

Field	Field value	Parameter	Size (byte)	Description
Value	→	suplSecurity	1	Indicates whether SUPL security is enabled. • 0x01 (TRUE) – SUPL security is enabled • 0x00 (FALSE) – SUPL security is disabled
Type	0x11		1	VX Version
Length	4		2	
Value	→	vxVersion	4	VX version. Valid values: • 0x00000001 – VX_VERSION_V1_ONLY • 0x00000002 – VX_VERSION_V2_ONLY
Type	0x12		1	SUPL Version
Length	4		2	
Value	→	suplVersion	4	SUPL version. Valid values: • 0x00000001 – SUPL_VERSION_1_0 • 0x00000002 – SUPL_VERSION_2_0

Error codes

QMI_ERR_NONE	No error in the request
QMI_ERR_INTERNAL	Unexpected error occurred during processing
QMI_ERR_MALFORMED_MSG	Message was not formulated correctly by the control point or the message was corrupted during transmission
QMI_ERR_NO_MEMORY	Device could not allocate memory to formulate a response
QMI_ERR_INVALID_HANDLE	Invalid client handle was received

3.55.3 Description of QMI_LOC_GET_PROTOCOL_CONFIG_PARAMETERS

This command is used by the control point to retrieve the configuration parameters that are stored in the nonvolatile memory. The command can be used to get one or more configuration parameters at a time.

If the implementation does not support multiple parameters, then eQMI_LOC_UNSUPPORTED error will be returned and no action will be taken.

3.56 QMI_LOC_SET_SENSOR_CONTROL_CONFIG

Sets the sensor control configuration.

LOC message ID

0x0056

Version introduced

Major - 2, Minor - 2

3.56.1 Request - QMI_LOC_SET_SENSOR_CONTROL_CONFIG_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

None

Optional TLVs

Name	Version last modified
Sensors Usage	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x10		1	Sensors Usage
Length	4		2	

Field	Field value	Parameter	Size (byte)	Description
Value	→	sensorsUsage	4	Controls how sensors are used to aid heading and positioning performance. Valid values: • 0x00000000 – SENSORS USE ENABLED: Sensors data is to be requested whenever a position request is received. If sensors data is injected, the GNSS location engine attempts to improve the heading and positioning performance using sensors. This is the default. • 0x00000001 – SENSORS USE DISABLED: Inertial sensors are not to be used to aid in heading and position improvement.

3.56.2 Indication - QMI_LOC_SET_SENSOR_CONTROL_CONFIG_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version last modified
Set Sensor Control Config Status	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x01		1	Set Sensor Control Config Status
Length	4		2	
Value	→	status	4	Status of the Set Sensor Control Configuration request. Valid values: • 0x00000000 – SUCCESS • 0x00000001 – GENERAL_FAILURE • 0x00000002 – UNSUPPORTED • 0x00000003 – INVALID_PARAMETER • 0x00000004 – ENGINE_BUSY • 0x00000005 – PHONE_OFFLINE • 0x00000006 – LOC_TIMEOUT

Optional TLVs

None

Error codes

QMI_ERR_NONE	No error in the request
QMI_ERR_INTERNAL	Unexpected error occurred during processing
QMI_ERR_MALFORMED_MSG	Message was not formulated correctly by the control point or the message was corrupted during transmission
QMI_ERR_NO_MEMORY	Device could not allocate memory to formulate a response
QMI_ERR_INVALID_HANDLE	Invalid client handle was received

3.56.3 Description of QMI_LOC_SET_SENSOR_CONTROL_CONFIG

This command is used to set the sensor control configuration. The request is acknowledged through the response, but the SUCCESS/FAILURE status is sent through the indication QMI_LOC_SET_SENSOR_CONTROL_CONFIG_IND.

This message is a global setting for sensors, i.e., it applies to all sensor clients. Consequently, THIS MESSAGE SHOULD ONLY BE SENT BY ONE CLIENT; its simultaneous use by multiple clients can cause undefined behavior.

3.57 QMI_LOC_GET_SENSOR_CONTROL_CONFIG

Retrieves the current sensor control configuration.

LOC message ID

0x0057

Version introduced

Major - 2, Minor - 2

3.57.1 Request - QMI_LOC_GET_SENSOR_CONTROL_CONFIG_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

None

Optional TLVs

None

3.57.2 Indication - QMI_LOC_GET_SENSOR_CONTROL_CONFIG_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version last modified
Get Sensor Control Config Status	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x01		1	Get Sensor Control Config Status
Length	4		2	
Value	→	status	4	Status of the Get Sensors Control Configuration request. Valid values: • 0x00000000 – SUCCESS • 0x00000001 – GENERAL_FAILURE • 0x00000002 – UNSUPPORTED • 0x00000003 – INVALID_PARAMETER • 0x00000004 – ENGINE_BUSY • 0x00000005 – PHONE_OFFLINE • 0x00000006 – LOC_TIMEOUT

Optional TLVs

At least one of the following optional TLVs is present if the status is eQMI_LOC_SUCCESS.

Name	Version last modified
Sensors Usage	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x10		1	Sensors Usage
Length	4		2	
Value	→	sensorsUsage	4	Controls how sensors are used to aid the heading and positioning performance. Valid values: • 0x00000000 – SENSORS USE ENABLED: Sensors data is to be requested whenever a position request is received. If sensors data is injected, the GNSS location engine attempts to improve the heading and positioning performance using sensors. This is the default. • 0x00000001 – SENSORS USE DISABLED: Inertial sensors are not to be used to aid in the heading and position improvement.

Error codes

QMI_ERR_NONE	No error in the request
QMI_ERR_INTERNAL	Unexpected error occurred during processing
QMI_ERR_MALFORMED_MSG	Message was not formulated correctly by the control point or the message was corrupted during transmission
QMI_ERR_NO_MEMORY	Device could not allocate memory to formulate a response
QMI_ERR_INVALID_HANDLE	Invalid client handle was received

3.57.3 Description of QMI_LOC_GET_SENSOR_CONTROL_CONFIG

This command is used to get the sensor control configuration. The request is acknowledged through the response, but the SUCCESS/FAILURE status is sent through the indication QMI_LOC_GET_SENSOR_CONTROL_CONFIG_IND.

3.58 QMI_LOC_SET_SENSOR_PROPERTIES

Sets the properties specific to the type of sensor used. The control point must set sensor properties before they can be used to aid in heading and positioning performance improvement.

LOC message ID

0x0058

Version introduced

Major - 2, Minor - 2

3.58.1 Request - QMI_LOC_SET_SENSOR_PROPERTIES_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

None

Optional TLVs

Name	Version last modified
Gyro Bias Variance	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x10		1	Gyro Bias Variance
Length	4		2	
Value	→	gyroBiasVarianceRandom Walk	4	Specifies the gyro bias random walk parameter as a positive floating-point value. This value does not have any internal defaults. The gyro bias variance random walk parameter is derived from either the sensors data sheet or a sensors conformance test. • Units: radians$\hat{2}$/second$\hat{4}$

3.58.2 Indication - QMI_LOC_SET_SENSOR_PROPERTIES_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version last modified
Set Sensor Properties Status	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x01		1	Set Sensor Properties Status
Length	4		2	
Value	→	status	4	Status of the Set Sensor Properties request. Valid values: • 0x00000000 – SUCCESS • 0x00000001 – GENERAL_FAILURE • 0x00000002 – UNSUPPORTED • 0x00000003 – INVALID_PARAMETER • 0x00000004 – ENGINE_BUSY • 0x00000005 – PHONE_OFFLINE • 0x00000006 – LOC_TIMEOUT

Optional TLVs

None

Error codes

QMI_ERR_NONE	No error in the request
QMI_ERR_INTERNAL	Unexpected error occurred during processing
QMI_ERR_MALFORMED_MSG	Message was not formulated correctly by the control point or the message was corrupted during transmission
QMI_ERR_NO_MEMORY	Device could not allocate memory to formulate a response
QMI_ERR_INVALID_HANDLE	Invalid client handle was received

3.58.3 Description of QMI_LOC_SET_SENSOR_PROPERTIES

This command is used to set the sensor properties. The request is acknowledged through the response, but the SUCCESS/FAILURE status is sent through the indication QMI_LOC_SET_SENSOR_PROPERTIES_IND.

These data are fundamental properties of the sensors and must be derived from either the data sheet or a conformance test.

The control point MUST provide these values, prefereably at initial bootup, for sensors integration to be active. If these values are set to 0 or not provided, it indicates that the control point has NOT initialized these values and, consequently, sensors will not be used to improve GNSS heading and positioning performance.

3.59 QMI_LOC_GET_SENSOR_PROPERTIES

Retrieves the current sensor properties.

LOC message ID

0x0059

Version introduced

Major - 2, Minor - 2

3.59.1 Request - QMI_LOC_GET_SENSOR_PROPERTIES_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

None

Optional TLVs

None

3.59.2 Indication - QMI_LOC_GET_SENSOR_PROPERTIES_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version last modified
Get Sensor Properties Status	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x01		1	Get Sensor Properties Status
Length	4		2	
Value	→	status	4	Status of the Get Sensors Properties request. Valid values: • 0x00000000 – SUCCESS • 0x00000001 – GENERAL_FAILURE • 0x00000002 – UNSUPPORTED • 0x00000003 – INVALID_PARAMETER • 0x00000004 – ENGINE_BUSY • 0x00000005 – PHONE_OFFLINE • 0x00000006 – LOC_TIMEOUT

Optional TLVs

At least one of the following optional TLVs is present if the status is eQMI_LOC_SUCCESS.

Name	Version last modified
Gyro Bias Variance	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x10		1	Gyro Bias Variance
Length	4		2	
Value	→	gyroBiasVarianceRandom Walk	4	Specifies the gyro bias random walk parameter as a positive floating-point value. This value does not have any internal defaults. The gyro bias variance random walk parameter is derived from either the sensors data sheet or a sensors conformance test. • Units: radians$\hat{2}$/seconds$\hat{4}$

Error codes

QMI_ERR_NONE	No error in the request
QMI_ERR_INTERNAL	Unexpected error occurred during processing
QMI_ERR_MALFORMED_MSG	Message was not formulated correctly by the control point or the message was corrupted during transmission
QMI_ERR_NO_MEMORY	Device could not allocate memory to formulate a response
QMI_ERR_INVALID_HANDLE	Invalid client handle was received

3.59.3 Description of QMI_LOC_GET_SENSOR_PROPERTIES

This command is used to get the sensor control configuration. The request is acknowledged through the response, but the SUCCESS/FAILURE status is sent through the indication QMI_LOC_GET_SENSOR_PROPERTIES_IND.

3.60 QMI_LOC_SET_SENSOR_PERFORMANCE_CONTROL_CONFIGURATION

Provides fine-grained control of sensor based positioning performance

LOC message ID

0x005A

Version introduced

Major - 2, Minor - 2

3.60.1 Request - QMI_LOC_SET_SENSOR_PERFORMANCE_CONTROL_CONFIGURATION_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

None

Optional TLVs

Name	Version last modified
Sensor Performance Control Mode	2.2
Accelerometer Sampling Specification	2.2
Gyrometer Sampling Specification	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x10		1	Sensor Performance Control Mode
Length	4		2	

Field	Field value	Parameter	Size (byte)	Description
Value	→	performanceControlMode	4	Controls when sensors data is requested during GNSS fix processing. This field is relevant only when sensors have been enabled using the sensors control configuration. Valid values: • 0x00000000 – AUTO: The GNSS location engine can decide when to request sensor data injection based on internal criteria. This is the default. • 0x00000001 – FORCED: The GNSS location engine must request use of sensors every time the GNSS location engine turns on.
Type	0x11		1	Accelerometer Sampling Specification Sets the nominal rate at which the GNSS location engine is to request acceleration data. The sensor data rate is specified in terms of the nominal number of samples per batch and the number of batches per second. However, the final control of the actual requested rate resides with the Sensors Manager Module/GNSS location engine.
Length	4		2	
Value	→	samplesPerBatch	2	Specifies the number of samples per batch the GNSS location engine is to receive. The sensor sampling frequency can be computed as follows: samplingFrequency = samplesPerBatch * batchesPerSecond samplesPerBatch must be a non-zero positive value.
		batchesPerSecond	2	Number of sensor-data batches the GNSS location engine is to receive per second. The rate is specified in integral number of batches per second (Hz). batchesPerSecond must be a non-zero positive value.

Field	Field value	Parameter	Size (byte)	Description
Type	0x12		1	Gyrometer Sampling Specification Sets the nominal rate at which the GNSS location engine is to request gyro data. The sensor data rate is specified in terms of the nominal number of samples per batch and the number of batches per second. However, the final control of the actual requested rate resides with the Sensors Manager Module/GNSS location engine.
Length	4		2	
Value	→	samplesPerBatch	2	Specifies the number of samples per batch the GNSS location engine is to receive. The sensor sampling frequency can be computed as follows: samplingFrequency = samplesPerBatch * batchesPerSecond samplesPerBatch must be a non-zero positive value.
		batchesPerSecond	2	Number of sensor-data batches the GNSS location engine is to receive per second. The rate is specified in integral number of batches per second (Hz). batchesPerSecond must be a non-zero positive value.

3.60.2 Indication - QMI_LOC_SET_SENSOR_PERFORMANCE_CONTROL_CONFIGURATION_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version last modified
Set Sensor Perf Control Config Status	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x01		1	Set Sensor Perf Control Config Status
Length	4		2	
Value	→	status	4	Status of the Set Sensor Performance Control Configuration request. Valid values: • 0x00000000 – SUCCESS • 0x00000001 – GENERAL_FAILURE • 0x00000002 – UNSUPPORTED • 0x00000003 – INVALID_PARAMETER • 0x00000004 – ENGINE_BUSY • 0x00000005 – PHONE_OFFLINE • 0x00000006 – LOC_TIMEOUT

Optional TLVs

At least one of the following optional TLVs is present if the status is not eQMI_LOC_SUCCESS.

Name	Version last modified
Failed Configuration	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x10		1	Failed Configuration
Length	4		2	
Value	→	failedConfiguration	4	This field is sent only if the status is not a success. When sent, this field identifies which configuration failed. Valid bitmasks: • 0x0000000000000001 – PERFORMANCE_MODE • 0x0000000000000002 – ACCEL_SAMPLING_SPEC • 0x0000000000000004 – GYRO_SAMPLING_SPEC

Error codes

QMI_ERR_NONE	No error in the request
QMI_ERR_INTERNAL	Unexpected error occurred during processing
QMI_ERR_MALFORMED_MSG	Message was not formulated correctly by the control point or the message was corrupted during transmission
QMI_ERR_NO_MEMORY	Device could not allocate memory to formulate a response
QMI_ERR_INVALID_HANDLE	Invalid client handle was received

3.60.3 Description of QMI_LOC_SET_SENSOR_PERFORMANCE_CONTROL_CONFIGURATION

This command is used to provide fine-grained control of sensor processing behavior. The request is acknowledged through the response, but the SUCCESS/FAILURE status is sent through the corresponding indication.

3.61 QMI_LOC_GET_SENSOR_PERFORMANCE_CONTROL_CONFIGURATION

Retrieves the current sensor performance control configuration.

LOC message ID

0x005B

Version introduced

Major - 2, Minor - 2

3.61.1 Request - QMI_LOC_GET_SENSOR_PERFORMANCE_CONTROL_CONFIGURATION_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

None

Optional TLVs

None

3.61.2 Indication - QMI_LOC_GET_SENSOR_PERFORMANCE_CONTROL_CONFIGURATION_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version last modified
Get Sensor Perf Control Config Status	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x01		1	Get Sensor Perf Control Config Status
Length	4		2	
Value	→	status	4	Status of the Get Sensor Performance Control Configuration request. Valid values: • 0x00000000 – SUCCESS • 0x00000001 – GENERAL_FAILURE • 0x00000002 – UNSUPPORTED • 0x00000003 – INVALID_PARAMETER • 0x00000004 – ENGINE_BUSY • 0x00000005 – PHONE_OFFLINE • 0x00000006 – LOC_TIMEOUT

Optional TLVs

At least one of the following optional TLVs is present if the status is eQMI_LOC_SUCCESS.

Name	Version last modified
Performance Control Mode	2.2
Accelerometer Sampling Specification	2.2
Gyrometer Sampling Specification	2.2

Field	Field value	Parameter	Size (byte)	Description
Type	0x10		1	Performance Control Mode
Length	4		2	
Value	→	performanceControlMode	4	Controls when sensor data is requested during GNSS fix processing. This field is relevant only when sensors have been enabled using the sensor control configuration. Valid values: • 0x00000000 – AUTO: The GNSS location engine can decide when to request sensor data injection based on internal criteria. This is the default. • 0x0000001 – FORCED: The GNSS location engine must request use of the sensors every time the GNSS location engine turns on.

Field	Field value	Parameter	Size (byte)	Description
Type	0x11		1	Accelerometer Sampling Specification Sets the nominal rate at which the GNSS location engine is to request acceleration data. The sensor data rate is specified in terms of the nominal number of samples per batch and the number of batches per second. However, the final control of the actual requested rate resides with the Sensors Manager Module/GNSS location engine.
Length	4		2	
Value	→	samplesPerBatch	2	Specifies the number of samples per batch the GNSS location engine is to receive. The sensor sampling frequency can be computed as follows: samplingFrequency = samplesPerBatch * batchesPerSecond samplesPerBatch must be a non-zero positive value.
		batchesPerSecond	2	Number of sensor-data batches the GNSS location engine is to receive per second. The rate is specified in integral number of batches per second (Hz). batchesPerSecond must be a non-zero positive value.
Type	0x12		1	Gyrometer Sampling Specification Sets the nominal rate at which the GNSS location engine is to request gyro data. The sensor data rate is specified in terms of the nominal number of samples per batch and the number of batches per second. However, the final control of the actual requested rate resides with the Sensors Manager Module/GNSS location engine.
Length	4		2	
Value	→	samplesPerBatch	2	Specifies the number of samples per batch the GNSS location engine is to receive. The sensor sampling frequency can be computed as follows: samplingFrequency = samplesPerBatch * batchesPerSecond samplesPerBatch must be a non-zero positive value.

Field	Field value	Parameter	Size (byte)	Description
		batchesPerSecond	2	Number of sensor-data batches the GNSS location engine is to receive per second. The rate is specified in integral number of batches per second (Hz). batchesPerSecond must be a non-zero positive value.

Error codes

QMI_ERR_NONE	No error in the request
QMI_ERR_INTERNAL	Unexpected error occurred during processing
QMI_ERR_MALFORMED_MSG	Message was not formulated correctly by the control point or the message was corrupted during transmission
QMI_ERR_NO_MEMORY	Device could not allocate memory to formulate a response
QMI_ERR_INVALID_HANDLE	Invalid client handle was received

3.61.3 Description of QMI_LOC_GET_SENSOR_PERFORMANCE_CONTROL_CONFIGURATION

This command is used to get the sensor performance control configuration. The request is acknowledged through the response, but the SUCCESS/FAILURE status is sent through the corresponding indication.

3.62 QMI_LOC_INJECT_SUPL_CERTIFICATE

Injects a SUPL certificate to be used in AGNSS sessions.

LOC message ID

0x005C

Version introduced

Major - 2, Minor - 3

3.62.1 Request - QMI_LOC_INJECT_SUPL_CERTIFICATE_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

Name	Version last modified
SUPL Certificate ID	2.3
SUPL Certificate Data	2.3

Field	Field value	Parameter	Size (byte)	Description
Type	0x01		1	SUPL Certificate ID
Length	1		2	
Value	→	suplCertId	1	Certificate ID of the SUPL certificate. Type: Unsigned integer Units: Bytes Range: 0 to 9
Type	0x02		1	SUPL Certificate Data
Length	Var		2	
Value	→	suplCertData_len	2	Number of sets of the following elements: • suplCertData
		suplCertData	Var	SUPL certificate contents. Type: Array of bytes. Maximum certificate size: 2000 bytes

Optional TLVs

None

3.62.2 Indication - QMI_LOC_INJECT_SUPL_CERTIFICATE_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version last modified
SUPL Certificate Injection Status	2.3

Field	Field value	Parameter	Size (byte)	Description
Type	0x01		1	SUPL Certificate Injection Status
Length	4		2	
Value	→	status	4	Status of the inject SUPL certificate request. Valid Values: • 0x00000000 – SUCCESS • 0x00000001 – GENERAL_FAILURE • 0x00000002 – UNSUPPORTED • 0x00000003 – INVALID_PARAMETER • 0x00000004 – ENGINE_BUSY • 0x00000005 – PHONE_OFFLINE • 0x00000006 – LOC_TIMEOUT

Optional TLVs

None

Error codes

QMI_ERR_NONE	No error in the request
QMI_ERR_INTERNAL	Unexpected error occurred during processing
QMI_ERR_MALFORMED_MSG	Message was not formulated correctly by the control point or the message was corrupted during transmission
QMI_ERR_NO_MEMORY	Device could not allocate memory to formulate a response
QMI_ERR_INVALID_HANDLE	Invalid client handle was received

3.62.3 Description of QMI_LOC_INJECT_SUPL_CERTIFICATE

This command is used to inject a SUPL certificate used by the Location Service for AGNSS sessions. The request is acknowledged through the response, but the SUCCESS/FAILURE status is sent through the indication QMI_LOC_INJECT_SUPL_CERTIFICATE_IND.

3.63 QMI_LOC_DELETE_SUPL_CERTIFICATE

Deletes a SUPL certificate.

LOC message ID

0x005D

Version introduced

Major - 2, Minor - 3

3.63.1 Request - QMI_LOC_DELETE_SUPL_CERTIFICATE_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

None

Optional TLVs

Name	Version last modified
SUPL Certificate ID	2.3

Field	Field value	Parameter	Size (byte)	Description
Type	0x10		1	SUPL Certificate ID
Length	1		2	
Value	→	suplCertId	1	Certificate ID of the SUPL certificate to be deleted. Type: Unsigned Integer Units: Bytes Range: 0 to 9 If suplCertId is not specified, all SUPL certificates are deleted.

3.63.2 Indication - QMI_LOC_DELETE_SUPL_CERTIFICATE_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version last modified
SUPL Certifcate Deletion Status	2.3

Field	Field value	Parameter	Size (byte)	Description
Type	0x01		1	SUPL Certifcate Deletion Status
Length	4		2	
Value	→	status	4	Status of the delete SUPL certificate request. Valid Values: • 0x00000000 – SUCCESS • 0x00000001 – GENERAL_FAILURE • 0x00000002 – UNSUPPORTED • 0x00000003 – INVALID_PARAMETER • 0x00000004 – ENGINE_BUSY • 0x00000005 – PHONE_OFFLINE • 0x00000006 – LOC_TIMEOUT

Optional TLVs

None

Error codes

QMI_ERR_NONE	No error in the request
QMI_ERR_INTERNAL	Unexpected error occurred during processing
QMI_ERR_MALFORMED_MSG	Message was not formulated correctly by the control point or the message was corrupted during transmission
QMI_ERR_NO_MEMORY	Device could not allocate memory to formulate a response
QMI_ERR_INVALID_HANDLE	Invalid client handle was received

3.63.3 Description of QMI_LOC_DELETE_SUPL_CERTIFICATE

This command is used to delete a SUPL certificate. The request is acknowledged through the response, but the SUCCESS/FAILURE status is sent through the indication QMI_LOC_DELETE_PERSISTENT_DATA_IND.

3.64 QMI_LOC_SET_POSITION_ENGINE_CONFIG_PARAMETERS

Used by the control point to configure position engine functionality.

LOC message ID

0x005E

Version introduced

Major - 2, Minor - 3

3.64.1 Request - QMI_LOC_SET_POSITION_ENGINE_CONFIG_PARAMETERS_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

None

Optional TLVs

Name	Version last modified
Injected Position Control	2.3
Filter SV Usage	2.3
Store Assist Data	2.3

Field	Field value	Parameter	Size (byte)	Description
Type	0x10		1	Injected Position Control
Length	1		2	

Field	Field value	Parameter	Size (byte)	Description
Value	→	injectedPositionControl	1	Controls how the injected position is used in the position engine. Valid values: • 0x01 (TRUE) – Use the injected position in direct position calculation. • 0x00 (FALSE) – Do not use the injected position in direct position calculation. The default value is TRUE.
Type	0x11		1	Filter SV Usage
Length	1		2	
Value	→	filterSvUsage	1	Controls whether SV usage is filtered in a position fix. Valid values: • 0x01 (TRUE) – Filter the usage of SVs in the fix. • 0x00 (FALSE) – Do not filter the usage of SVs in the fix. The default value is FALSE.
Type	0x12		1	Store Assist Data
Length	1		2	
Value	→	storeAssistData	1	Controls whether assistance data is to be stored in persistent memory. Valid values: • 0x01 (TRUE) – Store assistance data in persistent memory. • 0x00 (FALSE) – Do not store assistance data in persistent memory. The default value is TRUE.

3.64.2 Indication - QMI_LOC_SET_POSITION_ENGINE_CONFIG_PARAMETERS_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version last modified
Set Position Engine Configuration Status	2.3

Field	Field value	Parameter	Size (byte)	Description
Type	0x01		1	Set Position Engine Configuration Status
Length	4		2	
Value	→	status	4	Status of the Set Configuration Parameters request. Valid values: • 0x00000000 – SUCCESS • 0x00000001 – GENERAL_FAILURE • 0x00000002 – UNSUPPORTED • 0x00000003 – INVALID_PARAMETER • 0x00000004 – ENGINE_BUSY • 0x00000005 – PHONE_OFFLINE • 0x00000006 – LOC_TIMEOUT

Optional TLVs

Atleast one of the following optional TLVs are present if the status is not eQMI_LOC_SUCCESS.

Name	Version last modified
Failed Parameters	2.3

Field	Field value	Parameter	Size (byte)	Description
Type	0x10		1	Failed Parameters
Length	4		2	
Value	→	failedPositionEngine ConfigParamMask	4	Identifies the parameters that were not set successfully. This field is sent only if the status is not SUCCESS. Valid bitmasks: • 0x00000001 – INJECTED_POSITION_CONTROL • 0x00000002 – FILTER_SV_USAGE • 0x00000004 – STORE_ASSIST_DATA

Error codes

QMI_ERR_NONE	No error in the request
QMI_ERR_INTERNAL	Unexpected error occurred during processing
QMI_ERR_MALFORMED_MSG	Message was not formulated correctly by the control point or the message was corrupted during transmission
QMI_ERR_NO_MEMORY	Device could not allocate memory to formulate a response
QMI_ERR_INVALID_HANDLE	Invalid client handle was received

3.64.3 Description of QMI_LOC_SET_POSITION_ENGINE_CONFIG_PARAMETERS

This command is used by the control point to set the configuration parameters used by the position engine. The command can be used to set one or more configuration parameters at a time. If the implementation does not support multiple optional TLVs, eQMI_LOC_UNSUPPORTED error is returned and no action is taken.

3.65 QMI_LOC_GET_POSITION_ENGINE_CONFIG_PARAMETERS

Used by the control point to get the position engine configuration parameters.

LOC message ID

0x005F

Version introduced

Major - 2, Minor - 3

3.65.1 Request - QMI_LOC_GET_POSITION_ENGINE_CONFIG_PARAMETERS_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

Name	Version last modified
Config Parameters	2.3

Field	Field value	Parameter	Size (byte)	Description
Type	0x01		1	Config Parameters
Length	4		2	
Value	→	getPositionEngineConfig ParamMask	4	Mask denoting the configuration parameters to be retrieved. Valid bitmasks: • 0x00000001 – INJECTED_POSITION_ CONTROL • 0x00000002 – FILTER_SV_USAGE • 0x00000004 – STORE_ASSIST_DATA

Optional TLVs

None

3.65.2 Indication - QMI_LOC_GET_POSITION_ENGINE_CONFIG_PARAMETERS_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version last modified
Get Position Engine Configuration Status	2.3

Field	Field value	Parameter	Size (byte)	Description
Type	0x01		1	Get Position Engine Configuration Status
Length	4		2	
Value	→	status	4	Status of the Get Configuration Parameters request. Valid values: • 0x00000000 – SUCCESS • 0x00000001 – GENERAL_FAILURE • 0x00000002 – UNSUPPORTED • 0x00000003 – INVALID_PARAMETER • 0x00000004 – ENGINE_BUSY • 0x00000005 – PHONE_OFFLINE • 0x00000006 – LOC_TIMEOUT

Optional TLVs

Atleast one of the following optional TLVs is present if the status is eQMI_LOC_SUCCESS.

Name	Version last modified
Injected Position Control	2.3
Filter SV Usage	2.3
Store Assist Data	2.3

Field	Field value	Parameter	Size (byte)	Description
Type	0x10		1	Injected Position Control
Length	1		2	

Field	Field value	Parameter	Size (byte)	Description
Value	→	injectedPositionControl	1	Specifies whether the injected position is used for direct calculation in the position engine. Valid values: • 0x01 (TRUE) – The injected position is used in the direct position calculation. • 0x00 (FALSE) – The injected position is not used the direct position calculation. The default value is TRUE, which means the injected position is used in direct position calculation by default.
Type	0x11		1	Filter SV Usage
Length	1		2	
Value	→	filterSvUsage	1	Specifies whether SV usage is filtered in a position fix. Valid values: • 0x01 (TRUE) – SV usage is filtered in the fix. • 0x00 (FALSE) – SV usage is not filtered in the fix. The default value is FALSE.
Type	0x12		1	Store Assist Data
Length	1		2	
Value	→	storeAssistData	1	Specifies whether assistance data is stored in persistent memory. Valid values: 0x01 (TRUE) – Assistance data is stored in persistent memory. 0x00 (FALSE) – Assistance data is not stored in persistent memory. The default value is TRUE.

Error codes

QMI_ERR_NONE	No error in the request
QMI_ERR_INTERNAL	Unexpected error occurred during processing
QMI_ERR_MALFORMED_MSG	Message was not formulated correctly by the control point or the message was corrupted during transmission
QMI_ERR_NO_MEMORY	Device could not allocate memory to formulate a response
QMI_ERR_INVALID_HANDLE	Invalid client handle was received

3.65.3 Description of QMI_LOC_GET_POSITION_ENGINE_CONFIG_PARAMETERS

This command is used by the control point to retrieve the position engine configuration parameters. The command can be used to get one or more configuration parameters at a time.

If the implementation does not support multiple parameters, eQMI_LOC_UNSUPPORTED error is returned and no action is taken.