

QMI LOC 2.32 for MPSS.DI.1.0

QMI Location Svc Spec

80-ND600-17 E

August 11, 2014

Confidential and Proprietary - Qualcomm Technologies, Inc.

NO PUBLIC DISCLOSURE PERMITTED: Please report postings of this document on public servers or websites to: DocCtrlAgent@qualcomm.com.

Restricted Distribution. Not to be distributed to anyone who is not an employee of either Qualcomm or its subsidiaries without the express approval of Qualcomm's Configuration Management.

Not to be used, copied, reproduced, or modified in whole or in part, nor its contents revealed in any manner to others without the express written permission of Qualcomm Technologies, Inc.

Qualcomm reserves the right to make changes to the product(s) or information contained herein without notice. No liability is assumed for any damages arising directly or indirectly by their use or application. The information provided in this document is provided on an "as is" basis.

This document contains confidential and proprietary information and must be shredded when discarded.

Qualcomm and MSM are trademarks of QUALCOMM Incorporated, registered in the United States and other countries. All QUALCOMM Incorporated trademarks are used with permission. Other product and brand names may be trademarks or registered trademarks of their respective owners.

This technical data may be subject to U.S. and international export, re-export, or transfer ("export") laws. Diversion contrary to U.S. and international law is strictly prohibited.

**Qualcomm Technologies, Inc.
5775 Morehouse Drive
San Diego, CA 92121
U.S.A.**

**© 2012-2014 Qualcomm Technologies, Inc.
All rights reserved.**

Contents

1	Introduction	17
1.1	Purpose	17
1.2	Scope	17
1.3	Conventions	18
1.4	References	18
1.5	Technical Assistance	18
1.6	Acronyms	19
2	Theory of Operation	21
2.1	Generalized QMI Service Compliance	21
2.2	LOC Service Type	21
2.3	Message Definition Template	21
2.3.1	Response Message Result TLV	21
2.4	Backward Compatibility and Version Negotiation	22
2.5	Asynchronous Messaging Paradigm	23
2.6	Input Message Queuing	23
2.7	Error Messages	23
2.8	QMI_LOC Design Fundamentals	24
2.9	QMI_LOC Fundamental Positioning Concepts	25
2.9.1	GNSS	25
2.9.2	Position Determination Methods	25
2.9.2.1	MS-assisted PD	25
2.9.2.2	MS-based PD	26
2.9.2.3	Cell ID-based PD	27
2.9.2.4	Standalone Fix	28
2.9.2.5	WWAN-based PD	28
2.9.3	Multiple-Client Support	29
2.9.4	Single-Shot Position Fix Sessions	29
2.9.5	Auto-Tracking Sessions	29
2.9.6	NMEA Sentence Data	30
2.9.7	External Information Injection	30
2.9.7.1	External Time Injection	30
2.9.7.2	Coarse Position Injection	30
2.9.7.3	WiFi Position Injection	30
2.9.8	gpsOneXTRA Satellite Database Information	30
2.9.9	Satellite-Based Augmentation System Configuration	31
2.9.10	External Sensor Data Input	31
2.9.11	Gyroscope Measurements	34
2.9.12	Location Fix Batching	34

2.9.12.1	Batching Sessions	34
2.9.12.2	Event Notification Registration	34
2.9.12.3	Batching Operations	35
3	QMI_LOC Messages	37
3.1	QMI_LOC_GEN_RESP	43
3.1.1	Response - QMI_LOC_GEN_RESP	43
3.1.2	Description of QMI_LOC_GEN_RESP REQ/RESP	43
3.2	QMI_LOC_GET_SUPPORTED_MSGS	44
3.2.1	Request - QMI_LOC_GET_SUPPORTED_MSGS_REQ	44
3.2.2	Response - QMI_LOC_GET_SUPPORTED_MSGS_RESP	44
3.2.3	Description of QMI_LOC_GET_SUPPORTED_MSGS REQ/RESP	45
3.3	QMI_LOC_GET_SUPPORTED_FIELDS	46
3.3.1	Request - QMI_LOC_GET_SUPPORTED_FIELDS_REQ	46
3.3.2	Response - QMI_LOC_GET_SUPPORTED_FIELDS_RESP	46
3.3.3	Description of QMI_LOC_GET_SUPPORTED_FIELDS REQ/RESP	48
3.4	QMI_LOC_INFORM_CLIENT_REVISION	49
3.4.1	Request - QMI_LOC_INFORM_CLIENT_REVISION_REQ	49
3.4.2	Description of QMI_LOC_INFORM_CLIENT_REVISION REQ/RESP	50
3.5	QMI_LOC_REG_EVENTS	51
3.5.1	Request - QMI_LOC_REG_EVENTS_REQ	51
3.5.2	Description of QMI_LOC_REG_EVENTS REQ/RESP	56
3.6	QMI_LOC_START	57
3.6.1	Request - QMI_LOC_START_REQ	57
3.6.2	Description of QMI_LOC_START REQ/RESP	60
3.7	QMI_LOC_STOP	61
3.7.1	Request - QMI_LOC_STOP_REQ	61
3.7.2	Description of QMI_LOC_STOP REQ/RESP	62
3.8	QMI_LOC_EVENT_POSITION_REPORT	63
3.8.1	Indication - QMI_LOC_EVENT_POSITION_REPORT_IND	63
3.8.2	Description of QMI_LOC_EVENT_POSITION_REPORT	72
3.9	QMI_LOC_EVENT_GNSS_SV_INFO	73
3.9.1	Indication - QMI_LOC_EVENT_GNSS_SV_INFO_IND	73
3.9.2	Description of QMI_LOC_EVENT_GNSS_SV_INFO	76
3.10	QMI_LOC_EVENT_NMEA	77
3.10.1	Indication - QMI_LOC_EVENT_NMEA_IND	77
3.10.2	Description of QMI_LOC_EVENT_NMEA	78
3.11	QMI_LOC_EVENT_NI_NOTIFY_VERIFY_REQ	79
3.11.1	Indication - QMI_LOC_EVENT_NI_NOTIFY_VERIFY_REQ_IND	79
3.11.2	Description of QMI_LOC_EVENT_NI_NOTIFY_VERIFY_REQ	94
3.12	QMI_LOC_EVENT_INJECT_TIME_REQ	95
3.12.1	Indication - QMI_LOC_EVENT_INJECT_TIME_REQ_IND	95
3.12.2	Description of QMI_LOC_EVENT_INJECT_TIME_REQ	96
3.13	QMI_LOC_EVENT_INJECT_PREDICTED_ORBITS_REQ	97
3.13.1	Indication - QMI_LOC_EVENT_INJECT_PREDICTED_ORBITS_REQ_IND	97
3.13.2	Description of QMI_LOC_EVENT_INJECT_PREDICTED_ORBITS_REQ	98
3.14	QMI_LOC_EVENT_INJECT_POSITION_REQ	99
3.14.1	Indication - QMI_LOC_EVENT_INJECT_POSITION_REQ_IND	99
3.14.2	Description of QMI_LOC_EVENT_INJECT_POSITION_REQ	100
3.15	QMI_LOC_EVENT_ENGINE_STATE	101

3.15.1	Indication - QMI_LOC_EVENT_ENGINE_STATE_IND	101
3.15.2	Description of QMI_LOC_EVENT_ENGINE_STATE	102
3.16	QMI_LOC_EVENT_FIX_SESSION_STATE	103
3.16.1	Indication - QMI_LOC_EVENT_FIX_SESSION_STATE_IND	103
3.16.2	Description of QMI_LOC_EVENT_FIX_SESSION_STATE	104
3.17	QMI_LOC_EVENT_WIFI_REQ	105
3.17.1	Indication - QMI_LOC_EVENT_WIFI_REQ_IND	105
3.17.2	Description of QMI_LOC_EVENT_WIFI_REQ	106
3.18	QMI_LOC_EVENT_SENSOR_STREAMING_READY_STATUS	107
3.18.1	Indication - QMI_LOC_EVENT_SENSOR_STREAMING_READY_STATUS_IND	107
3.18.2	Description of QMI_LOC_EVENT_SENSOR_STREAMING_READY_STATUS	111
3.19	QMI_LOC_EVENT_TIME_SYNC_REQ	112
3.19.1	Indication - QMI_LOC_EVENT_TIME_SYNC_REQ_IND	112
3.19.2	Description of QMI_LOC_EVENT_TIME_SYNC_REQ	113
3.20	QMI_LOC_EVENT_SET_SPI_STREAMING_REPORT	114
3.20.1	Indication - QMI_LOC_EVENT_SET_SPI_STREAMING_REPORT_IND	114
3.20.2	Description of QMI_LOC_EVENT_SET_SPI_STREAMING_REPORT	115
3.21	QMI_LOC_EVENT_LOCATION_SERVER_CONNECTION_REQ	116
3.21.1	Indication - QMI_LOC_EVENT_LOCATION_SERVER_CONNECTION_REQ_IND	116
3.21.2	Description of QMI_LOC_EVENT_LOCATION_SERVER_CONNECTION_REQ	117
3.22	QMI_LOC_GET_SERVICE_REVISION	118
3.22.1	Request - QMI_LOC_GET_SERVICE_REVISION_REQ	118
3.22.2	Indication - QMI_LOC_GET_SERVICE_REVISION_IND	118
3.22.3	Description of QMI_LOC_GET_SERVICE_REVISION	121
3.23	QMI_LOC_GET_FIX_CRITERIA	122
3.23.1	Request - QMI_LOC_GET_FIX_CRITERIA_REQ	122
3.23.2	Indication - QMI_LOC_GET_FIX_CRITERIA_IND	122
3.23.3	Description of QMI_LOC_GET_FIX_CRITERIA	125
3.24	QMI_LOC_INFORM_NI_USER_RESPONSE	126
3.24.1	Request - QMI_LOC_NI_USER_RESPONSE_REQ	126
3.24.2	Indication - QMI_LOC_NI_USER_RESPONSE_IND	141
3.24.3	Description of QMI_LOC_INFORM_NI_USER_RESPONSE	143
3.25	QMI_LOC_INJECT_PREDICTED_ORBITS_DATA	144
3.25.1	Request - QMI_LOC_INJECT_PREDICTED_ORBITS_DATA_REQ	144
3.25.2	Indication - QMI_LOC_INJECT_PREDICTED_ORBITS_DATA_IND	145
3.25.3	Description of QMI_LOC_INJECT_PREDICTED_ORBITS_DATA	147
3.26	QMI_LOC_GET_PREDICTED_ORBITS_DATA_SOURCE	148
3.26.1	Request - QMI_LOC_GET_PREDICTED_ORBITS_DATA_SOURCE_REQ	148
3.26.2	Indication - QMI_LOC_GET_PREDICTED_ORBITS_DATA_SOURCE_IND	148
3.26.3	Description of QMI_LOC_GET_PREDICTED_ORBITS_DATA_SOURCE	150
3.27	QMI_LOC_GET_PREDICTED_ORBITS_DATA_VALIDITY	151
3.27.1	Request - QMI_LOC_GET_PREDICTED_ORBITS_DATA_VALIDITY_REQ	151
3.27.2	Indication - QMI_LOC_GET_PREDICTED_ORBITS_DATA_VALIDITY_IND	151
3.27.3	Description of QMI_LOC_GET_PREDICTED_ORBITS_DATA_VALIDITY	153
3.28	QMI_LOC_INJECT_UTC_TIME	154
3.28.1	Request - QMI_LOC_INJECT_UTC_TIME_REQ	154
3.28.2	Indication - QMI_LOC_INJECT_UTC_TIME_IND	154

3.28.3	Description of QMI_LOC_INJECT.UTC_TIME	156
3.29	QMI_LOC_INJECT_POSITION	157
3.29.1	Request - QMI_LOC_INJECT_POSITION_REQ	157
3.29.2	Indication - QMI_LOC_INJECT_POSITION_IND	163
3.29.3	Description of QMI_LOC_INJECT_POSITION	164
3.30	QMI_LOC_SET_ENGINE_LOCK	165
3.30.1	Request - QMI_LOC_SET_ENGINE_LOCK_REQ	165
3.30.2	Indication - QMI_LOC_SET_ENGINE_LOCK_IND	166
3.30.3	Description of QMI_LOC_SET_ENGINE_LOCK	167
3.31	QMI_LOC_GET_ENGINE_LOCK	168
3.31.1	Request - QMI_LOC_GET_ENGINE_LOCK_REQ	168
3.31.2	Indication - QMI_LOC_GET_ENGINE_LOCK_IND	168
3.31.3	Description of QMI_LOC_GET_ENGINE_LOCK	170
3.32	QMI_LOC_SET_SBAS_CONFIG	171
3.32.1	Request - QMI_LOC_SET_SBAS_CONFIG_REQ	171
3.32.2	Indication - QMI_LOC_SET_SBAS_CONFIG_IND	171
3.32.3	Description of QMI_LOC_SET_SBAS_CONFIG	173
3.33	QMI_LOC_GET_SBAS_CONFIG	174
3.33.1	Request - QMI_LOC_GET_SBAS_CONFIG_REQ	174
3.33.2	Indication - QMI_LOC_GET_SBAS_CONFIG_IND	174
3.33.3	Description of QMI_LOC_GET_SBAS_CONFIG	176
3.34	QMI_LOC_SET_NMEA_TYPES	177
3.34.1	Request - QMI_LOC_SET_NMEA_TYPES_REQ	177
3.34.2	Indication - QMI_LOC_SET_NMEA_TYPES_IND	178
3.34.3	Description of QMI_LOC_SET_NMEA_TYPES	179
3.35	QMI_LOC_GET_NMEA_TYPES	181
3.35.1	Request - QMI_LOC_GET_NMEA_TYPES_REQ	181
3.35.2	Indication - QMI_LOC_GET_NMEA_TYPES_IND	181
3.35.3	Description of QMI_LOC_GET_NMEA_TYPES	183
3.36	QMI_LOC_SET_LOW_POWER_MODE	184
3.36.1	Request - QMI_LOC_SET_LOW_POWER_MODE_REQ	184
3.36.2	Indication - QMI_LOC_SET_LOW_POWER_MODE_IND	184
3.36.3	Description of QMI_LOC_SET_LOW_POWER_MODE	186
3.37	QMI_LOC_GET_LOW_POWER_MODE	187
3.37.1	Request - QMI_LOC_GET_LOW_POWER_MODE_REQ	187
3.37.2	Indication - QMI_LOC_GET_LOW_POWER_MODE_IND	187
3.37.3	Description of QMI_LOC_GET_LOW_POWER_MODE	189
3.38	QMI_LOC_SET_SERVER	190
3.38.1	Request - QMI_LOC_SET_SERVER_REQ	190
3.38.2	Indication - QMI_LOC_SET_SERVER_IND	191
3.38.3	Description of QMI_LOC_SET_SERVER	193
3.39	QMI_LOC_GET_SERVER	194
3.39.1	Request - QMI_LOC_GET_SERVER_REQ	194
3.39.2	Indication - QMI_LOC_GET_SERVER_IND	195
3.39.3	Description of QMI_LOC_GET_SERVER	198
3.40	QMI_LOC_DELETE_ASSIST_DATA	199
3.40.1	Request - QMI_LOC_DELETE_ASSIST_DATA_REQ	199
3.40.2	Indication - QMI_LOC_DELETE_ASSIST_DATA_IND	205
3.40.3	Description of QMI_LOC_DELETE_ASSIST_DATA	207
3.41	QMI_LOC_SET_XTRA_T_SESSION_CONTROL	208

3.41.1	Request - QMI_LOC_SET_XTRA_T_SESSION_CONTROL_REQ	208
3.41.2	Indication - QMI_LOC_SET_XTRA_T_SESSION_CONTROL_IND	208
3.41.3	Description of QMI_LOC_SET_XTRA_T_SESSION_CONTROL	210
3.42	QMI_LOC_GET_XTRA_T_SESSION_CONTROL	211
3.42.1	Request - QMI_LOC_GET_XTRA_T_SESSION_CONTROL_REQ	211
3.42.2	Indication - QMI_LOC_GET_XTRA_T_SESSION_CONTROL_IND	211
3.42.3	Description of QMI_LOC_GET_XTRA_T_SESSION_CONTROL	213
3.43	QMI_LOC_INJECT_WIFI_POSITION	214
3.43.1	Request - QMI_LOC_INJECT_WIFI_POSITION_REQ	214
3.43.2	Indication - QMI_LOC_INJECT_WIFI_POSITION_IND	218
3.43.3	Description of QMI_LOC_INJECT_WIFI_POSITION	219
3.44	QMI_LOC_NOTIFY_WIFI_STATUS	220
3.44.1	Request - QMI_LOC_NOTIFY_WIFI_STATUS_REQ	220
3.44.2	Indication - QMI_LOC_NOTIFY_WIFI_STATUS_IND	220
3.44.3	Description of QMI_LOC_NOTIFY_WIFI_STATUS	222
3.45	QMI_LOC_GET_REGISTERED_EVENTS	223
3.45.1	Request - QMI_LOC_GET_REGISTERED_EVENTS_REQ	223
3.45.2	Indication - QMI_LOC_GET_REGISTERED_EVENTS_IND	223
3.45.3	Description of QMI_LOC_GET_REGISTERED_EVENTS	229
3.46	QMI_LOC_SET_OPERATION_MODE	230
3.46.1	Request - QMI_LOC_SET_OPERATION_MODE_REQ	230
3.46.2	Indication - QMI_LOC_SET_OPERATION_MODE_IND	231
3.46.3	Description of QMI_LOC_SET_OPERATION_MODE	232
3.47	QMI_LOC_GET_OPERATION_MODE	233
3.47.1	Request - QMI_LOC_GET_OPERATION_MODE_REQ	233
3.47.2	Indication - QMI_LOC_GET_OPERATION_MODE_IND	233
3.47.3	Description of QMI_LOC_GET_OPERATION_MODE	235
3.48	QMI_LOC_SET_SPI_STATUS	236
3.48.1	Request - QMI_LOC_SET_SPI_STATUS_REQ	236
3.48.2	Indication - QMI_LOC_SET_SPI_STATUS_IND	237
3.48.3	Description of QMI_LOC_SET_SPI_STATUS	238
3.49	QMI_LOC_INJECT_SENSOR_DATA	239
3.49.1	Request - QMI_LOC_INJECT_SENSOR_DATA_REQ	239
3.49.2	Indication - QMI_LOC_INJECT_SENSOR_DATA_IND	244
3.49.3	Description of QMI_LOC_INJECT_SENSOR_DATA	247
3.50	QMI_LOC_INJECT_TIME_SYNC_DATA	248
3.50.1	Request - QMI_LOC_INJECT_TIME_SYNC_DATA_REQ	248
3.50.2	Indication - QMI_LOC_INJECT_TIME_SYNC_DATA_IND	249
3.50.3	Description of QMI_LOC_INJECT_TIME_SYNC_DATA	250
3.51	QMI_LOC_SET_CRADLE_MOUNT_CONFIG	251
3.51.1	Request - QMI_LOC_SET_CRADLE_MOUNT_CONFIG_REQ	251
3.51.2	Indication - QMI_LOC_SET_CRADLE_MOUNT_CONFIG_IND	252
3.51.3	Description of QMI_LOC_SET_CRADLE_MOUNT_CONFIG	253
3.52	QMI_LOC_GET_CRADLE_MOUNT_CONFIG	254
3.52.1	Request - QMI_LOC_GET_CRADLE_MOUNT_CONFIG_REQ	254
3.52.2	Indication - QMI_LOC_GET_CRADLE_MOUNT_CONFIG_IND	254
3.52.3	Description of QMI_LOC_GET_CRADLE_MOUNT_CONFIG	256
3.53	QMI_LOC_SET_EXTERNAL_POWER_CONFIG	257
3.53.1	Request - QMI_LOC_SET_EXTERNAL_POWER_CONFIG_REQ	257
3.53.2	Indication - QMI_LOC_SET_EXTERNAL_POWER_CONFIG_IND	258

3.53.3	Description of QMI_LOC_SET_EXTERNAL_POWER_CONFIG	259
3.54	QMI_LOC_GET_EXTERNAL_POWER_CONFIG	260
3.54.1	Request - QMI_LOC_GET_EXTERNAL_POWER_CONFIG_REQ	260
3.54.2	Indication - QMI_LOC_GET_EXTERNAL_POWER_CONFIG_IND	260
3.54.3	Description of QMI_LOC_GET_EXTERNAL_POWER_CONFIG	262
3.55	QMI_LOC_INFORM_LOCATION_SERVER_CONN_STATUS	263
3.55.1	Request - QMI_LOC_INFORM_LOCATION_SERVER_CONN_STATUS_REQ	263
3.55.2	Indication - QMI_LOC_INFORM_LOCATION_SERVER_CONN_STATUS_IND	265
3.55.3	Description of QMI_LOC_INFORM_LOCATION_SERVER_CONN_STATUS	267
3.56	QMI_LOC_SET_PROTOCOL_CONFIG_PARAMETERS	268
3.56.1	Request - QMI_LOC_SET_PROTOCOL_CONFIG_PARAMETERS_REQ	268
3.56.2	Indication - QMI_LOC_SET_PROTOCOL_CONFIG_PARAMETERS_IND	271
3.56.3	Description of QMI_LOC_SET_PROTOCOL_CONFIG_PARAMETERS	274
3.57	QMI_LOC_GET_PROTOCOL_CONFIG_PARAMETERS	275
3.57.1	Request - QMI_LOC_GET_PROTOCOL_CONFIG_PARAMETERS_REQ	275
3.57.2	Indication - QMI_LOC_GET_PROTOCOL_CONFIG_PARAMETERS_IND	276
3.57.3	Description of QMI_LOC_GET_PROTOCOL_CONFIG_PARAMETERS	280
3.58	QMI_LOC_SET_SENSOR_CONTROL_CONFIG	281
3.58.1	Request - QMI_LOC_SET_SENSOR_CONTROL_CONFIG_REQ	281
3.58.2	Indication - QMI_LOC_SET_SENSOR_CONTROL_CONFIG_IND	282
3.58.3	Description of QMI_LOC_SET_SENSOR_CONTROL_CONFIG	284
3.59	QMI_LOC_GET_SENSOR_CONTROL_CONFIG	285
3.59.1	Request - QMI_LOC_GET_SENSOR_CONTROL_CONFIG_REQ	285
3.59.2	Indication - QMI_LOC_GET_SENSOR_CONTROL_CONFIG_IND	285
3.59.3	Description of QMI_LOC_GET_SENSOR_CONTROL_CONFIG	287
3.60	QMI_LOC_SET_SENSOR_PROPERTIES	288
3.60.1	Request - QMI_LOC_SET_SENSOR_PROPERTIES_REQ	288
3.60.2	Indication - QMI_LOC_SET_SENSOR_PROPERTIES_IND	292
3.60.3	Description of QMI_LOC_SET_SENSOR_PROPERTIES	295
3.61	QMI_LOC_GET_SENSOR_PROPERTIES	296
3.61.1	Request - QMI_LOC_GET_SENSOR_PROPERTIES_REQ	296
3.61.2	Indication - QMI_LOC_GET_SENSOR_PROPERTIES_IND	298
3.61.3	Description of QMI_LOC_GET_SENSOR_PROPERTIES	303
3.62	QMI_LOC_SET_SENSOR_PERFORMANCE_CONTROL_CONFIGURATION	304
3.62.1	Request - QMI_LOC_SET_SENSOR_PERFORMANCE_CONTROL_CONFIGURATION_REQ	304
3.62.2	Indication - QMI_LOC_SET_SENSOR_PERFORMANCE_CONTROL_CONFIGURATION_IND	308
3.62.3	Description of QMI_LOC_SET_SENSOR_PERFORMANCE_CONTROL_CONFIGURATION	310
3.63	QMI_LOC_GET_SENSOR_PERFORMANCE_CONTROL_CONFIGURATION	311
3.63.1	Request - QMI_LOC_GET_SENSOR_PERFORMANCE_CONTROL_CONFIGURATION_REQ	311
3.63.2	Indication - QMI_LOC_GET_SENSOR_PERFORMANCE_CONTROL_CONFIGURATION_IND	311
3.63.3	Description of QMI_LOC_GET_SENSOR_PERFORMANCE_CONTROL_CONFIGURATION	317
3.64	QMI_LOC_INJECT_SUPL_CERTIFICATE	318
3.64.1	Request - QMI_LOC_INJECT_SUPL_CERTIFICATE_REQ	318
3.64.2	Indication - QMI_LOC_INJECT_SUPL_CERTIFICATE_IND	319

3.64.3	Description of QMI_LOC_INJECT_SUPL_CERTIFICATE	320
3.65	QMI_LOC_DELETE_SUPL_CERTIFICATE	321
3.65.1	Request - QMI_LOC_DELETE_SUPL_CERTIFICATE_REQ	321
3.65.2	Indication - QMI_LOC_DELETE_SUPL_CERTIFICATE_IND	321
3.65.3	Description of QMI_LOC_DELETE_SUPL_CERTIFICATE	323
3.66	QMI_LOC_SET_POSITION_ENGINE_CONFIG_PARAMETERS	324
3.66.1	Request - QMI_LOC_SET_POSITION_ENGINE_CONFIG_PARAMETERS_-REQ	324
3.66.2	Indication - QMI_LOC_SET_POSITION_ENGINE_CONFIG_PARAMETERS_-IND	326
3.66.3	Description of QMI_LOC_SET_POSITION_ENGINE_CONFIG_-PARAMETERS	328
3.67	QMI_LOC_GET_POSITION_ENGINE_CONFIG_PARAMETERS	329
3.67.1	Request - QMI_LOC_GET_POSITION_ENGINE_CONFIG_PARAMETERS_-REQ	329
3.67.2	Indication - QMI_LOC_GET_POSITION_ENGINE_CONFIG_PARAMETERS_-IND	330
3.67.3	Description of QMI_LOC_GET_POSITION_ENGINE_CONFIG_-PARAMETERS	333
3.68	QMI_LOC_EVENT_NI_GEOFENCE_NOTIFICATION	334
3.68.1	Indication - QMI_LOC_EVENT_NI_GEOFENCE_NOTIFICATION_IND	334
3.68.2	Description of QMI_LOC_EVENT_NI_GEOFENCE_NOTIFICATION	335
3.69	QMI_LOC_EVENT_GEOFENCE_GEN_ALERT	336
3.69.1	Indication - QMI_LOC_EVENT_GEOFENCE_GEN_ALERT_IND	336
3.69.2	Description of QMI_LOC_EVENT_GEOFENCE_GEN_ALERT	337
3.70	QMI_LOC_EVENT_GEOFENCE_BREACH_NOTIFICATION	338
3.70.1	Indication - QMI_LOC_EVENT_GEOFENCE_BREACH_NOTIFICATION_IND	338
3.70.2	Description of QMI_LOC_EVENT_GEOFENCE_BREACH_NOTIFICATION	341
3.71	QMI_LOC_ADD_CIRCULAR_GEOFENCE	342
3.71.1	Request - QMI_LOC_ADD_CIRCULAR_GEOFENCE_REQ	342
3.71.2	Indication - QMI_LOC_ADD_CIRCULAR_GEOFENCE_IND	345
3.71.3	Description of QMI_LOC_ADD_CIRCULAR_GEOFENCE	347
3.72	QMI_LOC_DELETE_GEOFENCE	348
3.72.1	Request - QMI_LOC_DELETE_GEOFENCE_REQ	348
3.72.2	Indication - QMI_LOC_DELETE_GEOFENCE_IND	349
3.72.3	Description of QMI_LOC_DELETE_GEOFENCE	350
3.73	QMI_LOC_QUERY_GEOFENCE	351
3.73.1	Request - QMI_LOC_QUERY_GEOFENCE_REQ	351
3.73.2	Indication - QMI_LOC_QUERY_GEOFENCE_IND	352
3.73.3	Description of QMI_LOC_QUERY_GEOFENCE	354
3.74	QMI_LOC_EDIT_GEOFENCE	355
3.74.1	Request - QMI_LOC_EDIT_GEOFENCE_REQ	355
3.74.2	Indication - QMI_LOC_EDIT_GEOFENCE_IND	357
3.74.3	Description of QMI_LOC_EDIT_GEOFENCE	359
3.75	QMI_LOC_GET_BEST_AVAILABLE_POSITION	360
3.75.1	Request - QMI_LOC_GET_BEST_AVAILABLE_POSITION_REQ	360
3.75.2	Indication - QMI_LOC_GET_BEST_AVAILABLE_POSITION_IND	360
3.75.3	Description of QMI_LOC_GET_BEST_AVAILABLE_POSITION	370
3.76	QMI_LOC_INJECT_MOTION_DATA	371
3.76.1	Request - QMI_LOC_INJECT_MOTION_DATA_REQ	371

3.76.2	Indication - QMI_LOC_INJECT_MOTION_DATA_IND	374
3.76.3	Description of QMI_LOC_INJECT_MOTION_DATA	376
3.77	QMI_LOC_GET_NI_GEOFENCE_ID_LIST	377
3.77.1	Request - QMI_LOC_GET_NI_GEOFENCE_ID_LIST_REQ	377
3.77.2	Indication - QMI_LOC_GET_NI_GEOFENCE_ID_LIST_IND	377
3.77.3	Description of QMI_LOC_GET_NI_GEOFENCE_ID_LIST	379
3.78	QMI_LOC_INJECT_GSM_CELL_INFO	380
3.78.1	Request - QMI_LOC_INJECT_GSM_CELL_INFO_REQ	380
3.78.2	Indication - QMI_LOC_INJECT_GSM_CELL_INFO_IND	381
3.78.3	Description of QMI_LOC_INJECT_GSM_CELL_INFO	382
3.79	QMI_LOC_INJECT_NETWORK_INITIATED_MESSAGE	383
3.79.1	Request - QMI_LOC_INJECT_NETWORK_INITIATED_MESSAGE_REQ	383
3.79.2	Indication - QMI_LOC_INJECT_NETWORK_INITIATED_MESSAGE_IND	384
3.79.3	Description of QMI_LOC_INJECT_NETWORK_INITIATED_MESSAGE	385
3.80	QMI_LOC_WWAN_OUT_OF_SERVICE_NOTIFICATION	386
3.80.1	Request - QMI_LOC_WWAN_OUT_OF_SERVICE_NOTIFICATION_REQ	386
3.80.2	Indication - QMI_LOC_WWAN_OUT_OF_SERVICE_NOTIFICATION_IND	386
3.80.3	Description of QMI_LOC_WWAN_OUT_OF_SERVICE_NOTIFICATION	388
3.81	QMI_LOC_EVENT_PEDOMETER_CONTROL	389
3.81.1	Indication - QMI_LOC_EVENT_PEDOMETER_CONTROL_IND	389
3.81.2	Description of QMI_LOC_EVENT_PEDOMETER_CONTROL	390
3.82	QMI_LOC_EVENT_MOTION_DATA_CONTROL	391
3.82.1	Indication - QMI_LOC_EVENT_MOTION_DATA_CONTROL_IND	391
3.82.2	Description of QMI_LOC_EVENT_MOTION_DATA_CONTROL	391
3.83	QMI_LOC_PEDOMETER_REPORT	392
3.83.1	Request - QMI_LOC_PEDOMETER_REPORT_REQ	392
3.83.2	Indication - QMI_LOC_PEDOMETER_REPORT_IND	394
3.83.3	Description of QMI_LOC_PEDOMETER_REPORT	395
3.84	QMI_LOC_INJECT_WCDMA_CELL_INFO	396
3.84.1	Request - QMI_LOC_INJECT_WCDMA_CELL_INFO_REQ	396
3.84.2	Indication - QMI_LOC_INJECT_WCDMA_CELL_INFO_IND	397
3.84.3	Description of QMI_LOC_INJECT_WCDMA_CELL_INFO	399
3.85	QMI_LOC_INJECT_TDSCDMA_CELL_INFO	400
3.85.1	Request - QMI_LOC_INJECT_TDSCDMA_CELL_INFO_REQ	400
3.85.2	Indication - QMI_LOC_INJECT_TDSCDMA_CELL_INFO_IND	401
3.85.3	Description of QMI_LOC_INJECT_TDSCDMA_CELL_INFO	403
3.86	QMI_LOC_INJECT_SUBSCRIBER_ID	404
3.86.1	Request - QMI_LOC_INJECT_SUBSCRIBER_ID	404
3.86.2	Indication - QMI_LOC_INJECT_SUBSCRIBER_ID_IND	404
3.86.3	Description of QMI_LOC_INJECT_SUBSCRIBER_ID	406
3.87	QMI_LOC_SET_GEOFENCE_ENGINE_CONFIG	407
3.87.1	Request - QMI_LOC_SET_GEOFENCE_ENGINE_CONFIG_REQ	407
3.87.2	Indication - QMI_LOC_SET_GEOFENCE_ENGINE_CONFIG_IND	412
3.87.3	Description of QMI_LOC_SET_GEOFENCE_ENGINE_CONFIG	414
3.88	QMI_LOC_GET_GEOFENCE_ENGINE_CONFIG	415
3.88.1	Request - QMI_LOC_GET_GEOFENCE_ENGINE_CONFIG_REQ	415
3.88.2	Indication - QMI_LOC_GET_GEOFENCE_ENGINE_CONFIG_IND	415
3.88.3	Description of QMI_LOC_GET_GEOFENCE_ENGINE_CONFIG	418
3.89	QMI_LOC_GET_BATCH_SIZE	419
3.89.1	Request - QMI_LOC_GET_BATCH_SIZE_REQ	419

3.89.2	Indication - QMI_LOC_GET_BATCH_SIZE_IND	420
3.89.3	Description of QMI_LOC_GET_BATCH_SIZE	421
3.90	QMI_LOC_START_BATCHING	422
3.90.1	Request - QMI_LOC_START_BATCHING_REQ	422
3.90.2	Indication - QMI_LOC_START_BATCHING_IND	423
3.90.3	Description of QMI_LOC_START_BATCHING	425
3.91	QMI_LOC_EVENT_BATCH_FULL_NOTIFICATION	426
3.91.1	Indication - QMI_LOC_EVENT_BATCH_FULL_NOTIFICATION_IND	426
3.91.2	Description of QMI_LOC_EVENT_BATCH_FULL_NOTIFICATION	427
3.92	QMI_LOC_EVENT_LIVE_BATCHED_POSITION_REPORT	428
3.92.1	Indication - QMI_LOC_EVENT_LIVE_BATCHED_POSITION_REPORT_IND	428
3.92.2	Description of QMI_LOC_EVENT_LIVE_BATCHED_POSITION_REPORT	432
3.93	QMI_LOC_READ_FROM_BATCH	433
3.93.1	Request - QMI_LOC_READ_FROM_BATCH_REQ	433
3.93.2	Indication - QMI_LOC_READ_FROM_BATCH_IND	434
3.93.3	Description of QMI_LOC_READ_FROM_BATCH	440
3.94	QMI_LOC_STOP_BATCHING	441
3.94.1	Request - QMI_LOC_STOP_BATCHING_REQ	441
3.94.2	Indication - QMI_LOC_STOP_BATCHING_IND	441
3.94.3	Description of QMI_LOC_STOP_BATCHING	443
3.95	QMI_LOC_RELEASE_BATCH	444
3.95.1	Request - QMI_LOC_RELEASE_BATCH_REQ	444
3.95.2	Indication - QMI_LOC_RELEASE_BATCH_IND	444
3.95.3	Description of QMI_LOC_RELEASE_BATCH	446
3.96	QMI_LOC_EVENT_INJECT_WIFI_AP_DATA_REQ	447
3.96.1	Indication - QMI_LOC_EVENT_INJECT_WIFI_AP_DATA_REQ_IND	447
3.96.2	Description of QMI_LOC_EVENT_INJECT_WIFI_AP_DATA_REQ	447
3.97	QMI_LOC_INJECT_WIFI_AP_DATA	448
3.97.1	Request - QMI_LOC_INJECT_WIFI_AP_DATA_REQ	448
3.97.2	Indication - QMI_LOC_INJECT_WIFI_AP_DATA_IND	450
3.97.3	Description of QMI_LOC_INJECT_WIFI_AP_DATA	451
3.98	QMI_LOC_NOTIFY_WIFI_ATTACHMENT_STATUS	452
3.98.1	Request - QMI_LOC_NOTIFY_WIFI_ATTACHMENT_STATUS_REQ	452
3.98.2	Request - QMI_LOC_NOTIFY_WIFI_ATTACHMENT_STATUS_IND	453
3.98.3	Description of QMI_LOC_NOTIFY_WIFI_ATTACHMENT_STATUS REQ/RESP	454
3.99	QMI_LOC_NOTIFY_WIFI_ENABLED_STATUS	455
3.99.1	Request - QMI_LOC_NOTIFY_WIFI_ENABLED_STATUS_REQ	455
3.99.2	Indication - QMI_LOC_NOTIFY_WIFI_ENABLED_STATUS_IND	456
3.99.3	Description of QMI_LOC_NOTIFY_WIFI_ENABLED_STATUS	457
3.100	QMI_LOC_EVENT_GEOFENCE_BATCHED_BREACH_NOTIFICATION	458
3.100.1	Indication - QMI_LOC_EVENT_GEOFENCE_BATCHED_BREACH_NOTIFICATION_IND	458
3.100.2	Description of QMI_LOC_EVENT_GEOFENCE_BATCHED_BREACH_NOTIFICATION	463
3.101	QMI_LOC_EVENT_VEHICLE_DATA_READY_STATUS	464
3.101.1	Indication - QMI_LOC_EVENT_VEHICLE_DATA_READY_STATUS_IND	464
3.101.2	Description of QMI_LOC_EVENT_VEHICLE_DATA_READY_STATUS	465
3.102	QMI_LOC_INJECT_VEHICLE_SENSOR_DATA	466
3.102.1	Request - QMI_LOC_INJECT_VEHICLE_SENSOR_DATA_REQ	466
3.102.2	Indication - QMI_LOC_INJECT_VEHICLE_SENSOR_DATA_IND	472

3.102.3	Description of QMI_LOC_INJECT_VEHICLE_SENSOR_DATA	474
3.103	QMI_LOC_GET_AVAILABLE_WWAN_POSITION	475
3.103.1	Request - QMI_LOC_GET_AVAILABLE_WWAN_POSITION_REQ	475
3.103.2	Indication - QMI_LOC_GET_AVAILABLE_WWAN_POSITION_IND	475
3.103.3	Description of QMI_LOC_GET_AVAILABLE_WWAN_POSITION	482
3.104	QMI_LOC_SET_PREMIUM_SERVICES_CONFIG	483
3.104.1	Request - QMI_LOC_SET_PREMIUM_SERVICES_CONFIG_REQ	483
3.104.2	Indication - QMI_LOC_SET_PREMIUM_SERVICES_CONFIG_IND	484
3.105	QMI_LOC_SET_XTRA_VERSION_CHECK	486
3.105.1	Request - QMI_LOC_SET_XTRA_VERSION_CHECK_REQ	486
3.105.2	Indication - QMI_LOC_SET_XTRA_VERSION_CHECK_IND	487
3.105.3	Description of QMI_LOC_SET_XTRA_VERSION_CHECK	488
3.106	QMI_LOC_EVENT_GNSS_MEASUREMENT_REPORT_IND	489
3.106.1	Indication - QMI_LOC_EVENT_GNSS_MEASUREMENT_REPORT_IND	489
3.106.2	Description of QMI_LOC_EVENT_GNSS_MEASUREMENT_REPORT_IND	502
3.107	QMI_LOC_EVENT_SV_POLYNOMIAL_REPORT_IND	503
3.107.1	Indication - QMI_LOC_EVENT_SV_POLYNOMIAL_REPORT_IND	503
3.107.2	Description of QMI_LOC_EVENT_SV_POLYNOMIAL_REPORT_IND	507
3.108	QMI_LOC_SET_GNSS_CONSTELL_REPORT_CONFIG	508
3.108.1	Request - QMI_LOC_SET_GNSS_CONSTELL_REPORT_CONFIG	508
3.108.2	Indication - QMI_LOC_SET_GNSS_CONSTELL_REPORT_CONFIG_IND	509
3.108.3	Description of QMI_LOC_SET_GNSS_CONSTELL_REPORT_CONFIG	510
3.109	QMI_LOC_ADD_GEOFENCE_CONTEXT	511
3.109.1	QMI_LOC_EVENT_SV_POLYNOMIAL_REPORT_IND	511
3.109.2	Indication - QMI_LOC_ADD_GEOFENCE_CONTEXT_IND	514
3.109.3	Description of QMI_LOC_ADD_GEOFENCE_CONTEXT	516
3.110	QMI_LOC_SET_GEOFENCE_ENGINE_CONTEXT	517
3.110.1	- QMI_LOC_ADD_GEOFENCE_CONTEXT_IND	517
3.110.2	Indication - QMI_LOC_SET_GEOFENCE_ENGINE_CONTEXT_IND	518
3.110.3	Description of QMI_LOC_SET_GEOFENCE_ENGINE_CONTEXT	519
3.111	QMI_LOC_DELETE_GEOFENCE_CONTEXT	520
3.111.1	QMI_LOC_DELETE_GEOFENCE_CONTEXT_IND	520
3.111.2	Indication - QMI_LOC_DELETE_GEOFENCE_CONTEXT_IND	521
3.111.3	Description of QMI_LOC_DELETE_GEOFENCE_CONTEXT	523
3.112	QMI_LOC_EVENT_GEOFENCE_PROXIMITY - NOTIFICATION	524
3.112.1	Indication - QMI_LOC_EVENT_GEOFENCE_PROXIMITY_NOTIFICATION - IND	524
3.112.2	Description of QMI_LOC_EVENT_GEOFENCE_PROXIMITY_NOTIFICATION	525

List of Figures

2-1	MS-assisted call flow example	26
2-2	MS-based call flow example	27
2-3	Cell ID-based call flow example	27
2-4	Call flow for configuration, request, and injection of sensor and time sync data	32
2-5	Orientation of coordinate axes for acceleration measurements	33
2-6	Right-hand rule	34
2-7	Batching software architecture	36

List of Tables

1-1	Reference documents and standards	18
1-2	Acronyms	19
3-1	QMI_LOC messages	37

Revision History

Revision	Date	Description
A	Nov 2012	<p>Initial release. Created from 80-VB816-17 H.</p> <p>Updates for this revision include minor version 15 and minor version 16.</p> <p>Updated mandatory TLV Inject Motion Data Request Status (Section 3.74.2)</p> <p>Added new messages:</p> <ul style="list-style-type: none"> • QMI_LOC_INJECT_GSM_CELL_INFO (Section 3.76) • QMI_LOC_INJECT_NETWORK_INITIATED_MESSAGE (Section 3.77) • QMI_LOC_WWAN_OUT_OF_SERVICE_NOTIFICATION (Section 3.78)
B	Jan 2013	<p>Updates for this revision include minor version 17 and minor version 18.</p> <p>Updated:</p> <ul style="list-style-type: none"> • TLV Event Registration Mask (Sections 3.3.1 and 3.43.2) • TLV Time Source (Sections 3.6.1 and 3.73.2) • Description of QMI_LOC_INJECT_SENSOR_DATA (Section 3.47.3) • TLV Assisted GLONASS Protocol Mask (Sections 3.54.1 and 3.55.2) • TLV Failed Parameters (Section 3.54.2) • TLV Config Parameters (Section 3.55.1) <p>Added new TLVs:</p> <ul style="list-style-type: none"> • SUPL Emergency Notification (Sections 3.9.1 and 3.22.1) • Accelerometer Temperature Accept Ready (Section 3.16.1) • Gyroscope Temperature Accept Ready (Section 3.16.1) • 3-Axis Accelerometer Data Time Source (Section 3.47.1) • 3-Axis Gyroscope Data Time Source (Section 3.47.1) • Accelerometer Temperature Data (Section 3.47.1) • Gyroscope Temperature Data (Section 3.47.1) • Accelerometer Temperature Samples Accepted (Section 3.47.2) • Gyroscope Temperature Samples Accepted (Section 3.47.2) • SUPL Hash Algorithm (Sections 3.54.1 and 3.55.2) • SUPL TLS Version (Sections 3.54.1 and 3.55.2) • Emergency Protocol (Sections 3.54.1 and 3.55.2) • Timing Advance (Section 3.76.1) <p>Added new Messages:</p> <ul style="list-style-type: none"> • QMI_LOC_EVENT_PEDOMETER_CONTROL (Section 3.79) • QMI_LOC_EVENT_MOTION_DATA_CONTROL (Section 3.80) • QMI_LOC_PEDOMETER_REPORT (Section 3.81) • QMI_LOC_INJECT_WCDMA_CELL_INFO (Section 3.82) • QMI_LOC_INJECT_TDSCDMA_CELL_INFO (Section 3.83) • QMI_LOC_INJECT_SUBSCRIBER_ID (Section 3.84)

Revision	Date	Description
C	Mar 2013	<p>Updates for this revision include minor version 19 and minor version 20.</p> <p>Updated Section 2.3.1</p> <p>Updated TLVs:</p> <ul style="list-style-type: none"> • Technology Used (Section 3.8.1) • WWAN Type (Section 3.21.1) • Failed Parameters (Section 3.66.2) • Technology Used Mask (Section 3.75.2) <p>Added new TLVs:</p> <ul style="list-style-type: none"> • Altitude Assumed (Section 3.8.1) • Enable Faster TTFF (Sections 3.66.1 and 3.67.2) <p>Added Common Messages:</p> <ul style="list-style-type: none"> • QMI_LOC_GET_SUPPORTED_MESSAGES (Section 3.2) • QMI_LOC_GET_SUPPORTED_FIELDS (Section 3.3)
D	Nov 2013	<p>Updates for this revision include minor version 21 through minor version 24.</p> <p>Updated TLVs:</p> <ul style="list-style-type: none"> • Several Status TLVs (multiple sections) • Event Registration Mask (Sections 3.5.1 and 3.45.2) • Time Source (Sections 3.8.1 and 3.75.2) • SVs Used to Calculate the Fix (Sections 3.8.1 and 3.75.2) • Satellite Info (Section 3.9.1) • Delete SV Info (Section 3.40.1) • Delete GNSS Data (Section 3.40.1) • Delete Clock Info (Section 3.40.1) • Assisted GLONASS Protocol Mask (Sections 3.56.1 and 3.57.2) • Failed Parameters (Section 3.56.2) • Config Parameters (Section 3.57.1) • Failed Set Sensor Properties (Section 3.60.2) • Sensor Properties Config Parameters (Section 3.61.1) <p>Added new TLVs:</p> <ul style="list-style-type: none"> • Delete BDS SV Info (Section 3.40.1) • Wi-Fi Scan Injection Timeout Period (Sections 3.56.1 and 3.57.2) • Vehicle Data Use Control (Sections 3.60.1 and 3.61.2) • Vehicle Velocity Random Walk Spectral Density (Sections 3.60.1 and 3.61.2) • Vehicle Acceleration Random Walk Spectral Density (Sections 3.60.1 and 3.61.2) • Vehicle Angle Random Walk Spectral Density (Section 3.60.1s and 3.61.2) • Vehicle Angular Rate Random Walk Spectral Density (Sections 3.60.1 and 3.61.2) • Vehicle Odometry Scale Factor Random Walk Spectral Density (Sections 3.60.1 and 3.61.2) • Vehicle Odometry Variance (Sections 3.60.1 and 3.61.2) • Geofence Breach Confidence (Section 3.70.1) • Responsiveness (Section 3.74.1)

Revision	Date	Description
D (cont.)	Nov 2013 (cont.)	<p>Added new Messages:</p> <ul style="list-style-type: none"> • QMI_LOC_SET_GEOFENCE_ENGINE_CONFIG (Section 3.87) • QMI_LOC_GET_GEOFENCE_ENGINE_CONFIG (Section 3.88) • QMI_LOC_GET_BATCH_SIZE (Section 3.89) • QMI_LOC_START_BATCHING (Section 3.90) • QMI_LOC_BATCH_FULL_NOTIFICATION (Section 3.91) • QMI_LOC_EVENT_LIVE_BATCHED_POSITION_REPORT (Section 3.92) • QMI_LOC_READ_FROM_BATCH (Section 3.93) • QMI_LOC_STOP_BATCHING (Section 3.94) • QMI_LOC_RELEASE_BATCH (Section 3.95) • QMI_LOC_EVENT_INJECT_WIFI_AP_DATA_REQ (Section 3.96) • QMI_LOC_INJECT_WIFI_AP_DATA (Section 3.97) • QMI_LOC_NOTIFY_WIFI_ATTACHMENT_STATUS (Section 3.98) • QMI_LOC_NOTIFY_WIFI_ENABLED_STATUS (Section 3.99) • QMI_LOC_EVENT_GEOFENCE_BATCHED_BREACH_NOTIFICATION (Section 3.100) • QMI_LOC_EVENT_VEHICLE_DATA_READY_STATUS (Section 3.101) • QMI_LOC_INJECT_VEHICLE_SENSOR_DATA (Section 3.102)
E	Aug 2014	<p>Updates for this revision include minor version 25 through minor version 32.</p> <p>Updates for minor version 25.</p> <p>Added Section 2.9.12, Location Fix Batching.</p> <p>Updated TLV NMEA Sentence Types (Sections 3.34.1 and 3.35.2).</p> <p>Added new TLVs:</p> <ul style="list-style-type: none"> • Configuration for Altitude Assumed Info in GNSS SV Info Event (Section 3.6.1) • Sensors Provider (Sections 3.58.1 and 3.59.2) <p>Updates for minor version 26 through minor version 28.</p> <p>Updated TLVs:</p> <ul style="list-style-type: none"> • Responsiveness (Section 3.71.1) • Multiple Status TLVs were updated <p>Added new TLVs:</p> <ul style="list-style-type: none"> • Fix Session Timeout Period (Section 3.90.1) • Heading Uncertainty (Section 3.100.1) • Vertical Uncertainty (Section 3.100.1) • Speed Uncertainty (Section 3.100.1) • Horizontal Confidence (Section 3.100.1) • Vertical Confidence (Section 3.100.1) • Dilution of Precision (Section 3.100.1) • SVs Used to Calculate the Fix (Section 3.100.1) <p>Added new Messages:</p> <ul style="list-style-type: none"> • QMI_LOC_GET_AVAILABLE_WWAN_POSITION (Section 3.103) • QMI_LOC_SET_PREMIUM_SERVICES_CONFIG (Section 3.104) • QMI_LOC_SET_XTRA_VERSION_CHECK (Section 3.105)

Revision	Date	Description
E (cont.)	Aug 2014 (cont.)	<p>Updates for minor version 29 through minor version 32.</p> <p>Updated TLVs:</p> <ul style="list-style-type: none"> • Event Registration Mask (Sections 3.5.1 and 3.45.2) • Delete GNSS Data (Section 3.40.1) <p>Added new TLVs:</p> <ul style="list-style-type: none"> • Raw Circular Horizontal Uncertainty (Section 3.29.1) • Raw Horizontal Confidence (Section 3.29.1) • Raw HEPE (Section 3.43.1) • Wi-Fi AP SSID String (Sections 3.43.1 and 3.98.1) • GNSS Position Maximum Position Uncertainty Acceptable (Section 3.87.1) • Medium Responsiveness Value (Section 3.87.1) • Challenging GNSS Environment Minimum CPI Wait Interval (Section 3.87.1) • Geofence Motion State Information (Section 3.87.1) <p>Added new Messages:</p> <ul style="list-style-type: none"> • QMI_LOC_EVENT_GNSS_MEASUREMENT_REPORT_IND (Section 3.106) • QMI_LOC_EVENT_SV_POLYNOMIAL_REPORT_IND (Section 3.107) • QMI_LOC_SET_GNSS_CONSTELL_REPORT_CONFIG (Section 3.108) • QMI_LOC_ADD_GEOFENCE_CONTEXT (Section 3.109) • QMI_LOC_SET_GEOFENCE_ENGINE_CONTEXT (Section 3.110) • QMI_LOC_DELETE_GEOFENCE_CONTEXT (Section 3.111) • QMI_LOC_EVENT_GEOFENCE_PROXIMITY_NOTIFICATION (Section 3.112)

1 Introduction

1.1 Purpose

This specification documents Major Version 2 of the Qualcomm Messaging Interface Location Service (QMI_LOC). QMI_LOC provides location/position determination services on Qualcomm MSM™ and MDM devices.

QMI_LOC begins with Major Version 2 for two reasons. First, because QMI_LOC is built upon and supersedes an older position determination service, which was Major Version 1. Second, 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_LOC provides applications running on a tethered device or on the HLOS's side of a dual processor MSM device 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 – Chapter 2 provides the theory of operation of QMI_LOC. The chapter includes messaging conventions, assigned QMI service type, fundamental service concepts, and state variables related to the service.
- Message formats, syntax, and semantics – Chapter 3 provides the specific syntax and semantics of messages included in this version of the QMI_LOC specification.

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

Reference documents are listed in Table 1-1. Reference documents that are no longer applicable are deleted from this table; therefore, reference numbers may not be sequential.

Table 1-1 Reference documents and standards

Ref.	Document	
Qualcomm Technologies		
Q1	Application Note: Software Glossary for Customers	CL93-V3077-1
Q2	Qualcomm MSM Interface (QMI) Architecture	80-VB816-1
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
S5	UserPlane Location Protocol	OMA-TS-ULP-V2_0-20110527-A (Apr 2012)
S6	The international identification plan for public works and subscriptions	Recommendation ITU-T E.212
S7	Radio Resource Control (RRC); Protocol specification	3GPP TS 25.331
S8	Radio subsystem synchronization	3GPP TS 05.10
S9	Radio subsystem synchronization	3GPP TS 45.010
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 document, submit a case to Qualcomm Technologies at <https://support.cdmatech.com>.

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

1.6 Acronyms

For definitions of terms and abbreviations, refer to [Q1]. Table 1-2 lists terms that are specific to this document.

Table 1-2 Acronyms

Acronym	Definition
AFLT	advanced forward link trilateration
AGNSS	assisted GNSS
AP	access point
APN	access point name
APQ	application-only processor – Qualcomm
BDS	BeiDou Navigation Satellite System (a Chinese satellite navigation system)
CP	control point or control plane
DOP	dilution of precision
DS-DS	dual service - dual standby
ECID	exclusive chip ID
EOTD	enhanced observed time difference
ESLP	emergency SUPL location platform
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
IMSI	international mobile subscriber identity
ITU-T	International Telecommunication Union - Telecommunication Standardization Sector
LCS	location services
LOC	location
LPM	low power mode
LPP	LTE Positioning Protocol
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
MSISDN	mobile subscriber integrated services digital network number
MT	mobile-terminated
NI	network initiated
NMEA	National Marine Electronics Association
OMA	Open Mobile Alliance

Table 1-2 Acronyms (cont.)

Acronym	Definition
OTDOA	observed time delay of arrival
PDE	position determination entity
PDOP	position dilution of precision
PQXFI	proprietary Qualcomm extended fix information
PRN	pseudorandom noise
PSTIS	proprietary Snaptracks, Inc. session
QMI	Qualcomm messaging interface
QoP	quality of position
QoS	quality of service
QZSS	quasi-zenith satellite system
RAT	radio access technology
RMC	recommended minimum specific GPS/transit data
RRLP	Radio Resources LCS Protocol
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
TDSCDMA	time division synchronous CDMA
TLS	transport layer security
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, result, and error code values described in the QMI Generalized Message Protocol section of [Q2]. Extensions to the generalized QMI service theory of operation are noted in subsequent sections of this chapter.

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

LOC is assigned QMI service type 0x10.

2.3 Message Definition Template

2.3.1 Response Message Result TLV

All response messages returned by the QMI_LOC service are identical. All of them contain the mandatory Type-Length-Value (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 introduced	Version last modified
Result Code	Corresponding response's <i>Version introduced</i>	Corresponding response's <i>Version last modified</i>

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x02			1	Result Code
Length	4			2	
Value	→	uint16	qmi_result	2	Result code • QMI_RESULT_SUCCESS • QMI_RESULT_FAILURE
		uint16	qmi_error	2	Error code – Possible error code values are described in the error codes 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

The primary features of the QMI_LOC API are as follows:

- Acknowledgment of receipt of REQ messages 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.
- Each different asynchronous event is output with its own QMI message to allow for future extensibility.
- QMI_LOC has 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 from the previous position determination service to allow more freedom for QMI_LOC to choose the optimal positioning method "under the hood."
- Fix criteria is local to a client in QMI_LOC (with limitations to be described later in this document).
- Fix criteria is (an optional) part of the START request in QMI_LOC.
- Positioning mode is 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. Refer to [R1] for a detailed explanation.

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

2.9.2 Position Determination Methods

The location engine supports five methods for determining the position: MS-assisted, MS-based, Cell ID-based, Standalone, and WWAN-based. The primary difference between these methods is in how the final position calculations are performed. An MS-assisted fix is one in which a network entity (PDE or PDM) does the final position calculations. In an MS-based fix, the final position is calculated by the location engine, but the location engine may get additional assistance from the PDE/PDM. Cell ID-based positioning uses the current geographic site as a seed to find the current location. A standalone fix is one in which the entire position fix process is done within the location engine without interaction with the PDE/PDM. For WWAN-based positioning, the location engine uses WWAN measurement information to compute a location. Each has advantages and disadvantages. For example, MS-assisted fixes are usually capable of obtaining position information in harsher environments, such as indoors, at the expense of greater network traffic and a longer time-to-fix.

The following subsections describe these methods and provide example MS-assisted, MS-based, and Cell ID-based call flows. Since standalone and WWAN do not interact with the network (PDE, PDM, etc.), these have no call flows.

2.9.2.1 MS-assisted PD

In MS-assisted PD, the MS assists a PDE in determining the position. The device communicates with the PDE to get satellite acquisition assistance data to assist it in performing satellite pseudorange measurements. These measurements are performed and then sent to the PDE, where the MS position is calculated and returned to the MS. This procedure is repeated each time the MS position is requested, and for each fix, the location engine must communicate with the PDE over TCP/IP or via a control channel.

2.9.2.1.1 MS-assisted Call Flow

Figure 2-1 illustrates the steps required to perform an MS-assisted PD, where the MS position is calculated by the PDE.

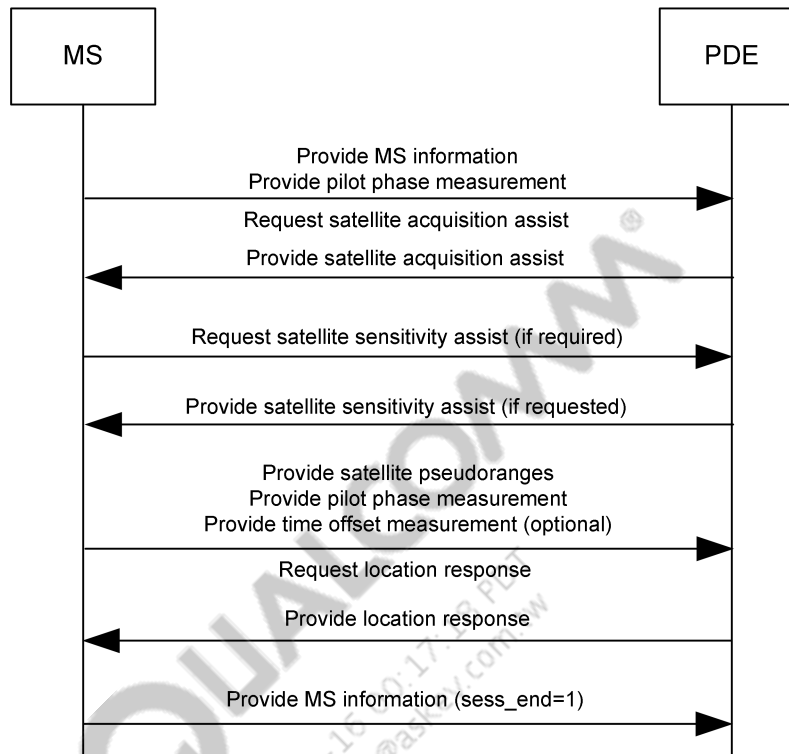


Figure 2-1 MS-assisted call flow example

2.9.2.1.2 Client Request

The client request for MS-assisted fixes is performed in the following sequence:

1. The client first selects the operation mode by sending in the message QMI_LOC_SET_OPERATION_MODE (`operationMode=0x00000003`).
2. The PD process begins when the client application sends the QMI_LOC_START message.

2.9.2.2 MS-based PD

In MS-based PD, the MS communicates with a PDE to acquire almanac and ephemeris information, which it then uses to generate satellite acquisition assistance. As in MS-assisted PD, the MS uses the ephemeris, reference time, and position to compute a final position.

To compute the position locally, the MS must also begin with a coarse estimate of its current location. This seed position is often obtained by performing an MS-assisted PD session prior to beginning an MS-based operation. Because the coarse position must be obtained prior to beginning an MS-based session, MS-based sessions are most useful when performed in the context of position tracking applications where the MS position is determined at regular intervals.

2.9.2.2.1 MS-based Call Flow

Figure 2-2 illustrates an example of an MS-based call flow.

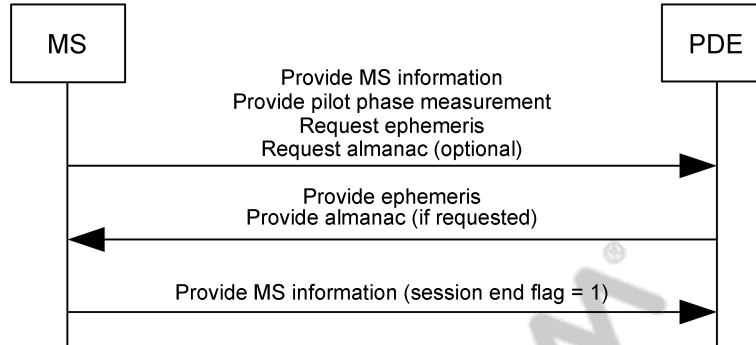


Figure 2-2 MS-based call flow example

2.9.2.2.2 Client Request

The client request for MS-based fixes is performed in the following sequence:

1. The client first selects the operation mode by sending in the message QMI_LOC_SET_OPERATION_MODE ([operationMode=0x00000002](#)).
2. The PD process begins when the client application sends the QMI_LOC_START message.

2.9.2.3 Cell ID-based PD

Cell ID-based positioning uses the current geographic site as a seed to find the current location. The location engine allows this positioning source when the phone is configured for User Plane MO method and, in that case, will perform a SUPL call flow to get a cell ID position. The Set ID Capabilities approach is used to get the Cell ID, i.e., the SUPL Start message sends a NULL in the Set Capabilities field, signaling the SLP to respond with a SUPL End message containing the Cell ID position.

2.9.2.3.1 Cell ID-based Call Flow

Figure 2-3 illustrates an example of a Cell ID-based call flow.

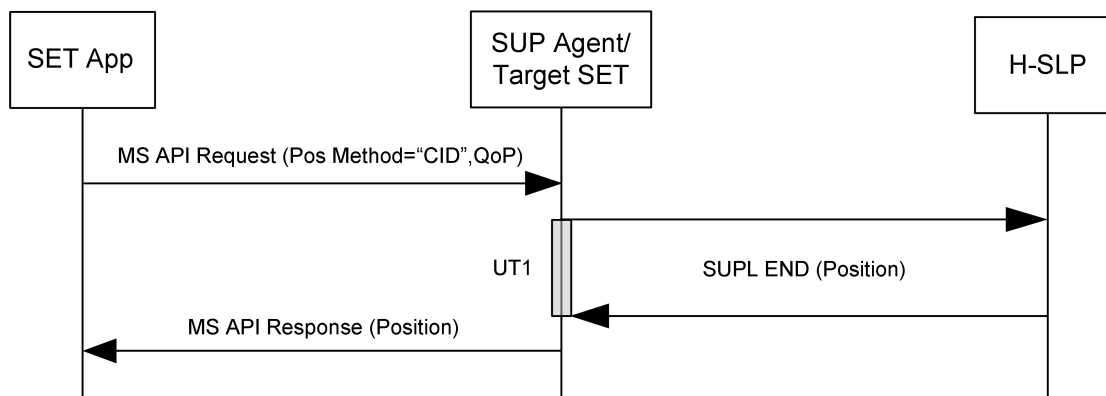


Figure 2-3 Cell ID-based call flow example

2.9.2.3.2 Client Request

The client request for Cell ID-based fixes is performed in the following sequence:

1. The client first selects the operation mode by sending in the message QMI_LOC_SET_OPERATION_MODE (*operationMode*=0x000005).
2. The PD process begins when the client application sends the QMI_LOC_START message.

2.9.2.4 Standalone Fix

A standalone fix is one in which the entire position fix calculation process is done within the location engine. This mode is particularly useful in applications where no PDE/PDM is available, or when network data traffic is to be minimized.

2.9.2.4.1 Client Request

The client request for standalone fixes is performed in the following sequence:

1. The client first selects the operation mode by sending in the message QMI_LOC_SET_OPERATION_MODE (*operationMode*=0x00000004).
2. The PD process begins when the client application sends the QMI_LOC_START message.

2.9.2.5 WWAN-based PD

In WWAN-based positioning, the location engine uses the WWAN measurements to compute position fixes. This mode is useful for cases where less power consumption is desired. The fixes generated when this mode is selected are less accurate than the modes where GNSS measurements are used to generate position fixes.

2.9.2.5.1 Client Request

The client request for WWAN-based fixes is performed in the following sequence:

1. The client first selects the operation mode by sending in the message QMI_LOC_SET_OPERATION_MODE (*operationMode*=0x000006).
2. The PD process begins when the client application sends the QMI_LOC_START message.

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

A QMI control point can inject time information to the GPS service.

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 ≤ 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 2-4 illustrates the call flow sequence for configuration, request, and injection of sensor and time sync data.

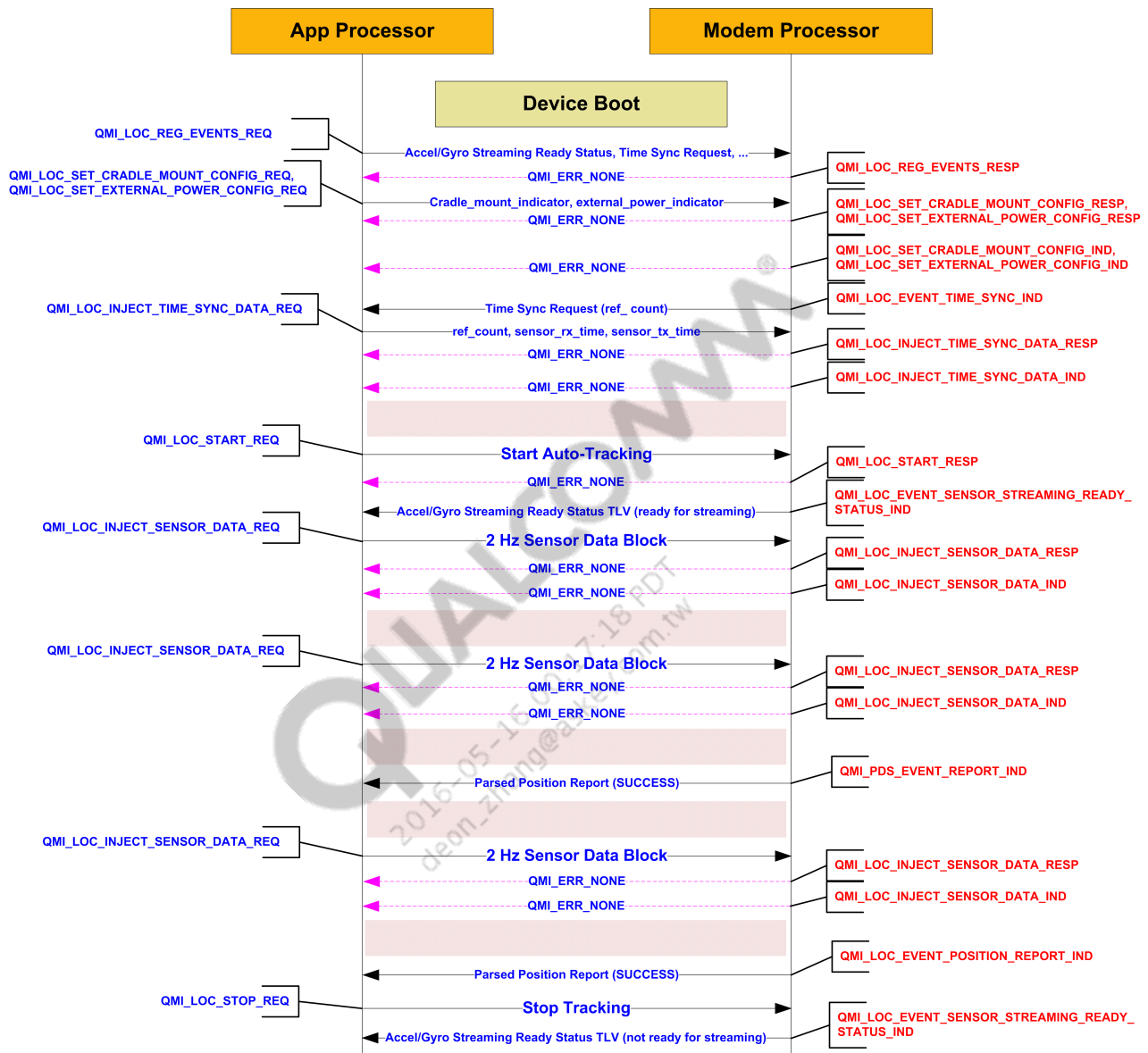


Figure 2-4 Call flow for configuration, request, and injection of sensor and time sync data

Figure 2-5 illustrates the coordinate axes orientation for acceleration measurements.

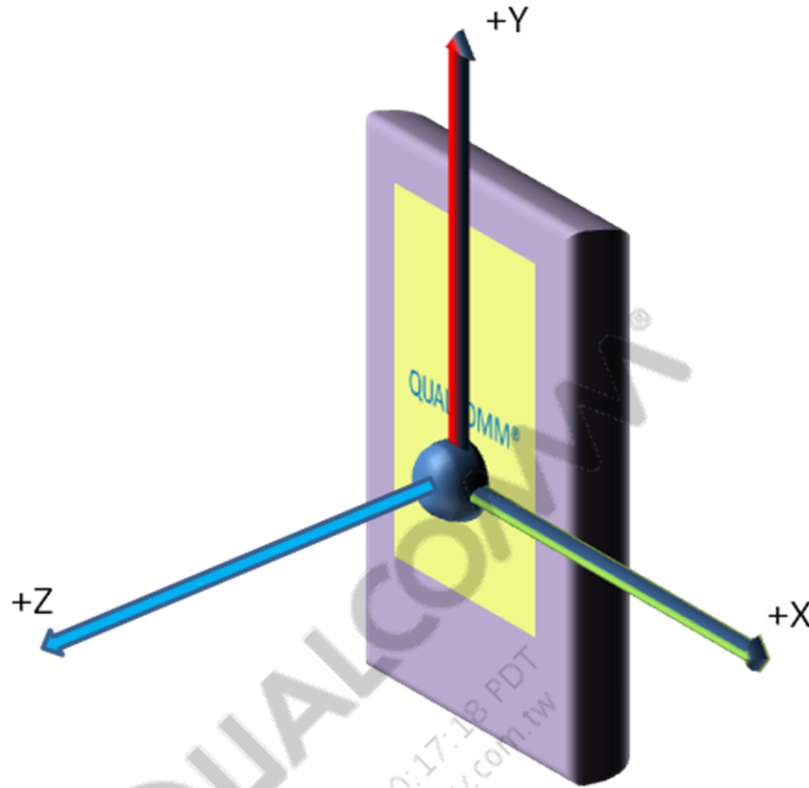


Figure 2-5 Orientation of coordinate axes for acceleration measurements

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 2-5 is approximately $+9.81 \text{ m/s}^2$ 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 2-6 illustrates positive rotation the right-hand rule.



Figure 2-6 Right-hand rule

If the gyroscope 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.

2.9.12 Location Fix Batching

2.9.12.1 Batching Sessions

The QMI_LOC control point can request the service to store position fixes without notifying the control point for each position fix that is generated. The service generates the position fixes and stores each fix in its internal buffer. This is a batching session. The control point configures the batch size, reporting interval, accuracy, batch full notifications, and live fix notifications using the APIs to the service.

The control point sends a QMI_LOC_GET_BATCH_SIZE_REQ request to allocate the buffer at the service. The request specifies the number of position fixes to be buffered. The service allocates the memory for the requested batch size, or a lesser value based on memory availability. It is the responsibility of the control point to configure the appropriate value of the batch size depending on the memory configuration on the product and the reporting interval that is being requested. If a control point chooses a large batch size value, any other control point requesting the batching operation, as well as overall system performance, may be affected.

2.9.12.2 Event Notification Registration

The control point can register for the QMI_LOC_EVENT_MASK_BATCH_FULL_NOTIFICATION event mask to receive notifications when the batching buffer is full. The service notifies the control point with a Batch Full indication event (QMI_LOC_EVENT_BATCH_FULL_NOTIFICATION_IND) if the control point registers for it. A client can handle this notification from the service by retrieving the location fixes from the batch using the QMI_LOC_READ_FROM_BATCH_REQ message.

The control point can also register for the QMI_LOC_EVENT_MASK_LIVE_BATCHED_POSITION_REPORT event mask to receive live batched position reports. The service notifies the control point with the live position report event QMI_LOC_EVENT_LIVE_BATCHED_POSITION_REPORT_IND if the control point registers for it.

2.9.12.3 Batching Operations

The control point sends a QMI_LOC_START_BATCHING_REQ request to the service to choose the fix criteria for generating the fixes for the batching operation. Upon receiving this request, the service starts generating the position fixes and stores each position fix in the batching buffer. The control point can use the same request command to change the criteria of an ongoing batching session.

The control point sends a QMI_LOC_READ_FROM_BATCH_REQ request to the service to read the location fix entries from the batching buffer. The control point can send this request to retrieve the entries asynchronously or when the service sends the event

QMI_LOC_EVENT_MASK_BATCH_FULL_NOTIFICATION_IND. If the control point sends this request when the batch is already empty, the service returns an indication with no entries. If the batch at the service has fewer entries than the requested value from the control point, the service only sends the fix entries that are available in the batch. The number of entries that can be retrieved from the service in each attempt is limited to QMI_LOC_READ_FROM_BATCH_MAX_SIZE entries.

The control point sends a QMI_LOC_STOP_BATCHING_REQ request to stop an ongoing batching session at the service. The service stops generating location fixes and storing them in the batch. Any existing location fixes in the batch are unaffected until QMI_LOC_RELEASE_BATCH is requested. The control point sends a QMI_LOC_RELEASE_BATCH_REQ request to release the batching buffer that was allocated at the service. The control point must stop the batching session before sending this release request. It is the responsibility of the control point to release the allocated memory at the service using QMI_LOC_RELEASE_BATCH.

Multiple control points can simultaneously have their own batching sessions. The same control point cannot have simultaneous auto-tracking and batching sessions. The batching session and the batching contents of a control point are automatically lost when the control point is disconnected or the device resets.

Figure 2-7 illustrates the software architecture of the batching commands.

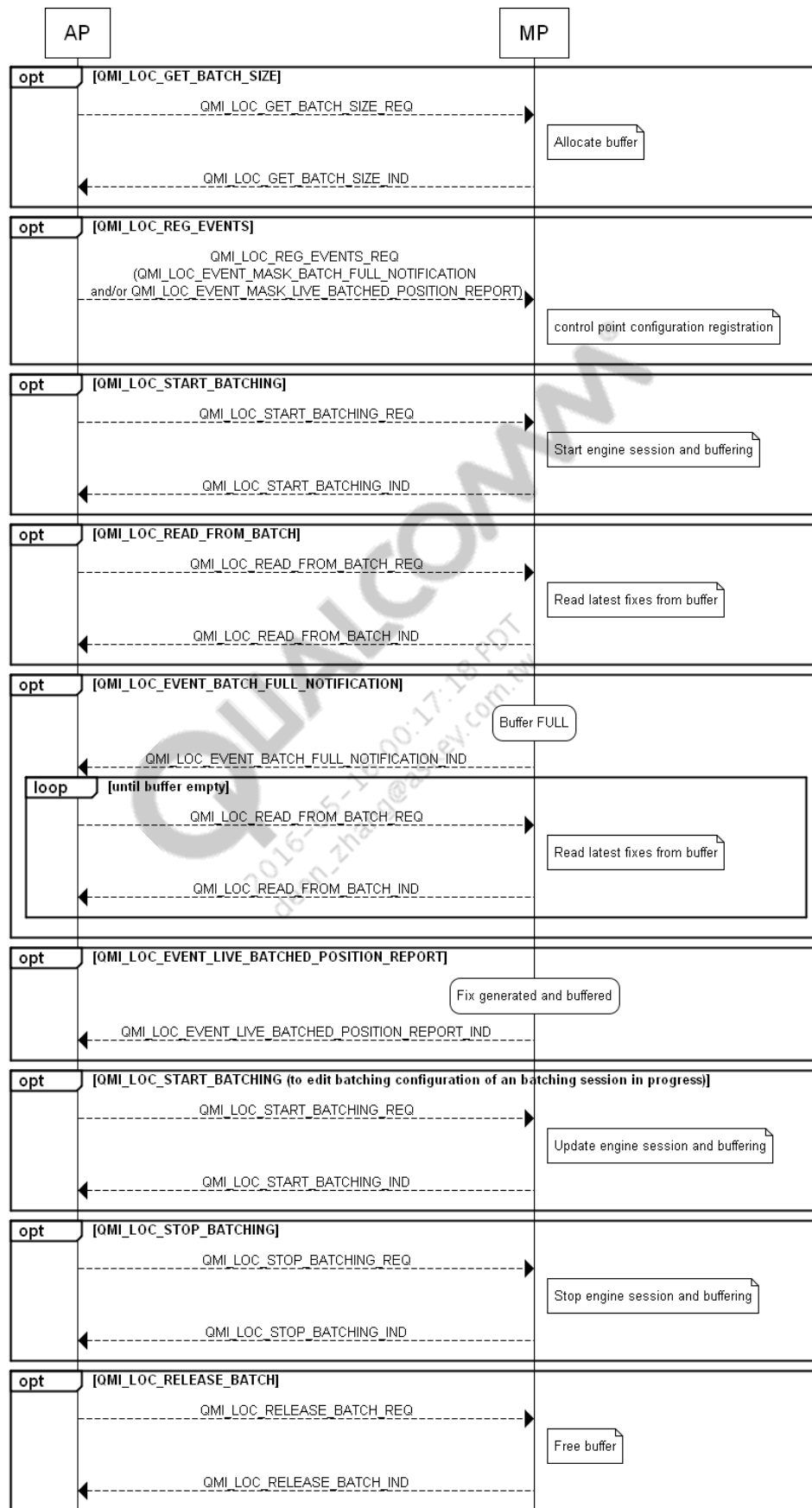


Figure 2-7 Batching software architecture

3 QMI_LOC Messages

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_GET_SUPPORTED_MSGS	0x001E	Queries the set of messages implemented by the currently running software.
QMI_LOC_GET_SUPPORTED_FIELDS	0x001F	Queries the fields supported for a single command as implemented by the currently running software.
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	Used by the control point to initiate a GPS session.
QMI_LOC_STOP	0x0023	Used by the control point to stop a GPS session.
QMI_LOC_EVENT_POSITION_REPORT	0x0024	Sends the position report to the control point.
QMI_LOC_EVENT_GNSS_SV_INFO	0x0025	Sends a satellite report to the control point.
QMI_LOC_EVENT_NMEA	0x0026	Sends 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.

Table 3-1 QMI_LOC messages (cont.)

Command	ID	Description
QMI_LOC_EVENT_WIFI_REQ	0x002D	Sends a Wi-Fi request to the control point.
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.

Table 3-1 QMI_LOC messages (cont.)

Command	ID	Description
QMI_LOC_SET_SERVER	0x0042	Specifies the A-GPS server type and address.
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 Wi-Fi position.
QMI_LOC_NOTIFY_WIFI_STATUS	0x0048	Notifies the location engine of the Wi-Fi 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.
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_SET_CRADLE_MOUNT_CONFIG	0x004F	Used by the control point to set the current cradle mount configuration.
QMI_LOC_GET_CRADLE_MOUNT_CONFIG	0x0050	Used by the control point to get the current cradle mount configuration.
QMI_LOC_SET_EXTERNAL_POWER_CONFIG	0x0051	Used by the control point to set the current external power configuration.
QMI_LOC_GET_EXTERNAL_POWER_CONFIG	0x0052	Used by the control point to get 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_CONNECTION_REQ_IND event.

Table 3-1 QMI_LOC messages (cont.)

Command	ID	Description
QMI_LOC_SET_PROTOCOL_CONFIG_PARAMETERS	0x0054	Used by the control point to configure parameters stored in the nonvolatile memory.
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.
QMI_LOC_EVENT_NI_GEOFENCE_NOTIFICATION	0x0060	Informs the control point about network-initiated Geofences.
QMI_LOC_EVENT_GEOFENCE_GEN_ALERT	0x0061	Notifies the control point of the Geofence status.
QMI_LOC_EVENT_GEOFENCE_BREACH_NOTIFICATION	0x0062	Notifies the control point of a Geofence breach event.
QMI_LOC_ADD_CIRCULAR_GEOFENCE	0x0063	Used by the control point to add a circular Geofence.
QMI_LOC_DELETE_GEOFENCE	0x0064	Used by the control point to delete a Geofence.
QMI_LOC_QUERY_GEOFENCE	0x0065	Used by the control point to query a Geofence.
QMI_LOC_EDIT_GEOFENCE	0x0066	Used by the control point to edit a Geofence.

Table 3-1 QMI_LOC messages (cont.)

Command	ID	Description
QMI_LOC_GET_BEST_AVAILABLE_POSITION	0x0067	Used by the control point to get the best available position estimate from the location engine.
QMI_LOC_INJECT_MOTION_DATA	0x0068	Injects motion data for MSM GPS service use.
QMI_LOC_GET_NI_GEOFENCE_ID_LIST	0x0069	Used by the control point to retrieve the list of network initiated Geofence IDs.
QMI_LOC_INJECT_GSM_CELL_INFO	0x006A	Injects GSM cell information into the location engine.
QMI_LOC_INJECT_NETWORK_INITIATED_MESSAGE	0x006B	Injects a network-initiated message into the location engine.
QMI_LOC_WWAN_OUT_OF_SERVICE_NOTIFICATION	0x006C	Notifies the location engine that the device is out of service.
QMI_LOC_EVENT_PEDOMETER_CONTROL	0x006D	Recommends how pedometer reports are to be sent to the location engine.
QMI_LOC_EVENT_MOTION_DATA_CONTROL	0x006E	Recommends how motion data reports are to be sent to the location engine.
QMI_LOC_PEDOMETER_REPORT	0x006F	Used by the control point to inject pedometer data into the location engine.
QMI_LOC_INJECT_WCDMA_CELL_INFO	0x0070	Injects WCDMA cell information into the location engine.
QMI_LOC_INJECT_TDSCDMA_CELL_INFO	0x0071	Injects TDSCDMA cell information into the location engine.
QMI_LOC_INJECT_SUBSCRIBER_ID	0x0072	Injects the phone's subscriber ID into the location engine.
QMI_LOC_SET_GEOFENCE_ENGINE_CONFIG	0x0073	Used by the control point to set the Geofence engine configuration.
QMI_LOC_GET_GEOFENCE_ENGINE_CONFIG	0x0074	Used by the control point to get the Geofence engine configuration.
QMI_LOC_GET_BATCH_SIZE	0x0075	Used by the control point to get the batching size.
QMI_LOC_START_BATCHING	0x0076	Used by the control point to initiate a batching session.
QMI_LOC_EVENT_BATCH_FULL_NOTIFICATION	0x0077	Used to notify the control point that the batched buffer is full.
QMI_LOC_EVENT_LIVE_BATCHED_POSITION_REPORT	0x0078	Used to notify the control point with the live batched position report.
QMI_LOC_READ_FROM_BATCH	0x0079	Used by the control point to retrieve fixes from the batch.
QMI_LOC_STOP_BATCHING	0x007A	Used by the control point to stop an ongoing batching session.
QMI_LOC_RELEASE_BATCH	0x007B	Used by the control point to release the batching buffer.
QMI_LOC_EVENT_INJECT_WIFI_AP_DATA_REQ	0x007C	Requests the control point to inject Wi-Fi AP data.

Table 3-1 QMI_LOC messages (cont.)

Command	ID	Description
QMI_LOC_INJECT_WIFI_AP_DATA	0x007D	Injects Wi-Fi AP data.
QMI_LOC_NOTIFY_WIFI_ATTACHMENT_STATUS	0x007E	Used by the control point to inject the Wi-Fi attachment status.
QMI_LOC_NOTIFY_WIFI_ENABLED_STATUS	0x007F	Used by the control point to inject the Wi-Fi enabled status.
QMI_LOC_EVENT_GEOFENCE_BATCHED_BREACH_NOTIFICATION	0x0080	Notifies the control point of a Geofence breach event by batching all the Geofences that were breached.
QMI_LOC_EVENT_VEHICLE_DATA_READY_STATUS	0x0081	Notifies the control point whether the GNSS location engine is ready to accept vehicle data.
QMI_LOC_INJECT_VEHICLE_SENSOR_DATA	0x0082	Injects on-vehicle sensor data into the location engine.
QMI_LOC_GET_AVAILABLE_WWAN_POSITION	0x0083	Used by the control point to get the first available WWAN position from the location engine.
QMI_LOC_SET_PREMIUM_SERVICES_CONFIG	0x0084	Used by the control point to set the configuration information for all iZat premium services to the location engine.
QMI_LOC_SET_XTRA_VERSION_CHECK	0x0085	Used by the control point to enable or disable XTRA version verification.
QMI_LOC_EVENT_GNSS_MEASUREMENT_REPORT_IND	0x0086	Sends a satellite measurement report to the control point.
QMI_LOC_EVENT_SV_POLYNOMIAL_REPORT_IND	0x0087	Sends a satellite polynomial report to the control point.
QMI_LOC_SET_GNSS_CONSTELL_REPORT_CONFIG	0x0088	Sets satellite constellations of interest for reporting.
QMI_LOC_ADD_GEOFENCE_CONTEXT	0x0089	Used by the control point to inject the Geofence context.
QMI_LOC_SET_GEOFENCE_ENGINE_CONTEXT	0x008A	Used by the control point to inject the Geofence engine context.
QMI_LOC_DELETE_GEOFENCE_CONTEXT	0x008B	Used by the control point to Delete the geofence context.
QMI_LOC_EVENT_GEOFENCE_PROXIMITY_NOTIFICATION	0x008C	Notifies the control point of a Geofence proximity event.

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

N/A

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_GET_SUPPORTED_MSGS

Queries the set of messages implemented by the currently running software.

LOC message ID

0x001E

Version introduced

Major - 2, Minor - 19

3.2.1 Request - QMI_LOC_GET_SUPPORTED_MSGS_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

None

Optional TLVs

None

3.2.2 Response - QMI_LOC_GET_SUPPORTED_MSGS_RESP

Message type

Response

Sender

Service

Mandatory TLVs

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

Name	Common version introduced	Common version last modified
Result Code	1.6	1.7

Optional TLVs

Name	Common version introduced	Common version last modified
List of Supported Messages	1.6	1.6

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	List of Supported Messages
Length	Var			2	
Value	→	uint16	supported_msgs_len	2	Number of sets of the following elements: • supported_msgs
		uint8	supported_msgs	Var	This array of uint8 is a bitmask where each bit represents a message ID, i.e., starting with the LSB, bit 0 represents message ID 0, bit 1 represents message ID 1, etc. The bit is set to 1 if the message is supported; otherwise, it is set to zero. For example, if a service supports exactly four messages with IDs 0, 1, 30, and 31 (decimal), the array (in hexadecimal) is 4 bytes [03 00 00 c0].

Error codes

QMI_ERR_NONE	No error in the request
QMI_ERR_INTERNAL	Unexpected error occurred during processing
QMI_ERR_NO_MEMORY	Device could not allocate memory to formulate a response
QMI_ERR_INFO_UNAVAILABLE	Information is not available

3.2.3 Description of QMI_LOC_GET_SUPPORTED_MSGS REQ/RESP

This command queries the set of messages implemented by the currently running software. This may be a subset of the messages defined in this revision of the service.

3.3 QMI_LOC_GET_SUPPORTED_FIELDS

Queries the fields supported for a single command as implemented by the currently running software.

LOC message ID

0x001F

Version introduced

Major - 2, Minor - 19

3.3.1 Request - QMI_LOC_GET_SUPPORTED_FIELDS_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

Name	Common version introduced	Common version last modified
Service Message ID	1.6	1.6

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Service Message ID
Length	2			2	
Value	→	uint16	msg_id	2	ID of the command for which the supported fields are requested.

Optional TLVs

None

3.3.2 Response - QMI_LOC_GET_SUPPORTED_FIELDS_RESP

Message type

Response

Sender

Service

Mandatory TLVs

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

Name	Common version introduced	Common version last modified
Result Code	1.6	1.7

Optional TLVs

Name	Common version introduced	Common version last modified
List of Supported Request Fields	1.6	1.6
List of Supported Response Fields	1.6	1.6
List of Supported Indication Fields	1.6	1.6

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	List of Supported Request Fields
Length	Var			2	
Value	→	uint8	request_fields_len	1	Number of sets of the following elements: • request_fields
		uint8	request_fields	Var	This field describes which optional field IDs are supported in the QMI request. The array of uint8 is a bitmask where each bit represents a field (TLV) ID. Because fields 0 to 15 (decimal) are mandatory by definition, the first bit represents field ID 16. Starting with the LSB, bit 0 represents field ID 16, bit 1 represents field ID 17, etc. The bit is set to 1 if the field ID is supported; otherwise, it is set to zero. For example, if a service supports exactly four fields with IDs 16, 17, 30, and 31 (decimal), the array (in hexadecimal) is 2 bytes [03 c0].
Type	0x11			1	List of Supported Response Fields
Length	Var			2	
Value	→	uint8	response_fields_len	1	Number of sets of the following elements: • response_fields
		uint8	response_fields	Var	This field describes which optional field IDs are supported in the QMI response. Its format is the same as request_fields.
Type	0x12			1	List of Supported Indication Fields
Length	Var			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	uint8	indication_fields_len	1	Number of sets of the following elements: • indication_fields
		uint8	indication_fields	Var	This field describes which optional field IDs are supported in the QMI indication. Its format is the same as request_fields.

Error codes

QMI_ERR_NONE	No error in the request
QMI_ERR_INTERNAL	Unexpected error occurred during processing
QMI_ERR_NO_MEMORY	Device could not allocate memory to formulate a response
QMI_ERR_REQUESTED_NUM_UNSUPPORTED	Requested message ID is not supported by the currently running software
QMI_ERR_MALFORMED_MSG	Message was not formulated correctly by the control point or the message was corrupted during transmission
QMI_ERR_INFO_UNAVAILABLE	Information is not available

3.3.3 Description of QMI_LOC_GET_SUPPORTED_FIELDS REQ/RESP

This command queries the fields supported for a single command as implemented by the currently running software.

If the request, response, or indication is supported for the given message ID, the corresponding optional array is included in QMI_LOC_GET_SUPPORTED_FIELDS_RESP, even if the message does not contain any optional fields. This enables the client to distinguish this case from one where the service does not support the request, response, or indication.

Examples are:

- If the specified message ID is not supported by the service, the response has qmi_result = QMI_RESULT_FAILURE and qmi_error = QMI_ERR_REQUESTED_NUM_UNSUPPORTED.
- If the specified message ID is an empty message, the response has qmi_result = QMI_RESULT_SUCCESS and qmi_error = QMI_ERR_NONE. None of the optional arrays are included.
- If the specified message ID supports the request with 0 optional fields, the response with 3 optional fields (16, 17, and 18 decimal), and does not support an indication, the response has the following:
 - qmi_result = QMI_RESULT_SUCCESS
 - qmi_error = QMI_ERR_NONE
 - request_fields array is included with length zero
 - response_fields array is included with length 1 value [07]
 - indication_fields array is not included

Trailing zero bytes are omitted from the response. For example, if the message defines 20 different fields but the response only contains 16 bits, the client is to assume the last four fields are not supported.

3.4 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 - 0

3.4.1 Request - QMI_LOC_INFORM_CLIENT_REVISION_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

Name	Version introduced	Version last modified
Revision	2.0	2.0

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Revision
Length	4			2	
Value	→	uint32	revision	4	Revision that the control point is using.

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

3.4.2 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. This message does not impact the global state of the service, and it is safe if more than one client sends this message.

QUALCOMM®
2016-05-16 00:17:18 PDT
deon_zhang@askey.com.tw

3.5 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 - 0

3.5.1 Request - QMI_LOC_REG_EVENTS_REQ

Message type

Request

Sender

Control Point

Mandatory TLVs

Name	Version introduced	Version last modified
Event Registration Mask	2.0	2.31

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Event Registration Mask
Length	8			2	
Value	→	mask	eventRegMask	8	Specifies the events that the control point is interested in receiving. Valid bitmasks: <ul style="list-style-type: none"> • QMI_LOC_EVENT_MASK_POSITION_REPORT (0x00000001) – The control point must enable this mask to receive position report event indications. • QMI_LOC_EVENT_MASK_GNSS_SV_INFO (0x00000002) – The control point must enable this mask to receive satellite report event indications. These reports are sent at a 1 Hz rate. • QMI_LOC_EVENT_MASK_NMEA (0x00000004) – The control point must enable this mask to receive NMEA reports for position and satellites in view. The report is at a 1 Hz rate.

Field	Field value	Field type	Parameter	Size (byte)	Description
			eventRegMask (cont.)		<ul style="list-style-type: none"> • QMI_LOC_EVENT_MASK_NI_NOTIFY_VERIFY_REQ (0x00000008) – The control point must enable this mask to receive NI Notify/Verify request event indications. • QMI_LOC_EVENT_MASK_INJECT_TIME_REQ (0x00000010) – The control point must enable this mask to receive time injection request event indications. • QMI_LOC_EVENT_MASK_INJECT_PREDICTED_ORBITS_REQ (0x00000020) – The control point must enable this mask to receive predicted orbits request event indications. • QMI_LOC_EVENT_MASK_INJECT_POSITION_REQ (0x00000040) – The control point must enable this mask to receive position injection request event indications. • QMI_LOC_EVENT_MASK_ENGINE_STATE (0x00000080) – The control point must enable this mask to receive engine state report event indications. • QMI_LOC_EVENT_MASK_FIX_SESSION_STATE (0x00000100) – The control point must enable this mask to receive fix session status report event indications. • QMI_LOC_EVENT_MASK_WIFI_REQ (0x00000200) – The control point must enable this mask to receive Wi-Fi position request event indications. • QMI_LOC_EVENT_MASK_SENSOR_STREAMING_READY_STATUS (0x00000400) – The control point must enable this mask to receive notifications from the location engine indicating its readiness to accept data from the sensors (accelerometer, gyroscope, etc.). • QMI_LOC_EVENT_MASK_TIME_SYNC_REQ (0x00000800) – The control point must enable this mask to receive time sync requests from the GPS engine. Time sync enables the GPS engine to synchronize its clock with the sensor processor's clock.

Field	Field value	Field type	Parameter	Size (byte)	Description
			eventRegMask (cont.)		<ul style="list-style-type: none"> • QMI_LOC_EVENT_MASK_SET_SPI_STREAMING_REPORT (0x00001000) – The control point must enable this mask to receive Stationary Position Indicator (SPI) streaming report indications. • QMI_LOC_EVENT_MASK_LOCATION_SERVER_CONNECTION_REQ (0x00002000) – The control point must enable this mask to receive location server requests. These requests are generated when the service wishes to establish a connection with a location server. • QMI_LOC_EVENT_MASK_NI_GEOFENCE_NOTIFICATION (0x00004000) – The control point must enable this mask to receive notifications related to network-initiated Geofences. These events notify the client when a network-initiated Geofence is added, deleted, or edited. • QMI_LOC_EVENT_MASK_GEOFENCE_GEN_ALERT (0x00008000) – The control point must enable this mask to receive Geofence alerts. These alerts are generated to inform the client of the changes that may affect a Geofence, e.g., if GPS is turned off or if the network is unavailable. • QMI_LOC_EVENT_MASK_GEOFENCE_BREACH_NOTIFICATION (0x00010000) – The control point must enable this mask to receive notifications when a Geofence is breached. These events are generated when a UE enters or leaves the perimeter of a Geofence. This breach report is for a single Geofence. • QMI_LOC_EVENT_MASK_PEDOMETER_CONTROL (0x00020000) – The control point must enable this mask to register for pedometer control requests from the location engine. The location engine sends this event to control the injection of pedometer reports.

Field	Field value	Field type	Parameter	Size (byte)	Description
			eventRegMask (cont.)		<ul style="list-style-type: none"> • QMI_LOC_EVENT_MASK_MOTION_DATA_CONTROL (0x00040000) – The control point must enable this mask to register for motion data control requests from the location engine. The location engine sends this event to control the injection of motion data. • QMI_LOC_EVENT_MASK_BATCH_FULL_NOTIFICATION (0x00080000) – The control point must enable this mask to receive notification when a batch is full. The location engine sends this event to notify of Batch Full for ongoing batching session. • QMI_LOC_EVENT_MASK_LIVE_BATCHED_POSITION_REPORT (0x00100000) – The control point must enable this mask to receive position report indications along with an ongoing batching session. The location engine sends this event to notify the batched position report while a batching session is ongoing. • QMI_LOC_EVENT_MASK_INJECT_WIFI_AP_DATA_REQ (0x00200000) – The control point must enable this mask to receive Wi-Fi Access Point (AP) data inject request event indications. • QMI_LOC_EVENT_MASK_GEOFENCE_BATCH_BREACH_NOTIFICATION (0x00400000) – The control point must enable this mask to receive notifications when a Geofence is breached. These events are generated when a UE enters or leaves the perimeter of a Geofence. This breach notification is for multiple Geofences. Breaches from multiple Geofences are all batched and sent in the same notification.

Field	Field value	Field type	Parameter	Size (byte)	Description
			eventRegMask (cont.)		<ul style="list-style-type: none"> • QMI_LOC_EVENT_MASK_VEHICLE_DATA_READY_STATUS (0x00800000) – The control point must enable this mask to receive notifications from the location engine indicating its readiness to accept vehicle data (vehicle accelerometer, vehicle angular rate, vehicle odometry, etc.). • QMI_LOC_EVENT_MASK_GNSS_MEASUREMENT_REPORT (0x01000000) – The control point must enable this mask to receive system clock and satellite measurement report events (system clock, SV time, Doppler, etc.). Reports are generated only for the GNSS satellite constellations that are enabled using QMI_LOC_SET_GNSS_CONSTELL_REPORT_CONFIG. • QMI_LOC_EVENT_MASK_GNSS_SV_POLYNOMIAL_REPORT (0x02000000) – The control point must enable this mask to receive satellite position reports as polynomials. Reports are generated only for the GNSS satellite constellations that are enabled using QMI_LOC_SET_GNSS_CONSTELL_REPORT_CONFIG. <p>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.</p>

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_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). This message does not impact the global state of the service, and it is safe if more than one client sends this message.

QUALCOMM®
2016-05-16 00:17:18 PDT
deon_zhang@askey.com.tw

3.6 QMI_LOC_START

Used by the control point to initiate a GPS session.

LOC message ID

0x0022

Version introduced

Major - 2, Minor - 0

3.6.1 Request - QMI_LOC_START_REQ

Message type

Request

Sender

Control Point

Mandatory TLVs

Name	Version introduced	Version last modified
Session ID	2.0	2.0

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Session ID
Length	1			2	
Value	→	uint8	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. • Range: 0 to 255

Optional TLVs

Name	Version introduced	Version last modified
Recurrence Type	2.0	2.0
Horizontal Accuracy	2.0	2.0
Enable/Disable Intermediate Reports	2.0	2.0
Minimum Interval Between Position Reports	2.0	2.0
ID of the Application that Sent this Request	2.6	2.6
Configuration for Altitude Assumed Info in GNSS SV Info Event	2.25	2.25

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Recurrence Type
Length	4			2	
Value	→	enum	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: • eQMI_LOC_RECURRENCE_PERIODIC (1) – Request periodic position fixes • eQMI_LOC_RECURRENCE_SINGLE (2) – Request a single position fix
Type	0x11			1	Horizontal Accuracy
Length	4			2	
Value	→	enum	horizontalAccuracyLevel	4	Specifies the horizontal accuracy level required by the control point. If not specified, accuracy defaults to LOW. Valid values: • eQMI_LOC_ACCURACY_LOW (1) – Low accuracy • eQMI_LOC_ACCURACY_MED (2) – Medium accuracy • eQMI_LOC_ACCURACY_HIGH (3) – High accuracy
Type	0x12			1	Enable/Disable Intermediate Reports
Length	4			2	
Value	→	enum	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: • eQMI_LOC_INTERMEDIATE_REPORTS_ON (1) – Intermediate reports are turned on • eQMI_LOC_INTERMEDIATE_REPORTS_OFF (2) – Intermediate reports are turned off

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x13			1	Minimum Interval Between Position Reports
Length	4			2	
Value	→	uint32	minInterval	4	Minimum time interval, specified by the control point, that must elapse between position reports. • 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	→	uint8	applicationProvider_len	1	Number of sets of the following elements: • applicationProvider
		string	applicationProvider	Var	Application provider.
		uint8	applicationName_len	1	Number of sets of the following elements: • applicationName
		string	applicationName	Var	Application name.
		boolean	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
		uint8	applicationVersion_len	1	Number of sets of the following elements: • applicationVersion
		string	applicationVersion	Var	Application version.
Type	0x15			1	Configuration for Altitude Assumed Info in GNSS SV Info Event
Length	4			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	enum	configAltitudeAssumed	4	<p>Specifies the configuration to include Altitude Assumed information in the GNSS SV Info Event. When enabled, an additional GNSS SV Info event indication is sent to the control point that also includes the altitude assumed information.</p> <p>If not specified, the configuration defaults to ENABLED.</p> <p>Valid values:</p> <ul style="list-style-type: none"> •eQMI_LOC_ALTITUDE_ASSUMED_IN_GNSS_SV_INFO_ENABLED (1) – Enable Altitude Assumed information in GNSS SV Info Event •eQMI_LOC_ALTITUDE_ASSUMED_IN_GNSS_SV_INFO_DISABLED (2) – Disable Altitude Assumed information in GNSS SV Info Event

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_START REQ/RESP

This message starts a positioning session with the specified configuration. It is safe if more than one client sends this message. Section 2.9.3 (Multiple-Client Support) describes how start requests from multiple clients are handled by the service.

3.7 QMI_LOC_STOP

Used by the control point to stop a GPS session.

LOC message ID

0x0023

Version introduced

Major - 2, Minor - 0

3.7.1 Request - QMI_LOC_STOP_REQ

Message type

Request

Sender

Control Point

Mandatory TLVs

Name	Version introduced	Version last modified
Session ID	2.0	2.0

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Session ID
Length	1			2	
Value	→	uint8	sessionId	1	ID of the session that was specified in the Start request (QMI_LOC_START_REQ). • 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

3.7.2 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, fix session, and NMEA report events. It is safe if more than one client sends this message.

QUALCOMM®
2016-05-16 00:17:18 PDT
deon_zhang@askey.com.tw

3.8 QMI_LOC_EVENT_POSITION_REPORT

Sends the position report to the control point.

LOC message ID

0x0024

Version introduced

Major - 2, Minor - 0

3.8.1 Indication - QMI_LOC_EVENT_POSITION_REPORT_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version introduced	Version last modified
Session Status	2.0	2.1
Session ID	2.0	2.0

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Session Status
Length	4			2	
Value	→	enum	sessionStatus	4	Session status. Valid values: <ul style="list-style-type: none"> • eQMI_LOC_SESS_STATUS_SUCCESS (0) – Session was successful • eQMI_LOC_SESS_STATUS_IN_PROGRESS (1) – Session is still in progress; further position reports will be generated until either the fix criteria specified by the client are met or the client response timeout occurs • eQMI_LOC_SESS_STATUS_GENERAL_FAILURE (2) – Session failed • eQMI_LOC_SESS_STATUS_TIMEOUT (3) – Fix request failed because the session timed out

Field	Field value	Field type	Parameter	Size (byte)	Description
			sessionStatus (cont.)		<ul style="list-style-type: none"> • eQMI_LOC_SESS_STATUS_USER_END (4) – Fix request failed because the session was ended by the user • eQMI_LOC_SESS_STATUS_BAD_PARAMETER (5) – Fix request failed due to bad parameters in the request • eQMI_LOC_SESS_STATUS_PHONE_OFFLINE (6) – Fix request failed because the phone is offline • eQMI_LOC_SESS_STATUS_ENGINE_LOCKED (7) – Fix request failed because the engine is locked
Type	0x02			1	Session ID
Length	1			2	
Value	→	uint8	sessionId	1	ID of the session that was specified in the Start request QMI_LOC_START_REQ. • Range: 0 to 255

Optional TLVs

Name	Version introduced	Version last modified
Latitude	2.0	2.0
Longitude	2.0	2.0
Circular Horizontal Position Uncertainty	2.0	2.0
Horizontal Elliptical Uncertainty	2.0	2.0
Horizontal Elliptical Uncertainty	2.0	2.0
Elliptical Horizontal Uncertainty Azimuth	2.0	2.0
Horizontal Confidence	2.0	2.0
Horizontal Reliability	2.0	2.1
Horizontal Speed	2.0	2.0
Speed Uncertainty	2.0	2.0
Altitude With Respect to Ellipsoid	2.0	2.0
Altitude With Respect to Sea Level	2.0	2.0
Vertical Uncertainty	2.0	2.0
Vertical Confidence	2.0	2.0
Vertical Reliability	2.0	2.1
Vertical Speed	2.0	2.0
Heading	2.0	2.0
Heading Uncertainty	2.0	2.0
Magnetic Deviation	2.0	2.0
Technology Used	2.0	2.20
Dilution of Precision	2.0	2.0
UTC Timestamp	2.0	2.0
Leap Seconds	2.0	2.0

Name	Version introduced	Version last modified
GPS Time	2.0	2.0
Time Uncertainty	2.0	2.0
Time Source	2.0	2.22
Sensor Data Usage	2.0	2.0
Fix Count for This Session	2.2	2.2
SVs Used to Calculate the Fix	2.9	2.22
Altitude Assumed	2.20	2.20

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Latitude
Length	8			2	
Value	→	double	latitude	8	Latitude (specified in WGS84 datum). <ul style="list-style-type: none"> • Type: Floating point • Units: Degrees • Range: -90.0 to 90.0 <ul style="list-style-type: none"> – Positive values indicate northern latitude – Negative values indicate southern latitude
Type	0x11			1	Longitude
Length	8			2	
Value	→	double	longitude	8	Longitude (specified in WGS84 datum). <ul style="list-style-type: none"> • Type: Floating point • Units: Degrees • Range: -180.0 to 180.0 <ul style="list-style-type: none"> – Positive values indicate eastern longitude – Negative values indicate western longitude
Type	0x12			1	Circular Horizontal Position Uncertainty
Length	4			2	
Value	→	float	horUncCircular	4	Horizontal position uncertainty (circular). <ul style="list-style-type: none"> • Units: Meters
Type	0x13			1	Horizontal Elliptical Uncertainty (Semi-Minor Axis)
Length	4			2	
Value	→	float	horUncEllipseSemiMinor	4	Semi-minor axis of horizontal elliptical uncertainty. <ul style="list-style-type: none"> • Units: Meters
Type	0x14			1	Horizontal Elliptical Uncertainty (Semi-Major Axis)
Length	4			2	
Value	→	float	horUncEllipseSemiMajor	4	Semi-major axis of horizontal elliptical uncertainty. <ul style="list-style-type: none"> • Units: Meters

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x15			1	Elliptical Horizontal Uncertainty Azimuth
Length	4			2	
Value	→	float	horUncEllipseOrient Azimuth	4	Elliptical horizontal uncertainty azimuth of orientation. • Units: Decimal degrees • Range: 0 to 180
Type	0x16			1	Horizontal Confidence
Length	1			2	
Value	→	uint8	horConfidence	1	Horizontal uncertainty confidence. If both elliptical and horizontal uncertainties are specified in this message, the confidence corresponds to the elliptical uncertainty. • Units: Percent • Range: 0 to 99
Type	0x17			1	Horizontal Reliability
Length	4			2	
Value	→	enum	horReliability	4	Specifies the reliability of the horizontal position. Valid values: • eQMI_LOC_RELIABILITY_NOT_SET (0) – Location reliability is not set • eQMI_LOC_RELIABILITY_VERY_LOW (1) – Location reliability is very low; use it at your own risk • eQMI_LOC_RELIABILITY_LOW (2) – Location reliability is low; little or no cross-checking is possible • eQMI_LOC_RELIABILITY_MEDIUM (3) – Location reliability is medium; limited cross-check passed • eQMI_LOC_RELIABILITY_HIGH (4) – Location reliability is high; strong cross-check passed
Type	0x18			1	Horizontal Speed
Length	4			2	
Value	→	float	speedHorizontal	4	Horizontal speed. • Units: Meters/second
Type	0x19			1	Speed Uncertainty
Length	4			2	
Value	→	float	speedUnc	4	3-D Speed uncertainty. • Units: Meters/second
Type	0x1A			1	Altitude With Respect to Ellipsoid
Length	4			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	float	altitudeWrtEllipsoid	4	Altitude with respect to the WGS84 ellipsoid. • Units: Meters • Range: -500 to 15883
Type	0x1B			1	Altitude With Respect to Sea Level
Length	4			2	
Value	→	float	altitudeWrtMeanSeaLevel	4	Altitude with respect to mean sea level. • Units: Meters
Type	0x1C			1	Vertical Uncertainty
Length	4			2	
Value	→	float	vertUnc	4	Vertical uncertainty. • Units: Meters
Type	0x1D			1	Vertical Confidence
Length	1			2	
Value	→	uint8	vertConfidence	1	Vertical uncertainty confidence. • Units: Percent • Range: 0 to 99
Type	0x1E			1	Vertical Reliability
Length	4			2	
Value	→	enum	vertReliability	4	Specifies the reliability of the vertical position. Valid values: • eQMI_LOC_RELIABILITY_NOT_SET (0) – Location reliability is not set • eQMI_LOC_RELIABILITY_VERY_LOW (1) – Location reliability is very low; use it at your own risk • eQMI_LOC_RELIABILITY_LOW (2) – Location reliability is low; little or no cross-checking is possible • eQMI_LOC_RELIABILITY_MEDIUM (3) – Location reliability is medium; limited cross-check passed • eQMI_LOC_RELIABILITY_HIGH (4) – Location reliability is high; strong cross-check passed
Type	0x1F			1	Vertical Speed
Length	4			2	
Value	→	float	speedVertical	4	Vertical speed. • Units: Meters/second
Type	0x20			1	Heading
Length	4			2	
Value	→	float	heading	4	Heading. • Units: Degrees • Range: 0 to 359.999
Type	0x21			1	Heading Uncertainty
Length	4			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	float	headingUnc	4	Heading uncertainty. • Units: Degrees • Range: 0 to 359.999
Type	0x22			1	Magnetic Deviation
Length	4			2	
Value	→	float	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	0x23			1	Technology Used
Length	4			2	
Value	→	mask32	technologyMask	4	Technology used in computing this fix. Valid bitmasks: • QMI_LOC_POS_TECH_MASK_SATELLITE (0x00000001) – Satellites were used to generate the fix • QMI_LOC_POS_TECH_MASK_CELLID (0x00000002) – Cell towers were used to generate the fix • QMI_LOC_POS_TECH_MASK_WIFI (0x00000004) – Wi-Fi access points were used to generate the fix • QMI_LOC_POS_TECH_MASK_SENSORS (0x00000008) – Sensors were used to generate the fix • QMI_LOC_POS_TECH_MASK_REFERENCE_LOCATION (0x00000010) – Reference Location was used to generate the fix • QMI_LOC_POS_TECH_MASK_INJECTED_COARSE_POSITION (0x00000020) – Coarse position injected into the location engine was used to generate the fix • QMI_LOC_POS_TECH_MASK_AFLT (0x00000040) – AFLT was used to generate the fix • QMI_LOC_POS_TECH_MASK_HYBRID (0x00000080) – GNSS and network-provided measurements were used to generate the fix
Type	0x24			1	Dilution of Precision
					Dilution of precision associated with this position.
Length	12			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	float	PDOP	4	Position dilution of precision. • Range: 1 (highest accuracy) to 50 (lowest accuracy) • PDOP = square root of (HDOP ² + VDOP ²)
		float	HDOP	4	Horizontal dilution of precision. • Range: 1 (highest accuracy) to 50 (lowest accuracy)
		float	VDOP	4	Vertical dilution of precision. • Range: 1 (highest accuracy) to 50 (lowest accuracy)
Type	0x25			1	UTC Timestamp
Length	8			2	
Value	→	uint64	timestampUtc	8	UTC timestamp. • Units: Milliseconds since Jan. 1, 1970
Type	0x26			1	Leap Seconds
Length	1			2	
Value	→	uint8	leapSeconds	1	Leap second information. If leapSeconds is not available, timestampUtc is calculated based on a hard-coded value for leap seconds. • 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	→	uint16	gpsWeek	2	Current GPS week as calculated from midnight, Jan. 6, 1980. • Units: Weeks
		uint32	gpsTimeOfWeekMs	4	Amount of time into the current GPS week. • Units: Milliseconds
Type	0x28			1	Time Uncertainty
Length	4			2	
Value	→	float	timeUnc	4	Time uncertainty. • Units: Milliseconds
Type	0x29			1	Time Source
Length	4			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	enum	timeSrc	4	<p>Time source. Valid values:</p> <ul style="list-style-type: none"> • eQMI_LOC_TIME_SRC_INVALID (0) – Invalid time. • eQMI_LOC_TIME_SRC_NETWORK_TIME_TRANSFER (1) – Time is set by the 1X system • eQMI_LOC_TIME_SRC_NETWORK_TIME_TAGGING (2) – Time is set by WCDMA/GSM time tagging (i.e., associating network time with GPS time) • eQMI_LOC_TIME_SRC_EXTERNAL_INPUT (3) – Time is set by an external injection • eQMI_LOC_TIME_SRC_TOW_DECODE (4) – Time is set after decoding over-the-air GPS navigation data from one GPS satellite • eQMI_LOC_TIME_SRC_TOW_CONFIRMED (5) – Time is set after decoding over-the-air GPS navigation data from multiple satellites • eQMI_LOC_TIME_SRC_TOW_AND_WEEK_CONFIRMED (6) – Both time of the week and the GPS week number are known • eQMI_LOC_TIME_SRC_NAV_SOLUTION (7) – Time is set by the position engine after the fix is obtained • eQMI_LOC_TIME_SRC_SOLVE_FOR_TIME (8) – Time is set by the position engine after performing SFT; this is done when the clock time uncertainty is large • eQMI_LOC_TIME_SRC_GLO_TOW_DECODE (9) – Time is set after decoding GLO satellites • eQMI_LOC_TIME_SRC_TIME_TRANSFORM (10) – Time is set after transforming the GPS to GLO time • eQMI_LOC_TIME_SRC_WCDMA_SLEEP_TIME_TAGGING (11) – Time is set by the sleep time tag provided by the WCDMA network • eQMI_LOC_TIME_SRC_GSM_SLEEP_TIME_TAGGING (12) – Time is set by the sleep time tag provided by the GSM network

Field	Field value	Field type	Parameter	Size (byte)	Description
			timeSrc (cont.)		<ul style="list-style-type: none"> • eQMI_LOC_TIME_SRC_UNKNOWN (13) – Source of the time is unknown • eQMI_LOC_TIME_SRC_SYSTEM_TIMETICK (14) – Time is derived from the system clock (better known as the slow clock); GNSS time is maintained irrespective of the GNSS receiver state • eQMI_LOC_TIME_SRC_QZSS_TOW_DECODE (15) – Time is set after decoding QZSS satellites • eQMI_LOC_TIME_SRC_BDS_TOW_DECODE (16) – Time is set after decoding BDS satellites
Type	0x2A			1	Sensor Data Usage Indicates whether sensor data was used in computing the position in this position report.
Length	8			2	
Value	→	mask32	usageMask	4	Specifies which sensors were used in calculating the position in the position report. Valid bitmasks: <ul style="list-style-type: none"> • 0x00000001 – SENSOR_USED_ACCEL • 0x00000002 – SENSOR_USED_GYRO
		mask32	aidingIndicatorMask	4	Specifies which results were aided by sensors. Valid bitmasks: <ul style="list-style-type: none"> • 0x00000001 – AIDED_HEADING • 0x00000002 – AIDED_SPEED • 0x00000004 – AIDED_POSITION • 0x00000008 – AIDED_VELOCITY
Type	0x2B			1	Fix Count for This Session
Length	4			2	
Value	→	uint32	fixId	4	Fix count for the session. Starts with 0 and increments by one for each successive position report for a particular session.
Type	0x2C			1	SVs Used to Calculate the Fix
Length	Var			2	
Value	→	uint8	gnssSvUsedList_len	1	Number of sets of the following elements: <ul style="list-style-type: none"> • gnssSvUsedList

Field	Field value	Field type	Parameter	Size (byte)	Description
		uint16	gnssSvUsedList	Var	Each entry in the list contains the SV ID of a satellite used for calculating this position report. The following information is associated with each SV ID: Range: <ul style="list-style-type: none"> • For GPS: 1 to 32 • For SBAS: 33 to 64 • For GLONASS: 65 to 96 • For QZSS: 193 to 197 • For BDS: 201 to 237
Type	0x2D			1	Altitude Assumed
Length	1			2	
Value	→	boolean	altitudeAssumed	1	Indicates whether altitude is assumed or calculated: <ul style="list-style-type: none"> • 0x00 (FALSE) – Altitude is calculated • 0x01 (TRUE) – Altitude is assumed; there may not be enough satellites to determine the precise altitude

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_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. The position reports are sent only to the control point that sent the QMI_LOC_START message that generated the position report.

3.9 QMI_LOC_EVENT_GNSS_SV_INFO

Sends a satellite report to the control point.

LOC message ID

0x0025

Version introduced

Major - 2, Minor - 0

3.9.1 Indication - QMI_LOC_EVENT_GNSS_SV_INFO_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version introduced	Version last modified
Altitude Assumed	2.0	2.0

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Altitude Assumed
Length	1			2	
Value	→	boolean	altitudeAssumed	1	Indicates whether altitude is assumed or calculated: <ul style="list-style-type: none"> 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 introduced	Version last modified
Satellite Info	2.0	2.22

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Satellite Info SV information list.

Field	Field value	Field type	Parameter	Size (byte)	Description
Length	Var			2	
Value	→	uint8	svList_len	1	Number of sets of the following elements: <ul style="list-style-type: none"> • validMask • system • gnssSvId • healthStatus • svStatus • svInfoMask • elevation • azimuth • snr
		mask32	validMask	4	Bitmask indicating which of the fields in this TLV are valid. Valid bitmasks: <ul style="list-style-type: none"> • 0x00000001 – VALID_SYSTEM • 0x00000002 – VALID_GNSS_SVID • 0x00000004 – VALID_HEALTH_STATUS • 0x00000008 – VALID_PROCESS_STATUS • 0x00000010 – VALID_SVINFORMASK • 0x00000020 – VALID_ELEVATION • 0x00000040 – VALID_AZIMUTH • 0x00000080 – VALID_SNR
		enum	system	4	Indicates to which constellation this SV belongs. Valid values: <ul style="list-style-type: none"> • eQMI_LOC_SV_SYSTEM_GPS (1) – GPS satellite • eQMI_LOC_SV_SYSTEM_GALILEO (2) – GALILEO satellite • eQMI_LOC_SV_SYSTEM_SBAS (3) – SBAS satellite • eQMI_LOC_SV_SYSTEM_COMPASS (4) – COMPASS satellite • eQMI_LOC_SV_SYSTEM_GLONASS (5) – GLONASS satellite • eQMI_LOC_SV_SYSTEM_BDS (6) – BDS satellite

Field	Field value	Field type	Parameter	Size (byte)	Description
		uint16	gnssSvId	2	GNSS SV ID. <ul style="list-style-type: none"> Range: <ul style="list-style-type: none"> For GPS: 1 to 32 For GLONASS: 1 to 32 For SBAS: 120 to 151 For BDS: 201 to 237 The GPS and GLONASS SVs can be disambiguated using the system field.
		uint8	healthStatus	1	Health status. <ul style="list-style-type: none"> Range: 0 to 1; 0 = unhealthy, 1 = healthy
		enum	svStatus	4	SV processing status. Valid values: <ul style="list-style-type: none"> eQMI_LOC_SV_STATUS_IDLE (1) – SV is not being actively processed eQMI_LOC_SV_STATUS_SEARCH (2) – The system is searching for this SV eQMI_LOC_SV_STATUS_TRACK (3) – SV is being tracked
		mask8	svInfoMask	1	Indicates whether almanac and ephemeris information is available. Valid bitmasks: <ul style="list-style-type: none"> 0x01 – SVINFO_HAS_EPHEMERIS 0x02 – SVINFO_HAS_ALMANAC
		float	elevation	4	SV elevation angle. <ul style="list-style-type: none"> Units: Degrees Range: 0 to 90
		float	azimuth	4	SV azimuth angle. <ul style="list-style-type: none"> Units: Degrees Range: 0 to 360
		float	snr	4	SV signal-to-noise ratio. <ul style="list-style-type: none"> 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

3.9.2 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. The satellite reports are sent only to the control point that sent the QMI_LOC_START message that generated the satellite report.

QUALCOMM®
2016-05-16 00:17:18 PDT
deon_zhang@askey.com.tw

3.10 QMI_LOC_EVENT_NMEA

Sends NMEA sentences to the control point

LOC message ID

0x0026

Version introduced

Major - 2, Minor - 0

3.10.1 Indication - QMI_LOC_EVENT_NMEA_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version introduced	Version last modified
NMEA String	2.0	2.1

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	NMEA String
Length	Var			2	
Value	→	string	nmea	Var	NMEA string. <ul style="list-style-type: none"> 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

3.10.2 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. The NMEA reports are sent only to the control point that sent the QMI_LOC_START message that generated the NMEA report.

QUALCOMM®
2016-05-16 00:17:18 PDT
deon_zhang@askey.com.tw

3.11 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 - 0

3.11.1 Indication - QMI_LOC_EVENT_NI_NOTIFY_VERIFY_REQ_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version introduced	Version last modified
Notification Type	2.1	2.1

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Notification Type
Length	4			2	
Value	→	enum	notificationType	4	Type of notification/verification performed. Valid values: <ul style="list-style-type: none"> • eQMI_LOC_NI_USER_NO_NOTIFY_NO_VERIFY (1) – No notification and no verification required • eQMI_LOC_NI_USER_NOTIFY_ONLY (2) – Notify only; no verification required • eQMI_LOC_NI_USER_NOTIFY_VERIFY_ALLOW_NO_RESP (3) – Notify and verify, but no response required. • eQMI_LOC_NI_USER_NOTIFY_VERIFY_NOT_ALLOW_NO_RESP (4) – Notify and verify, and require a response • eQMI_LOC_NI_USER_NOTIFY_VERIFY_PRIVACY_OVERRIDE (5) – Notify and verify; privacy override

Optional TLVs

Name	Version introduced	Version last modified
Network Initiated Vx Request	2.0	2.1
Network Initiated SUPL Request	2.0	2.2
Network Initiated UMTS Control Plane Request	2.0	2.2
Network Initiated Service Interaction Request	2.0	2.1
Network Initiated SUPL Version 2 Extension	2.10	2.10
SUPL Emergency Notification	2.17	2.17

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Network Initiated Vx Request Optional NI Vx request payload.
Length	Var			2	
Value	→	boolean	posQosIncl	1	Indicates whether quality of service is included: • 0x01 (TRUE) – QoS is included • 0x00 (FALSE) – QoS is not included
		uint8	posQos	1	Position QoS timeout. • Units: Seconds • Range: 0 to 255
		uint32	numFixes	4	Number of fixes allowed.
		uint32	timeBetweenFixes	4	Time between fixes. • Units: Seconds
		enum	posMode	4	Position mode. Valid values: • eQMI_LOC_NI_VX_MS_ASSISTED_ONLY (1) – MS-assisted only allowed • eQMI_LOC_NI_VX_MS_BASED_ONLY (2) – MS-based only allowed • eQMI_LOC_NI_VX_MS_ASSISTED_PREFERRED_MS_BASED_ALLOWED (3) – MS-assisted preferred, but MS-based allowed • eQMI_LOC_NI_VX_MS_BASED_PREFERRED_MS_ASSISTED_ALLOWED (4) – MS-based preferred, but MS-assisted allowed

Field	Field value	Field type	Parameter	Size (byte)	Description
		enum	encodingScheme	4	VX encoding scheme. Valid values: <ul style="list-style-type: none"> • eQMI_LOC_NI_VX_OCTET (0) – Encoding is Octet • eQMI_LOC_NI_VX_EXN_PROTOCOL_MSG (1) – Encoding is EXN protocol message • eQMI_LOC_NI_VX_ASCII (2) – Encoding is ASCII • eQMI_LOC_NI_VX_IA5 (3) – Encoding is IA5 • eQMI_LOC_NI_VX_UNICODE (4) – Encoding is Unicode • eQMI_LOC_NI_VX_SHIFT_JIS (5) – Encoding is Shift JIS • eQMI_LOC_NI_VX_KOREAN (6) – Encoding is Korean • eQMI_LOC_NI_VX_LATIN_HEBREW (7) – Encoding is Latin Hebrew • eQMI_LOC_NI_VX_LATIN (8) – Encoding is Latin • eQMI_LOC_NI_VX_GSM (9) – Encoding is GSM
		uint8	requestorId_len	1	Number of sets of the following elements: <ul style="list-style-type: none"> • requestorId
		uint8	requestorId	Var	Requestor ID. <ul style="list-style-type: none"> • Type: Array of bytes • Maximum array length: 200
		uint16	userRespTimerInSeconds	2	Time to wait for the user to respond. <ul style="list-style-type: none"> • Units: Seconds
Type	0x11			1	Network Initiated SUPL Request Optional NI SUPL request payload.
Length	Var			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	mask32	valid_flags	4	Indicates which of the following fields are present in this value. Valid bitmasks: <ul style="list-style-type: none"> • 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
		mask8	suplServerAddrTypeMask	1	Mask specifying the valid fields in this value. Valid bitmasks: <ul style="list-style-type: none"> • 0x01 – IPv4 • 0x02 – IPv6 • 0x04 – URL
		uint32	addr	4	IPv4 address.
		uint16	port	2	IPv4 port.
		uint16	addr	16	IPv6 address. <ul style="list-style-type: none"> • Type: Array of unsigned integers • Maximum length of the array: 8
		uint32	port	4	IPv6 port.
		uint8	urlAddr_len	1	Number of sets of the following elements: <ul style="list-style-type: none"> • urlAddr
		string	urlAddr	Var	URL. <ul style="list-style-type: none"> • Type: NULL-terminated string • Maximum string length (including NULL terminator): 256
		uint8	suplSessionId	4	SUPL session ID. <ul style="list-style-type: none"> • Type: Array of unsigned integers • Maximum length of the array: 4
		uint8	suplHash	8	Hash for SUPL_INIT; used to validate that the message was not corrupted. <ul style="list-style-type: none"> • Type: Array of unsigned integers • Length of the array: 8

Field	Field value	Field type	Parameter	Size (byte)	Description
		enum	posMethod	4	<p>GPS mode to be used for the fix.</p> <p>Valid values:</p> <ul style="list-style-type: none"> •eQMI_LOC_NI_SUPL_POSMETHOD_AGPS_SETASSISTED (1) – Set assisted •eQMI_LOC_NI_SUPL_POSMETHOD_AGPS_SETBASED (2) – Set based •eQMI_LOC_NI_SUPL_POSMETHOD_AGPS_SETASSISTED_PREF (3) – Set assisted preferred •eQMI_LOC_NI_SUPL_POSMETHOD_AGPS_SETBASED_PREF (4) – Set based preferred •eQMI_LOC_NI_SUPL_POSMETHOD_AUTONOMOUS_GPS (5) – Standalone GPS •eQMI_LOC_NI_SUPL_POSMETHOD_AFLT (6) – Advanced forward link trilateration •eQMI_LOC_NI_SUPL_POSMETHOD_ECID (7) – Exclusive chip ID •eQMI_LOC_NI_SUPL_POSMETHOD_EOTD (8) – Enhanced observed time difference •eQMI_LOC_NI_SUPL_POSMETHOD_OTDOA (9) – Observed time delay of arrival •eQMI_LOC_NI_SUPL_POSMETHOD_NO_POSITION (10) – No position

Field	Field value	Field type	Parameter	Size (byte)	Description
		enum	dataCodingScheme	4	<p>Data coding scheme applies to both the requestor ID and the client name.</p> <p>Valid values:</p> <ul style="list-style-type: none"> • eQMI_LOC_NI_SS_GERMAN (12) – Language is German • eQMI_LOC_NI_SS_ENGLISH (13) – Language is English • eQMI_LOC_NI_SS_ITALIAN (14) – Language is Italian • eQMI_LOC_NI_SS_FRENCH (15) – Language is French • eQMI_LOC_NI_SS_SPANISH (16) – Language is Spanish • eQMI_LOC_NI_SS_DUTCH (17) – Language is Dutch • eQMI_LOC_NI_SS_SWEDISH (18) – Language is Swedish • eQMI_LOC_NI_SS_DANISH (19) – Language is Danish • eQMI_LOC_NI_SS_PORTUGUESE (20) – Language is Portuguese • eQMI_LOC_NI_SS_FINNISH (21) – Language is Finnish • eQMI_LOC_NI_SS_NORWEGIAN (22) – Language is Norwegian • eQMI_LOC_NI_SS_GREEK (23) – Language is Greek • eQMI_LOC_NI_SS_TURKISH (24) – Language is Turkish • eQMI_LOC_NI_SS_HUNGARIAN (25) – Language is Hungarian • eQMI_LOC_NI_SS_POLISH (26) – Language is Polish • eQMI_LOC_NI_SS_LANGUAGE_UNSPEC (27) – Language is unspecified • eQMI_LOC_NI_SUPL_UTF8 (28) – Encoding is UTF 8 • eQMI_LOC_NI_SUPL_UCS2 (29) – Encoding is UCS 2 • eQMI_LOC_NI_SUPL_GSM_DEFAULT (30) – Encoding is GSM default

Field	Field value	Field type	Parameter	Size (byte)	Description
		enum	formatType	4	Format of the formatted string. Valid values: <ul style="list-style-type: none"> • eQMI_LOC_NI_SUPL_FORMAT_LOGICAL_NAME (0) – SUPL logical name format • eQMI_LOC_NI_SUPL_FORMAT_EMAIL_ADDRESS (1) – SUPL email address format • eQMI_LOC_NI_SUPL_FORMAT_MSISDN (2) – SUPL MS-ISDN format • eQMI_LOC_NI_SUPL_FORMAT_URL (3) – SUPL URL format • eQMI_LOC_NI_SUPL_FORMAT_SIP_URL (4) – SUPL SIP URL format • eQMI_LOC_NI_SUPL_FORMAT_MIN (5) – SUPL MIN format • eQMI_LOC_NI_SUPL_FORMAT_MDN (6) – SUPL MDN format • eQMI_LOC_NI_SUPL_FORMAT_IMSPUBLIC_IDENTITY (7) – SUPL IMS public identity • eQMI_LOC_NI_SUPL_FORMAT_OSS_UNKNOWN (2147483647) – SUPL unknown format
		uint8	formattedString_len	1	Number of sets of the following elements: <ul style="list-style-type: none"> • formattedString
		uint8	formattedString	Var	Formatted string. <ul style="list-style-type: none"> • Type: Byte array • Maximum string length: 64

Field	Field value	Field type	Parameter	Size (byte)	Description
		enum	formatType	4	Format of the formatted string. Valid values: <ul style="list-style-type: none"> • eQMI_LOC_NI_SUPL_FORMAT_LOGICAL_NAME (0) – SUPL logical name format • eQMI_LOC_NI_SUPL_FORMAT_EMAIL_ADDRESS (1) – SUPL email address format • eQMI_LOC_NI_SUPL_FORMAT_MSISDN (2) – SUPL MS-ISDN format • eQMI_LOC_NI_SUPL_FORMAT_URL (3) – SUPL URL format • eQMI_LOC_NI_SUPL_FORMAT_SIP_URL (4) – SUPL SIP URL format • eQMI_LOC_NI_SUPL_FORMAT_MIN (5) – SUPL MIN format • eQMI_LOC_NI_SUPL_FORMAT_MDN (6) – SUPL MDN format • eQMI_LOC_NI_SUPL_FORMAT_IMSPUBLIC_IDENTITY (7) – SUPL IMS public identity • eQMI_LOC_NI_SUPL_FORMAT_OSS_UNKNOWN (2147483647) – SUPL unknown format
		uint8	formattedString_len	1	Number of sets of the following elements: <ul style="list-style-type: none"> • formattedString
		uint8	formattedString	Var	Formatted string. <ul style="list-style-type: none"> • Type: Byte array • Maximum string length: 64
		mask8	validMask	1	Bit field indicating which fields are valid in this value. Valid bitmasks: <ul style="list-style-type: none"> • 0x01 – QOP_HORZ_ACC_VALID • 0x02 – QOP_VER_ACC_VALID • 0x04 – QOP_MAXAGE_VALID • 0x08 – QOP_DELAY_VALID
		uint8	horizontalAccuracy	1	Horizontal accuracy. <ul style="list-style-type: none"> • Units: Meters
		uint8	verticalAccuracy	1	Vertical accuracy. <ul style="list-style-type: none"> • Units: Meters
		uint16	maxLocAge	2	Maximum age of the location if the engine sends a previously computed position. <ul style="list-style-type: none"> • Units: Seconds
		uint8	delay	1	Delay the server is willing to tolerate for the fix. <ul style="list-style-type: none"> • Units: Seconds

Field	Field value	Field type	Parameter	Size (byte)	Description
		uint16	userResponseTimer	2	Time to wait for the user to respond. • Units: Seconds
Type	0x12			1	Network Initiated UMTS Control Plane Request Optional NI UMTS-CP request payload.
Length	Var			2	
Value	→	mask16	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
		uint8	invokeId	1	Supplementary Services invoke ID.
		enum	dataCodingScheme	4	Type of data encoding scheme for the text. Applies to both the notification text and the client address. Valid values: • eQMI_LOC_NI_SS_GERMAN (12) – Language is German • eQMI_LOC_NI_SS_ENGLISH (13) – Language is English • eQMI_LOC_NI_SS_ITALIAN (14) – Language is Italian • eQMI_LOC_NI_SS_FRENCH (15) – Language is French • eQMI_LOC_NI_SS_SPANISH (16) – Language is Spanish • eQMI_LOC_NI_SS_DUTCH (17) – Language is Dutch • eQMI_LOC_NI_SS_SWEDISH (18) – Language is Swedish • eQMI_LOC_NI_SS_DANISH (19) – Language is Danish • eQMI_LOC_NI_SS_PORTUGUESE (20) – Language is Portuguese • eQMI_LOC_NI_SS_FINNISH (21) – Language is Finnish

Field	Field value	Field type	Parameter	Size (byte)	Description
			dataCodingScheme (cont.)		<ul style="list-style-type: none"> • eQMI_LOC_NI_SS_NORWEGIAN (22) – Language is Norwegian • eQMI_LOC_NI_SS_GREEK (23) – Language is Greek • eQMI_LOC_NI_SS_TURKISH (24) – Language is Turkish • eQMI_LOC_NI_SS_HUNGARIAN (25) – Language is Hungarian • eQMI_LOC_NI_SS_POLISH (26) – Language is Polish • eQMI_LOC_NI_SS_LANGUAGE_UNSPEC (27) – Language is unspecified • eQMI_LOC_NI_SUPL_UTF8 (28) – Encoding is UTF 8 • eQMI_LOC_NI_SUPL_UCS2 (29) – Encoding is UCS 2 • eQMI_LOC_NI_SUPL_GSM_DEFAULT (30) – Encoding is GSM default
		uint8	notificationText_len	1	Number of sets of the following elements: <ul style="list-style-type: none"> • notificationText
		uint8	notificationText	Var	Notification text; the encoding method is specified in dataCodingScheme. <ul style="list-style-type: none"> • Type: Array of bytes • Maximum array length: 64
		uint8	clientAddress_len	1	Number of sets of the following elements: <ul style="list-style-type: none"> • clientAddress
		uint8	clientAddress	Var	Client address; the encoding method is specified in dataCodingScheme. <ul style="list-style-type: none"> • Maximum array length: 20
		enum	locationType	4	Location type. Valid values: <ul style="list-style-type: none"> • eQMI_LOC_NI_LOCATIONTYPE_CURRENT_LOCATION (1) – Current location • eQMI_LOC_NI_LOCATIONTYPE_CURRENT_OR_LAST_KNOWN_LOCATION (2) – Last known location; may be the current location • eQMI_LOC_NI_LOCATIONTYPE_INITIAL_LOCATION (3) – Initial location

Field	Field value	Field type	Parameter	Size (byte)	Description
		enum	dataCodingScheme	4	Identifies the coding scheme of the coded string. Valid values: <ul style="list-style-type: none"> • eQMI_LOC_NI_SS_GERMAN (12) – Language is German • eQMI_LOC_NI_SS_ENGLISH (13) – Language is English • eQMI_LOC_NI_SS_ITALIAN (14) – Language is Italian • eQMI_LOC_NI_SS_FRENCH (15) – Language is French • eQMI_LOC_NI_SS_SPANISH (16) – Language is Spanish • eQMI_LOC_NI_SS_DUTCH (17) – Language is Dutch • eQMI_LOC_NI_SS_SWEDISH (18) – Language is Swedish • eQMI_LOC_NI_SS_DANISH (19) – Language is Danish • eQMI_LOC_NI_SS_PORTUGUESE (20) – Language is Portuguese • eQMI_LOC_NI_SS_FINNISH (21) – Language is Finnish • eQMI_LOC_NI_SS_NORWEGIAN (22) – Language is Norwegian • eQMI_LOC_NI_SS_GREEK (23) – Language is Greek • eQMI_LOC_NI_SS_TURKISH (24) – Language is Turkish • eQMI_LOC_NI_SS_HUNGARIAN (25) – Language is Hungarian • eQMI_LOC_NI_SS_POLISH (26) – Language is Polish • eQMI_LOC_NI_SS_LANGUAGE_UNSPEC (27) – Language is unspecified • eQMI_LOC_NI_SUPL_UTF8 (28) – Encoding is UTF 8 • eQMI_LOC_NI_SUPL_UCS2 (29) – Encoding is UCS 2 • eQMI_LOC_NI_SUPL_GSM_DEFAULT (30) – Encoding is GSM default
		uint8	codedString_len	1	Number of sets of the following elements: <ul style="list-style-type: none"> • codedString
		uint8	codedString	Var	Coded string. <ul style="list-style-type: none"> • Type: Array of bytes • Maximum string length: 20

Field	Field value	Field type	Parameter	Size (byte)	Description
		enum	dataCodingScheme	4	Identifies the coding scheme of the coded string. Valid values: <ul style="list-style-type: none"> • eQMI_LOC_NI_SS_GERMAN (12) – Language is German • eQMI_LOC_NI_SS_ENGLISH (13) – Language is English • eQMI_LOC_NI_SS_ITALIAN (14) – Language is Italian • eQMI_LOC_NI_SS_FRENCH (15) – Language is French • eQMI_LOC_NI_SS_SPANISH (16) – Language is Spanish • eQMI_LOC_NI_SS_DUTCH (17) – Language is Dutch • eQMI_LOC_NI_SS_SWEDISH (18) – Language is Swedish • eQMI_LOC_NI_SS_DANISH (19) – Language is Danish • eQMI_LOC_NI_SS_PORTUGUESE (20) – Language is Portuguese • eQMI_LOC_NI_SS_FINNISH (21) – Language is Finnish • eQMI_LOC_NI_SS_NORWEGIAN (22) – Language is Norwegian • eQMI_LOC_NI_SS_GREEK (23) – Language is Greek • eQMI_LOC_NI_SS_TURKISH (24) – Language is Turkish • eQMI_LOC_NI_SS_HUNGARIAN (25) – Language is Hungarian • eQMI_LOC_NI_SS_POLISH (26) – Language is Polish • eQMI_LOC_NI_SS_LANGUAGE_UNSPEC (27) – Language is unspecified • eQMI_LOC_NI_SUPL_UTF8 (28) – Encoding is UTF 8 • eQMI_LOC_NI_SUPL_UCS2 (29) – Encoding is UCS 2 • eQMI_LOC_NI_SUPL_GSM_DEFAULT (30) – Encoding is GSM default
		uint8	codedString_len	1	Number of sets of the following elements: <ul style="list-style-type: none"> • codedString
		uint8	codedString	Var	Coded string. <ul style="list-style-type: none"> • Type: Array of bytes • Maximum string length: 20

Field	Field value	Field type	Parameter	Size (byte)	Description
		uint8	lcsServiceTypeId	1	Service type ID.
		uint16	userResponseTimer	2	Time to wait for the user to respond. • Units: Seconds
Type	0x13			1	Network Initiated Service Interaction Request Optional NI service interaction payload.
Length	Var			2	
Value	→	boolean	posQosIncl	1	Indicates whether quality of service is included: • 0x01 (TRUE) – QoS is included • 0x00 (FALSE) – QoS is not included
		uint8	posQos	1	Position QoS timeout. • Units: Seconds • Range: 0 to 255
		uint32	numFixes	4	Number of fixes allowed.
		uint32	timeBetweenFixes	4	Time between fixes. • Units: Seconds
		enum	posMode	4	Position mode. Valid values: • eQMI_LOC_NI_VX_MS_ASSISTED_ONLY (1) – MS-assisted only allowed • eQMI_LOC_NI_VX_MS_BASED_ONLY (2) – MS-based only allowed • eQMI_LOC_NI_VX_MS_ASSISTED_PREFERRED_MS_BASED_ALLOWED (3) – MS-assisted preferred, but MS-based allowed • eQMI_LOC_NI_VX_MS_BASED_PREFERRED_MS_ASSISTED_ALLOWED (4) – MS-based preferred, but MS-assisted allowed

Field	Field value	Field type	Parameter	Size (byte)	Description
		enum	encodingScheme	4	VX encoding scheme. Valid values: <ul style="list-style-type: none"> • eQMI_LOC_NI_VX_OCTET (0) – Encoding is Octet • eQMI_LOC_NI_VX_EXN_PROTOCOL_MSG (1) – Encoding is EXN protocol message • eQMI_LOC_NI_VX_ASCII (2) – Encoding is ASCII • eQMI_LOC_NI_VX_IA5 (3) – Encoding is IA5 • eQMI_LOC_NI_VX_UNICODE (4) – Encoding is Unicode • eQMI_LOC_NI_VX_SHIFT_JIS (5) – Encoding is Shift JIS • eQMI_LOC_NI_VX_KOREAN (6) – Encoding is Korean • eQMI_LOC_NI_VX_LATIN_HEBREW (7) – Encoding is Latin Hebrew • eQMI_LOC_NI_VX_LATIN (8) – Encoding is Latin • eQMI_LOC_NI_VX_GSM (9) – Encoding is GSM
		uint8	requestorId_len	1	Number of sets of the following elements: <ul style="list-style-type: none"> • requestorId
		uint8	requestorId	Var	Requestor ID. <ul style="list-style-type: none"> • Type: Array of bytes • Maximum array length: 200
		uint16	userRespTimerInSeconds	2	Time to wait for the user to respond. <ul style="list-style-type: none"> • Units: Seconds
		enum	serviceInteractionType	4	Service interaction type specified in qmiLocNiServiceInteractionEnumT. Valid values: <ul style="list-style-type: none"> • eQMI_LOC_NI_SERVICE_INTERACTION_ONGOING_NI_INCOMING_MO (1) – Service interaction between ongoing NI and incoming MO sessions.
Type	0x14			1	Network Initiated SUPL Version 2 Extension Optional NI SUPL Version 2 Extension payload. When present, this payload is to be used in conjunction with the SUPL indication payload.
Length	8			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	mask16	supportedNetworksMask	2	<p>Specifies which type of network measurements are allowed to be sent as part of the Location ID or Multiple Location IDs parameter in the SUPL_POS_INIT message (see [S4]).</p> <p>Valid bitmasks:</p> <ul style="list-style-type: none"> • 0x0001 – SUPPORTED_NETWORK_WLAN • 0x0002 – SUPPORTED_NETWORK_GSM • 0x0004 – SUPPORTED_NETWORK_WCDMA • 0x0008 – SUPPORTED_NETWORK_CDMA • 0x0010 – SUPPORTED_NETWORK_HRDP • 0x0020 – SUPPORTED_NETWORK_UMB • 0x0040 – SUPPORTED_NETWORK_LTE • 0x0080 – SUPPORTED_NETWORK_WIMAX • 0x0100 – SUPPORTED_NETWORK_HISTORIC • 0x0200 – SUPPORTED_NETWORK_NONSVRV
		enum	triggerType	4	<p>Specifies the type of session trigger requested in the SUPL_POS_INIT message (refer to [S4]).</p> <p>Valid values:</p> <ul style="list-style-type: none"> • eQMI_LOC_SUPL_VER_2_EXT_TRIGGER_TYPE_SINGLE_SHOT (-1) – SUPL INIT message indicates a request for a single shot triggered session • eQMI_LOC_SUPL_VER_2_EXT_TRIGGER_TYPE_PERIODIC (0) – SUPL INIT message indicates a request for a periodic triggered session • eQMI_LOC_SUPL_VER_2_EXT_TRIGGER_TYPE_AREA_EVENT (1) – SUPL INIT message indicates a request for an area event triggered session

Field	Field value	Field type	Parameter	Size (byte)	Description
		mask16	gnssType	2	Specifies which GNSS technologies are allowed as positioning technologies. Valid bitmasks: <ul style="list-style-type: none"> • 0x0001 – GNSS_GPS • 0x0002 – GNSS_GLONASS • 0x0004 – GNSS_GALILEO • 0x0008 – GNSS_SBAS • 0x0010 – GNSS_QZSS • 0x0020 – GNSS_MODERN_GPS
Type	0x15			1	SUPL Emergency Notification This specifies that the corresponding NI notification is an emergency notification. Emergency notification can be given even without an Emergency SUPL Location Platform (ESLP) address.
Length	Var			2	
Value	→	uint8	eslpUrl_len	1	Number of sets of the following elements: <ul style="list-style-type: none"> • eslpUrl
		string	eslpUrl	Var	ESLP URL. Maximum length: 255 bytes

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_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. A client that receives this event is expected to send the QMI_LOC_INFORM_NI_USER_RESPONSE message containing the user response to the notify/verify request. Only one response may be sent by the client per NI notify/verify request. It is safe if multiple clients register for this event, as long as only one client responds to a particular NI request.

3.12 QMI_LOC_EVENT_INJECT_TIME_REQ

Requests the control point to inject time information.

LOC message ID

0x0028

Version introduced

Major - 2, Minor - 0

3.12.1 Indication - QMI_LOC_EVENT_INJECT_TIME_REQ_IND

Message type

Indication

Sender

Service

Mandatory TLVs

None

Optional TLVs

Name	Version introduced	Version last modified
Time Server Info	2.1	2.1

Field	Field value	Field type	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	→	uint32	delayThreshold	4	The time server is to be skipped if a one-way delay to the server exceeds this threshold. • Units: Milliseconds
		uint8	timeServerList_len	1	Number of sets of the following elements: • serverUrl_len • serverUrl
		uint8	serverUrl_len	1	Number of sets of the following elements: • serverUrl

Field	Field value	Field type	Parameter	Size (byte)	Description
		string	serverUrl	Var	Assistance server URL. <ul style="list-style-type: none"> • 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

3.12.2 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. A control point must send the QMI_LOC_INJECT_UTC_TIME to satisfy this request.

3.13 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 - 0

3.13.1 Indication - QMI_LOC_EVENT_INJECT_PREDICTED_ORBITS_REQ_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version introduced	Version last modified
Allowed Sizes	2.0	2.0

Field	Field value	Field type	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	→	uint32	maxFileSizeInBytes	4	Maximum allowable predicted orbits file size (in bytes).
		uint32	maxPartSize	4	Maximum allowable predicted orbits file chunk size (in bytes).

Optional TLVs

Name	Version introduced	Version last modified
Server List	2.0	2.1

Field	Field value	Field type	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	→	uint8	serverList_len	1	Number of sets of the following elements: • serverUrl_len • serverUrl
		uint8	serverUrl_len	1	Number of sets of the following elements: • serverUrl
		string	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

3.13.2 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. A client can satisfy this request from the service by sending the QMI_LOC_INJECT_PREDICTED_ORBITS_DATA message. It is not safe for multiple clients to inject predicted orbits data into the engine, hence only one client may handle this request.

3.14 QMI_LOC_EVENT_INJECT_POSITION_REQ

Requests the control point to inject a position.

LOC message ID

0x002A

Version introduced

Major - 2, Minor - 0

3.14.1 Indication - QMI_LOC_EVENT_INJECT_POSITION_REQ_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version introduced	Version last modified
Latitude	2.0	2.2
Longitude	2.0	2.0
Circular Horizontal Uncertainty	2.0	2.0
UTC Timestamp	2.0	2.2

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Latitude
Length	8			2	
Value	→	double	latitude	8	Latitude (specified in WGS84 datum). <ul style="list-style-type: none"> Type: Floating point Units: Degrees Range: -90.0 to 90.0 <ul style="list-style-type: none"> Positive values indicate northern latitude Negative values indicate southern latitude
Type	0x02			1	Longitude
Length	8			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	double	longitude	8	Longitude (specified in WGS84 datum). <ul style="list-style-type: none"> • Type: Floating point • Units: Degrees • Range: -180.0 to 180.0 <ul style="list-style-type: none"> – Positive values indicate eastern longitude – Negative values indicate western longitude
Type	0x03			1	Circular Horizontal Uncertainty
Length	4			2	
Value	→	float	horUncCircular	4	Horizontal position uncertainty (circular). <ul style="list-style-type: none"> • Units: Meters
Type	0x04			1	UTC Timestamp
Length	8			2	
Value	→	uint64	timestampUtc	8	UTC timestamp. <ul style="list-style-type: none"> • 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

3.14.2 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. The client must send QMI_LOC_INJECT_POSITION to satisfy this request from the service.

3.15 QMI_LOC_EVENT_ENGINE_STATE

Sends the engine state to the control point.

LOC message ID

0x002B

Version introduced

Major - 2, Minor - 0

3.15.1 Indication - QMI_LOC_EVENT_ENGINE_STATE_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version introduced	Version last modified
Engine State	2.0	2.1

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Engine State
Length	4			2	
Value	→	enum	engineState	4	Location engine state. Valid values: • eQMI_LOC_ENGINE_STATE_ON (1) – Location engine is on • eQMI_LOC_ENGINE_STATE_OFF (2) – Location engine is 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

3.15.2 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

QUALCOMM®
2016-05-16 00:17:18 PDT
deon_zhang@askey.com.tw

3.16 QMI_LOC_EVENT_FIX_SESSION_STATE

Sends the fix session state to the control point.

LOC message ID

0x002C

Version introduced

Major - 2, Minor - 0

3.16.1 Indication - QMI_LOC_EVENT_FIX_SESSION_STATE_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version introduced	Version last modified
Session State	2.0	2.1

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Session State
Length	4			2	
Value	→	enum	sessionState	4	LOC fix session state. Valid values: • eQMI_LOC_FIX_SESSION_STARTED (1) – Location fix session has started • eQMI_LOC_FIX_SESSION_FINISHED (2) – Location fix session has ended

Optional TLVs

Name	Version introduced	Version last modified
Session ID	2.0	2.0

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Session ID
Length	1			2	
Value	→	uint8	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. • 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

3.16.2 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. This event is only sent to the client that sent the QMI_LOC_START_REQ message that generated this event.

3.17 QMI_LOC_EVENT_WIFI_REQ

Sends a Wi-Fi request to the control point.

LOC message ID

0x002D

Version introduced

Major - 2, Minor - 0

3.17.1 Indication - QMI_LOC_EVENT_WIFI_REQ_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version introduced	Version last modified
Request Type	2.0	2.1

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Request Type
Length	4			2	
Value	→	enum	requestType	4	Request type. Valid values: • eQMI_LOC_WIFI_START_PERIODIC_HI_FREQ_FIXES (0) – Start periodic fixes with high frequency • eQMI_LOC_WIFI_START_PERIODIC_KEEP_WARM (1) – Keep warm for low frequency fixes without data downloads • eQMI_LOC_WIFI_STOP_PERIODIC_FIXES (2) – Stop periodic fixes request

Optional TLVs

Name	Version introduced	Version last modified
Time Between Fixes	2.0	2.0

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Time Between Fixes
Length	2			2	
Value	→	uint16	tbfInMs	2	Time between fixes for a periodic request. • 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

3.17.2 Description of QMI_LOC_EVENT_WIFI_REQ

This command sends a Wi-Fi 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. A client can send the QMI_LOC_INJECT_WIFI_POSITION_REQ message to satisfy this request from the service.

3.18 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

3.18.1 Indication - QMI_LOC_EVENT_SENSOR_STREAMING_READY_STATUS_IND

Message type

Indication

Sender

Service

Mandatory TLVs

None

Optional TLVs

Name	Version introduced	Version last modified
Accelerometer Accept Ready	2.0	2.2
Gyroscope Accept Ready	2.0	2.2
Accelerometer Temperature Accept Ready	2.17	2.17
Gyroscope Temperature Accept Ready	2.17	2.17

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

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	boolean	injectEnable	1	Indicates whether the GNSS location engine is ready to accept data from this sensor. Valid values: <ul style="list-style-type: none"> • 0x01 (TRUE) – GNSS location engine is ready to accept sensor data • 0x00 (FALSE) – GNSS location engine is not ready to accept sensor data
		uint16	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: $\text{samplingFrequency} = \text{samplesPerBatch} * \text{batchesPerSecond}$ samplesPerBatch must be a nonzero positive value.
		uint16	batchesPerSecond	2	Number of sensor-data batches the GNSS location engine is to receive per second. The rate is specified in an integral number of batches per second (Hz). batchesPerSecond must be a nonzero positive value.
Type	0x11			1	Gyroscope Accept Ready Indicates whether the GNSS location engine is ready to accept gyroscope sensor data.
Length	5			2	
Value	→	boolean	injectEnable	1	Indicates whether the GNSS location engine is ready to accept data from this sensor. Valid values: <ul style="list-style-type: none"> • 0x01 (TRUE) – GNSS location engine is ready to accept sensor data • 0x00 (FALSE) – GNSS location engine is not ready to accept sensor data

Field	Field value	Field type	Parameter	Size (byte)	Description
		uint16	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 nonzero positive value.
		uint16	batchesPerSecond	2	Number of sensor-data batches the GNSS location engine is to receive per second. The rate is specified in an integral number of batches per second (Hz). batchesPerSecond must be a nonzero positive value.
Type	0x12			1	Accelerometer Temperature Accept Ready Indicates whether the GNSS location engine is ready to accept accelerometer temperature data.
Length	5			2	
Value	→	boolean	injectEnable	1	Indicates whether the GNSS location engine is ready to accept data from this sensor. Valid values: <ul style="list-style-type: none"> • 0x01 (TRUE) – GNSS location engine is ready to accept sensor data • 0x00 (FALSE) – GNSS location engine is not ready to accept sensor data
		uint16	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 nonzero positive value.

Field	Field value	Field type	Parameter	Size (byte)	Description
		uint16	batchesPerSecond	2	Number of sensor-data batches the GNSS location engine is to receive per second. The rate is specified in an integral number of batches per second (Hz). batchesPerSecond must be a nonzero positive value.
Type	0x13			1	Gyroscope Temperature Accept Ready Indicates whether the GNSS location engine is ready to accept gyroscope temperature data.
Length	5			2	
Value	→	boolean	injectEnable	1	Indicates whether the GNSS location engine is ready to accept data from this sensor. Valid values: <ul style="list-style-type: none"> • 0x01 (TRUE) – GNSS location engine is ready to accept sensor data • 0x00 (FALSE) – GNSS location engine is not ready to accept sensor data
		uint16	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: $\text{samplingFrequency} = \text{samplesPerBatch} * \text{batchesPerSecond}$ samplesPerBatch must be a nonzero positive value.
		uint16	batchesPerSecond	2	Number of sensor-data batches the GNSS location engine is to receive per second. The rate is specified in an integral number of batches per second (Hz). batchesPerSecond must be a nonzero 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

3.18.2 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.

QUALCOMM®
2016-05-16 00:17:18 PDT
deon_zhang@askey.com.tw

3.19 QMI_LOC_EVENT_TIME_SYNC_REQ

Notifies the control point to inject time synchronization data.

LOC message ID

0x002F

Version introduced

Major - 2, Minor - 0

3.19.1 Indication - QMI_LOC_EVENT_TIME_SYNC_REQ_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version introduced	Version last modified
Opaque Time Sync Reference Counter	2.0	2.0

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Opaque Time Sync Reference Counter
Length	4			2	
Value	→	uint32	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.

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_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. The control point is expected to send the QMI_LOC_INJECT_TIME_SYNC_DATA_REQ message to satisfy this request from the service.

3.20 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 - 0

3.20.1 Indication - QMI_LOC_EVENT_SET_SPI_STREAMING_REPORT - IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version introduced	Version last modified
Enable/Disable SPI Requests	2.0	2.0

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Enable/Disable SPI Requests
Length	1			2	
Value	→	boolean	enable	1	Indicates whether the client is to start or stop sending an SPI status stream. <ul style="list-style-type: none"> • 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

3.20.2 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. The client is expected to use the QMI_LOC_SET_SPI_STATUS message to inject SPI reports.

3.21 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 - 1

3.21.1 Indication - QMI_LOC_EVENT_LOCATION_SERVER_CONNECTION_REQ_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version introduced	Version last modified
Connection Handle	2.1	2.1
Request Type	2.1	2.1
WWAN Type	2.1	2.20

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Connection Handle
Length	4			2	
Value	→	uint32	connHandle	4	Identifies a connection across Open and Close request events.
Type	0x02			1	Request Type
Length	4			2	
Value	→	enum	requestType	4	Open or close a connection to the location server. Valid values: • eQMI_LOC_SERVER_REQUEST_OPEN (1) – Open a connection to the location server • eQMI_LOC_SERVER_REQUEST_CLOSE (2) – Close a connection to the location server

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x03			1	WWAN Type
Length	4			2	
Value	→	enum	wwanType	4	Identifies the WWAN type for this request. Valid values: <ul style="list-style-type: none"> • eQMI_LOC_WWAN_TYPE_INTERNET (0) – Bring up the WWAN type used for an Internet connection • eQMI_LOC_WWAN_TYPE_AGNSS (1) – Bring up the WWAN type used for AGNSS connections • eQMI_LOC_WWAN_TYPE_AGNSS_EMERGENCY (2) – Bring up the WWAN type used for AGNSS Emergency connections

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

3.21.2 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. The client is expected to send the QMI_LOC_INFORM_LOCATION_SERVER_CONN_STATUS message to inform the service whether this request was successful. It is recommended that only one client respond to this request.

3.22 QMI_LOC_GET_SERVICE_REVISION

Client can query the service revision using this message.

LOC message ID

0x0032

Version introduced

Major - 2, Minor - 0

3.22.1 Request - QMI_LOC_GET_SERVICE_REVISION_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

None

Optional TLVs

None

3.22.2 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 introduced	Version last modified
Get Revision Status	2.0	2.28
Interface Definition Minor Revision	2.0	2.0

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Get Revision Status
Length	4			2	
Value	→	enum	status	4	<p>Status of the Get Revision request.</p> <p>Valid values:</p> <ul style="list-style-type: none"> • eQMI_LOC_SUCCESS (0) – Request was completed successfully • eQMI_LOC_GENERAL_FAILURE (1) – Request failed because of a general failure • eQMI_LOC_UNSUPPORTED (2) – Request failed because it is not supported • eQMI_LOC_INVALID_PARAMETER (3) – Request failed because it contained invalid parameters • eQMI_LOC_ENGINE_BUSY (4) – Request failed because the engine is busy • eQMI_LOC_PHONE_OFFLINE (5) – Request failed because the phone is offline • eQMI_LOC_TIMEOUT (6) – Request failed because it timed out • eQMI_LOC_CONFIG_NOT_SUPPORTED (7) – Request failed because an undefined configuration was requested • eQMI_LOC_INSUFFICIENT_MEMORY (8) – Request failed because the engine could not allocate sufficient memory for the request • eQMI_LOC_MAX_GEOFENCE_PROGRAMMED (9) – Request failed because the maximum number of Geofences are already programmed • eQMI_LOC_XTRA_VERSION_CHECK_FAILURE (10) – Location service failed because of an XTRA version-based file format check failure
Type	0x02			1	Interface Definition Minor Revision
Length	4			2	
Value	→	uint32	revision	4	<p>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.</p>

Optional TLVs

Name	Version introduced	Version last modified
GNSS Measurement Engine Firmware Version String	2.8	2.8
GNSS Hosted Software Version String	2.8	2.8
GNSS Software Version String	2.8	2.8

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	GNSS Measurement Engine Firmware Version String
Length	Var			2	
Value	→	string	gnssMeFWVerString	Var	<p>Version of the GNSS measurement engine software running under the LOC API.</p> <ul style="list-style-type: none"> Type: NULL-terminated string Maximum string length (including NULL terminator): 128 <p>Note: This string is only provided on platforms that have a measurement engine that supports this version string. On all other platforms, this optional TLV is not provided.</p>
Type	0x11			1	GNSS Hosted Software Version String
Length	Var			2	
Value	→	string	gnssHostSWVerString	Var	<p>Version of the GNSS hosted software running under the LOC API.</p> <ul style="list-style-type: none"> Type: NULL-terminated string Maximum string length (including NULL terminator): 128 <p>Note: This string is only provided on hosted architectures (measurement and position engine running on different processors) that support this version string. On all other platforms, this optional TLV is not provided.</p>
Type	0x12			1	GNSS Software Version String
Length	Var			2	
Value	→	string	gnssSWVerString	Var	<p>Aggregate version of the GNSS software.</p> <ul style="list-style-type: none"> 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

3.22.3 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. This command can be safely used by multiple clients.

3.23 QMI_LOC_GET_FIX_CRITERIA

Gets the fix criteria from the location engine.

LOC message ID

0x0033

Version introduced

Major - 2, Minor - 0

3.23.1 Request - QMI_LOC_GET_FIX_CRITERIA_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

None

Optional TLVs

None

3.23.2 Indication - QMI_LOC_GET_FIX_CRITERIA_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version introduced	Version last modified
Get Fix Criteria Status	2.0	2.28

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

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	enum	status	4	<p>Status of the Get Fix Criteria request.</p> <p>Valid values:</p> <ul style="list-style-type: none"> • eQMI_LOC_SUCCESS (0) – Request was completed successfully • eQMI_LOC_GENERAL_FAILURE (1) – Request failed because of a general failure • eQMI_LOC_UNSUPPORTED (2) – Request failed because it is not supported • eQMI_LOC_INVALID_PARAMETER (3) – Request failed because it contained invalid parameters • eQMI_LOC_ENGINE_BUSY (4) – Request failed because the engine is busy • eQMI_LOC_PHONE_OFFLINE (5) – Request failed because the phone is offline • eQMI_LOC_TIMEOUT (6) – Request failed because it timed out • eQMI_LOC_CONFIG_NOT_SUPPORTED (7) – Request failed because an undefined configuration was requested • eQMI_LOC_INSUFFICIENT_MEMORY (8) – Request failed because the engine could not allocate sufficient memory for the request • eQMI_LOC_MAX_GEOFENCE_PROGRAMMED (9) – Request failed because the maximum number of Geofences are already programmed • eQMI_LOC_XTRA_VERSION_CHECK_FAILURE (10) – Location service failed because of an XTRA version-based file format check failure

Optional TLVs

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

Name	Version introduced	Version last modified
Horizontal Accuracy	2.0	2.0
Enable/Disable Intermediate Fixes	2.0	2.0
Minimum Interval Between Fixes	2.0	2.0
ID of the Application that Sent the Position Request	2.6	2.6

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Horizontal Accuracy
Length	4			2	
Value	→	enum	horizontalAccuracyLevel	4	Horizontal accuracy level. Valid values: • eQMI_LOC_ACCURACY_LOW (1) – Low accuracy • eQMI_LOC_ACCURACY_MED (2) – Medium accuracy • eQMI_LOC_ACCURACY_HIGH (3) – High accuracy
Type	0x11			1	Enable/Disable Intermediate Fixes
Length	4			2	
Value	→	enum	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: • eQMI_LOC_INTERMEDIATE_REPORTS_ON (1) – Intermediate reports are turned on • eQMI_LOC_INTERMEDIATE_REPORTS_OFF (2) – Intermediate reports are turned off
Type	0x12			1	Minimum Interval Between Fixes
Length	4			2	
Value	→	uint32	minInterval	4	Time that must elapse before alerting the client. • Units: Milliseconds
Type	0x13			1	ID of the Application that Sent the Position Request
Length	Var			2	Application provider, name, and version.
Value	→	uint8	applicationProvider_len	1	Number of sets of the following elements: • applicationProvider
		string	applicationProvider	Var	Application provider.
		uint8	applicationName_len	1	Number of sets of the following elements: • applicationName
		string	applicationName	Var	Application name.

Field	Field value	Field type	Parameter	Size (byte)	Description
		boolean	applicationVersion_valid	1	Specifies whether the application version string contains a valid value: <ul style="list-style-type: none"> • 0x00 (FALSE) – Application version string is invalid • 0x01 (TRUE) – Application version string is valid
		uint8	applicationVersion_len	1	Number of sets of the following elements: <ul style="list-style-type: none"> • applicationVersion
		string	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

3.23.3 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. This command can be safely used by multiple clients.

3.24 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 - 0

3.24.1 Request - QMI_LOC_NI_USER_RESPONSE_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

Name	Version introduced	Version last modified
User Response	2.0	2.1
Notification Type	2.1	2.1

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	User Response
Length	4			2	
Value	→	enum	userResp	4	User accepted or denied. Valid values: <ul style="list-style-type: none"> • eQMI_LOC_NI_LCS_NOTIFY_VERIFY_ACCEPT (1) – User accepted the Notify/Verify request • eQMI_LOC_NI_LCS_NOTIFY_VERIFY_DENY (2) – User denied the Notify/Verify request • eQMI_LOC_NI_LCS_NOTIFY_VERIFY_NORESP (3) – User did not respond to the Notify/Verify request
Type	0x02			1	Notification Type
Length	4			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	enum	notificationType	4	Type of notification/verification performed. Valid values: <ul style="list-style-type: none"> • eQMI_LOC_NI_USER_NO_NOTIFY_NO_VERIFY (1) – No notification and no verification required • eQMI_LOC_NI_USER_NOTIFY_ONLY (2) – Notify only; no verification required • eQMI_LOC_NI_USER_NOTIFY_VERIFY_ALLOW_NO_RESP (3) – Notify and verify, but no response required. • eQMI_LOC_NI_USER_NOTIFY_VERIFY_NOT_ALLOW_NO_RESP (4) – Notify and verify, and require a response • eQMI_LOC_NI_USER_NOTIFY_VERIFY_PRIVACY_OVERRIDE (5) – Notify and verify; privacy override

Optional TLVs

Name	Version introduced	Version last modified
Network Initiated Vx Request	2.0	2.1
Network Initiated SUPL Request	2.0	2.2
Network Initiated UMTS Control Plane Request	2.0	2.2
Network Initiated Service Interaction Request	2.0	2.1
Network Initiated SUPL Version 2 Extension	2.10	2.10
SUPL Emergency Notification	2.17	2.17

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Network Initiated Vx Request Optional NI VX request payload.
Length	Var			2	
Value	→	boolean	posQosIncl	1	Indicates whether quality of service is included: <ul style="list-style-type: none"> • 0x01 (TRUE) – QoS is included • 0x00 (FALSE) – QoS is not included
		uint8	posQos	1	Position QoS timeout. <ul style="list-style-type: none"> • Units: Seconds • Range: 0 to 255
		uint32	numFixes	4	Number of fixes allowed.
		uint32	timeBetweenFixes	4	Time between fixes. <ul style="list-style-type: none"> • Units: Seconds

Field	Field value	Field type	Parameter	Size (byte)	Description
		enum	posMode	4	Position mode. Valid values: <ul style="list-style-type: none"> • eQMI_LOC_NI_VX_MS_ASSISTED_ONLY (1) – MS-assisted only allowed • eQMI_LOC_NI_VX_MS_BASED_ONLY (2) – MS-based only allowed • eQMI_LOC_NI_VX_MS_ASSISTED_PREFERRED_MS_BASED_ALLOWED (3) – MS-assisted preferred, but MS-based allowed • eQMI_LOC_NI_VX_MS_BASED_PREFERRED_MS_ASSISTED_ALLOWED (4) – MS-based preferred, but MS-assisted allowed
		enum	encodingScheme	4	VX encoding scheme. Valid values: <ul style="list-style-type: none"> • eQMI_LOC_NI_VX_OCTET (0) – Encoding is Octet • eQMI_LOC_NI_VX_EXN_PROTOCOL_MSG (1) – Encoding is EXN protocol message • eQMI_LOC_NI_VX_ASCII (2) – Encoding is ASCII • eQMI_LOC_NI_VX_IA5 (3) – Encoding is IA5 • eQMI_LOC_NI_VX_UNICODE (4) – Encoding is Unicode • eQMI_LOC_NI_VX_SHIFT_JIS (5) – Encoding is Shift JIS • eQMI_LOC_NI_VX_KOREAN (6) – Encoding is Korean • eQMI_LOC_NI_VX_LATIN_HEBREW (7) – Encoding is Latin Hebrew • eQMI_LOC_NI_VX_LATIN (8) – Encoding is Latin • eQMI_LOC_NI_VX_GSM (9) – Encoding is GSM
		uint8	requestorId_len	1	Number of sets of the following elements: <ul style="list-style-type: none"> • requestorId
		uint8	requestorId	Var	Requestor ID. <ul style="list-style-type: none"> • Type: Array of bytes • Maximum array length: 200
		uint16	userRespTimerInSeconds	2	Time to wait for the user to respond. <ul style="list-style-type: none"> • Units: Seconds

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x11			1	Network Initiated SUPL Request Optional NI SUPL request payload.
Length	Var			2	
Value	→	mask32	valid_flags	4	Indicates which of the following fields are present in this value. Valid bitmasks: <ul style="list-style-type: none"> • 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
		mask8	suplServerAddrTypeMask	1	Mask specifying the valid fields in this value. Valid bitmasks: <ul style="list-style-type: none"> • 0x01 – IPv4 • 0x02 – IPv6 • 0x04 – URL
		uint32	addr	4	IPv4 address.
		uint16	port	2	IPv4 port.
		uint16	addr	16	IPv6 address. <ul style="list-style-type: none"> • Type: Array of unsigned integers • Maximum length of the array: 8
		uint32	port	4	IPv6 port.
		uint8	urlAddr_len	1	Number of sets of the following elements: <ul style="list-style-type: none"> • urlAddr
		string	urlAddr	Var	URL. <ul style="list-style-type: none"> • Type: NULL-terminated string • Maximum string length (including NULL terminator): 256
		uint8	suplSessionId	4	SUPL session ID. <ul style="list-style-type: none"> • Type: Array of unsigned integers • Maximum length of the array: 4
		uint8	suplHash	8	Hash for SUPL_INIT; used to validate that the message was not corrupted. <ul style="list-style-type: none"> • Type: Array of unsigned integers • Length of the array: 8

Field	Field value	Field type	Parameter	Size (byte)	Description
		enum	posMethod	4	<p>GPS mode to be used for the fix.</p> <p>Valid values:</p> <ul style="list-style-type: none"> •eQMI_LOC_NI_SUPL_POSMETHOD_AGPS_SETASSISTED (1) – Set assisted •eQMI_LOC_NI_SUPL_POSMETHOD_AGPS_SETBASED (2) – Set based •eQMI_LOC_NI_SUPL_POSMETHOD_AGPS_SETASSISTED_PREF (3) – Set assisted preferred •eQMI_LOC_NI_SUPL_POSMETHOD_AGPS_SETBASED_PREF (4) – Set based preferred •eQMI_LOC_NI_SUPL_POSMETHOD_AUTONOMOUS_GPS (5) – Standalone GPS •eQMI_LOC_NI_SUPL_POSMETHOD_AFLT (6) – Advanced forward link trilateration •eQMI_LOC_NI_SUPL_POSMETHOD_ECID (7) – Exclusive chip ID •eQMI_LOC_NI_SUPL_POSMETHOD_EOTD (8) – Enhanced observed time difference •eQMI_LOC_NI_SUPL_POSMETHOD_OTDOA (9) – Observed time delay of arrival •eQMI_LOC_NI_SUPL_POSMETHOD_NO_POSITION (10) – No position
		enum	dataCodingScheme	4	<p>Data coding scheme applies to both the requestor ID and the client name.</p> <p>Valid values:</p> <ul style="list-style-type: none"> •eQMI_LOC_NI_SS_GERMAN (12) – Language is German •eQMI_LOC_NI_SS_ENGLISH (13) – Language is English •eQMI_LOC_NI_SS_ITALIAN (14) – Language is Italian •eQMI_LOC_NI_SS_FRENCH (15) – Language is French •eQMI_LOC_NI_SS_SPANISH (16) – Language is Spanish •eQMI_LOC_NI_SS_DUTCH (17) – Language is Dutch •eQMI_LOC_NI_SS_SWEDISH (18) – Language is Swedish •eQMI_LOC_NI_SS_DANISH (19) – Language is Danish

Field	Field value	Field type	Parameter	Size (byte)	Description
			dataCodingScheme (cont.)		<ul style="list-style-type: none"> • eQMI_LOC_NI_SS_PORTUGUESE (20) – Language is Portuguese • eQMI_LOC_NI_SS_FINNISH (21) – Language is Finnish • eQMI_LOC_NI_SS_NORWEGIAN (22) – Language is Norwegian • eQMI_LOC_NI_SS_GREEK (23) – Language is Greek • eQMI_LOC_NI_SS_TURKISH (24) – Language is Turkish • eQMI_LOC_NI_SS_HUNGARIAN (25) – Language is Hungarian • eQMI_LOC_NI_SS_POLISH (26) – Language is Polish • eQMI_LOC_NI_SS_LANGUAGE_UNSPEC (27) – Language is unspecified • eQMI_LOC_NI_SUPL_UTF8 (28) – Encoding is UTF 8 • eQMI_LOC_NI_SUPL_UCS2 (29) – Encoding is UCS 2 • eQMI_LOC_NI_SUPL_GSM_DEFAULT (30) – Encoding is GSM default
		enum	formatType	4	Format of the formatted string. Valid values: <ul style="list-style-type: none"> • eQMI_LOC_NI_SUPL_FORMAT_LOGICAL_NAME (0) – SUPL logical name format • eQMI_LOC_NI_SUPL_FORMAT_EMAIL_ADDRESS (1) – SUPL email address format • eQMI_LOC_NI_SUPL_FORMAT_MSISDN (2) – SUPL MS-ISDN format • eQMI_LOC_NI_SUPL_FORMAT_URL (3) – SUPL URL format • eQMI_LOC_NI_SUPL_FORMAT_SIP_URL (4) – SUPL SIP URL format • eQMI_LOC_NI_SUPL_FORMAT_MIN (5) – SUPL MIN format • eQMI_LOC_NI_SUPL_FORMAT_MDN (6) – SUPL MDN format • eQMI_LOC_NI_SUPL_FORMAT_IMSPUBLIC_IDENTITY (7) – SUPL IMS public identity • eQMI_LOC_NI_SUPL_FORMAT_OSS_UNKNOWN (2147483647) – SUPL unknown format

Field	Field value	Field type	Parameter	Size (byte)	Description
		uint8	formattedString_len	1	Number of sets of the following elements: • formattedString
		uint8	formattedString	Var	Formatted string. • Type: Byte array • Maximum string length: 64
		enum	formatType	4	Format of the formatted string. Valid values: • eQMI_LOC_NI_SUPL_FORMAT_LOGICAL_NAME (0) – SUPL logical name format • eQMI_LOC_NI_SUPL_FORMAT_EMAIL_ADDRESS (1) – SUPL email address format • eQMI_LOC_NI_SUPL_FORMAT_MSISDN (2) – SUPL MS-ISDN format • eQMI_LOC_NI_SUPL_FORMAT_URL (3) – SUPL URL format • eQMI_LOC_NI_SUPL_FORMAT_SIP_URL (4) – SUPL SIP URL format • eQMI_LOC_NI_SUPL_FORMAT_MIN (5) – SUPL MIN format • eQMI_LOC_NI_SUPL_FORMAT_MDN (6) – SUPL MDN format • eQMI_LOC_NI_SUPL_FORMAT_IMSPUBLIC_IDENTITY (7) – SUPL IMS public identity • eQMI_LOC_NI_SUPL_FORMAT_OSS_UNKNOWN (2147483647) – SUPL unknown format
		uint8	formattedString_len	1	Number of sets of the following elements: • formattedString
		uint8	formattedString	Var	Formatted string. • Type: Byte array • Maximum string length: 64
		mask8	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
		uint8	horizontalAccuracy	1	Horizontal accuracy. • Units: Meters
		uint8	verticalAccuracy	1	Vertical accuracy. • Units: Meters

Field	Field value	Field type	Parameter	Size (byte)	Description
		uint16	maxLocAge	2	Maximum age of the location if the engine sends a previously computed position. • Units: Seconds
		uint8	delay	1	Delay the server is willing to tolerate for the fix. • Units: Seconds
		uint16	userResponseTimer	2	Time to wait for the user to respond. • Units: Seconds
Type	0x12			1	Network Initiated UMTS Control Plane Request Optional NI UMTS-CP request payload.
Length	Var			2	
Value	→	mask16	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
		uint8	invokeId	1	Supplementary Services invoke ID.

Field	Field value	Field type	Parameter	Size (byte)	Description
		enum	dataCodingScheme	4	<p>Type of data encoding scheme for the text. Applies to both the notification text and the client address.</p> <p>Valid values:</p> <ul style="list-style-type: none"> • eQMI_LOC_NI_SS_GERMAN (12) – Language is German • eQMI_LOC_NI_SS_ENGLISH (13) – Language is English • eQMI_LOC_NI_SS_ITALIAN (14) – Language is Italian • eQMI_LOC_NI_SS_FRENCH (15) – Language is French • eQMI_LOC_NI_SS_SPANISH (16) – Language is Spanish • eQMI_LOC_NI_SS_DUTCH (17) – Language is Dutch • eQMI_LOC_NI_SS_SWEDISH (18) – Language is Swedish • eQMI_LOC_NI_SS_DANISH (19) – Language is Danish • eQMI_LOC_NI_SS_PORTUGUESE (20) – Language is Portuguese • eQMI_LOC_NI_SS_FINNISH (21) – Language is Finnish • eQMI_LOC_NI_SS_NORWEGIAN (22) – Language is Norwegian • eQMI_LOC_NI_SS_GREEK (23) – Language is Greek • eQMI_LOC_NI_SS_TURKISH (24) – Language is Turkish • eQMI_LOC_NI_SS_HUNGARIAN (25) – Language is Hungarian • eQMI_LOC_NI_SS_POLISH (26) – Language is Polish • eQMI_LOC_NI_SS_LANGUAGE_UNSPEC (27) – Language is unspecified • eQMI_LOC_NI_SUPL_UTF8 (28) – Encoding is UTF 8 • eQMI_LOC_NI_SUPL_UCS2 (29) – Encoding is UCS 2 • eQMI_LOC_NI_SUPL_GSM_DEFAULT (30) – Encoding is GSM default
		uint8	notificationText_len	1	<p>Number of sets of the following elements:</p> <ul style="list-style-type: none"> • notificationText

Field	Field value	Field type	Parameter	Size (byte)	Description
		uint8	notificationText	Var	Notification text; the encoding method is specified in dataCodingScheme. • Type: Array of bytes • Maximum array length: 64
		uint8	clientAddress_len	1	Number of sets of the following elements: • clientAddress
		uint8	clientAddress	Var	Client address; the encoding method is specified in dataCodingScheme. • Maximum array length: 20
		enum	locationType	4	Location type. Valid values: • eQMI_LOC_NI_LOCATIONTYPE_CURRENT_LOCATION (1) – Current location • eQMI_LOC_NI_LOCATIONTYPE_CURRENT_OR_LAST_KNOWN_LOCATION (2) – Last known location; may be the current location • eQMI_LOC_NI_LOCATIONTYPE_INITIAL_LOCATION (3) – Initial location

Field	Field value	Field type	Parameter	Size (byte)	Description
		enum	dataCodingScheme	4	Identifies the coding scheme of the coded string. Valid values: <ul style="list-style-type: none"> • eQMI_LOC_NI_SS_GERMAN (12) – Language is German • eQMI_LOC_NI_SS_ENGLISH (13) – Language is English • eQMI_LOC_NI_SS_ITALIAN (14) – Language is Italian • eQMI_LOC_NI_SS_FRENCH (15) – Language is French • eQMI_LOC_NI_SS_SPANISH (16) – Language is Spanish • eQMI_LOC_NI_SS_DUTCH (17) – Language is Dutch • eQMI_LOC_NI_SS_SWEDISH (18) – Language is Swedish • eQMI_LOC_NI_SS_DANISH (19) – Language is Danish • eQMI_LOC_NI_SS_PORTUGUESE (20) – Language is Portuguese • eQMI_LOC_NI_SS_FINNISH (21) – Language is Finnish • eQMI_LOC_NI_SS_NORWEGIAN (22) – Language is Norwegian • eQMI_LOC_NI_SS_GREEK (23) – Language is Greek • eQMI_LOC_NI_SS_TURKISH (24) – Language is Turkish • eQMI_LOC_NI_SS_HUNGARIAN (25) – Language is Hungarian • eQMI_LOC_NI_SS_POLISH (26) – Language is Polish • eQMI_LOC_NI_SS_LANGUAGE_UNSPEC (27) – Language is unspecified • eQMI_LOC_NI_SUPL_UTF8 (28) – Encoding is UTF 8 • eQMI_LOC_NI_SUPL_UCS2 (29) – Encoding is UCS 2 • eQMI_LOC_NI_SUPL_GSM_DEFAULT (30) – Encoding is GSM default
		uint8	codedString_len	1	Number of sets of the following elements: <ul style="list-style-type: none"> • codedString
		uint8	codedString	Var	Coded string. <ul style="list-style-type: none"> • Type: Array of bytes • Maximum string length: 20

Field	Field value	Field type	Parameter	Size (byte)	Description
		enum	dataCodingScheme	4	Identifies the coding scheme of the coded string. Valid values: <ul style="list-style-type: none"> • eQMI_LOC_NI_SS_GERMAN (12) – Language is German • eQMI_LOC_NI_SS_ENGLISH (13) – Language is English • eQMI_LOC_NI_SS_ITALIAN (14) – Language is Italian • eQMI_LOC_NI_SS_FRENCH (15) – Language is French • eQMI_LOC_NI_SS_SPANISH (16) – Language is Spanish • eQMI_LOC_NI_SS_DUTCH (17) – Language is Dutch • eQMI_LOC_NI_SS_SWEDISH (18) – Language is Swedish • eQMI_LOC_NI_SS_DANISH (19) – Language is Danish • eQMI_LOC_NI_SS_PORTUGUESE (20) – Language is Portuguese • eQMI_LOC_NI_SS_FINNISH (21) – Language is Finnish • eQMI_LOC_NI_SS_NORWEGIAN (22) – Language is Norwegian • eQMI_LOC_NI_SS_GREEK (23) – Language is Greek • eQMI_LOC_NI_SS_TURKISH (24) – Language is Turkish • eQMI_LOC_NI_SS_HUNGARIAN (25) – Language is Hungarian • eQMI_LOC_NI_SS_POLISH (26) – Language is Polish • eQMI_LOC_NI_SS_LANGUAGE_UNSPEC (27) – Language is unspecified • eQMI_LOC_NI_SUPL_UTF8 (28) – Encoding is UTF 8 • eQMI_LOC_NI_SUPL_UCS2 (29) – Encoding is UCS 2 • eQMI_LOC_NI_SUPL_GSM_DEFAULT (30) – Encoding is GSM default
		uint8	codedString_len	1	Number of sets of the following elements: <ul style="list-style-type: none"> • codedString
		uint8	codedString	Var	Coded string. <ul style="list-style-type: none"> • Type: Array of bytes • Maximum string length: 20

Field	Field value	Field type	Parameter	Size (byte)	Description
		uint8	lcsServiceTypeId	1	Service type ID.
		uint16	userResponseTimer	2	Time to wait for the user to respond. • Units: Seconds
Type	0x13			1	Network Initiated Service Interaction Request Optional NI service interaction payload.
Length	Var			2	
Value	→	boolean	posQosIncl	1	Indicates whether quality of service is included: • 0x01 (TRUE) – QoS is included • 0x00 (FALSE) – QoS is not included
		uint8	posQos	1	Position QoS timeout. • Units: Seconds • Range: 0 to 255
		uint32	numFixes	4	Number of fixes allowed.
		uint32	timeBetweenFixes	4	Time between fixes. • Units: Seconds
		enum	posMode	4	Position mode. Valid values: • eQMI_LOC_NI_VX_MS_ASSISTED_ONLY (1) – MS-assisted only allowed • eQMI_LOC_NI_VX_MS_BASED_ONLY (2) – MS-based only allowed • eQMI_LOC_NI_VX_MS_ASSISTED_PREFERRED_MS_BASED_ALLOWED (3) – MS-assisted preferred, but MS-based allowed • eQMI_LOC_NI_VX_MS_BASED_PREFERRED_MS_ASSISTED_ALLOWED (4) – MS-based preferred, but MS-assisted allowed

Field	Field value	Field type	Parameter	Size (byte)	Description
		enum	encodingScheme	4	VX encoding scheme. Valid values: <ul style="list-style-type: none"> • eQMI_LOC_NI_VX_OCTET (0) – Encoding is Octet • eQMI_LOC_NI_VX_EXN_PROTOCOL_MSG (1) – Encoding is EXN protocol message • eQMI_LOC_NI_VX_ASCII (2) – Encoding is ASCII • eQMI_LOC_NI_VX_IA5 (3) – Encoding is IA5 • eQMI_LOC_NI_VX_UNICODE (4) – Encoding is Unicode • eQMI_LOC_NI_VX_SHIFT_JIS (5) – Encoding is Shift JIS • eQMI_LOC_NI_VX_KOREAN (6) – Encoding is Korean • eQMI_LOC_NI_VX_LATIN_HEBREW (7) – Encoding is Latin Hebrew • eQMI_LOC_NI_VX_LATIN (8) – Encoding is Latin • eQMI_LOC_NI_VX_GSM (9) – Encoding is GSM
		uint8	requestorId_len	1	Number of sets of the following elements: <ul style="list-style-type: none"> • requestorId
		uint8	requestorId	Var	Requestor ID. <ul style="list-style-type: none"> • Type: Array of bytes • Maximum array length: 200
		uint16	userRespTimerInSeconds	2	Time to wait for the user to respond. <ul style="list-style-type: none"> • Units: Seconds
		enum	serviceInteractionType	4	Service interaction type specified in qmiLocNiServiceInteractionEnumT. Valid values: <ul style="list-style-type: none"> • eQMI_LOC_NI_SERVICE_INTERACTION_ONGOING_NI_INCOMING_MO (1) – Service interaction between ongoing NI and incoming MO sessions.
Type	0x14			1	Network Initiated SUPL Version 2 Extension Optional SUPL Version 2 Extension payload.
Length	8			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	mask16	supportedNetworksMask	2	<p>Specifies which type of network measurements are allowed to be sent as part of the Location ID or Multiple Location IDs parameter in the SUPL_POS_INIT message (see [S4]).</p> <p>Valid bitmasks:</p> <ul style="list-style-type: none"> • 0x0001 – SUPPORTED_NETWORK_WLAN • 0x0002 – SUPPORTED_NETWORK_GSM • 0x0004 – SUPPORTED_NETWORK_WCDMA • 0x0008 – SUPPORTED_NETWORK_CDMA • 0x0010 – SUPPORTED_NETWORK_HRDP • 0x0020 – SUPPORTED_NETWORK_UMB • 0x0040 – SUPPORTED_NETWORK_LTE • 0x0080 – SUPPORTED_NETWORK_WIMAX • 0x0100 – SUPPORTED_NETWORK_HISTORIC • 0x0200 – SUPPORTED_NETWORK_NONSVRV
		enum	triggerType	4	<p>Specifies the type of session trigger requested in the SUPL_POS_INIT message (refer to [S4]).</p> <p>Valid values:</p> <ul style="list-style-type: none"> • eQMI_LOC_SUPL_VER_2_EXT_TRIGGER_TYPE_SINGLE_SHOT (-1) – SUPL INIT message indicates a request for a single shot triggered session • eQMI_LOC_SUPL_VER_2_EXT_TRIGGER_TYPE_PERIODIC (0) – SUPL INIT message indicates a request for a periodic triggered session • eQMI_LOC_SUPL_VER_2_EXT_TRIGGER_TYPE_AREA_EVENT (1) – SUPL INIT message indicates a request for an area event triggered session

Field	Field value	Field type	Parameter	Size (byte)	Description
		mask16	gnssType	2	Specifies which GNSS technologies are allowed as positioning technologies. Valid bitmasks: <ul style="list-style-type: none"> • 0x0001 – GNSS_GPS • 0x0002 – GNSS_GLONASS • 0x0004 – GNSS_GALILEO • 0x0008 – GNSS_SBAS • 0x0010 – GNSS_QZSS • 0x0020 – GNSS_MODERN_GPS
Type	0x15			1	SUPL Emergency Notification SUPL emergency notification payload. Emergency notification can be given even without an ESLP address
Length	Var			2	
Value	→	uint8	eslpUrl_len	1	Number of sets of the following elements: <ul style="list-style-type: none"> • eslpUrl
		string	eslpUrl	Var	ESLP URL. Maximum length: 255 bytes

3.24.2 Indication - QMI_LOC_NI_USER_RESPONSE_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version introduced	Version last modified
NI User Response Status	2.1	2.28

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	NI User Response Status
Length	4			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	enum	status	4	<p>Status of the NI User Response request.</p> <p>Valid values:</p> <ul style="list-style-type: none"> • eQMI_LOC_SUCCESS (0) – Request was completed successfully • eQMI_LOC_GENERAL_FAILURE (1) – Request failed because of a general failure • eQMI_LOC_UNSUPPORTED (2) – Request failed because it is not supported • eQMI_LOC_INVALID_PARAMETER (3) – Request failed because it contained invalid parameters • eQMI_LOC_ENGINE_BUSY (4) – Request failed because the engine is busy • eQMI_LOC_PHONE_OFFLINE (5) – Request failed because the phone is offline • eQMI_LOC_TIMEOUT (6) – Request failed because it timed out • eQMI_LOC_CONFIG_NOT_SUPPORTED (7) – Request failed because an undefined configuration was requested • eQMI_LOC_INSUFFICIENT_MEMORY (8) – Request failed because the engine could not allocate sufficient memory for the request • eQMI_LOC_MAX_GEOFENCE_PROGRAMMED (9) – Request failed because the maximum number of Geofences are already programmed • eQMI_LOC_XTRA_VERSION_CHECK_FAILURE (10) – Location service failed because of an XTRA version-based file format check failure

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

3.24.3 Description of QMI_LOC_INFORM_NI_USER_RESPONSE

This command sends the user response to the engine corresponding to a prior NI Notify/Verify request. The indication contains the status of the User Response request. The user response payload must contain the same optional fields as were passed in the Notify/Verify request. Only one client may respond to an NI Notify/Verify request.

QUALCOMM®
2016-05-16 00:17:18 PDT
deon_zhang@askey.com.tw

3.25 QMI_LOC_INJECT_PREDICTED_ORBITS_DATA

Injects predicted orbits data.

LOC message ID

0x0035

Version introduced

Major - 2, Minor - 0

3.25.1 Request - QMI_LOC_INJECT_PREDICTED_ORBITS_DATA_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

Name	Version introduced	Version last modified
Total Size	2.0	2.0
Total Parts	2.0	2.0
Part Number	2.0	2.0
Data	2.0	2.0

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Total Size
Length	4			2	
Value	→	uint32	totalSize	4	Total size of the predicted orbits data to be injected. • Units: Bytes
Type	0x02			1	Total Parts
Length	2			2	
Value	→	uint16	totalParts	2	Total number of parts into which the predicted orbits data is divided.
Type	0x03			1	Part Number
Length	2			2	
Value	→	uint16	partNum	2	Number of the current predicted orbits data part; starts at 1.
Type	0x04			1	Data
Length	Var			2	
Value	→	uint16	partData_len	2	Number of sets of the following elements: • partData

Field	Field value	Field type	Parameter	Size (byte)	Description
		char	partData	Var	Predicted orbits data. • Type: Array of bytes • Maximum length of the array: 1024

Optional TLVs

Name	Version introduced	Version last modified
Format Type	2.1	2.1

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Format Type
Length	4			2	
Value	→	enum	formatType	4	Predicted orbits data format. Valid values: • eQMI_LOC_PREDICTED_ORBITS_XTRA (0) – Default is QCOM-XTRA format.

3.25.2 Indication - QMI_LOC_INJECT_PREDICTED_ORBITS_DATA_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version introduced	Version last modified
Data Injection Status	2.0	2.28

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

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	enum	status	4	<p>Status of the Data Injection request.</p> <p>Valid values:</p> <ul style="list-style-type: none"> • eQMI_LOC_SUCCESS (0) – Request was completed successfully • eQMI_LOC_GENERAL_FAILURE (1) – Request failed because of a general failure • eQMI_LOC_UNSUPPORTED (2) – Request failed because it is not supported • eQMI_LOC_INVALID_PARAMETER (3) – Request failed because it contained invalid parameters • eQMI_LOC_ENGINE_BUSY (4) – Request failed because the engine is busy • eQMI_LOC_PHONE_OFFLINE (5) – Request failed because the phone is offline • eQMI_LOC_TIMEOUT (6) – Request failed because it timed out • eQMI_LOC_CONFIG_NOT_SUPPORTED (7) – Request failed because an undefined configuration was requested • eQMI_LOC_INSUFFICIENT_MEMORY (8) – Request failed because the engine could not allocate sufficient memory for the request • eQMI_LOC_MAX_GEOFENCE_PROGRAMMED (9) – Request failed because the maximum number of Geofences are already programmed • eQMI_LOC_XTRA_VERSION_CHECK_FAILURE (10) – Location service failed because of an XTRA version-based file format check failure

Optional TLVs

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

Name	Version introduced	Version last modified
Part Number	2.0	2.1

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Part Number
Length	2			2	
Value	→	uint16	partNum	2	Number of the predicted orbits data part for which this indication is sent; starts at 1.

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_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. Only one client may inject the predicted orbits data into the service at a time.

3.26 QMI_LOC_GET_PREDICTED_ORBITS_DATA_SOURCE

Gets the predicted orbits data source.

LOC message ID

0x0036

Version introduced

Major - 2, Minor - 0

3.26.1 Request - QMI_LOC_GET_PREDICTED_ORBITS_DATA_SOURCE_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

None

Optional TLVs

None

3.26.2 Indication - QMI_LOC_GET_PREDICTED_ORBITS_DATA_SOURCE_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version introduced	Version last modified
Predicted Orbits Data Source Status	2.0	2.28

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Predicted Orbits Data Source Status
Length	4			2	
Value	→	enum	status	4	<p>Status of the query request for a predicted orbits data source.</p> <p>Valid values:</p> <ul style="list-style-type: none"> • eQMI_LOC_SUCCESS (0) – Request was completed successfully • eQMI_LOC_GENERAL_FAILURE (1) – Request failed because of a general failure • eQMI_LOC_UNSUPPORTED (2) – Request failed because it is not supported • eQMI_LOC_INVALID_PARAMETER (3) – Request failed because it contained invalid parameters • eQMI_LOC_ENGINE_BUSY (4) – Request failed because the engine is busy • eQMI_LOC_PHONE_OFFLINE (5) – Request failed because the phone is offline • eQMI_LOC_TIMEOUT (6) – Request failed because it timed out • eQMI_LOC_CONFIG_NOT_SUPPORTED (7) – Request failed because an undefined configuration was requested • eQMI_LOC_INSUFFICIENT_MEMORY (8) – Request failed because the engine could not allocate sufficient memory for the request • eQMI_LOC_MAX_GEOFENCE_PROGRAMMED (9) – Request failed because the maximum number of Geofences are already programmed • eQMI_LOC_XTRA_VERSION_CHECK_FAILURE (10) – Location service failed because of an XTRA version-based file format check failure

Optional TLVs

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

Name	Version introduced	Version last modified
Allowed Sizes	2.0	2.0
Server List	2.0	2.0

Field	Field value	Field type	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	→	uint32	maxFileSizeInBytes	4	Maximum allowable predicted orbits file size (in bytes).
		uint32	maxPartSize	4	Maximum allowable predicted orbits file chunk size (in bytes).
Type	0x11			1	Server List List of servers that can be used by the client to download predicted orbits data.
Length	Var			2	
Value	→	uint8	serverList_len	1	Number of sets of the following elements: • serverUrl_len • serverUrl
		uint8	serverUrl_len	1	Number of sets of the following elements: • serverUrl
		string	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

3.26.3 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. It is safe for multiple clients to use this command.

3.27 QMI_LOC_GET_PREDICTED_ORBITS_DATA_VALIDITY

Gets the predicted orbits data validity.

LOC message ID

0x0037

Version introduced

Major - 2, Minor - 0

3.27.1 Request - QMI_LOC_GET_PREDICTED_ORBITS_DATA_VALIDITY_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

None

Optional TLVs

None

3.27.2 Indication - QMI_LOC_GET_PREDICTED_ORBITS_DATA_VALIDITY_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version introduced	Version last modified
Predicted Orbits Data Validity Status	2.0	2.28

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Predicted Orbits Data Validity Status
Length	4			2	
Value	→	enum	status	4	<p>Status of the query request for predicted orbits data validity.</p> <p>Valid values:</p> <ul style="list-style-type: none"> • eQMI_LOC_SUCCESS (0) – Request was completed successfully • eQMI_LOC_GENERAL_FAILURE (1) – Request failed because of a general failure • eQMI_LOC_UNSUPPORTED (2) – Request failed because it is not supported • eQMI_LOC_INVALID_PARAMETER (3) – Request failed because it contained invalid parameters • eQMI_LOC_ENGINE_BUSY (4) – Request failed because the engine is busy • eQMI_LOC_PHONE_OFFLINE (5) – Request failed because the phone is offline • eQMI_LOC_TIMEOUT (6) – Request failed because it timed out • eQMI_LOC_CONFIG_NOT_SUPPORTED (7) – Request failed because an undefined configuration was requested • eQMI_LOC_INSUFFICIENT_MEMORY (8) – Request failed because the engine could not allocate sufficient memory for the request • eQMI_LOC_MAX_GEOFENCE_PROGRAMMED (9) – Request failed because the maximum number of Geofences are already programmed • eQMI_LOC_XTRA_VERSION_CHECK_FAILURE (10) – Location service failed because of an XTRA version-based file format check failure

Optional TLVs

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

Name	Version introduced	Version last modified
Validity Info	2.0	2.0

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Validity Info
Length	10			2	
Value	→	uint64	startTimeInUTC	8	Predicted orbits data is valid starting from this time. • Units: Seconds (since Jan. 1, 1970)
		uint16	durationHours	2	Duration from the start time for which the data is valid. • 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

3.27.3 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. It is safe for multiple clients to use this command.

3.28 QMI_LOC_INJECT.UTC_TIME

Injects UTC time in the location engine.

LOC message ID

0x0038

Version introduced

Major - 2, Minor - 0

3.28.1 Request - QMI_LOC_INJECT.UTC_TIME_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

Name	Version introduced	Version last modified
UTC Time	2.0	2.0
Time Uncertainty	2.0	2.0

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	UTC Time
Length	8			2	
Value	→	uint64	timeUtc	8	UTC time since Jan. 1, 1970. • Units: Milliseconds
Type	0x02			1	Time Uncertainty
Length	4			2	
Value	→	uint32	timeUnc	4	Time uncertainty. • Units: Milliseconds

Optional TLVs

None

3.28.2 Indication - QMI_LOC_INJECT.UTC_TIME_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version introduced	Version last modified
UTC Time Injection Status	2.0	2.28

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	UTC Time Injection Status
Length	4			2	
Value	→	enum	status	4	<p>Status of the UTC Time Injection request.</p> <p>Valid values:</p> <ul style="list-style-type: none"> • eQMI_LOC_SUCCESS (0) – Request was completed successfully • eQMI_LOC_GENERAL_FAILURE (1) – Request failed because of a general failure • eQMI_LOC_UNSUPPORTED (2) – Request failed because it is not supported • eQMI_LOC_INVALID_PARAMETER (3) – Request failed because it contained invalid parameters • eQMI_LOC_ENGINE_BUSY (4) – Request failed because the engine is busy • eQMI_LOC_PHONE_OFFLINE (5) – Request failed because the phone is offline • eQMI_LOC_TIMEOUT (6) – Request failed because it timed out • eQMI_LOC_CONFIG_NOT_SUPPORTED (7) – Request failed because an undefined configuration was requested • eQMI_LOC_INSUFFICIENT_MEMORY (8) – Request failed because the engine could not allocate sufficient memory for the request • eQMI_LOC_MAX_GEOFENCE_PROGRAMMED (9) – Request failed because the maximum number of Geofences are already programmed • eQMI_LOC_XTRA_VERSION_CHECK_FAILURE (10) – Location service failed because of an XTRA version-based file format check failure

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_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. It is recommended that only one client inject the UTC time into the service, since this impacts the global state of the service.

3.29 QMI_LOC_INJECT_POSITION

Injects a position to the location engine.

LOC message ID

0x0039

Version introduced

Major - 2, Minor - 0

3.29.1 Request - QMI_LOC_INJECT_POSITION_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

None

Optional TLVs

Name	Version introduced	Version last modified
Latitude	2.0	2.0
Longitude	2.0	2.0
Circular Horizontal Uncertainty	2.0	2.0
Horizontal Confidence	2.0	2.0
Horizontal Reliability	2.0	2.1
Altitude With Respect to Ellipsoid	2.0	2.0
Altitude With Respect to Sea Level	2.0	2.0
Vertical Uncertainty	2.0	2.0
Vertical Confidence	2.0	2.0
Vertical Reliability	2.0	2.1
Altitude Source Info	2.0	2.1
UTC Timestamp	2.0	2.0
Position Age	2.0	2.0
Position Source	2.4	2.4
Raw Circular Horizontal Uncertainty	2.29	2.29
Raw Horizontal Confidence	2.29	2.29

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Latitude
Length	8			2	
Value	→	double	latitude	8	Latitude (specified in WGS84 datum). <ul style="list-style-type: none"> • Type: Floating point • Units: Degrees • Range: -90.0 to 90.0 <ul style="list-style-type: none"> – Positive values indicate northern latitude – Negative values indicate southern latitude
Type	0x11			1	Longitude
Length	8			2	
Value	→	double	longitude	8	Longitude (specified in WGS84 datum). <ul style="list-style-type: none"> • Type: Floating point • Units: Degrees • Range: -180.0 to 180.0 <ul style="list-style-type: none"> – Positive values indicate eastern longitude – Negative values indicate western longitude
Type	0x12			1	Circular Horizontal Uncertainty
Length	4			2	
Value	→	float	horUncCircular	4	Horizontal position uncertainty (circular). <ul style="list-style-type: none"> • Units: Meters
Type	0x13			1	Horizontal Confidence
Length	1			2	
Value	→	uint8	horConfidence	1	Horizontal confidence, as defined by ETSI TS 101 109 ([S4]). <ul style="list-style-type: none"> • Units: Percent (1 to 99) • 0, 101 to 255 – invalid value • If 100 is received, reinterpret to 99 This field must be specified together with horizontal uncertainty. If not specified when horUncCircular is set, the default value is 50.
Type	0x14			1	Horizontal Reliability
Length	4			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	enum	horReliability	4	Specifies the reliability of the horizontal position. Valid values: <ul style="list-style-type: none"> • eQMI_LOC_RELIABILITY_NOT_SET (0) – Location reliability is not set • eQMI_LOC_RELIABILITY_VERY_LOW (1) – Location reliability is very low; use it at your own risk • eQMI_LOC_RELIABILITY_LOW (2) – Location reliability is low; little or no cross-checking is possible • eQMI_LOC_RELIABILITY_MEDIUM (3) – Location reliability is medium; limited cross-check passed • eQMI_LOC_RELIABILITY_HIGH (4) – Location reliability is high; strong cross-check passed
Type	0x15			1	Altitude With Respect to Ellipsoid
Length	4			2	
Value	→	float	altitudeWrtEllipsoid	4	Altitude with respect to the WGS84 ellipsoid. <ul style="list-style-type: none"> • Units: Meters <ul style="list-style-type: none"> – Positive = height – Negative = depth
Type	0x16			1	Altitude With Respect to Sea Level
Length	4			2	
Value	→	float	altitudeWrtMeanSeaLevel	4	Altitude with respect to mean sea level. <ul style="list-style-type: none"> • Units: Meters
Type	0x17			1	Vertical Uncertainty
Length	4			2	
Value	→	float	vertUnc	4	Vertical uncertainty. This is mandatory if either altitudeWrtEllipsoid or altitudeWrtMeanSeaLevel is specified. <ul style="list-style-type: none"> • Units: Meters
Type	0x18			1	Vertical Confidence
Length	1			2	
Value	→	uint8	vertConfidence	1	Vertical confidence, as defined by ETSI TS 101 109 ([S4]). <ul style="list-style-type: none"> • Units: Percent (0-99) • 0 – invalid value • 100 to 256 – not used • If 100 is received, reinterpret to 99 <p>This field must be specified together with the vertical uncertainty. If not specified, the default value will be 50.</p>
Type	0x19			1	Vertical Reliability

Field	Field value	Field type	Parameter	Size (byte)	Description
Length	4			2	
Value	→	enum	vertReliability	4	Specifies the reliability of the vertical position. Valid values: <ul style="list-style-type: none"> • eQMI_LOC_RELIABILITY_NOT_SET (0) – Location reliability is not set • eQMI_LOC_RELIABILITY_VERY_LOW (1) – Location reliability is very low; use it at your own risk • eQMI_LOC_RELIABILITY_LOW (2) – Location reliability is low; little or no cross-checking is possible • eQMI_LOC_RELIABILITY_MEDIUM (3) – Location reliability is medium; limited cross-check passed • eQMI_LOC_RELIABILITY_HIGH (4) – Location reliability is high; strong cross-check passed
Type	0x1A			1	Altitude Source Info Specifies information regarding the altitude source.
Length	12			2	
Value	→	enum	source	4	Specifies the source of the altitude. Valid values: <ul style="list-style-type: none"> • eQMI_LOC_ALT_SRC_UNKNOWN (0) – Source is unknown • eQMI_LOC_ALT_SRC_GPS (1) – GPS is the source • eQMI_LOC_ALT_SRC_CELL_ID (2) – Cell ID provided the source • eQMI_LOC_ALT_SRC_ENHANCED_CELL_ID (3) – Source is enhanced cell ID • eQMI_LOC_ALT_SRC_WIFI (4) – Wi-Fi is the source • eQMI_LOC_ALT_SRC_TERRESTRIAL (5) – Terrestrial source • eQMI_LOC_ALT_SRC_TERRESTRIAL_HYBRID (6) – Hybrid terrestrial source • eQMI_LOC_ALT_SRC_ALTITUDE_DATABASE (7) – Altitude database is the source • eQMI_LOC_ALT_SRC_BAROMETRIC_ALTIMETER (8) – Barometric altimeter is the source • eQMI_LOC_ALT_SRC_OTHER (9) – Other sources

Field	Field value	Field type	Parameter	Size (byte)	Description
		enum	linkage	4	Specifies the dependency between the horizontal and altitude position components. Valid values: <ul style="list-style-type: none"> • eQMI_LOC_ALT_SRC_LINKAGE_NOT_SPECIFIED (0) – Not specified • eQMI_LOC_ALT_SRC_LINKAGE_FULLY_INTERDEPENDENT (1) – Fully interdependent • eQMI_LOC_ALT_SRC_LINKAGE_DEPENDS_ON_LAT_LONG (2) – Depends on latitude and longitude • eQMI_LOC_ALT_SRC_LINKAGE_FULLY_INDEPENDENT (3) – Fully independent
		enum	coverage	4	Specifies the region of uncertainty. Valid values: <ul style="list-style-type: none"> • eQMI_LOC_ALT_UNCERTAINTY_NOT_SPECIFIED (0) – Not specified • eQMI_LOC_ALT_UNCERTAINTY_POINT (1) – Altitude uncertainty is valid at the injected horizontal position coordinates only • eQMI_LOC_ALT_UNCERTAINTY_FULL (2) – 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	→	uint64	timestampUtc	8	UTC timestamp. <ul style="list-style-type: none"> • Units: Milliseconds (since Jan. 1, 1970)
Type	0x1C			1	Position Age
Length	4			2	
Value	→	int32	timestampAge	4	Position age, which is an estimate of how long ago this fix was made. <ul style="list-style-type: none"> • Units: Milliseconds
Type	0x1D			1	Position Source
Length	4			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	enum	positionSrc	4	<p>Source from which this position was obtained.</p> <p>Valid values:</p> <ul style="list-style-type: none"> • eQMI_LOC_POSITION_SRC_GNSS (0) – Position source is GNSS • eQMI_LOC_POSITION_SRC_CELLID (1) – Position source is Cell ID • eQMI_LOC_POSITION_SRC_ENH_CELLID (2) – Position source is Enhanced Cell ID • eQMI_LOC_POSITION_SRC_WIFI (3) – Position source is Wi-Fi • eQMI_LOC_POSITION_SRC_TERRESTRIAL (4) – Position source is Terrestrial • eQMI_LOC_POSITION_SRC_GNSS_TERRESTRIAL_HYBRID (5) – Position source is GNSS Terrestrial Hybrid • eQMI_LOC_POSITION_SRC_OTHER (6) – Other sources. <p>If altitude is specified and the altitude source is not specified, the engine assumes that the altitude was obtained using the specified position source.</p> <p>If both altitude and altitude source are specified, the engine assumes that only latitude and longitude were obtained using the specified position source.</p>
Type	0x1E			1	Raw Circular Horizontal Uncertainty
Length	4			2	
Value	→	float	rawHorUncCircular	4	<p>Horizontal position uncertainty (circular) without any optimization.</p> <ul style="list-style-type: none"> • Units: Meters
Type	0x1F			1	Raw Horizontal Confidence
Length	1			2	
Value	→	uint8	rawHorConfidence	1	<p>Horizontal confidence associated with raw horizontal uncertainty, as defined by ETSI TS 101 109 ([S4]).</p> <ul style="list-style-type: none"> • Units: Percent (1 to 99) • 0, 101 to 255 – invalid value • If 100 is received, reinterpret to 99 <p>This field must be specified together with raw horizontal uncertainty. If not specified when rawHorUncCircular is set, the default value is 50.</p>

3.29.2 Indication - QMI_LOC_INJECT_POSITION_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version introduced	Version last modified
UTC Position Injection Status	2.0	2.28

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	UTC Position Injection Status
Length	4			2	
Value	→	enum	status	4	<p>Status of the UTC Position Injection request.</p> <p>Valid values:</p> <ul style="list-style-type: none"> • eQMI_LOC_SUCCESS (0) – Request was completed successfully • eQMI_LOC_GENERAL_FAILURE (1) – Request failed because of a general failure • eQMI_LOC_UNSUPPORTED (2) – Request failed because it is not supported • eQMI_LOC_INVALID_PARAMETER (3) – Request failed because it contained invalid parameters • eQMI_LOC_ENGINE_BUSY (4) – Request failed because the engine is busy • eQMI_LOC_PHONE_OFFLINE (5) – Request failed because the phone is offline • eQMI_LOC_TIMEOUT (6) – Request failed because it timed out • eQMI_LOC_CONFIG_NOT_SUPPORTED (7) – Request failed because an undefined configuration was requested • eQMI_LOC_INSUFFICIENT_MEMORY (8) – Request failed because the engine could not allocate sufficient memory for the request

Field	Field value	Field type	Parameter	Size (byte)	Description
			status (cont.)		<ul style="list-style-type: none"> • eQMI_LOC_MAX_GEOFENCE_PROGRAMMED (9) – Request failed because the maximum number of Geofences are already programmed • eQMI_LOC_XTRA_VERSION_CHECK_FAILURE (10) – Location service failed because of an XTRA version-based file format check failure

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

3.29.3 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. A client that injects a position affects the global state of the engine, thereby impacting all other clients.

3.30 QMI_LOC_SET_ENGINE_LOCK

Sets the location engine lock.

LOC message ID

0x003A

Version introduced

Major - 2, Minor - 0

3.30.1 Request - QMI_LOC_SET_ENGINE_LOCK_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

Name	Version introduced	Version last modified
Lock Type	2.0	2.1

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Lock Type
Length	4			2	
Value	→	enum	lockType	4	Type of lock. Valid values: <ul style="list-style-type: none"> • eQMI_LOC_LOCK_NONE (1) – Do not lock any position sessions • eQMI_LOC_LOCK_MI (2) – Lock mobile-initiated position sessions • eQMI_LOC_LOCK_MT (3) – Lock mobile-terminated position sessions • eQMI_LOC_LOCK_ALL (4) – Lock all position sessions

Optional TLVs

None

3.30.2 Indication - QMI_LOC_SET_ENGINE_LOCK_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version introduced	Version last modified
Set Engine Lock Status	2.0	2.28

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Set Engine Lock Status
Length	4			2	
Value	→	enum	status	4	<p>Status of the Set Engine Lock request.</p> <p>Valid values:</p> <ul style="list-style-type: none"> • eQMI_LOC_SUCCESS (0) – Request was completed successfully • eQMI_LOC_GENERAL_FAILURE (1) – Request failed because of a general failure • eQMI_LOC_UNSUPPORTED (2) – Request failed because it is not supported • eQMI_LOC_INVALID_PARAMETER (3) – Request failed because it contained invalid parameters • eQMI_LOC_ENGINE_BUSY (4) – Request failed because the engine is busy • eQMI_LOC_PHONE_OFFLINE (5) – Request failed because the phone is offline • eQMI_LOC_TIMEOUT (6) – Request failed because it timed out • eQMI_LOC_CONFIG_NOT_SUPPORTED (7) – Request failed because an undefined configuration was requested • eQMI_LOC_INSUFFICIENT_MEMORY (8) – Request failed because the engine could not allocate sufficient memory for the request

Field	Field value	Field type	Parameter	Size (byte)	Description
			status (cont.)		<ul style="list-style-type: none"> • eQMI_LOC_MAX_GEOFENCE_PROGRAMMED (9) – Request failed because the maximum number of Geofences are already programmed • eQMI_LOC_XTRA_VERSION_CHECK_FAILURE (10) – Location service failed because of an XTRA version-based file format check failure

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_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. Only one client may control the location engine lock, since the lock significantly impacts the operation of all clients.

3.31 QMI_LOC_GET_ENGINE_LOCK

Gets the location engine lock.

LOC message ID

0x003B

Version introduced

Major - 2, Minor - 0

3.31.1 Request - QMI_LOC_GET_ENGINE_LOCK_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

None

Optional TLVs

None

3.31.2 Indication - QMI_LOC_GET_ENGINE_LOCK_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version introduced	Version last modified
Get Engine Lock Status	2.0	2.28

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

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	enum	status	4	<p>Status of the Get Engine Lock request.</p> <p>Valid values:</p> <ul style="list-style-type: none"> • eQMI_LOC_SUCCESS (0) – Request was completed successfully • eQMI_LOC_GENERAL_FAILURE (1) – Request failed because of a general failure • eQMI_LOC_UNSUPPORTED (2) – Request failed because it is not supported • eQMI_LOC_INVALID_PARAMETER (3) – Request failed because it contained invalid parameters • eQMI_LOC_ENGINE_BUSY (4) – Request failed because the engine is busy • eQMI_LOC_PHONE_OFFLINE (5) – Request failed because the phone is offline • eQMI_LOC_TIMEOUT (6) – Request failed because it timed out • eQMI_LOC_CONFIG_NOT_SUPPORTED (7) – Request failed because an undefined configuration was requested • eQMI_LOC_INSUFFICIENT_MEMORY (8) – Request failed because the engine could not allocate sufficient memory for the request • eQMI_LOC_MAX_GEOFENCE_PROGRAMMED (9) – Request failed because the maximum number of Geofences are already programmed • eQMI_LOC_XTRA_VERSION_CHECK_FAILURE (10) – Location service failed because of an XTRA version-based file format check failure

Optional TLVs

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

Name	Version introduced	Version last modified
Lock Type	2.0	2.1

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Lock Type
Length	4			2	
Value	→	enum	lockType	4	Type of lock. Valid values: <ul style="list-style-type: none"> • eQMI_LOC_LOCK_NONE (1) – Do not lock any position sessions • eQMI_LOC_LOCK_MI (2) – Lock mobile-initiated position sessions • eQMI_LOC_LOCK_MT (3) – Lock mobile-terminated position sessions • eQMI_LOC_LOCK_ALL (4) – Lock all position sessions

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_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. It is safe for multiple clients to use this command.

3.32 QMI_LOC_SET_SBAS_CONFIG

Sets the SBAS configuration.

LOC message ID

0x003C

Version introduced

Major - 2, Minor - 0

3.32.1 Request - QMI_LOC_SET_SBAS_CONFIG_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

Name	Version introduced	Version last modified
SBAS Config	2.0	2.0

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	SBAS Config
Length	1			2	
Value	→	boolean	sbasConfig	1	Indicates whether SBAS configuration is enabled. <ul style="list-style-type: none"> • 0x01 (TRUE) – SBAS configuration is enabled • 0x00 (FALSE) – SBAS configuration is disabled

Optional TLVs

None

3.32.2 Indication - QMI_LOC_SET_SBAS_CONFIG_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version introduced	Version last modified
Set SBAS Config Status	2.0	2.28

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Set SBAS Config Status
Length	4			2	
Value	→	enum	status	4	<p>Status of the Set SBAS Configuration request.</p> <p>Valid values:</p> <ul style="list-style-type: none"> • eQMI_LOC_SUCCESS (0) – Request was completed successfully • eQMI_LOC_GENERAL_FAILURE (1) – Request failed because of a general failure • eQMI_LOC_UNSUPPORTED (2) – Request failed because it is not supported • eQMI_LOC_INVALID_PARAMETER (3) – Request failed because it contained invalid parameters • eQMI_LOC_ENGINE_BUSY (4) – Request failed because the engine is busy • eQMI_LOC_PHONE_OFFLINE (5) – Request failed because the phone is offline • eQMI_LOC_TIMEOUT (6) – Request failed because it timed out • eQMI_LOC_CONFIG_NOT_SUPPORTED (7) – Request failed because an undefined configuration was requested • eQMI_LOC_INSUFFICIENT_MEMORY (8) – Request failed because the engine could not allocate sufficient memory for the request • eQMI_LOC_MAX_GEOFENCE_PROGRAMMED (9) – Request failed because the maximum number of Geofences are already programmed • eQMI_LOC_XTRA_VERSION_CHECK_FAILURE (10) – Location service failed because of an XTRA version-based file format check failure

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_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. It is recommended that only one client control the SBAS configuration, since it impacts the global state of the location service.

3.33 QMI_LOC_GET_SBAS_CONFIG

Gets the SBAS configuration from the location engine.

LOC message ID

0x003D

Version introduced

Major - 2, Minor - 0

3.33.1 Request - QMI_LOC_GET_SBAS_CONFIG_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

None

Optional TLVs

None

3.33.2 Indication - QMI_LOC_GET_SBAS_CONFIG_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version introduced	Version last modified
Get SBAS Config Status	2.0	2.28

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

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	enum	status	4	<p>Status of the Get SBAS Configuration request.</p> <p>Valid values:</p> <ul style="list-style-type: none"> • eQMI_LOC_SUCCESS (0) – Request was completed successfully • eQMI_LOC_GENERAL_FAILURE (1) – Request failed because of a general failure • eQMI_LOC_UNSUPPORTED (2) – Request failed because it is not supported • eQMI_LOC_INVALID_PARAMETER (3) – Request failed because it contained invalid parameters • eQMI_LOC_ENGINE_BUSY (4) – Request failed because the engine is busy • eQMI_LOC_PHONE_OFFLINE (5) – Request failed because the phone is offline • eQMI_LOC_TIMEOUT (6) – Request failed because it timed out • eQMI_LOC_CONFIG_NOT_SUPPORTED (7) – Request failed because an undefined configuration was requested • eQMI_LOC_INSUFFICIENT_MEMORY (8) – Request failed because the engine could not allocate sufficient memory for the request • eQMI_LOC_MAX_GEOFENCE_PROGRAMMED (9) – Request failed because the maximum number of Geofences are already programmed • eQMI_LOC_XTRA_VERSION_CHECK_FAILURE (10) – Location service failed because of an XTRA version-based file format check failure

Optional TLVs

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

Name	Version introduced	Version last modified
SBAS Config	2.0	2.0

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	SBAS Config
Length	1			2	
Value	→	boolean	sbasConfig	1	Indicates whether SBAS configuration is enabled. <ul style="list-style-type: none"> • 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

3.33.3 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. It is safe for multiple clients to use this command.

3.34 QMI_LOC_SET_NMEA_TYPES

Sets the NMEA types.

LOC message ID

0x003E

Version introduced

Major - 2, Minor - 0

3.34.1 Request - QMI_LOC_SET_NMEA_TYPES_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

Name	Version introduced	Version last modified
NMEA Sentence Types	2.0	2.25

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	NMEA Sentence Types
Length	4			2	
Value	→	mask32	nmeaSentenceType	4	Bitmasks of NMEA types to enable. Valid bitmasks: <ul style="list-style-type: none"> • QMI_LOC_NMEA_MASK_GGA (0x00000001) – Enable GGA type • QMI_LOC_NMEA_MASK_RMC (0x00000002) – Enable RMC type • QMI_LOC_NMEA_MASK_GSV (0x00000004) – Enable GSV type • QMI_LOC_NMEA_MASK_GSA (0x00000008) – Enable GSA type • QMI_LOC_NMEA_MASK_VTG (0x00000010) – Enable VTG type • QMI_LOC_NMEA_MASK_PQXFI (0x00000020) – Enable PQXFI type • QMI_LOC_NMEA_MASK_PSTIS (0x00000040) – Enable PSTIS type • QMI_LOC_NMEA_MASK_GLSV (0x00000080) – Enable GLGSV type

Field	Field value	Field type	Parameter	Size (byte)	Description
			nmeaSentenceType (cont.)		<ul style="list-style-type: none"> • QMI_LOC_NMEA_MASK_GNGSA (0x00000100) – Enable GNGSA type • QMI_LOC_NMEA_MASK_GNGNS (0x00000200) – Enable GNGNS type

Optional TLVs

None

3.34.2 Indication - QMI_LOC_SET_NMEA_TYPES_IND**Message type**

Indication

Sender

Service

Mandatory TLVs

Name	Version introduced	Version last modified
Set NMEA Types Status	2.0	2.28

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Set NMEA Types Status
Length	4			2	
Value	→	enum	status	4	Status of Set NMEA Types request. Valid values: <ul style="list-style-type: none"> • eQMI_LOC_SUCCESS (0) – Request was completed successfully • eQMI_LOC_GENERAL_FAILURE (1) – Request failed because of a general failure • eQMI_LOC_UNSUPPORTED (2) – Request failed because it is not supported • eQMI_LOC_INVALID_PARAMETER (3) – Request failed because it contained invalid parameters • eQMI_LOC_ENGINE_BUSY (4) – Request failed because the engine is busy • eQMI_LOC_PHONE_OFFLINE (5) – Request failed because the phone is offline • eQMI_LOC_TIMEOUT (6) – Request failed because it timed out

Field	Field value	Field type	Parameter	Size (byte)	Description
			status (cont.)		<ul style="list-style-type: none"> • eQMI_LOC_CONFIG_NOT_SUPPORTED (7) – Request failed because an undefined configuration was requested • eQMI_LOC_INSUFFICIENT_MEMORY (8) – Request failed because the engine could not allocate sufficient memory for the request • eQMI_LOC_MAX_GEOFENCE_PROGRAMMED (9) – Request failed because the maximum number of Geofences are already programmed • eQMI_LOC_XTRA_VERSION_CHECK_FAILURE (10) – Location service failed because of an XTRA version-based file format check failure

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_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. It is recommended that only one client control the NMEA types, since it impacts the NMEA sentence generation for all clients.

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

3.35 QMI_LOC_GET_NMEA_TYPES

Gets the NMEA types from the location engine.

LOC message ID

0x003F

Version introduced

Major - 2, Minor - 0

3.35.1 Request - QMI_LOC_GET_NMEA_TYPES_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

None

Optional TLVs

None

3.35.2 Indication - QMI_LOC_GET_NMEA_TYPES_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version introduced	Version last modified
Get NMEA Types Status	2.0	2.28

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

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	enum	status	4	<p>Status of the Get NMEA Types request.</p> <p>Valid values:</p> <ul style="list-style-type: none"> • eQMI_LOC_SUCCESS (0) – Request was completed successfully • eQMI_LOC_GENERAL_FAILURE (1) – Request failed because of a general failure • eQMI_LOC_UNSUPPORTED (2) – Request failed because it is not supported • eQMI_LOC_INVALID_PARAMETER (3) – Request failed because it contained invalid parameters • eQMI_LOC_ENGINE_BUSY (4) – Request failed because the engine is busy • eQMI_LOC_PHONE_OFFLINE (5) – Request failed because the phone is offline • eQMI_LOC_TIMEOUT (6) – Request failed because it timed out • eQMI_LOC_CONFIG_NOT_SUPPORTED (7) – Request failed because an undefined configuration was requested • eQMI_LOC_INSUFFICIENT_MEMORY (8) – Request failed because the engine could not allocate sufficient memory for the request • eQMI_LOC_MAX_GEOFENCE_PROGRAMMED (9) – Request failed because the maximum number of Geofences are already programmed • eQMI_LOC_XTRA_VERSION_CHECK_FAILURE (10) – Location service failed because of an XTRA version-based file format check failure

Optional TLVs

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

Name	Version introduced	Version last modified
NMEA Sentence Types	2.0	2.25

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	NMEA Sentence Types
Length	4			2	
Value	→	mask32	nmeaSentenceType	4	NMEA types to enable. Valid bitmasks: <ul style="list-style-type: none"> • QMI_LOC_NMEA_MASK_GGA (0x00000001) – Enable GGA type • QMI_LOC_NMEA_MASK_RMC (0x00000002) – Enable RMC type • QMI_LOC_NMEA_MASK_GSV (0x00000004) – Enable GSV type • QMI_LOC_NMEA_MASK_GSA (0x00000008) – Enable GSA type • QMI_LOC_NMEA_MASK_VTG (0x00000010) – Enable VTG type • QMI_LOC_NMEA_MASK_PQXFI (0x00000020) – Enable PQXFI type • QMI_LOC_NMEA_MASK_PSTIS (0x00000040) – Enable PSTIS type • QMI_LOC_NMEA_MASK_GLGSV (0x00000080) – Enable GLGSV type • QMI_LOC_NMEA_MASK_GNGSA (0x00000100) – Enable GNGSA type • QMI_LOC_NMEA_MASK_GNGNS (0x00000200) – Enable GNGNS type

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_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. This command can safely be called by multiple clients.

3.36 QMI_LOC_SET_LOW_POWER_MODE

Enables/disables Low Power Mode (LPM) configuration.

LOC message ID

0x0040

Version introduced

Major - 2, Minor - 0

3.36.1 Request - QMI_LOC_SET_LOW_POWER_MODE_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

Name	Version introduced	Version last modified
Enable Low Power Mode	2.0	2.0

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

Optional TLVs

None

3.36.2 Indication - QMI_LOC_SET_LOW_POWER_MODE_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version introduced	Version last modified
Set LPM Status	2.0	2.28

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Set LPM Status
Length	4			2	
Value	→	enum	status	4	<p>Status of the Set Low Power Mode request.</p> <p>Valid values:</p> <ul style="list-style-type: none"> • eQMI_LOC_SUCCESS (0) – Request was completed successfully • eQMI_LOC_GENERAL_FAILURE (1) – Request failed because of a general failure • eQMI_LOC_UNSUPPORTED (2) – Request failed because it is not supported • eQMI_LOC_INVALID_PARAMETER (3) – Request failed because it contained invalid parameters • eQMI_LOC_ENGINE_BUSY (4) – Request failed because the engine is busy • eQMI_LOC_PHONE_OFFLINE (5) – Request failed because the phone is offline • eQMI_LOC_TIMEOUT (6) – Request failed because it timed out • eQMI_LOC_CONFIG_NOT_SUPPORTED (7) – Request failed because an undefined configuration was requested • eQMI_LOC_INSUFFICIENT_MEMORY (8) – Request failed because the engine could not allocate sufficient memory for the request • eQMI_LOC_MAX_GEOFENCE_PROGRAMMED (9) – Request failed because the maximum number of Geofences are already programmed • eQMI_LOC_XTRA_VERSION_CHECK_FAILURE (10) – Location service failed because of an XTRA version-based file format check failure

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_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. It is recommended that only one client control the low power mode, since it impacts the global state of the location service.

3.37 QMI_LOC_GET_LOW_POWER_MODE

Gets the LPM status from the location engine.

LOC message ID

0x0041

Version introduced

Major - 2, Minor - 0

3.37.1 Request - QMI_LOC_GET_LOW_POWER_MODE_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

None

Optional TLVs

None

3.37.2 Indication - QMI_LOC_GET_LOW_POWER_MODE_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version introduced	Version last modified
Get LPM Status	2.0	2.28

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

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	enum	status	4	<p>Status of the Get LPM request.</p> <p>Valid values:</p> <ul style="list-style-type: none"> • eQMI_LOC_SUCCESS (0) – Request was completed successfully • eQMI_LOC_GENERAL_FAILURE (1) – Request failed because of a general failure • eQMI_LOC_UNSUPPORTED (2) – Request failed because it is not supported • eQMI_LOC_INVALID_PARAMETER (3) – Request failed because it contained invalid parameters • eQMI_LOC_ENGINE_BUSY (4) – Request failed because the engine is busy • eQMI_LOC_PHONE_OFFLINE (5) – Request failed because the phone is offline • eQMI_LOC_TIMEOUT (6) – Request failed because it timed out • eQMI_LOC_CONFIG_NOT_SUPPORTED (7) – Request failed because an undefined configuration was requested • eQMI_LOC_INSUFFICIENT_MEMORY (8) – Request failed because the engine could not allocate sufficient memory for the request • eQMI_LOC_MAX_GEOFENCE_PROGRAMMED (9) – Request failed because the maximum number of Geofences are already programmed • eQMI_LOC_XTRA_VERSION_CHECK_FAILURE (10) – Location service failed because of an XTRA version-based file format check failure

Optional TLVs

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

Name	Version introduced	Version last modified
Enable/Disable LPM	2.0	2.0

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Enable/Disable LPM
Length	1			2	
Value	→	boolean	lowPowerMode	1	Indicates whether to enable Low Power mode: <ul style="list-style-type: none"> • 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

3.37.3 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). This command can safely be called by multiple clients.

3.38 QMI_LOC_SET_SERVER

Specifies the A-GPS server type and address.

LOC message ID

0x0042

Version introduced

Major - 2, Minor - 0

3.38.1 Request - QMI_LOC_SET_SERVER_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

Name	Version introduced	Version last modified
Server Type	2.0	2.1

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Server Type
Length	4			2	
Value	→	enum	serverType	4	Type of server. Valid values: <ul style="list-style-type: none"> • eQMI_LOC_SERVER_TYPE_CDMA_PDE (1) – Server type is CDMA PDE • eQMI_LOC_SERVER_TYPE_CDMA_MPC (2) – Server type is CDMA MPC • eQMI_LOC_SERVER_TYPE_UMTS_SLP (3) – Server type is UMTS SLP • eQMI_LOC_SERVER_TYPE_CUSTOM_PDE (4) – Server type is custom PDE

Optional TLVs

Name	Version introduced	Version last modified
IPv4 Address	2.0	2.0
IPv6 Address	2.1	2.1
Uniform Resource Locator	2.0	2.0

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	IPv4 Address IPv4 address and port.
Length	6			2	
Value	→	uint32	addr	4	IPv4 address.
		uint16	port	2	IPv4 port.
Type	0x11			1	IPv6 Address IPv6 address and port.
Length	20			2	
Value	→	uint16	addr	16	IPv6 address. • Type: Array of unsigned integers • Maximum length of the array: 8
		uint32	port	4	IPv6 port.
Type	0x12			1	Uniform Resource Locator
Length	Var			2	
Value	→	string	urlAddr	Var	URL address. • Type: NULL-terminated string • Maximum string length (including NULL terminator): 256

3.38.2 Indication - QMI_LOC_SET_SERVER_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version introduced	Version last modified
Set Server Status	2.0	2.28

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Set Server Status

Field	Field value	Field type	Parameter	Size (byte)	Description
Length	4			2	
Value	→	enum	status	4	<p>Status of the Set Server request.</p> <p>Valid values:</p> <ul style="list-style-type: none"> • eQMI_LOC_SUCCESS (0) – Request was completed successfully • eQMI_LOC_GENERAL_FAILURE (1) – Request failed because of a general failure • eQMI_LOC_UNSUPPORTED (2) – Request failed because it is not supported • eQMI_LOC_INVALID_PARAMETER (3) – Request failed because it contained invalid parameters • eQMI_LOC_ENGINE_BUSY (4) – Request failed because the engine is busy • eQMI_LOC_PHONE_OFFLINE (5) – Request failed because the phone is offline • eQMI_LOC_TIMEOUT (6) – Request failed because it timed out • eQMI_LOC_CONFIG_NOT_SUPPORTED (7) – Request failed because an undefined configuration was requested • eQMI_LOC_INSUFFICIENT_MEMORY (8) – Request failed because the engine could not allocate sufficient memory for the request • eQMI_LOC_MAX_GEOFENCE_PROGRAMMED (9) – Request failed because the maximum number of Geofences are already programmed • eQMI_LOC_XTRA_VERSION_CHECK_FAILURE (10) – Location service failed because of an XTRA version-based file format check failure

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_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. It is recommended that only one client control the AGPS server configuration, since the same configuration is used across all clients.

3.39 QMI_LOC_GET_SERVER

Gets the location server from the location engine.

LOC message ID

0x0043

Version introduced

Major - 2, Minor - 0

3.39.1 Request - QMI_LOC_GET_SERVER_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

Name	Version introduced	Version last modified
Server Type	2.0	2.1

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Server Type
Length	4			2	
Value	→	enum	serverType	4	Type of server. Valid values: <ul style="list-style-type: none"> • eQMI_LOC_SERVER_TYPE_CDMA_PDE (1) – Server type is CDMA PDE • eQMI_LOC_SERVER_TYPE_CDMA_MPC (2) – Server type is CDMA MPC • eQMI_LOC_SERVER_TYPE_UMTS_SLP (3) – Server type is UMTS SLP • eQMI_LOC_SERVER_TYPE_CUSTOM_PDE (4) – Server type is custom PDE

Optional TLVs

Name	Version introduced	Version last modified
Server Address Type	2.0	2.1

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Server Address Type
Length	1			2	
Value	→	mask8	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: <ul style="list-style-type: none"> • 0x01 – IPv4 • 0x02 – IPv6 • 0x04 – URL

3.39.2 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 introduced	Version last modified
Get Server Status	2.0	2.28
Server Type	2.0	2.1

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

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	enum	status	4	<p>Status of the Get Server request.</p> <p>Valid values:</p> <ul style="list-style-type: none"> • eQMI_LOC_SUCCESS (0) – Request was completed successfully • eQMI_LOC_GENERAL_FAILURE (1) – Request failed because of a general failure • eQMI_LOC_UNSUPPORTED (2) – Request failed because it is not supported • eQMI_LOC_INVALID_PARAMETER (3) – Request failed because it contained invalid parameters • eQMI_LOC_ENGINE_BUSY (4) – Request failed because the engine is busy • eQMI_LOC_PHONE_OFFLINE (5) – Request failed because the phone is offline • eQMI_LOC_TIMEOUT (6) – Request failed because it timed out • eQMI_LOC_CONFIG_NOT_SUPPORTED (7) – Request failed because an undefined configuration was requested • eQMI_LOC_INSUFFICIENT_MEMORY (8) – Request failed because the engine could not allocate sufficient memory for the request • eQMI_LOC_MAX_GEOFENCE_PROGRAMMED (9) – Request failed because the maximum number of Geofences are already programmed • eQMI_LOC_XTRA_VERSION_CHECK_FAILURE (10) – Location service failed because of an XTRA version-based file format check failure
Type	0x02			1	Server Type
Length	4			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	enum	serverType	4	Type of server. Valid values: <ul style="list-style-type: none"> • eQMI_LOC_SERVER_TYPE_CDMA_PDE (1) – Server type is CDMA PDE • eQMI_LOC_SERVER_TYPE_CDMA_MPC (2) – Server type is CDMA MPC • eQMI_LOC_SERVER_TYPE_UMTS_SLP (3) – Server type is UMTS SLP • eQMI_LOC_SERVER_TYPE_CUSTOM_PDE (4) – Server type is custom PDE

Optional TLVs

Name	Version introduced	Version last modified
IPv4 Address	2.0	2.1
IPv6 Address	2.1	2.1
Uniform Resource Locator	2.0	2.0

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	IPv4 Address IPv4 address and port.
Length	6			2	
Value	→	uint32	addr	4	IPv4 address.
		uint16	port	2	IPv4 port.
Type	0x11			1	IPv6 Address IPv6 address and port.
Length	20			2	
Value	→	uint16	addr	16	IPv6 address. <ul style="list-style-type: none"> • Type: Array of unsigned integers • Maximum length of the array: 8
		uint32	port	4	IPv6 port.
Type	0x12			1	Uniform Resource Locator
Length	Var			2	
Value	→	string	urlAddr	Var	URL. <ul style="list-style-type: none"> • 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

3.39.3 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. This command can safely be called by multiple clients.

3.40 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 - 0

3.40.1 Request - QMI_LOC_DELETE_ASSIST_DATA_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

Name	Version introduced	Version last modified
Delete All	2.1	2.1

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Delete All
Length	1			2	
Value	→	boolean	deleteAllFlag	1	Indicates whether all assistance data is to be deleted. Valid values: <ul style="list-style-type: none"> 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 introduced	Version last modified
Delete SV Info	2.1	2.1
Delete GNSS Data	2.1	2.30
Delete Cell Database	2.1	2.1

Name	Version introduced	Version last modified
Delete Clock Info	2.1	2.1
Delete BDS SV Info	2.22	2.22

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Delete SV Info List of satellites for which the assistance data is to be deleted.
Length	Var			2	
Value	→	uint8	deleteSvInfoList_len	1	Number of sets of the following elements: • gnssSvId • system • deleteSvInfoMask
		uint16	gnssSvId	2	SV ID of the satellite whose data is to be deleted. • Range: – For GPS: 1 to 32 – For SBAS: 33 to 64 – For GLONASS: 65 to 96
		enum	system	4	Indicates to which constellation this SV belongs. Valid values: • eQMI_LOC_SV_SYSTEM_GPS (1) – GPS satellite • eQMI_LOC_SV_SYSTEM_GALILEO (2) – GALILEO satellite • eQMI_LOC_SV_SYSTEM_SBAS (3) – SBAS satellite • eQMI_LOC_SV_SYSTEM_COMPASS (4) – COMPASS satellite • eQMI_LOC_SV_SYSTEM_GLONASS (5) – GLONASS satellite • eQMI_LOC_SV_SYSTEM_BDS (6) – BDS satellite
		mask8	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	Field type	Parameter	Size (byte)	Description
Value	→	mask	deleteGnssDataMask	8	<p>Mask for the GNSS data that is to be deleted.</p> <p>Valid values:</p> <ul style="list-style-type: none"> • QMI_LOC_MASK_DELETE_GPS_SVDIR (0x00000001) – Mask to delete GPS SVDIR • QMI_LOC_MASK_DELETE_GPS_SVSTEER (0x00000002) – Mask to delete GPS SVSTEER • QMI_LOC_MASK_DELETE_GPS_TIME (0x00000004) – Mask to delete GPS time • QMI_LOC_MASK_DELETE_GPS_ALM_CORR (0x00000008) – Mask to delete almanac correlation • QMI_LOC_MASK_DELETE_GLO_SVDIR (0x00000010) – Mask to delete GLONASS SVDIR • QMI_LOC_MASK_DELETE_GLO_SVSTEER (0x00000020) – Mask to delete GLONASS SVSTEER • QMI_LOC_MASK_DELETE_GLO_TIME (0x00000040) – Mask to delete GLONASS time • QMI_LOC_MASK_DELETE_GLO_ALM_CORR (0x00000080) – Mask to delete GLONASS almanac correlation • QMI_LOC_MASK_DELETE_SBAS_SVDIR (0x00000100) – Mask to delete SBAS SVDIR • QMI_LOC_MASK_DELETE_SBAS_SVSTEER (0x00000200) – Mask to delete SBAS SVSTEER • QMI_LOC_MASK_DELETE_POSITION (0x00000400) – Mask to delete position estimate • QMI_LOC_MASK_DELETE_TIME (0x00000800) – Mask to delete time estimate • QMI_LOC_MASK_DELETE_IONO (0x00001000) – Mask to delete IONO • QMI_LOC_MASK_DELETE_UTC (0x00002000) – Mask to delete UTC estimate • QMI_LOC_MASK_DELETE_HEALTH (0x00004000) – Mask to delete SV health record

Field	Field value	Field type	Parameter	Size (byte)	Description
			deleteGnssDataMask (cont.)		<ul style="list-style-type: none"> • QMI_LOC_MASK_DELETE_SADATA (0x00008000) – Mask to delete SADATA • QMI_LOC_MASK_DELETE_RTI (0x00010000) – Mask to delete RTI • QMI_LOC_MASK_DELETE_SV_NO_EXIST (0x00020000) – Mask to delete SV_NO_EXIST • QMI_LOC_MASK_DELETE_FREQ_BIAS_EST (0x00040000) – Mask to delete frequency bias estimate • QMI_LOC_MASK_DELETE_BDS_SVDIR (0x00080000) – Mask to delete BDS SVDIR • QMI_LOC_MASK_DELETE_BDS_SVSTEER (0x00100000) – Mask to delete BDS SVSTEER • QMI_LOC_MASK_DELETE_BDS_TIME (0x00200000) – Mask to delete BDS time • QMI_LOC_MASK_DELETE_BDS_ALM_CORR (0x00400000) – Mask to delete BDS almanac correlation • QMI_LOC_MASK_DELETE_GNSS_SV_BLACKLIST_GPS (0x00800000) – Mask to delete GNSS SV blacklist GPS • QMI_LOC_MASK_DELETE_GNSS_SV_BLACKLIST_GLO (0x01000000) – Mask to delete GNSS SV blacklist GLO • QMI_LOC_MASK_DELETE_GNSS_SV_BLACKLIST_BDS (0x02000000) – Mask to delete GNSS SV blacklist BDS
Type	0x12			1	Delete Cell Database
Length	4			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	mask32	deleteCellDbDataMask	4	Mask for the cell database assistance data that is to be deleted. Valid values: <ul style="list-style-type: none"> • 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	
Value	→	mask32	deleteClockInfoMask	4	Mask for the clock information assistance data that is to be deleted. Valid bitmasks: <ul style="list-style-type: none"> • QMI_LOC_MASK_DELETE_CLOCK_INFO_TIME_EST (0x00000001) – Mask to delete time estimate from clock information • QMI_LOC_MASK_DELETE_CLOCK_INFO_FREQ_EST (0x00000002) – Mask to delete frequency estimate from clock information • QMI_LOC_MASK_DELETE_CLOCK_INFO_WEEK_NUMBER (0x00000004) – Mask to delete week number from clock information • QMI_LOC_MASK_DELETE_CLOCK_INFO_RTC_TIME (0x00000008) – Mask to delete RTC time from clock information

Field	Field value	Field type	Parameter	Size (byte)	Description
			deleteClockInfoMask (cont.)		<ul style="list-style-type: none"> • QMI_LOC_MASK_DELETE_CLOCK_INFO_TIME_TRANSFER (0x00000010) – Mask to delete time transfer from clock information • QMI_LOC_MASK_DELETE_CLOCK_INFO_GPSTIME_EST (0x00000020) – Mask to delete GPS time estimate from clock information • QMI_LOC_MASK_DELETE_CLOCK_INFO_GLOTIME_EST (0x00000040) – Mask to delete GLONASS time estimate from clock information • QMI_LOC_MASK_DELETE_CLOCK_INFO_GLODAY_NUMBER (0x00000080) – Mask to delete GLONASS day number from clock information • QMI_LOC_MASK_DELETE_CLOCK_INFO_GLO4YEAR_NUMBER (0x00000100) – Mask to delete GLONASS four year number from clock information • QMI_LOC_MASK_DELETE_CLOCK_INFO_GLO_RF_GRP_DELAY (0x00000200) – Mask to delete GLONASS RF GRP delay from clock information • QMI_LOC_MASK_DELETE_CLOCK_INFO_DISABLE_TT (0x00000400) – Mask to delete disable TT from clock information • QMI_LOC_MASK_DELETE_CLOCK_INFO_GG_LEAPSEC (0x00000800) – Mask to delete a BDS time estimate from the clock information • QMI_LOC_MASK_DELETE_CLOCK_INFO_GG_GGTB (0x00001000) – Mask to delete a BDS time estimate from the clock information • QMI_LOC_MASK_DELETE_CLOCK_INFO_BDSTIME_EST (0x00002000) – Mask to delete a BDS time estimate from the clock information • QMI_LOC_MASK_DELETE_CLOCK_INFO_GB_GBTB (0x00004000) – Mask to delete Glonass-to-BDS time bias-related information from the clock information

Field	Field value	Field type	Parameter	Size (byte)	Description
			deleteClockInfoMask (cont.)		<ul style="list-style-type: none"> • QMI_LOC_MASK_DELETE_CLOCK_INFO_BG_BGTB (0x00008000) – Mask to delete BDS-to-GLONASS time bias-related information from the clock information • QMI_LOC_MASK_DELETE_CLOCK_INFO_BDSWEEK_NUMBER (0x00010000) – Mask to delete the BDS week number from the clock information • QMI_LOC_MASK_DELETE_CLOCK_INFO_BDS_RF_GRP_DELAY (0x00020000) – Mask to delete the BDS RF GRP delay from the clock information
Type	0x14			1	Delete BDS SV Info List of BDS satellites for which the assistance data is to be deleted.
Length	Var			2	
Value	→	uint8	deleteBdsSvInfoList_len	1	Number of sets of the following elements: <ul style="list-style-type: none"> • gnssSvId • deleteSvInfoMask
		uint16	gnssSvId	2	SV ID of the satellite whose data is to be deleted. Range for BDS: 201 to 237
		mask8	deleteSvInfoMask	1	Indicates if the ephemeris or almanac for a satellite is to be deleted. Valid values: <ul style="list-style-type: none"> • QMI_LOC_MASK_DELETE_EPHEMERIS (0x01) – Delete ephemeris for the satellite • QMI_LOC_MASK_DELETE_ALMANAC (0x02) – Delete almanac for the satellite

3.40.2 Indication - QMI_LOC_DELETE_ASSIST_DATA_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version introduced	Version last modified
Delete Assist Data Status	2.0	2.28

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Delete Assist Data Status
Length	4			2	
Value	→	enum	status	4	<p>Status of the Delete Assist Data request. Valid values:</p> <ul style="list-style-type: none"> • eQMI_LOC_SUCCESS (0) – Request was completed successfully • eQMI_LOC_GENERAL_FAILURE (1) – Request failed because of a general failure • eQMI_LOC_UNSUPPORTED (2) – Request failed because it is not supported • eQMI_LOC_INVALID_PARAMETER (3) – Request failed because it contained invalid parameters • eQMI_LOC_ENGINE_BUSY (4) – Request failed because the engine is busy • eQMI_LOC_PHONE_OFFLINE (5) – Request failed because the phone is offline • eQMI_LOC_TIMEOUT (6) – Request failed because it timed out • eQMI_LOC_CONFIG_NOT_SUPPORTED (7) – Request failed because an undefined configuration was requested • eQMI_LOC_INSUFFICIENT_MEMORY (8) – Request failed because the engine could not allocate sufficient memory for the request • eQMI_LOC_MAX_GEOFENCE_PROGRAMMED (9) – Request failed because the maximum number of Geofences are already programmed • eQMI_LOC_XTRA_VERSION_CHECK_FAILURE (10) – Location service failed because of an XTRA version-based file format check failure

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

3.40.3 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. Deleting assistance data will impact the time to first fix for all other positioning clients, hence it is recommended that only one client delete assistance data.

3.41 QMI_LOC_SET_XTRA_T_SESSION_CONTROL

Enables/disables XTRA-T session control.

LOC message ID

0x0045

Version introduced

Major - 2, Minor - 0

3.41.1 Request - QMI_LOC_SET_XTRA_T_SESSION_CONTROL_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

Name	Version introduced	Version last modified
Enable XTRA-T	2.0	2.0

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

Optional TLVs

None

3.41.2 Indication - QMI_LOC_SET_XTRA_T_SESSION_CONTROL_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version introduced	Version last modified
Set XTRA-T Session Control Status	2.0	2.28

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Set XTRA-T Session Control Status
Length	4			2	
Value	→	enum	status	4	<p>Status of the Set XTRA-T Session Control request.</p> <p>Valid values:</p> <ul style="list-style-type: none"> • eQMI_LOC_SUCCESS (0) – Request was completed successfully • eQMI_LOC_GENERAL_FAILURE (1) – Request failed because of a general failure • eQMI_LOC_UNSUPPORTED (2) – Request failed because it is not supported • eQMI_LOC_INVALID_PARAMETER (3) – Request failed because it contained invalid parameters • eQMI_LOC_ENGINE_BUSY (4) – Request failed because the engine is busy • eQMI_LOC_PHONE_OFFLINE (5) – Request failed because the phone is offline • eQMI_LOC_TIMEOUT (6) – Request failed because it timed out • eQMI_LOC_CONFIG_NOT_SUPPORTED (7) – Request failed because an undefined configuration was requested • eQMI_LOC_INSUFFICIENT_MEMORY (8) – Request failed because the engine could not allocate sufficient memory for the request • eQMI_LOC_MAX_GEOFENCE_PROGRAMMED (9) – Request failed because the maximum number of Geofences are already programmed • eQMI_LOC_XTRA_VERSION_CHECK_FAILURE (10) – Location service failed because of an XTRA version-based file format check failure

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_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. It is recommended that only one client control the XTRA-T configuration, since the same configuration is used across all clients.

3.42 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 - 0

3.42.1 Request - QMI_LOC_GET_XTRA_T_SESSION_CONTROL_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

None

Optional TLVs

None

3.42.2 Indication - QMI_LOC_GET_XTRA_T_SESSION_CONTROL_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version introduced	Version last modified
Get XTRA-T Session Control Status	2.0	2.28

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Get XTRA-T Session Control Status
Length	4			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	enum	status	4	<p>Status of the Get XTRA-T Session Control request.</p> <p>Valid values:</p> <ul style="list-style-type: none"> • eQMI_LOC_SUCCESS (0) – Request was completed successfully • eQMI_LOC_GENERAL_FAILURE (1) – Request failed because of a general failure • eQMI_LOC_UNSUPPORTED (2) – Request failed because it is not supported • eQMI_LOC_INVALID_PARAMETER (3) – Request failed because it contained invalid parameters • eQMI_LOC_ENGINE_BUSY (4) – Request failed because the engine is busy • eQMI_LOC_PHONE_OFFLINE (5) – Request failed because the phone is offline • eQMI_LOC_TIMEOUT (6) – Request failed because it timed out • eQMI_LOC_CONFIG_NOT_SUPPORTED (7) – Request failed because an undefined configuration was requested • eQMI_LOC_INSUFFICIENT_MEMORY (8) – Request failed because the engine could not allocate sufficient memory for the request • eQMI_LOC_MAX_GEOFENCE_PROGRAMMED (9) – Request failed because the maximum number of Geofences are already programmed • eQMI_LOC_XTRA_VERSION_CHECK_FAILURE (10) – Location service failed because of an XTRA version-based file format check failure

Optional TLVs

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

Name	Version introduced	Version last modified
Enable/Disable XTRA-T	2.0	2.0

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Enable/Disable XTRA-T
Length	1			2	
Value	→	boolean	xtraTSessionControl	1	Indicates 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

3.42.3 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). This command can safely be called by multiple clients.

3.43 QMI_LOC_INJECT_WIFI_POSITION

Injects the Wi-Fi position.

LOC message ID

0x0047

Version introduced

Major - 2, Minor - 0

3.43.1 Request - QMI_LOC_INJECT_WIFI_POSITION_REQ

Message type

Request

Sender

Control Point

Mandatory TLVs

None

Optional TLVs

Name	Version introduced	Version last modified
Wi-Fi Fix Time	2.0	2.0
Wi-Fi Position	2.0	2.1
Wi-Fi Access Point Information	2.0	2.1
Horizontal Reliability	2.0	2.1
Raw HEPE	2.29	2.29
Wi-Fi AP SSID String	2.32	2.32

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Wi-Fi Fix Time Time of Wi-Fi position fix.
Length	4			2	
Value	→	uint32	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	0x11			1	Wi-Fi Position Wi-Fi position fix.
Length	23			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	double	lat	8	Wi-Fi position latitude. • Type: Floating point • Units: Degrees
		double	lon	8	Wi-Fi position longitude. • Type: Floating point • Units: Degrees
		uint16	hepe	2	Wi-Fi position HEPE. • Units: Meters
		uint8	numApsUsed	1	Number of Access Points (AP) used to generate a fix.
		enum	fixErrorCode	4	Wi-Fi 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 Wi-Fi positioning system can be provided here. Valid values: • eQMI_LOC_WIFI_FIX_ERROR_SUCCESS (0) – Wi-Fi fix is successful. • eQMI_LOC_WIFI_FIX_ERROR_WIFI_NOT_AVAILABLE (1) – Wi-Fi fix failed because Wi-Fi is not available on the device. • eQMI_LOC_WIFI_FIX_ERROR_NO_AP_FOUND (2) – Wi-Fi fix failed because no access points were found. • eQMI_LOC_WIFI_FIX_ERROR_UNAUTHORIZED (3) – Wi-Fi fix failed because the server denied access due to bad authorization code. • eQMI_LOC_WIFI_FIX_ERROR_SERVER_UNAVAILABLE (4) – Wi-Fi fix failed because the Wi-Fi server was unavailable. • eQMI_LOC_WIFI_FIX_ERROR_LOCATION_CANNOT_BE_DETERMINED (5) – Wi-Fi fix failed even though APs were found and the server could be reached. This may be because the APs found are not in the database. • eQMI_LOC_WIFI_FIX_ERROR_UNKNOWN (6) – Wi-Fi fix failed, but the cause could not be determined.

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x12			1	Wi-Fi Access Point Information AP scan list. SSID of the Wi-Fi AP. The ordering of the Wi-Fi AP SSID list should match the Wi-Fi AP MAC address list if both are provided, i.e., the first element of the Wi-Fi AP SSID list must be the SSID of the AP whose MAC address is in the first element in the Wi-Fi AP Info MAC Address, etc.
Length	Var			2	
Value	→	uint8	apInfo_len	1	Number of sets of the following elements: • macAddr • rssi • channel • apQualifier
		uint8	macAddr	6	Associated MAC address of the AP. • Type: Array of unsigned integers • Address length: 6
		int32	rssi	4	Receive signal strength indicator. • Units: dBm (offset with +100 dB)
		uint16	channel	2	Wi-Fi channel on which a beacon was received.
		mask8	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	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	enum	horizontalReliability	4	Specifies the reliability of the horizontal position. Valid values: <ul style="list-style-type: none"> • eQMI_LOC_RELIABILITY_NOT_SET (0) – Location reliability is not set • eQMI_LOC_RELIABILITY_VERY_LOW (1) – Location reliability is very low; use it at your own risk • eQMI_LOC_RELIABILITY_LOW (2) – Location reliability is low; little or no cross-checking is possible • eQMI_LOC_RELIABILITY_MEDIUM (3) – Location reliability is medium; limited cross-check passed • eQMI_LOC_RELIABILITY_HIGH (4) – Location reliability is high; strong cross-check passed
Type	0x14			1	Raw HEPE
Length	2			2	
Value	→	uint16	rawHepe	2	Wi-Fi position raw HEPE, which has no optimization. <ul style="list-style-type: none"> • Units: Meters
Type	0x15			1	Wi-Fi AP SSID String The ordering of the Wi-Fi AP SSID list should match the Wi-Fi AP MAC address list if both are provided, i.e., the first element of the Wi-Fi AP SSID list must be the SSID of the AP whose MAC address is in the first element in the Wi-Fi AP Info MAC address, etc.
Length	Var			2	
Value	→	uint8	wifiApSsidInfo_len	1	Number of sets of the following elements: <ul style="list-style-type: none"> • ssid_len • ssid
		uint8	ssid_len	1	Number of sets of the following elements: <ul style="list-style-type: none"> • ssid
		string	ssid	Var	NULL-terminated SSID string of the Wi-Fi AP. Its maximum length according to the ASCII standard is 32 octets.

3.43.2 Indication - QMI_LOC_INJECT_WIFI_POSITION_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version introduced	Version last modified
Inject Wi-Fi Position Status	2.0	2.28

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Inject Wi-Fi Position Status
Length	4			2	
Value	→	enum	status	4	<p>Status of the Inject Wi-Fi Position request.</p> <p>Valid values:</p> <ul style="list-style-type: none"> • eQMI_LOC_SUCCESS (0) – Request was completed successfully • eQMI_LOC_GENERAL_FAILURE (1) – Request failed because of a general failure • eQMI_LOC_UNSUPPORTED (2) – Request failed because it is not supported • eQMI_LOC_INVALID_PARAMETER (3) – Request failed because it contained invalid parameters • eQMI_LOC_ENGINE_BUSY (4) – Request failed because the engine is busy • eQMI_LOC_PHONE_OFFLINE (5) – Request failed because the phone is offline • eQMI_LOC_TIMEOUT (6) – Request failed because it timed out • eQMI_LOC_CONFIG_NOT_SUPPORTED (7) – Request failed because an undefined configuration was requested • eQMI_LOC_INSUFFICIENT_MEMORY (8) – Request failed because the engine could not allocate sufficient memory for the request

Field	Field value	Field type	Parameter	Size (byte)	Description
			status (cont.)		<ul style="list-style-type: none"> • eQMI_LOC_MAX_GEOFENCE_PROGRAMMED (9) – Request failed because the maximum number of Geofences are already programmed • eQMI_LOC_XTRA_VERSION_CHECK_FAILURE (10) – Location service failed because of an XTRA version-based file format check failure

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

3.43.3 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. A client injecting the coarse position affects the global state of the engine, thereby impacting all other clients.

How to set the reliability indicator:

It is suggested that this command 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 optional TLVs for the request).

3.44 QMI_LOC_NOTIFY_WIFI_STATUS

Notifies the location engine of the Wi-Fi status.

LOC message ID

0x0048

Version introduced

Major - 2, Minor - 0

3.44.1 Request - QMI_LOC_NOTIFY_WIFI_STATUS_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

Name	Version introduced	Version last modified
Availability of Wi-Fi	2.0	2.1

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Availability of Wi-Fi
Length	4			2	
Value	→	enum	wifiStatus	4	Wi-Fi status information. Valid values: • eQMI_LOC_WIFI_STATUS_AVAILABLE (1) – Wi-Fi is available • eQMI_LOC_WIFI_STATUS_UNAVAILABLE (2) – Wi-Fi is not available

Optional TLVs

None

3.44.2 Indication - QMI_LOC_NOTIFY_WIFI_STATUS_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version introduced	Version last modified
Status of Notify Wi-Fi Status Request	2.0	2.28

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Status of Notify Wi-Fi Status Request
Length	4			2	
Value	→	enum	status	4	<p>Status of the Notify Wi-Fi Status request.</p> <p>Valid values:</p> <ul style="list-style-type: none"> • eQMI_LOC_SUCCESS (0) – Request was completed successfully • eQMI_LOC_GENERAL_FAILURE (1) – Request failed because of a general failure • eQMI_LOC_UNSUPPORTED (2) – Request failed because it is not supported • eQMI_LOC_INVALID_PARAMETER (3) – Request failed because it contained invalid parameters • eQMI_LOC_ENGINE_BUSY (4) – Request failed because the engine is busy • eQMI_LOC_PHONE_OFFLINE (5) – Request failed because the phone is offline • eQMI_LOC_TIMEOUT (6) – Request failed because it timed out • eQMI_LOC_CONFIG_NOT_SUPPORTED (7) – Request failed because an undefined configuration was requested • eQMI_LOC_INSUFFICIENT_MEMORY (8) – Request failed because the engine could not allocate sufficient memory for the request • eQMI_LOC_MAX_GEOFENCE_PROGRAMMED (9) – Request failed because the maximum number of Geofences are already programmed • eQMI_LOC_XTRA_VERSION_CHECK_FAILURE (10) – Location service failed because of an XTRA version-based file format check failure

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_NOTIFY_WIFI_STATUS

This command is used to notify the location engine of the Wi-Fi status. The request is acknowledged through the response, but the SUCCESS/FAILURE status is sent through the indication QMI_LOC_NOTIFY_WIFI_STATUS_IND. The Wi-Fi status is a global state in the engine, so it is recommended that only one client notify the engine of the Wi-Fi status.

3.45 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 - 0

3.45.1 Request - QMI_LOC_GET_REGISTERED_EVENTS_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

None

Optional TLVs

None

3.45.2 Indication - QMI_LOC_GET_REGISTERED_EVENTS_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version introduced	Version last modified
Get Registered Events Status	2.0	2.28

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

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	enum	status	4	<p>Status of the Get Registered Events request.</p> <p>Valid values:</p> <ul style="list-style-type: none"> • eQMI_LOC_SUCCESS (0) – Request was completed successfully • eQMI_LOC_GENERAL_FAILURE (1) – Request failed because of a general failure • eQMI_LOC_UNSUPPORTED (2) – Request failed because it is not supported • eQMI_LOC_INVALID_PARAMETER (3) – Request failed because it contained invalid parameters • eQMI_LOC_ENGINE_BUSY (4) – Request failed because the engine is busy • eQMI_LOC_PHONE_OFFLINE (5) – Request failed because the phone is offline • eQMI_LOC_TIMEOUT (6) – Request failed because it timed out • eQMI_LOC_CONFIG_NOT_SUPPORTED (7) – Request failed because an undefined configuration was requested • eQMI_LOC_INSUFFICIENT_MEMORY (8) – Request failed because the engine could not allocate sufficient memory for the request • eQMI_LOC_MAX_GEOFENCE_PROGRAMMED (9) – Request failed because the maximum number of Geofences are already programmed • eQMI_LOC_XTRA_VERSION_CHECK_FAILURE (10) – Location service failed because of an XTRA version-based file format check failure

Optional TLVs

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

Name	Version introduced	Version last modified
Event Registration Mask	2.0	2.31

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Event Registration Mask
Length	8			2	
Value	→	mask	eventRegMask	8	<p>Event registration mask. Valid bitmasks:</p> <ul style="list-style-type: none"> • QMI_LOC_EVENT_MASK_POSITION_REPORT (0x00000001) – The control point must enable this mask to receive position report event indications. • QMI_LOC_EVENT_MASK_GNSS_SV_INFO (0x00000002) – The control point must enable this mask to receive satellite report event indications. These reports are sent at a 1 Hz rate. • QMI_LOC_EVENT_MASK_NMEA (0x00000004) – The control point must enable this mask to receive NMEA reports for position and satellites in view. The report is at a 1 Hz rate. • QMI_LOC_EVENT_MASK_NI_NOTIFY_VERIFY_REQ (0x00000008) – The control point must enable this mask to receive NI Notify/Verify request event indications. • QMI_LOC_EVENT_MASK_INJECT_TIME_REQ (0x00000010) – The control point must enable this mask to receive time injection request event indications. • QMI_LOC_EVENT_MASK_INJECT_PREDICTED_ORBITS_REQ (0x00000020) – The control point must enable this mask to receive predicted orbits request event indications. • QMI_LOC_EVENT_MASK_INJECT_POSITION_REQ (0x00000040) – The control point must enable this mask to receive position injection request event indications. • QMI_LOC_EVENT_MASK_ENGINE_STATE (0x00000080) – The control point must enable this mask to receive engine state report event indications. • QMI_LOC_EVENT_MASK_FIX_SESSION_STATE (0x00000100) – The control point must enable this mask to receive fix session status report event indications.

Field	Field value	Field type	Parameter	Size (byte)	Description
			eventRegMask (cont.)		<ul style="list-style-type: none"> • QMI_LOC_EVENT_MASK_WIFI_REQ (0x00000200) – The control point must enable this mask to receive Wi-Fi position request event indications. • QMI_LOC_EVENT_MASK_SENSOR_STREAMING_READY_STATUS (0x00000400) – The control point must enable this mask to receive notifications from the location engine indicating its readiness to accept data from the sensors (accelerometer, gyroscope, etc.). • QMI_LOC_EVENT_MASK_TIME_SYNC_REQ (0x00000800) – The control point must enable this mask to receive time sync requests from the GPS engine. Time sync enables the GPS engine to synchronize its clock with the sensor processor's clock. • QMI_LOC_EVENT_MASK_SET_SPI_STREAMING_REPORT (0x00001000) – The control point must enable this mask to receive Stationary Position Indicator (SPI) streaming report indications. • QMI_LOC_EVENT_MASK_LOCATION_SERVER_CONNECTION_REQ (0x00002000) – The control point must enable this mask to receive location server requests. These requests are generated when the service wishes to establish a connection with a location server. • QMI_LOC_EVENT_MASK_NI_GEOFENCE_NOTIFICATION (0x00004000) – The control point must enable this mask to receive notifications related to network-initiated Geofences. These events notify the client when a network-initiated Geofence is added, deleted, or edited. • QMI_LOC_EVENT_MASK_GEOFENCE_GEN_ALERT (0x00008000) – The control point must enable this mask to receive Geofence alerts. These alerts are generated to inform the client of the changes that may affect a Geofence, e.g., if GPS is turned off or if the network is unavailable.

Field	Field value	Field type	Parameter	Size (byte)	Description
			eventRegMask (cont.)		<ul style="list-style-type: none"> • QMI_LOC_EVENT_MASK_GEOFENCE_BREACH_NOTIFICATION (0x00010000) – The control point must enable this mask to receive notifications when a Geofence is breached. These events are generated when a UE enters or leaves the perimeter of a Geofence. This breach report is for a single Geofence . • QMI_LOC_EVENT_MASK_PEDOMETER_CONTROL (0x00020000) – The control point must enable this mask to register for pedometer control requests from the location engine. The location engine sends this event to control the injection of pedometer reports. • QMI_LOC_EVENT_MASK_MOTION_DATA_CONTROL (0x00040000) – The control point must enable this mask to register for motion data control requests from the location engine. The location engine sends this event to control the injection of motion data. • QMI_LOC_EVENT_MASK_BATCH_FULL_NOTIFICATION (0x00080000) – The control point must enable this mask to receive notification when a batch is full. The location engine sends this event to notify of Batch Full for ongoing batching session. • QMI_LOC_EVENT_MASK_LIVE_BATCHED_POSITION_REPORT (0x00100000) – The control point must enable this mask to receive position report indications along with an ongoing batching session. The location engine sends this event to notify the batched position report while a batching session is ongoing. • QMI_LOC_EVENT_MASK_INJECT_WIFI_AP_DATA_REQ (0x00200000) – The control point must enable this mask to receive Wi-Fi AP data inject request event indications.

Field	Field value	Field type	Parameter	Size (byte)	Description
			eventRegMask (cont.)		<ul style="list-style-type: none"> • QMI_LOC_EVENT_MASK_GEOFENCE_BATCH_BREACH_NOTIFICATION (0x00400000) – The control point must enable this mask to receive notifications when a Geofence is breached. These events are generated when a UE enters or leaves the perimeter of a Geofence. This breach notification is for multiple Geofences. Breaches from multiple Geofences are all batched and sent in the same notification . • QMI_LOC_EVENT_MASK_VEHICLE_DATA_READY_STATUS (0x00800000) – The control point must enable this mask to receive notifications from the location engine indicating its readiness to accept vehicle data (vehicle accelerometer, vehicle angular rate, vehicle odometry, etc.). • QMI_LOC_EVENT_MASK_GNSS_MEASUREMENT_REPORT (0x01000000) – The control point must enable this mask to receive system clock and satellite measurement report events (system clock, SV time, Doppler, etc.). Reports are generated only for the GNSS satellite constellations that are enabled using QMI_LOC_SET_GNSS_CONSTELL_REPORT_CONFIG. • QMI_LOC_EVENT_MASK_GNSS_SV_POLYNOMIAL_REPORT (0x02000000) – The control point must enable this mask to receive satellite position reports as polynomials. Reports are generated only for the GNSS satellite constellations that are enabled using QMI_LOC_SET_GNSS_CONSTELL_REPORT_CONFIG.

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_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. It is safe for multiple clients to use this command.

QUALCOMM®
2016-05-16 00:17:18 PDT
deon_zhang@askey.com.tw

3.46 QMI_LOC_SET_OPERATION_MODE

Tells the engine to use the specified operation mode while making the position fixes.

LOC message ID

0x004A

Version introduced

Major - 2, Minor - 0

3.46.1 Request - QMI_LOC_SET_OPERATION_MODE_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

Name	Version introduced	Version last modified
Operation Mode	2.0	2.13

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Operation Mode
Length	4			2	
Value	→	enum	operationMode	4	Preferred operation mode. Valid values: <ul style="list-style-type: none"> • eQMI_LOC_OPER_MODE_DEFAULT (1) – Use the default engine mode • eQMI_LOC_OPER_MODE_MSB (2) – Use the MS-based mode • eQMI_LOC_OPER_MODE_MSA (3) – Use the MS-assisted mode • eQMI_LOC_OPER_MODE_STANDALONE (4) – Use Standalone mode • eQMI_LOC_OPER_MODE_CELL_ID (5) – Use cell ID; this mode is only valid for GSM/UMTS networks • eQMI_LOC_OPER_MODE_WWAN (6) – Use WWAN measurements to calculate the position; if this mode is set, AFLT will be used for 1X networks and OTDOA will be used for LTE networks

Optional TLVs

None

3.46.2 Indication - QMI_LOC_SET_OPERATION_MODE_IND**Message type**

Indication

Sender

Service

Mandatory TLVs

Name	Version introduced	Version last modified
Set Operation Mode Status	2.0	2.28

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Set Operation Mode Status
Length	4			2	
Value	→	enum	status	4	<p>Status of the Set Operation Mode request.</p> <p>Valid values:</p> <ul style="list-style-type: none"> • eQMI_LOC_SUCCESS (0) – Request was completed successfully • eQMI_LOC_GENERAL_FAILURE (1) – Request failed because of a general failure • eQMI_LOC_UNSUPPORTED (2) – Request failed because it is not supported • eQMI_LOC_INVALID_PARAMETER (3) – Request failed because it contained invalid parameters • eQMI_LOC_ENGINE_BUSY (4) – Request failed because the engine is busy • eQMI_LOC_PHONE_OFFLINE (5) – Request failed because the phone is offline • eQMI_LOC_TIMEOUT (6) – Request failed because it timed out • eQMI_LOC_CONFIG_NOT_SUPPORTED (7) – Request failed because an undefined configuration was requested

Field	Field value	Field type	Parameter	Size (byte)	Description
			status (cont.)		<ul style="list-style-type: none"> • eQMI_LOC_INSUFFICIENT_MEMORY (8) – Request failed because the engine could not allocate sufficient memory for the request • eQMI_LOC_MAX_GEOFENCE_PROGRAMMED (9) – Request failed because the maximum number of Geofences are already programmed • eQMI_LOC_XTRA_VERSION_CHECK_FAILURE (10) – Location service failed because of an XTRA version-based file format check failure

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_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.

3.47 QMI_LOC_GET_OPERATION_MODE

Gets the current operation mode from the engine.

LOC message ID

0x004B

Version introduced

Major - 2, Minor - 0

3.47.1 Request - QMI_LOC_GET_OPERATION_MODE_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

None

Optional TLVs

None

3.47.2 Indication - QMI_LOC_GET_OPERATION_MODE_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version introduced	Version last modified
Get Operation Mode Status	2.0	2.28

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

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	enum	status	4	<p>Status of the Get Operation Mode request.</p> <p>Valid values:</p> <ul style="list-style-type: none"> • eQMI_LOC_SUCCESS (0) – Request was completed successfully • eQMI_LOC_GENERAL_FAILURE (1) – Request failed because of a general failure • eQMI_LOC_UNSUPPORTED (2) – Request failed because it is not supported • eQMI_LOC_INVALID_PARAMETER (3) – Request failed because it contained invalid parameters • eQMI_LOC_ENGINE_BUSY (4) – Request failed because the engine is busy • eQMI_LOC_PHONE_OFFLINE (5) – Request failed because the phone is offline • eQMI_LOC_TIMEOUT (6) – Request failed because it timed out • eQMI_LOC_CONFIG_NOT_SUPPORTED (7) – Request failed because an undefined configuration was requested • eQMI_LOC_INSUFFICIENT_MEMORY (8) – Request failed because the engine could not allocate sufficient memory for the request • eQMI_LOC_MAX_GEOFENCE_PROGRAMMED (9) – Request failed because the maximum number of Geofences are already programmed • eQMI_LOC_XTRA_VERSION_CHECK_FAILURE (10) – Location service failed because of an XTRA version-based file format check failure

Optional TLVs

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

Name	Version introduced	Version last modified
Operation Mode	2.0	2.13

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Operation Mode
Length	4			2	
Value	→	enum	operationMode	4	Current operation mode. Valid values: <ul style="list-style-type: none"> • eQMI_LOC_OPER_MODE_DEFAULT (1) – Use the default engine mode • eQMI_LOC_OPER_MODE_MSB (2) – Use the MS-based mode • eQMI_LOC_OPER_MODE_MSA (3) – Use the MS-assisted mode • eQMI_LOC_OPER_MODE_STANDALONE (4) – Use Standalone mode • eQMI_LOC_OPER_MODE_CELL_ID (5) – Use cell ID; this mode is only valid for GSM/UMTS networks • eQMI_LOC_OPER_MODE_WWAN (6) – Use WWAN measurements to calculate the position; if this mode is set, AFLT will be used for 1X networks and OTDOA will be used for LTE networks

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_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. It is safe for multiple clients to use this command.

3.48 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 - 0

3.48.1 Request - QMI_LOC_SET_SPI_STATUS_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

Name	Version introduced	Version last modified
Stationary Status	2.0	2.0

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Stationary Status
Length	1			2	
Value	→	boolean	stationary	1	Indicates whether the device is stationary: <ul style="list-style-type: none"> 0x00 (FALSE) – Device is not stationary 0x01 (TRUE) – Device is stationary

Optional TLVs

Name	Version introduced	Version last modified
Confidence	2.0	2.0

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Confidence
Length	1			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	uint8	confidenceStationary	1	Confidence in the Stationary state expressed as a percentage. • Range: 0 to 100

3.48.2 Indication - QMI_LOC_SET_SPI_STATUS_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version introduced	Version last modified
Status of SPI Status Request	2.0	2.28

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Status of SPI Status Request
Length	4			2	
Value	→	enum	status	4	Status of the SPI Status request. Valid values: • eQMI_LOC_SUCCESS (0) – Request was completed successfully • eQMI_LOC_GENERAL_FAILURE (1) – Request failed because of a general failure • eQMI_LOC_UNSUPPORTED (2) – Request failed because it is not supported • eQMI_LOC_INVALID_PARAMETER (3) – Request failed because it contained invalid parameters • eQMI_LOC_ENGINE_BUSY (4) – Request failed because the engine is busy • eQMI_LOC_PHONE_OFFLINE (5) – Request failed because the phone is offline • eQMI_LOC_TIMEOUT (6) – Request failed because it timed out • eQMI_LOC_CONFIG_NOT_SUPPORTED (7) – Request failed because an undefined configuration was requested

Field	Field value	Field type	Parameter	Size (byte)	Description
			status (cont.)		<ul style="list-style-type: none"> • eQMI_LOC_INSUFFICIENT_MEMORY (8) – Request failed because the engine could not allocate sufficient memory for the request • eQMI_LOC_MAX_GEOFENCE_PROGRAMMED (9) – Request failed because the maximum number of Geofences are already programmed • eQMI_LOC_XTRA_VERSION_CHECK_FAILURE (10) – Location service failed because of an XTRA version-based file format check failure

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_SET_SPI_STATUS

This command is used by the control point to inject the current SPI status. The SPI status is a global state in the engine, so it is recommended that only one client notify the engine of the SPI status

3.49 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 - 0

3.49.1 Request - QMI_LOC_INJECT_SENSOR_DATA_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

None

Optional TLVs

Name	Version introduced	Version last modified
Opaque Identifier	2.0	2.0
3-Axis Accelerometer Data	2.0	2.11
3-Axis Gyroscope Data	2.0	2.11
3-Axis Accelerometer Data Time Source	2.17	2.17
3-Axis Gyroscope Data Time Source	2.17	2.17
Accelerometer Temperature Data	2.17	2.17
Gyroscope Temperature Data	2.17	2.17

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Opaque Identifier
Length	4			2	
Value	→	uint32	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	0x11			1	3-Axis Accelerometer Data
Length	Var			2	Accelerometer sensor samples.

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	uint32	timeOfFirstSample	4	Denotes a full 32-bit timestamp of the first (oldest) sample in this message. The timestamp is in the time reference scale that is used by the sensor time source. • Units: Milliseconds
		mask8	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: • QMI_LOC_SENSOR_DATA_FLAG_SIGN_REVERSAL (0x01) – Bitmask to specify that a sign reversal is required while interpreting the sensor data; only applies to the accelerometer samples • QMI_LOC_SENSOR_DATA_FLAG_SENSOR_TIME_IS_MODEM_TIME (0x02) – Bitmask to specify that the sensor time stamp is the same as the modem time stamp
		uint8	sensorData_len	1	Number of sets of the following elements: • timeOffset • xAxis • yAxis • zAxis
		uint16	timeOffset	2	Sample time offset. This time offset must be relative to the timestamp of the first sensor data sample. • Units: Milliseconds
		float	xAxis	4	Sensor x-axis sample. • Units Accelerometer: Meters/seconds ² • Units Gyroscope: Radians/second
		float	yAxis	4	Sensor y-axis sample. • Units Accelerometer: Meters/seconds ² • Units Gyroscope: Radians/second
		float	zAxis	4	Sensor z-axis sample. • Units Accelerometer: Meters/seconds ² • Units Gyroscope: Radians/second
Type	0x12			1	3-Axis Gyroscope Data Gyroscope sensor samples.
Length	Var			2	
Value	→	uint32	timeOfFirstSample	4	Denotes a full 32-bit timestamp of the first (oldest) sample in this message. The timestamp is in the time reference scale that is used by the sensor time source. • Units: Milliseconds

Field	Field value	Field type	Parameter	Size (byte)	Description
		mask8	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: <ul style="list-style-type: none"> • QMI_LOC_SENSOR_DATA_FLAG_SIGN_REVERSAL (0x01) – Bitmask to specify that a sign reversal is required while interpreting the sensor data; only applies to the accelerometer samples • QMI_LOC_SENSOR_DATA_FLAG_SENSOR_TIME_IS_MODEM_TIME (0x02) – Bitmask to specify that the sensor time stamp is the same as the modem time stamp
		uint8	sensorData_len	1	Number of sets of the following elements: <ul style="list-style-type: none"> • timeOffset • xAxis • yAxis • zAxis
		uint16	timeOffset	2	Sample time offset. This time offset must be relative to the timestamp of the first sensor data sample. <ul style="list-style-type: none"> • Units: Milliseconds
		float	xAxis	4	Sensor x-axis sample. <ul style="list-style-type: none"> • Units Accelerometer: Meters/seconds² • Units Gyroscope: Radians/second
		float	yAxis	4	Sensor y-axis sample. <ul style="list-style-type: none"> • Units Accelerometer: Meters/seconds² • Units Gyroscope: Radians/second
		float	zAxis	4	Sensor z-axis sample. <ul style="list-style-type: none"> • Units Accelerometer: Meters/seconds² • Units Gyroscope: Radians/second
Type	0x13			1	3-Axis Accelerometer Data Time Source
Length	4			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	enum	threeAxisAccelDataTime Source	4	Time source for the 3-axis accelerometer data. The location service uses this field to identify the time reference used in the accelerometer data timestamps. If not specified, the location service assumes that the time source for the accelerometer data is unknown. Values: <ul style="list-style-type: none"> • eQMI_LOC_SENSOR_TIME_SOURCE_UNSPECIFIED (0) – Sensor time source is unspecified • eQMI_LOC_SENSOR_TIME_SOURCE_COMMON (1) – Time source is common between the sensors and the location engine
Type	0x14			1	3-Axis Gyroscope Data Time Source
Length	4			2	
Value	→	enum	threeAxisGyroDataTime Source	4	Time source for the 3-axis gyroscope data. The location service uses this field to identify the time reference used in the gyroscope data timestamps. If not specified, the location service assumes that the time source for the gyroscope data is unknown. Values: <ul style="list-style-type: none"> • eQMI_LOC_SENSOR_TIME_SOURCE_UNSPECIFIED (0) – Sensor time source is unspecified • eQMI_LOC_SENSOR_TIME_SOURCE_COMMON (1) – Time source is common between the sensors and the location engine
Type	0x15			1	Accelerometer Temperature Data Accelerometer temperature samples. This data is optional and does not have to be included in the message along with accelerometer data.
Length	Var			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	enum	timeSource	4	Denotes the time source of the sensor data. Location service will use this field to identify the time reference used in the sensor data timestamps. Valid values: • eQMI_LOC_SENSOR_TIME_SOURCE_UNSPECIFIED (0) – Sensor time source is unspecified • eQMI_LOC_SENSOR_TIME_SOURCE_COMMON (1) – Time source is common between the sensors and the location engine
		uint32	timeOfFirstSample	4	Denotes a full 32-bit timestamp of the first (oldest) sample in this message. The timestamp is in the time reference scale that is used by the sensor time source. • Units: Milliseconds
		uint8	temperatureData_len	1	Number of sets of the following elements: • timeOffset • temperature
		uint16	timeOffset	2	Sample time offset. This time offset must be relative to the timestamp of the first sensor sample. • Units: Milliseconds
		float	temperature	4	Sensor temperature. • Type: Floating point • Units: Degrees Celsius • Range: -50 to +100.00
Type	0x16			1	Gyroscope Temperature Data Gyroscope temperature samples. This data is optional and does not have to be included in the message along with gyroscope data.
Length	Var			2	
Value	→	enum	timeSource	4	Denotes the time source of the sensor data. Location service will use this field to identify the time reference used in the sensor data timestamps. Valid values: • eQMI_LOC_SENSOR_TIME_SOURCE_UNSPECIFIED (0) – Sensor time source is unspecified • eQMI_LOC_SENSOR_TIME_SOURCE_COMMON (1) – Time source is common between the sensors and the location engine

Field	Field value	Field type	Parameter	Size (byte)	Description
		uint32	timeOfFirstSample	4	Denotes a full 32-bit timestamp of the first (oldest) sample in this message. The timestamp is in the time reference scale that is used by the sensor time source. • Units: Milliseconds
		uint8	temperatureData_len	1	Number of sets of the following elements: • timeOffset • temperature
		uint16	timeOffset	2	Sample time offset. This time offset must be relative to the timestamp of the first sensor sample. • Units: Milliseconds
		float	temperature	4	Sensor temperature. • Type: Floating point • Units: Degrees Celsius • Range: -50 to +100.00

3.49.2 Indication - QMI_LOC_INJECT_SENSOR_DATA_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version introduced	Version last modified
Inject Sensor Data Status	2.0	2.28

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

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	enum	status	4	<p>Status of the Inject Sensor Data request.</p> <p>Valid values:</p> <ul style="list-style-type: none"> • eQMI_LOC_SUCCESS (0) – Request was completed successfully • eQMI_LOC_GENERAL_FAILURE (1) – Request failed because of a general failure • eQMI_LOC_UNSUPPORTED (2) – Request failed because it is not supported • eQMI_LOC_INVALID_PARAMETER (3) – Request failed because it contained invalid parameters • eQMI_LOC_ENGINE_BUSY (4) – Request failed because the engine is busy • eQMI_LOC_PHONE_OFFLINE (5) – Request failed because the phone is offline • eQMI_LOC_TIMEOUT (6) – Request failed because it timed out • eQMI_LOC_CONFIG_NOT_SUPPORTED (7) – Request failed because an undefined configuration was requested • eQMI_LOC_INSUFFICIENT_MEMORY (8) – Request failed because the engine could not allocate sufficient memory for the request • eQMI_LOC_MAX_GEOFENCE_PROGRAMMED (9) – Request failed because the maximum number of Geofences are already programmed • eQMI_LOC_XTRA_VERSION_CHECK_FAILURE (10) – Location service failed because of an XTRA version-based file format check failure

Optional TLVs

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

Name	Version introduced	Version last modified
Opaque Identifier	2.0	2.0
Accelerometer Samples Accepted	2.0	2.0
Gyroscope Samples Accepted	2.0	2.0
Accelerometer Temperature Samples Accepted	2.17	2.17
Gyroscope Temperature Samples Accepted	2.17	2.17

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Opaque Identifier
Length	4			2	
Value	→	uint32	opaqueIdentifier	4	Opaque identifier that was sent in by the client echoed so the client can relate the indication to the request.
Type	0x11			1	Accelerometer Samples Accepted
Length	1			2	
Value	→	uint8	threeAxisAccelSamples Accepted	1	Lets the client know how many 3-axis accelerometer samples were accepted. This field is present only if the accelerometer samples were sent in the request.
Type	0x12			1	Gyroscope Samples Accepted
Length	1			2	
Value	→	uint8	threeAxisGyroSamples Accepted	1	Lets the client know how many 3-axis gyroscope samples were accepted. This field is present only if the gyroscope samples were sent in the request.
Type	0x13			1	Accelerometer Temperature Samples Accepted
Length	1			2	
Value	→	uint8	accelTemperatureSamples Accepted	1	Lets the client know how many accelerometer temperature samples were accepted. This field is present only if the accelerometer temperature samples were sent in the request.
Type	0x14			1	Gyroscope Temperature Samples Accepted
Length	1			2	
Value	→	uint8	gyroTemperatureSamples Accepted	1	Lets the client know how many gyroscope temperature samples were accepted. This field is present only if the gyroscope temperature samples were sent in the request.

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_INJECT_SENSOR_DATA

This command is used by the control point to inject sensor data into the location engine. The `timeOfFirstSample` field must be the time stamp of the oldest sample in the message, regardless of the sample type. Thus, the oldest sample in the message must have a `timeOffset` value of zero, and all other values of `timeOffset` must be nonzero. Samples for each sample type must be provided in chronological order.

The sensor data must only be sent if the location service indicates its readiness to accept sensor data in the `QMI_LOC_EVENT_SENSOR_STREAMING_READY_STATUS` indication.

The time stamps associated with the injected sensor samples are expected to be monotonically increasing and to increase at approximately the same rate as GPS time as determined by the location service. The initial relationship of GPS time to sensor time is established by the `QMI_LOC_INJECT_TIME_SYNC` command.

The sensor time stamps of this command are expected to have an average time between samples of no less than 8 ms (approximately 125 Hz).

QUALCOMM
2016-05-16 00:17:18 PDT
deon_zhang@askey.com.tw

3.50 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 - 0

3.50.1 Request - QMI_LOC_INJECT_TIME_SYNC_DATA_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

Name	Version introduced	Version last modified
Reference Time Sync Counter	2.0	2.0
Sensor Receive Time	2.0	2.0
Sensor Transmit Time	2.0	2.0

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Reference Time Sync Counter
Length	4			2	
Value	→	uint32	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	0x02			1	Sensor Receive Time
Length	4			2	
Value	→	uint32	sensorProcRxTime	4	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. • Units: Milliseconds
Type	0x03			1	Sensor Transmit Time
Length	4			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	uint32	sensorProcTxTime	4	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. • Units: Milliseconds

Optional TLVs

None

3.50.2 Indication - QMI_LOC_INJECT_TIME_SYNC_DATA_IND**Message type**

Indication

Sender

Service

Mandatory TLVs

Name	Version introduced	Version last modified
Inject Time Sync Data Status	2.0	2.28

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Inject Time Sync Data Status
Length	4			2	
Value	→	enum	status	4	Status of the Inject Time Sync Data request. Valid values: • eQMI_LOC_SUCCESS (0) – Request was completed successfully • eQMI_LOC_GENERAL_FAILURE (1) – Request failed because of a general failure • eQMI_LOC_UNSUPPORTED (2) – Request failed because it is not supported • eQMI_LOC_INVALID_PARAMETER (3) – Request failed because it contained invalid parameters • eQMI_LOC_ENGINE_BUSY (4) – Request failed because the engine is busy

Field	Field value	Field type	Parameter	Size (byte)	Description
			status (cont.)		<ul style="list-style-type: none"> • eQMI_LOC_PHONE_OFFLINE (5) – Request failed because the phone is offline • eQMI_LOC_TIMEOUT (6) – Request failed because it timed out • eQMI_LOC_CONFIG_NOT_SUPPORTED (7) – Request failed because an undefined configuration was requested • eQMI_LOC_INSUFFICIENT_MEMORY (8) – Request failed because the engine could not allocate sufficient memory for the request • eQMI_LOC_MAX_GEOFENCE_PROGRAMMED (9) – Request failed because the maximum number of Geofences are already programmed • eQMI_LOC_XTRA_VERSION_CHECK_FAILURE (10) – Location service failed because of an XTRA version-based file format check failure

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_INJECT_TIME_SYNC_DATA

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

3.51 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 - 0

3.51.1 Request - QMI_LOC_SET_CRADLE_MOUNT_CONFIG_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

Name	Version introduced	Version last modified
Cradle Mount State	2.0	2.2

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Cradle Mount State
Length	4			2	
Value	→	enum	cradleMountState	4	Cradle Mount state set by the control point. Valid values: <ul style="list-style-type: none"> • eQMI_LOC_CRADLE_STATE_NOT_MOUNTED (0) – Device is mounted on the cradle • eQMI_LOC_CRADLE_STATE_MOUNTED (1) – Device is not mounted on the cradle • eQMI_LOC_CRADLE_STATE_UNKNOWN (2) – Unknown cradle mount state

Optional TLVs

Name	Version introduced	Version last modified
Cradle Mount Confidence	2.2	2.2

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Cradle Mount Confidence
Length	1			2	
Value	→	uint8	confidenceCradleMount State	1	Confidence in the Cradle Mount state expressed as a percentage. • Range: 0 to 100

3.51.2 Indication - QMI_LOC_SET_CRADLE_MOUNT_CONFIG_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version introduced	Version last modified
Set Cradle Mount Config Status	2.0	2.28

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Set Cradle Mount Config Status
Length	4			2	
Value	→	enum	status	4	Status of the Set Cradle Mount Configuration request. Valid values: • eQMI_LOC_SUCCESS (0) – Request was completed successfully • eQMI_LOC_GENERAL_FAILURE (1) – Request failed because of a general failure • eQMI_LOC_UNSUPPORTED (2) – Request failed because it is not supported • eQMI_LOC_INVALID_PARAMETER (3) – Request failed because it contained invalid parameters • eQMI_LOC_ENGINE_BUSY (4) – Request failed because the engine is busy • eQMI_LOC_PHONE_OFFLINE (5) – Request failed because the phone is offline • eQMI_LOC_TIMEOUT (6) – Request failed because it timed out

Field	Field value	Field type	Parameter	Size (byte)	Description
			status (cont.)		<ul style="list-style-type: none"> • eQMI_LOC_CONFIG_NOT_SUPPORTED (7) – Request failed because an undefined configuration was requested • eQMI_LOC_INSUFFICIENT_MEMORY (8) – Request failed because the engine could not allocate sufficient memory for the request • eQMI_LOC_MAX_GEOFENCE_PROGRAMMED (9) – Request failed because the maximum number of Geofences are already programmed • eQMI_LOC_XTRA_VERSION_CHECK_FAILURE (10) – Location service failed because of an XTRA version-based file format check failure

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

3.51.3 Description of QMI_LOC_SET_CRADLE_MOUNT_CONFIG

This command is used by the control point to set the current cradle mount configuration. The cradle mount configuration is a global state in the engine, so it is recommended that only one client control the cradle mount. configuration.

3.52 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 - 0

3.52.1 Request - QMI_LOC_GET_CRADLE_MOUNT_CONFIG_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

None

Optional TLVs

None

3.52.2 Indication - QMI_LOC_GET_CRADLE_MOUNT_CONFIG_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version introduced	Version last modified
Get Cradle Mount Config Status	2.0	2.28

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Get Cradle Mount Config Status
Length	4			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	enum	status	4	<p>Status of the Get Cradle Mount Configuration request.</p> <p>Valid values:</p> <ul style="list-style-type: none"> • eQMI_LOC_SUCCESS (0) – Request was completed successfully • eQMI_LOC_GENERAL_FAILURE (1) – Request failed because of a general failure • eQMI_LOC_UNSUPPORTED (2) – Request failed because it is not supported • eQMI_LOC_INVALID_PARAMETER (3) – Request failed because it contained invalid parameters • eQMI_LOC_ENGINE_BUSY (4) – Request failed because the engine is busy • eQMI_LOC_PHONE_OFFLINE (5) – Request failed because the phone is offline • eQMI_LOC_TIMEOUT (6) – Request failed because it timed out • eQMI_LOC_CONFIG_NOT_SUPPORTED (7) – Request failed because an undefined configuration was requested • eQMI_LOC_INSUFFICIENT_MEMORY (8) – Request failed because the engine could not allocate sufficient memory for the request • eQMI_LOC_MAX_GEOFENCE_PROGRAMMED (9) – Request failed because the maximum number of Geofences are already programmed • eQMI_LOC_XTRA_VERSION_CHECK_FAILURE (10) – Location service failed because of an XTRA version-based file format check failure

Optional TLVs

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

Name	Version introduced	Version last modified
Cradle Mount State	2.0	2.2
Cradle Mount Confidence	2.2	2.2

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Cradle Mount State
Length	4			2	
Value	→	enum	cradleMountState	4	Cradle Mount state set by the control point. Valid values: <ul style="list-style-type: none"> • eQMI_LOC_CRADLE_STATE_NOT_MOUNTED (0) – Device is mounted on the cradle • eQMI_LOC_CRADLE_STATE_MOUNTED (1) – Device is not mounted on the cradle • eQMI_LOC_CRADLE_STATE_UNKNOWN (2) – Unknown cradle mount state
Type	0x11			1	Cradle Mount Confidence
Length	1			2	
Value	→	uint8	confidenceCradleMount State	1	Confidence of the Cradle Mount state expressed as a percentage. <ul style="list-style-type: none"> • 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

3.52.3 Description of QMI_LOC_GET_CRADLE_MOUNT_CONFIG

This command is used by the control point to get the current cradle mount configuration. It is safe for multiple clients to use this command.

3.53 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 - 0

3.53.1 Request - QMI_LOC_SET_EXTERNAL_POWER_CONFIG_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

Name	Version introduced	Version last modified
External Power State	2.0	2.1

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	External Power State
Length	4			2	
Value	→	enum	externalPowerState	4	Power state; injected by the control point. Valid values: <ul style="list-style-type: none"> • eQMI_LOC_EXTERNAL_POWER_NOT_CONNECTED (0) – Device is not connected to an external power source • eQMI_LOC_EXTERNAL_POWER_CONNECTED (1) – Device is connected to an external power source • eQMI_LOC_EXTERNAL_POWER_UNKNOWN (2) – Unknown external power state

Optional TLVs

None

3.53.2 Indication - QMI_LOC_SET_EXTERNAL_POWER_CONFIG_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version introduced	Version last modified
Set Ext Power Config Status	2.0	2.28

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Set Ext Power Config Status
Length	4			2	
Value	→	enum	status	4	<p>Status of the Set External Power Configuration request.</p> <p>Valid values:</p> <ul style="list-style-type: none"> • eQMI_LOC_SUCCESS (0) – Request was completed successfully • eQMI_LOC_GENERAL_FAILURE (1) – Request failed because of a general failure • eQMI_LOC_UNSUPPORTED (2) – Request failed because it is not supported • eQMI_LOC_INVALID_PARAMETER (3) – Request failed because it contained invalid parameters • eQMI_LOC_ENGINE_BUSY (4) – Request failed because the engine is busy • eQMI_LOC_PHONE_OFFLINE (5) – Request failed because the phone is offline • eQMI_LOC_TIMEOUT (6) – Request failed because it timed out • eQMI_LOC_CONFIG_NOT_SUPPORTED (7) – Request failed because an undefined configuration was requested • eQMI_LOC_INSUFFICIENT_MEMORY (8) – Request failed because the engine could not allocate sufficient memory for the request

Field	Field value	Field type	Parameter	Size (byte)	Description
			status (cont.)		<ul style="list-style-type: none"> • eQMI_LOC_MAX_GEOFENCE_PROGRAMMED (9) – Request failed because the maximum number of Geofences are already programmed • eQMI_LOC_XTRA_VERSION_CHECK_FAILURE (10) – Location service failed because of an XTRA version-based file format check failure

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_SET_EXTERNAL_POWER_CONFIG

This command is used by the control point to set the current external power configuration. The external power configuration is a global state in the engine, so it is recommended that only one client control the external power. configuration.

3.54 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 - 0

3.54.1 Request - QMI_LOC_GET_EXTERNAL_POWER_CONFIG_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

None

Optional TLVs

None

3.54.2 Indication - QMI_LOC_GET_EXTERNAL_POWER_CONFIG_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version introduced	Version last modified
Get Ext Power Config Status	2.0	2.28

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Get Ext Power Config Status
Length	4			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	enum	status	4	<p>Status of the Get External Power Configuration request.</p> <p>Valid values:</p> <ul style="list-style-type: none"> • eQMI_LOC_SUCCESS (0) – Request was completed successfully • eQMI_LOC_GENERAL_FAILURE (1) – Request failed because of a general failure • eQMI_LOC_UNSUPPORTED (2) – Request failed because it is not supported • eQMI_LOC_INVALID_PARAMETER (3) – Request failed because it contained invalid parameters • eQMI_LOC_ENGINE_BUSY (4) – Request failed because the engine is busy • eQMI_LOC_PHONE_OFFLINE (5) – Request failed because the phone is offline • eQMI_LOC_TIMEOUT (6) – Request failed because it timed out • eQMI_LOC_CONFIG_NOT_SUPPORTED (7) – Request failed because an undefined configuration was requested • eQMI_LOC_INSUFFICIENT_MEMORY (8) – Request failed because the engine could not allocate sufficient memory for the request • eQMI_LOC_MAX_GEOFENCE_PROGRAMMED (9) – Request failed because the maximum number of Geofences are already programmed • eQMI_LOC_XTRA_VERSION_CHECK_FAILURE (10) – Location service failed because of an XTRA version-based file format check failure

Optional TLVs

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

Name	Version introduced	Version last modified
External Power State	2.0	2.1

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	External Power State
Length	4			2	
Value	→	enum	externalPowerState	4	Power state; injected by the control point. Valid values: <ul style="list-style-type: none"> • eQMI_LOC_EXTERNAL_POWER_NOT_CONNECTED (0) – Device is not connected to an external power source • eQMI_LOC_EXTERNAL_POWER_CONNECTED (1) – Device is connected to an external power source • eQMI_LOC_EXTERNAL_POWER_UNKNOWN (2) – Unknown external power state

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_GET_EXTERNAL_POWER_CONFIG

This command is used by the control point to get the current external power configuration. It is safe for multiple clients to use this command.

3.55 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_CONNECTION_REQ_IND event.

LOC message ID

0x0053

Version introduced

Major - 2, Minor - 1

3.55.1 Request - QMI_LOC_INFORM_LOCATION_SERVER_CONN_STATUS_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

Name	Version introduced	Version last modified
Connection Handle	2.1	2.1
Request Type	2.1	2.1
Connection Status	2.1	2.1

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Connection Handle
Length	4			2	
Value	→	uint32	connHandle	4	Connection handle that the service specified in the Location Server Connection request event.
Type	0x02			1	Request Type
Length	4			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	enum	requestType	4	Type of connection request service that was specified in the Location Server Connection Request event. Valid values: • eQMI_LOC_SERVER_REQUEST_OPEN (1) – Open a connection to the location server • eQMI_LOC_SERVER_REQUEST_CLOSE (2) – Close a connection to the location server
Type	0x03			1	Connection Status
Length	4			2	
Value	→	enum	statusType	4	Status of the Connection request. Valid values: • eQMI_LOC_SERVER_REQ_STATUS_SUCCESS (1) – Location server request was successful • eQMI_LOC_SERVER_REQ_STATUS_FAILURE (2) – Location server request failed

Optional TLVs

Name	Version introduced	Version last modified
APN Profile	2.1	2.1

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	APN Profile Access Point Name (APN) profile information is present only when requestType is OPEN and statusType is SUCCESS.
Length	Var			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	enum	pdnType	4	PDN type of the APN profile. Valid values: <ul style="list-style-type: none"> • eQMI_LOC_APN_PROFILE_PDN_TYPE_IPV4 (0x01) – IPv4 PDN type • eQMI_LOC_APN_PROFILE_PDN_TYPE_IPV6 (0x02) – IPv6 PDN type • eQMI_LOC_APN_PROFILE_PDN_TYPE_IPV4V6 (0x03) – IPv4v6 PDN type • eQMI_LOC_APN_PROFILE_PDN_TYPE_PPP (0x04) – PPP PDN type
		uint8	apnName_len	1	Number of sets of the following elements: <ul style="list-style-type: none"> • apnName
		string	apnName	Var	APN name. <ul style="list-style-type: none"> • Type: NULL-terminated string • Maximum string length (including NULL terminator): 101

3.55.2 Indication - QMI_LOC_INFORM_LOCATION_SERVER_CONN_STATUS_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version introduced	Version last modified
Status of Inform Loc Server Conn Status	2.1	2.28

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Status of Inform Loc Server Conn Status
Length	4			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	enum	status	4	<p>Status of the Inform Location Server Connection Status request.</p> <p>Valid values:</p> <ul style="list-style-type: none"> • eQMI_LOC_SUCCESS (0) – Request was completed successfully • eQMI_LOC_GENERAL_FAILURE (1) – Request failed because of a general failure • eQMI_LOC_UNSUPPORTED (2) – Request failed because it is not supported • eQMI_LOC_INVALID_PARAMETER (3) – Request failed because it contained invalid parameters • eQMI_LOC_ENGINE_BUSY (4) – Request failed because the engine is busy • eQMI_LOC_PHONE_OFFLINE (5) – Request failed because the phone is offline • eQMI_LOC_TIMEOUT (6) – Request failed because it timed out • eQMI_LOC_CONFIG_NOT_SUPPORTED (7) – Request failed because an undefined configuration was requested • eQMI_LOC_INSUFFICIENT_MEMORY (8) – Request failed because the engine could not allocate sufficient memory for the request • eQMI_LOC_MAX_GEOFENCE_PROGRAMMED (9) – Request failed because the maximum number of Geofences are already programmed • eQMI_LOC_XTRA_VERSION_CHECK_FAILURE (10) – Location service failed because of an XTRA version-based file format check failure

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

3.55.3 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_CONNECTION_REQ_IND. It is recommended that only one client register for and respond to the location server connection request.

3.56 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 - 1

3.56.1 Request - QMI_LOC_SET_PROTOCOL_CONFIG_PARAMETERS - REQ

Message type

Request

Sender

Control point

Mandatory TLVs

None

Optional TLVs

Name	Version introduced	Version last modified
SUPL Security	2.1	2.1
VX Version	2.1	2.1
SUPL Version	2.2	2.2
LPP Configuration	2.9	2.9
Assisted GLONASS Protocol Mask	2.13	2.21
SUPL Hash Algorithm	2.17	2.17
SUPL TLS Version	2.17	2.17
Emergency Protocol	2.17	2.17
Wi-Fi Scan Injection Timeout Period	2.24	2.24

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

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	boolean	suplSecurity	1	Indicates whether SUPL security is enabled. <ul style="list-style-type: none"> • 0x01 (TRUE) – SUPL security is enabled • 0x00 (FALSE) – SUPL security is disabled
Type	0x11			1	VX Version
Length	4			2	
Value	→	enum	vxVersion	4	VX version. Valid values: <ul style="list-style-type: none"> • eQMI_LOC_VX_VERSION_V1_ONLY (1) – V1 VX version • eQMI_LOC_VX_VERSION_V2_ONLY (2) – V2 VX version
Type	0x12			1	SUPL Version
Length	4			2	
Value	→	enum	suplVersion	4	SUPL version. Valid values: <ul style="list-style-type: none"> • eQMI_LOC_SUPL_VERSION_1_0 (1) – SUPL version 1.0 • eQMI_LOC_SUPL_VERSION_2_0 (2) – SUPL version 2.0
Type	0x13			1	LPP Configuration
Length	4			2	
Value	→	mask32	lppConfig	4	LTE Positioning Profile (LPP) configuration. Valid bitmasks: <ul style="list-style-type: none"> • 0x00000001 – LPP_CONFIG_ENABLE_USER_PLANE • 0x00000002 – LPP_CONFIG_ENABLE_CONTROL_PLANE
Type	0x14			1	Assisted GLONASS Protocol Mask
Length	4			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	mask32	assistedGlonassProtocolMask	4	Configures the protocols that the location service supports for assisted GLONASS. Valid bitmasks: <ul style="list-style-type: none"> • QMI_LOC_ASSISTED_GLONASS_PROTOCOL_MASK_RRC_CP (0x00000001) – Assisted GLONASS is supported over RRC in the control plane • QMI_LOC_ASSISTED_GLONASS_PROTOCOL_MASK_RRLP_UP (0x00000002) – Assisted GLONASS is supported over RRLP in the user plane • QMI_LOC_ASSISTED_GLONASS_PROTOCOL_MASK_LPP_UP (0x00000004) – Assisted GLONASS is supported over LPP in the user plane; QMI_LOC_LPP_CONFIG_ENABLE_USER_PLANE must be set in the LPP configuration for this to take effect • QMI_LOC_ASSISTED_GLONASS_PROTOCOL_MASK_LPP_CP (0x00000008) – Assisted GLONASS is supported over LPP in the control plane; QMI_LOC_LPP_CONFIG_ENABLE_CONTROL_PLANE must be set in the LPP configuration for this to take effect
Type	0x15			1	SUPL Hash Algorithm
Length	4			2	
Value	→	enum	suplHashAlgo	4	SUPL hash algorithm to be used. Valid values: <ul style="list-style-type: none"> • eQMI_LOC_SUPL_HASH_ALGO_SHA1 (0) – SHA-1 hash algorithm for SUPL version 2.0 or later • eQMI_LOC_SUPL_HASH_ALGO_SHA256 (1) – SHA-256 hash algorithm for SUPL version 2.0 or later
Type	0x16			1	SUPL TLS Version
Length	4			2	
Value	→	enum	suplTlsVersion	4	SUPL Transport Layer Security (TLS) version. This configuration is only applicable to SUPL 2.0 or later, as SUPL 1.0 always uses TLS version 1.0. Valid values: <ul style="list-style-type: none"> • eQMI_LOC_SUPL_TLS_VERSION_1_0 (0) – SUPL TLS version 1.0 • eQMI_LOC_SUPL_TLS_VERSION_1_1 (1) – SUPL TLS version 1.1
Type	0x17			1	Emergency Protocol

Field	Field value	Field type	Parameter	Size (byte)	Description
Length	4			2	
Value	→	enum	emergencyProtocol	4	Configures the protocol to be used during an emergency. Note: Currently, this can only be selected on WCDMA. For GSM and 1X, the UE only allows a control plane NI trigger for positioning. For LTE, the UE allows either a SUPL or a control plane NI trigger. Valid values: <ul style="list-style-type: none"> • eQMI_LOC_EMERGENCY_PROTOCOL_WCDMA_CP (0) – Use Control Plane Protocol during an emergency while on WCDMA • eQMI_LOC_EMERGENCY_PROTOCOL_WCDMA_UP (1) – Use SUPL 2.0 emergency services during an emergency while on WCDMA
Type	0x18			1	Wi-Fi Scan Injection Timeout Period
Length	1			2	
Value	→	uint8	wifiScanInjectTimeout	1	Configures the timeout duration that the service waits for scan results injection from the control point after the event notification is sent. Note: The timeout value is in seconds. Values: 0 to 10 seconds The minimum value (0 seconds) is the default. At this value, the service disables sending the Wi-Fi scan injection notification and ignores any scan results injection request.

3.56.2 Indication - QMI_LOC_SET_PROTOCOL_CONFIG_PARAMETERS_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version introduced	Version last modified
Set Config Params Status	2.1	2.28

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Set Config Params Status
Length	4			2	
Value	→	enum	status	4	<p>Status of the Set Configuration Parameters request.</p> <p>Valid values:</p> <ul style="list-style-type: none"> • eQMI_LOC_SUCCESS (0) – Request was completed successfully • eQMI_LOC_GENERAL_FAILURE (1) – Request failed because of a general failure • eQMI_LOC_UNSUPPORTED (2) – Request failed because it is not supported • eQMI_LOC_INVALID_PARAMETER (3) – Request failed because it contained invalid parameters • eQMI_LOC_ENGINE_BUSY (4) – Request failed because the engine is busy • eQMI_LOC_PHONE_OFFLINE (5) – Request failed because the phone is offline • eQMI_LOC_TIMEOUT (6) – Request failed because it timed out • eQMI_LOC_CONFIG_NOT_SUPPORTED (7) – Request failed because an undefined configuration was requested • eQMI_LOC_INSUFFICIENT_MEMORY (8) – Request failed because the engine could not allocate sufficient memory for the request • eQMI_LOC_MAX_GEOFENCE_PROGRAMMED (9) – Request failed because the maximum number of Geofences are already programmed • eQMI_LOC_XTRA_VERSION_CHECK_FAILURE (10) – Location service failed because of an XTRA version-based file format check failure

Optional TLVs

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

Name	Version introduced	Version last modified
Failed Parameters	2.1	2.24

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Failed Parameters
Length	8			2	
Value	→	mask	failedProtocolConfigParam Mask	8	<p>Identifies parameters that were not set successfully. This field is sent only if the status is not SUCCESS.</p> <p>Valid bitmasks:</p> <ul style="list-style-type: none"> • QMI_LOC_PROTOCOL_CONFIG_PARAM_MASK_SUPL_SECURITY (0x0000000000000001) – Mask for the SUPL security configuration parameter • QMI_LOC_PROTOCOL_CONFIG_PARAM_MASK_VX_VERSION (0x0000000000000002) – Mask for the VX version configuration parameter • QMI_LOC_PROTOCOL_CONFIG_PARAM_MASK_SUPL_VERSION (0x0000000000000004) – Mask for the SUPL version configuration parameter • QMI_LOC_PROTOCOL_CONFIG_PARAM_MASK_LPP_CONFIG (0x0000000000000008) – Mask for the LPP configuration parameter • QMI_LOC_PROTOCOL_CONFIG_PARAM_MASK_ASSISTED_GLONASS_PROTOCOL (0x0000000000000010) – Mask for the assisted GLONASS configuration parameter • QMI_LOC_PROTOCOL_CONFIG_PARAM_MASK_SUPL_HASH_ALGO (0x0000000000000020) – Mask for the SUPL hash algorithm configuration parameter • QMI_LOC_PROTOCOL_CONFIG_PARAM_MASK_SUPL_TLS_VERSION (0x0000000000000040) – Mask for the SUPL TLS version configuration parameter

Field	Field value	Field type	Parameter	Size (byte)	Description
			failedProtocolConfigParamMask (cont.)		<ul style="list-style-type: none"> • QMI_LOC_PROTOCOL_CONFIG_PARAM_MASK_EMERGENCY_PROTOCOL (0x0000000000000080) – Mask for the emergency protocol configuration parameter • QMI_LOC_PROTOCOL_CONFIG_PARAM_MASK_WIFI_SCAN_INJECT_TIMEOUT (0x0000000000000100) – Mask for the Wi-Fi scan injection timeout configuration parameter

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_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. Multiple clients should not set protocol configurations that conflict with each other, since these impact the global state of the location engine.

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

3.57 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 - 1

3.57.1 Request - QMI_LOC_GET_PROTOCOL_CONFIG_PARAMETERS - REQ

Message type

Request

Sender

Control point

Mandatory TLVs

Name	Version introduced	Version last modified
Config Parameters	2.1	2.24

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Config Parameters
Length	8			2	
Value	→	mask	getProtocolConfigParam Mask	8	Mask denoting the configuration parameters to be retrieved. Valid bitmasks: <ul style="list-style-type: none"> • QMI_LOC_PROTOCOL_CONFIG_PARAM_MASK_SUPL_SECURITY (0x0000000000000001) – Mask for the SUPL security configuration parameter • QMI_LOC_PROTOCOL_CONFIG_PARAM_MASK_VX_VERSION (0x0000000000000002) – Mask for the VX version configuration parameter • QMI_LOC_PROTOCOL_CONFIG_PARAM_MASK_SUPL_VERSION (0x0000000000000004) – Mask for the SUPL version configuration parameter

Field	Field value	Field type	Parameter	Size (byte)	Description
			getProtocolConfigParam Mask (cont.)		<ul style="list-style-type: none"> • QMI_LOC_PROTOCOL_CONFIG_PARAM_MASK_LPP_CONFIG (0x0000000000000008) – Mask for the LPP configuration parameter • QMI_LOC_PROTOCOL_CONFIG_PARAM_MASK_ASSISTED_GLONASS_PROTOCOL (0x0000000000000010) – Mask for the assisted GLONASS configuration parameter • QMI_LOC_PROTOCOL_CONFIG_PARAM_MASK_SUPL_HASH_ALGO (0x0000000000000020) – Mask for the SUPL hash algorithm configuration parameter • QMI_LOC_PROTOCOL_CONFIG_PARAM_MASK_SUPL_TLS_VERSION (0x0000000000000040) – Mask for the SUPL TLS version configuration parameter • QMI_LOC_PROTOCOL_CONFIG_PARAM_MASK_EMERGENCY_PROTOCOL (0x0000000000000080) – Mask for the emergency protocol configuration parameter • QMI_LOC_PROTOCOL_CONFIG_PARAM_MASK_WIFI_SCAN_INJECT_TIMEOUT (0x0000000000000100) – Mask for the Wi-Fi scan injection timeout configuration parameter

Optional TLVs

None

3.57.2 Indication - QMI_LOC_GET_PROTOCOL_CONFIG_PARAMETERS - IND**Message type**

Indication

Sender

Service

Mandatory TLVs

Name	Version introduced	Version last modified
Get Config Params Status	2.1	2.28

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Get Config Params Status
Length	4			2	
Value	→	enum	status	4	<p>Status of the Get Configuration Parameters request.</p> <p>Valid values:</p> <ul style="list-style-type: none"> • eQMI_LOC_SUCCESS (0) – Request was completed successfully • eQMI_LOC_GENERAL_FAILURE (1) – Request failed because of a general failure • eQMI_LOC_UNSUPPORTED (2) – Request failed because it is not supported • eQMI_LOC_INVALID_PARAMETER (3) – Request failed because it contained invalid parameters • eQMI_LOC_ENGINE_BUSY (4) – Request failed because the engine is busy • eQMI_LOC_PHONE_OFFLINE (5) – Request failed because the phone is offline • eQMI_LOC_TIMEOUT (6) – Request failed because it timed out • eQMI_LOC_CONFIG_NOT_SUPPORTED (7) – Request failed because an undefined configuration was requested • eQMI_LOC_INSUFFICIENT_MEMORY (8) – Request failed because the engine could not allocate sufficient memory for the request • eQMI_LOC_MAX_GEOFENCE_PROGRAMMED (9) – Request failed because the maximum number of Geofences are already programmed • eQMI_LOC_XTRA_VERSION_CHECK_FAILURE (10) – Location service failed because of an XTRA version-based file format check failure

Optional TLVs

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

Name	Version introduced	Version last modified
SUPL Security	2.1	2.1
VX Version	2.1	2.1
SUPL Version	2.2	2.2
LPP Configuration	2.9	2.9
Assisted GLONASS Protocol Mask	2.13	2.21
SUPL Hash Algorithm	2.17	2.17
SUPL TLS Version	2.17	2.17
Emergency Protocol	2.17	2.17
Wi-Fi Scan Injection Timeout Period	2.24	2.24

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	SUPL Security
Length	1			2	
Value	→	boolean	suplSecurity	1	Indicates whether SUPL security is enabled. <ul style="list-style-type: none"> • 0x01 (TRUE) – SUPL security is enabled • 0x00 (FALSE) – SUPL security is disabled
Type	0x11			1	VX Version
Length	4			2	
Value	→	enum	vxVersion	4	VX version. Valid values: <ul style="list-style-type: none"> • eQMI_LOC_VX_VERSION_V1_ONLY (1) – V1 VX version • eQMI_LOC_VX_VERSION_V2_ONLY (2) – V2 VX version
Type	0x12			1	SUPL Version
Length	4			2	
Value	→	enum	suplVersion	4	SUPL version. Valid values: <ul style="list-style-type: none"> • eQMI_LOC_SUPL_VERSION_1_0 (1) – SUPL version 1.0 • eQMI_LOC_SUPL_VERSION_2_0 (2) – SUPL version 2.0
Type	0x13			1	LPP Configuration
Length	4			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	mask32	lppConfig	4	LTE Positioning Profile (LPP) configuration. Valid bitmasks: <ul style="list-style-type: none"> • 0x00000001 – LPP_CONFIG_ENABLE_USER_PLANE • 0x00000002 – LPP_CONFIG_ENABLE_CONTROL_PLANE
Type	0x14			1	Assisted GLONASS Protocol Mask
Length	4			2	
Value	→	mask32	assistedGlonassProtocolMask	4	Assisted GLONASS Protocol mask. Valid bitmasks: <ul style="list-style-type: none"> • QMI_LOC_ASSISTED_GLO_NASS_PROTOCOL_MASK_RRC_CP (0x00000001) – Assisted GLONASS is supported over RRC in the control plane • QMI_LOC_ASSISTED_GLO_NASS_PROTOCOL_MASK_RRLP_UP (0x00000002) – Assisted GLONASS is supported over RRLP in the user plane • QMI_LOC_ASSISTED_GLO_NASS_PROTOCOL_MASK_LPP_UP (0x00000004) – Assisted GLONASS is supported over LPP in the user plane; QMI_LOC_LPP_CONFIG_ENABLE_USER_PLANE must be set in the LPP configuration for this to take effect • QMI_LOC_ASSISTED_GLO_NASS_PROTOCOL_MASK_LPP_CP (0x00000008) – Assisted GLONASS is supported over LPP in the control plane; QMI_LOC_LPP_CONFIG_ENABLE_CONTROL_PLANE must be set in the LPP configuration for this to take effect
Type	0x15			1	SUPL Hash Algorithm
Length	4			2	
Value	→	enum	suplHashAlgo	4	SUPL hash algorithm to be used. Valid values: <ul style="list-style-type: none"> • eQMI_LOC_SUPL_HASH_ALGO_SHA1 (0) – SHA-1 hash algorithm for SUPL version 2.0 or later • eQMI_LOC_SUPL_HASH_ALGO_SHA256 (1) – SHA-256 hash algorithm for SUPL version 2.0 or later
Type	0x16			1	SUPL TLS Version
Length	4			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	enum	suplTlsVersion	4	SUPL TLS version. This configuration is only applicable to SUPL 2.0 or later, as SUPL 1.0 always uses TLS version 1.0. Valid values: • eQMI_LOC_SUPL_TLS_VERSION_1_0 (0) – SUPL TLS version 1.0 • eQMI_LOC_SUPL_TLS_VERSION_1_1 (1) – SUPL TLS version 1.1
Type	0x17			1	Emergency Protocol
Length	4			2	
Value	→	enum	emergencyProtocol	4	Protocol to be used during emergency. Valid values: • eQMI_LOC_EMERGENCY_PROTOCOL_WCDMA_CP (0) – Use Control Plane Protocol during an emergency while on WCDMA • eQMI_LOC_EMERGENCY_PROTOCOL_WCDMA_UP (1) – Use SUPL 2.0 emergency services during an emergency while on WCDMA
Type	0x18			1	Wi-Fi Scan Injection Timeout Period
Length	1			2	
Value	→	uint8	wifiScanInjectTimeout	1	Timeout duration that the service waits for a scan results injection from the control point after the event notification is sent. Values: 0 to 10 seconds

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_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, eQMI_LOC_UNSUPPORTED is returned and no action is taken. It is safe for multiple clients to use this command.

3.58 QMI_LOC_SET_SENSOR_CONTROL_CONFIG

Sets the sensor control configuration.

LOC message ID

0x0056

Version introduced

Major - 2, Minor - 2

3.58.1 Request - QMI_LOC_SET_SENSOR_CONTROL_CONFIG_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

None

Optional TLVs

Name	Version introduced	Version last modified
Sensors Usage	2.2	2.2
Sensors Provider	2.25	2.25

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

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	enum	sensorsUsage	4	Controls how sensors are used to aid heading and positioning performance. Valid values: <ul style="list-style-type: none"> • eQMI_LOC_SENSOR_CONFIG_SENSOR_USE_ENABLE (0) – Sensors data should be requested whenever a position request is received. If sensor data are injected, the positioning engine attempts to improve the heading and positioning performance using sensors. This is the default. • eQMI_LOC_SENSOR_CONFIG_SENSOR_USE_DISABLE (1) – Inertial sensors are not to be used to aid heading and position improvement.
Type	0x11			1	Sensors Provider
Length	4			2	
Value	→	enum	sensorProvider	4	Controls which sensors data provider is to be used. Valid values: <ul style="list-style-type: none"> • eQMI_LOC_SENSOR_CONFIG_USE_PROVIDER_SSC (0) – Sensors data provider is Snapdragon Sensor Core (SSC); this is the default • eQMI_LOC_SENSOR_CONFIG_USE_PROVIDER_NATIVE (1) – Sensors data provider is on the host processor

3.58.2 Indication - QMI_LOC_SET_SENSOR_CONTROL_CONFIG_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version introduced	Version last modified
Set Sensor Control Config Status	2.2	2.28

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Set Sensor Control Config Status
Length	4			2	
Value	→	enum	status	4	<p>Status of the Set Sensor Control Configuration request.</p> <p>Valid values:</p> <ul style="list-style-type: none"> • eQMI_LOC_SUCCESS (0) – Request was completed successfully • eQMI_LOC_GENERAL_FAILURE (1) – Request failed because of a general failure • eQMI_LOC_UNSUPPORTED (2) – Request failed because it is not supported • eQMI_LOC_INVALID_PARAMETER (3) – Request failed because it contained invalid parameters • eQMI_LOC_ENGINE_BUSY (4) – Request failed because the engine is busy • eQMI_LOC_PHONE_OFFLINE (5) – Request failed because the phone is offline • eQMI_LOC_TIMEOUT (6) – Request failed because it timed out • eQMI_LOC_CONFIG_NOT_SUPPORTED (7) – Request failed because an undefined configuration was requested • eQMI_LOC_INSUFFICIENT_MEMORY (8) – Request failed because the engine could not allocate sufficient memory for the request • eQMI_LOC_MAX_GEOFENCE_PROGRAMMED (9) – Request failed because the maximum number of Geofences are already programmed • eQMI_LOC_XTRA_VERSION_CHECK_FAILURE (10) – Location service failed because of an XTRA version-based file format check failure

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_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.

3.59 QMI_LOC_GET_SENSOR_CONTROL_CONFIG

Retrieves the current sensor control configuration.

LOC message ID

0x0057

Version introduced

Major - 2, Minor - 2

3.59.1 Request - QMI_LOC_GET_SENSOR_CONTROL_CONFIG_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

None

Optional TLVs

None

3.59.2 Indication - QMI_LOC_GET_SENSOR_CONTROL_CONFIG_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version introduced	Version last modified
Get Sensor Control Config Status	2.2	2.28

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Get Sensor Control Config Status
Length	4			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	enum	status	4	<p>Status of the Get Sensors Control Configuration request.</p> <p>Valid values:</p> <ul style="list-style-type: none"> • eQMI_LOC_SUCCESS (0) – Request was completed successfully • eQMI_LOC_GENERAL_FAILURE (1) – Request failed because of a general failure • eQMI_LOC_UNSUPPORTED (2) – Request failed because it is not supported • eQMI_LOC_INVALID_PARAMETER (3) – Request failed because it contained invalid parameters • eQMI_LOC_ENGINE_BUSY (4) – Request failed because the engine is busy • eQMI_LOC_PHONE_OFFLINE (5) – Request failed because the phone is offline • eQMI_LOC_TIMEOUT (6) – Request failed because it timed out • eQMI_LOC_CONFIG_NOT_SUPPORTED (7) – Request failed because an undefined configuration was requested • eQMI_LOC_INSUFFICIENT_MEMORY (8) – Request failed because the engine could not allocate sufficient memory for the request • eQMI_LOC_MAX_GEOFENCE_PROGRAMMED (9) – Request failed because the maximum number of Geofences are already programmed • eQMI_LOC_XTRA_VERSION_CHECK_FAILURE (10) – Location service failed because of an XTRA version-based file format check failure

Optional TLVs

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

Name	Version introduced	Version last modified
Sensors Usage	2.2	2.2
Sensors Provider	2.25	2.25

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Sensors Usage
Length	4			2	
Value	→	enum	sensorsUsage	4	Controls how sensors are used to aid the heading and positioning performance. Valid values: <ul style="list-style-type: none"> • eQMI_LOC_SENSOR_CONFIG_SENSOR_USE_ENABLE (0) – Sensors data should be requested whenever a position request is received. If sensor data are injected, the positioning engine attempts to improve the heading and positioning performance using sensors. This is the default. • eQMI_LOC_SENSOR_CONFIG_SENSOR_USE_DISABLE (1) – Inertial sensors are not to be used to aid heading and position improvement.
Type	0x11			1	Sensors Provider
Length	4			2	
Value	→	enum	sensorProvider	4	Controls which sensors data provider to be used. Valid values: <ul style="list-style-type: none"> • eQMI_LOC_SENSOR_CONFIG_USE_PROVIDER_SSC (0) – Sensors data provider is Snapdragon Sensor Core (SSC); this is the default • eQMI_LOC_SENSOR_CONFIG_USE_PROVIDER_NATIVE (1) – Sensors data provider is on the host processor

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_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. It is safe for multiple clients to use this command.

3.60 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

3.60.1 Request - QMI_LOC_SET_SENSOR_PROPERTIES_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

None

Optional TLVs

Name	Version introduced	Version last modified
Gyro Bias Random Walk Variance	2.2	2.7
Velocity Random Walk Spectral Density	2.7	2.7
Acceleration Random Walk Spectral Density	2.7	2.7
Angle Random Walk Spectral Density	2.7	2.7
Rate Random Walk Spectral Density	2.7	2.7
Vehicle Data Use Control	2.24	2.24
Vehicle Velocity Random Walk Spectral Density	2.24	2.24
Vehicle Acceleration Random Walk Spectral Density	2.24	2.24
Vehicle Angle Random Walk Spectral Density	2.24	2.24
Vehicle Angular Rate Random Walk Spectral Density	2.24	2.24
Vehicle Odometry Scale Factor Random Walk Spectral Density	2.24	2.24
Vehicle Odometry Variance	2.24	2.24

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Gyro Bias Random Walk Variance
Length	4			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	float	gyroBiasVarianceRandomWalk	4	Specifies the gyro bias random walk variance parameter as a positive floating-point value. This value has internal default value $1.0e-5$ radians ² /seconds ⁴ . The gyro bias variance random walk parameter is derived from either the sensors data sheet or a sensors conformance test. • Units: Radians ² /seconds ⁴
Type	0x11			1	Velocity Random Walk Spectral Density
Length	4			2	
Value	→	float	velocityRandomWalkSpectralDensity	4	Specifies the velocity random walk spectral density parameter as a positive floating-point value. This value does not have any internal defaults. The velocity random walk spectral density parameter is derived from either the sensors data sheet or a sensors conformance test. • Units: Meters/seconds ² /Hertz ^{0.5}
Type	0x12			1	Acceleration Random Walk Spectral Density
Length	4			2	
Value	→	float	accelerationRandomWalkSpectralDensity	4	Specifies the acceleration random walk spectral density parameter as a positive floating-point value. This value does not have any internal defaults. The acceleration random walk spectral density parameter is derived from either the sensors data sheet or a sensors conformance test. • Units: Meters/seconds ³ /Hertz ^{0.5}
Type	0x13			1	Angle Random Walk Spectral Density
Length	4			2	
Value	→	float	angleRandomWalkSpectralDensity	4	Specifies the angle random walk spectral density parameter as a positive floating-point value. This value does not have any internal defaults. The angle random walk spectral density parameter is derived from either the sensors data sheet or a sensors conformance test. • Units: Radians/seconds/Hertz ^{0.5}
Type	0x14			1	Rate Random Walk Spectral Density
Length	4			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	float	rateRandomWalkSpectral Density	4	Specifies the rate random walk spectral density parameter as a positive floating-point value. This value does not have any internal defaults. The rate random walk spectral density parameter is derived from either the sensors data sheet or a sensors conformance test. • Units: Radians/seconds ² /Hertz ^{0.5}
Type	0x15			1	Vehicle Data Use Control
Length	8			2	
Value	→	mask	vehicleDataUse	8	Identifies which portions of the vehicle data to use in location estimation (information provided by the message QMI_LOC_INJECT_VEHICLE_SENSOR_DATA). Valid bitmasks: • QMI_LOC_VEHICLE_DATA_ENABLE_USE_MASK_ACCEL_X_AXIS (0x0000000000000001) – Enable use of X-axis vehicle acceleration sensor data • QMI_LOC_VEHICLE_DATA_ENABLE_USE_MASK_ACCEL_Y_AXIS (0x0000000000000002) – Enable use of Y-axis vehicle acceleration sensor data • QMI_LOC_VEHICLE_DATA_ENABLE_USE_MASK_ACCEL_Z_AXIS (0x0000000000000004) – Enable use of Z-axis vehicle acceleration sensor data • QMI_LOC_VEHICLE_DATA_ENABLE_USE_MASK_GYRO_X_AXIS (0x0000000000000010) – Enable use of X-axis vehicle gyroscope data • QMI_LOC_VEHICLE_DATA_ENABLE_USE_MASK_GYRO_Y_AXIS (0x0000000000000020) – Enable use of Y-axis vehicle gyroscope data • QMI_LOC_VEHICLE_DATA_ENABLE_USE_MASK_GYRO_Z_AXIS (0x0000000000000040) – Enable use of Z-axis vehicle gyroscope data

Field	Field value	Field type	Parameter	Size (byte)	Description
			vehicleDataUse (cont.)		<ul style="list-style-type: none"> • QMI_LOC_VEHICLE_DATA_ENABLE_USE_MASK_ODOMETRY (0x0000000000000100) – Enable use of odometry data Note: All other bits are reserved for future use and are to be set to 0.
Type	0x16			1	Vehicle Velocity Random Walk Spectral Density
Length	4			2	
Value	→	float	vehicleVelocityRandomWalkSpectralDensity	4	Vehicle velocity random walk spectral density. <ul style="list-style-type: none"> • Type: 32-bit float • Units: Meters/seconds²/Hz^{0.5} • Valid values: Positive values • Default: None
Type	0x17			1	Vehicle Acceleration Random Walk Spectral Density
Length	4			2	
Value	→	float	vehicleAccelRandomWalkSpectralDensity	4	Vehicle accelerometer random walk spectral density. <ul style="list-style-type: none"> • Type: 32-bit float • Units: Meters/seconds³/Hz^{0.5} • Valid values: Positive values • Default: None
Type	0x18			1	Vehicle Angle Random Walk Spectral Density
Length	4			2	
Value	→	float	vehicleAngleRandomWalkSpectralDensity	4	Vehicle angle random walk spectral density. <ul style="list-style-type: none"> • Type: 32-bit float • Units: Radians/seconds/Hz^{0.5} • Valid values: Positive values • Default: None
Type	0x19			1	Vehicle Angular Rate Random Walk Spectral Density
Length	4			2	
Value	→	float	vehicleAngularRateRandomWalkSpectralDensity	4	Vehicle angular rate random walk spectral density. <ul style="list-style-type: none"> • Type: 32-bit float • Units: Radians/seconds²/Hz^{0.5} • Valid values: Positive values • Default: None
Type	0x1A			1	Vehicle Odometry Scale Factor Random Walk Spectral Density
Length	4			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	float	vehicleOdometryScaleFactorRandomWalkSpectralDensity	4	Vehicle odometry scale factor random walk spectral density. <ul style="list-style-type: none"> • Type: 32-bit float • Units: (1/seconds)/Hz^{0.5} • Range: Approximately 0.0001 to 0.001 • Default: 0.001 (actual calibration recommended)
Type	0x1B			1	Vehicle Odometry Variance
Length	4			2	
Value	→	float	vehicleOdometryVariance	4	Vehicle odometry variance of each odometry sample (coarseness of measurement). <ul style="list-style-type: none"> • Type: 32-bit float • Units: Meters² • Valid values: Positive values • Default: None

3.60.2 Indication - QMI_LOC_SET_SENSOR_PROPERTIES_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version introduced	Version last modified
Set Sensor Properties Status	2.2	2.28

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Set Sensor Properties Status
Length	4			2	
Value	→	enum	status	4	Status of the Set Sensor Properties request. Valid values: <ul style="list-style-type: none"> • eQMI_LOC_SUCCESS (0) – Request was completed successfully • eQMI_LOC_GENERAL_FAILURE (1) – Request failed because of a general failure • eQMI_LOC_UNSUPPORTED (2) – Request failed because it is not supported

Field	Field value	Field type	Parameter	Size (byte)	Description
			status (cont.)		<ul style="list-style-type: none"> • eQMI_LOC_INVALID_PARAMETER (3) – Request failed because it contained invalid parameters • eQMI_LOC_ENGINE_BUSY (4) – Request failed because the engine is busy • eQMI_LOC_PHONE_OFFLINE (5) – Request failed because the phone is offline • eQMI_LOC_TIMEOUT (6) – Request failed because it timed out • eQMI_LOC_CONFIG_NOT_SUPPORTED (7) – Request failed because an undefined configuration was requested • eQMI_LOC_INSUFFICIENT_MEMORY (8) – Request failed because the engine could not allocate sufficient memory for the request • eQMI_LOC_MAX_GEOFENCE_PROGRAMMED (9) – Request failed because the maximum number of Geofences are already programmed • eQMI_LOC_XTRA_VERSION_CHECK_FAILURE (10) – Location service failed because of an XTRA version-based file format check failure

Optional TLVs

Name	Version introduced	Version last modified
Failed Set Sensor Properties	2.7	2.24

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Failed Set Sensor Properties
Length	4			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	mask32	failedSensorPropertiesMask	4	<p>This field is sent only if the status is not SUCCESS. Identifies the parameters that were not set successfully.</p> <p>Valid bitmasks:</p> <ul style="list-style-type: none"> • QMI_LOC_SENSOR_PROPERTIES_MASK_GYRO_BIAS_VARIANCE_RANDOM_WALK (0x00000001) – Denotes the gyro bias variance random walk parameter • QMI_LOC_SENSOR_PROPERTIES_MASK_VELOCITY_RANDOM_WALK_SPECTRAL_DENSITY (0x00000002) – Denotes the velocity random walk spectral density parameter • QMI_LOC_SENSOR_PROPERTIES_MASK_ACCELERATION_RANDOM_WALK_SPECTRAL_DENSITY (0x00000004) – Denotes the acceleration random walk spectral density parameter • QMI_LOC_SENSOR_PROPERTIES_MASK_ANGLE_RANDOM_WALK_SPECTRAL_DENSITY (0x00000008) – Denotes the angle random walk spectral density parameter • QMI_LOC_SENSOR_PROPERTIES_MASK_RATE_RANDOM_WALK_SPECTRAL_DENSITY (0x00000010) – Denotes the rate random walk spectral density parameter • QMI_LOC_SENSOR_PROPERTIES_MASK_VEHICLE_DATA_USE_CONTROL (0x00000020) – Denotes the vehicle data use control parameter • QMI_LOC_SENSOR_PROPERTIES_MASK_VEHICLE_VELOCITY_RWSD (0x00000040) – Denotes the vehicle velocity random walk spectral density • QMI_LOC_SENSOR_PROPERTIES_MASK_VEHICLE_ACCEL_RWSD (0x00000080) – Denotes the vehicle accelerometer random walk spectral density • QMI_LOC_SENSOR_PROPERTIES_MASK_VEHICLE_ANGLE_RWSD (0x00000100) – Denotes the vehicle angle random walk spectral density

Field	Field value	Field type	Parameter	Size (byte)	Description
			failedSensorProperties Mask (cont.)		<ul style="list-style-type: none"> • QMI_LOC_SENSOR_PROPERTIES_MASK_VEHICLE_ANGULAR_RATE_RWSD (0x00000200) – Denotes the vehicle angular rate random walk spectral density • QMI_LOC_SENSOR_PROPERTIES_MASK_VEHICLE_ODOMETRY_SCALE_RWSD (0x00000400) – Denotes the vehicle odometry scale random walk spectral density • QMI_LOC_SENSOR_PROPERTIES_MASK_VEHICLE_ODOMETRY_VARIANCE (0x00000800) – Denotes the vehicle odometry variance

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_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, preferably 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.

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.

3.61 QMI_LOC_GET_SENSOR_PROPERTIES

Retrieves the current sensor properties.

LOC message ID

0x0059

Version introduced

Major - 2, Minor - 2

3.61.1 Request - QMI_LOC_GET_SENSOR_PROPERTIES_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

Name	Version introduced	Version last modified
Sensor Properties Config Parameters	2.7	2.24

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Sensor Properties Config Parameters
Length	4			2	
Value	→	mask32	getSensorPropertiesMask	4	Mask denoting the sensor properties parameters to be retrieved. Valid bitmasks: <ul style="list-style-type: none"> • QMI_LOC_SENSOR_PROPERTIES_MASK_GYRO_BIAS_VARIANCE_RANDOM_WALK (0x00000001) – Denotes the gyro bias variance random walk parameter • QMI_LOC_SENSOR_PROPERTIES_MASK_VELOCITY_RANDOM_WALK_SPECTRAL_DENSITY (0x00000002) – Denotes the velocity random walk spectral density parameter • QMI_LOC_SENSOR_PROPERTIES_MASK_ACCELERATION_RANDOM_WALK_SPECTRAL_DENSITY (0x00000004) – Denotes the acceleration random walk spectral density parameter

Field	Field value	Field type	Parameter	Size (byte)	Description
			getSensorPropertiesMask (cont.)		<ul style="list-style-type: none"> • QMI_LOC_SENSOR_PROPERTIES_MASK_ANGLE_RANDOM_WALK_SPECTRAL_DENSITY (0x00000008) – Denotes the angle random walk spectral density parameter • QMI_LOC_SENSOR_PROPERTIES_MASK_RATE_RANDOM_WALK_SPECTRAL_DENSITY (0x00000010) – Denotes the rate random walk spectral density parameter • QMI_LOC_SENSOR_PROPERTIES_MASK_VEHICLE_DATA_USE_CONTROL (0x00000020) – Denotes the vehicle data use control parameter • QMI_LOC_SENSOR_PROPERTIES_MASK_VEHICLE_VELOCITY_RWSD (0x00000040) – Denotes the vehicle velocity random walk spectral density • QMI_LOC_SENSOR_PROPERTIES_MASK_VEHICLE_ACCEL_RWSD (0x00000080) – Denotes the vehicle accelerometer random walk spectral density • QMI_LOC_SENSOR_PROPERTIES_MASK_VEHICLE_ANGLE_RWSD (0x00000100) – Denotes the vehicle angle random walk spectral density • QMI_LOC_SENSOR_PROPERTIES_MASK_VEHICLE_ANGULAR_RATE_RWSD (0x00000200) – Denotes the vehicle angular rate random walk spectral density • QMI_LOC_SENSOR_PROPERTIES_MASK_VEHICLE_ODOMETRY_SCALE_RWSD (0x00000400) – Denotes the vehicle odometry scale random walk spectral density • QMI_LOC_SENSOR_PROPERTIES_MASK_VEHICLE_ODOMETRY_VARIANCE (0x00000800) – Denotes the vehicle odometry variance

Optional TLVs

None

3.61.2 Indication - QMI_LOC_GET_SENSOR_PROPERTIES_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version introduced	Version last modified
Get Sensor Properties Status	2.2	2.28

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Get Sensor Properties Status
Length	4			2	
Value	→	enum	status	4	<p>Status of the Get Sensors Properties request.</p> <p>Valid values:</p> <ul style="list-style-type: none"> • eQMI_LOC_SUCCESS (0) – Request was completed successfully • eQMI_LOC_GENERAL_FAILURE (1) – Request failed because of a general failure • eQMI_LOC_UNSUPPORTED (2) – Request failed because it is not supported • eQMI_LOC_INVALID_PARAMETER (3) – Request failed because it contained invalid parameters • eQMI_LOC_ENGINE_BUSY (4) – Request failed because the engine is busy • eQMI_LOC_PHONE_OFFLINE (5) – Request failed because the phone is offline • eQMI_LOC_TIMEOUT (6) – Request failed because it timed out • eQMI_LOC_CONFIG_NOT_SUPPORTED (7) – Request failed because an undefined configuration was requested • eQMI_LOC_INSUFFICIENT_MEMORY (8) – Request failed because the engine could not allocate sufficient memory for the request

Field	Field value	Field type	Parameter	Size (byte)	Description
			status (cont.)		<ul style="list-style-type: none"> • eQMI_LOC_MAX_GEOFENCE_PROGRAMMED (9) – Request failed because the maximum number of Geofences are already programmed • eQMI_LOC_XTRA_VERSION_CHECK_FAILURE (10) – Location service failed because of an XTRA version-based file format check failure

Optional TLVs

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

Name	Version introduced	Version last modified
Gyro Bias Random Walk Variance	2.2	2.7
Velocity Random Walk Spectral Density	2.7	2.7
Acceleration Random Walk Spectral Density	2.7	2.7
Angle Random Walk Spectral Density	2.7	2.7
Rate Random Walk Spectral Density	2.7	2.7
Vehicle Data Use Control	2.24	2.24
Vehicle Velocity Random Walk Spectral Density	2.24	2.24
Vehicle Acceleration Random Walk Spectral Density	2.24	2.24
Vehicle Angle Random Walk Spectral Density	2.24	2.24
Vehicle Angular Rate Random Walk Spectral Density	2.24	2.24
Vehicle Odometry Scale Factor Random Walk Spectral Density	2.24	2.24
Vehicle Odometry Variance	2.24	2.24

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Gyro Bias Random Walk Variance
Length	4			2	
Value	→	float	gyroBiasVarianceRandomWalk	4	Specifies the gyro bias random walk variance parameter as a positive floating-point value. This value has internal default value 1.0e-5 radians ² /seconds ⁴ . The gyro bias variance random walk parameter is derived from either the sensors data sheet or a sensors conformance test. <ul style="list-style-type: none"> • Units: Radians²/seconds⁴
Type	0x11			1	Velocity Random Walk Spectral Density
Length	4			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	float	velocityRandomWalkSpectralDensity	4	Specifies the velocity random walk spectral density parameter as a positive floating-point value. This value does not have any internal defaults. The velocity random walk spectral density parameter is derived from either the sensors data sheet or a sensors conformance test. • Units: Meters/seconds ² /Hertz ^{0.5}
Type	0x12			1	Acceleration Random Walk Spectral Density
Length	4			2	
Value	→	float	accelerationRandomWalkSpectralDensity	4	Specifies the acceleration random walk spectral density parameter as a positive floating-point value. This value does not have any internal defaults. The acceleration random walk spectral density parameter is derived from either the sensors data sheet or a sensors conformance test. • Units: Meters/seconds ³ /Hertz ^{0.5}
Type	0x13			1	Angle Random Walk Spectral Density
Length	4			2	
Value	→	float	angleRandomWalkSpectralDensity	4	Specifies the angle random walk spectral density parameter as a positive floating-point value. This value does not have any internal defaults. The angle random walk spectral density parameter is derived from either the sensors data sheet or a sensors conformance test. • Units: Radians/seconds/Hertz ^{0.5}
Type	0x14			1	Rate Random Walk Spectral Density
Length	4			2	
Value	→	float	rateRandomWalkSpectralDensity	4	Specifies the rate random walk spectral density parameter as a positive floating-point value. This value does not have any internal defaults. The rate random walk spectral density parameter is derived from either the sensors data sheet or a sensors conformance test. • Units: Radians/seconds ² /Hertz ^{0.5}
Type	0x15			1	Vehicle Data Use Control
Length	8			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	mask	vehicleDataUse	8	<p>Identifies which portions of the vehicle data to use in location estimation (information provided by message QMI_LOC_INJECT_VEHICLE_SENSOR_DATA).</p> <p>Valid bitmasks:</p> <ul style="list-style-type: none"> • QMI_LOC_VEHICLE_DATA_ENABLE_USE_MASK_ACCEL_X_AXIS (0x0000000000000001) – Enable use of X-axis vehicle acceleration sensor data • QMI_LOC_VEHICLE_DATA_ENABLE_USE_MASK_ACCEL_Y_AXIS (0x0000000000000002) – Enable use of Y-axis vehicle acceleration sensor data • QMI_LOC_VEHICLE_DATA_ENABLE_USE_MASK_ACCEL_Z_AXIS (0x0000000000000004) – Enable use of Z-axis vehicle acceleration sensor data • QMI_LOC_VEHICLE_DATA_ENABLE_USE_MASK_GYRO_X_AXIS (0x0000000000000010) – Enable use of X-axis vehicle gyroscope data • QMI_LOC_VEHICLE_DATA_ENABLE_USE_MASK_GYRO_Y_AXIS (0x0000000000000020) – Enable use of Y-axis vehicle gyroscope data • QMI_LOC_VEHICLE_DATA_ENABLE_USE_MASK_GYRO_Z_AXIS (0x0000000000000040) – Enable use of Z-axis vehicle gyroscope data • QMI_LOC_VEHICLE_DATA_ENABLE_USE_MASK_ODOMETRY (0x0000000000000100) – Enable use of odometry data <p>Note: All other bits are reserved for future use and are to be set to 0.</p>
Type	0x16			1	Vehicle Velocity Random Walk Spectral Density
Length	4			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	float	vehicleVelocityRandomWalkSpectralDensity	4	Vehicle velocity random walk spectral density. • Type: 32-bit float • Units: Meters/seconds ² /Hz ^{0.5} • Valid values: Positive values • Default: None
Type	0x17			1	Vehicle Acceleration Random Walk Spectral Density
Length	4			2	
Value	→	float	vehicleAccelRandomWalkSpectralDensity	4	Vehicle accelerometer random walk spectral density. • Type: 32-bit float • Units: Meters/seconds ³ /Hz ^{0.5} • Valid values: Positive values • Default: None
Type	0x18			1	Vehicle Angle Random Walk Spectral Density
Length	4			2	
Value	→	float	vehicleAngleRandomWalkSpectralDensity	4	Vehicle angle random walk spectral density. • Type: 32-bit float • Units: Radians/seconds/Hz ^{0.5} • Valid values: Positive values • Default: None
Type	0x19			1	Vehicle Angular Rate Random Walk Spectral Density
Length	4			2	
Value	→	float	vehicleAngularRateRandomWalkSpectralDensity	4	Vehicle angular rate random walk spectral density. • Type: 32-bit float • Units: Radians/seconds ² /Hz ^{0.5} • Valid values: Positive values • Default: None
Type	0x1A			1	Vehicle Odometry Scale Factor Random Walk Spectral Density
Length	4			2	
Value	→	float	vehicleOdometryScaleFactorRandomWalkSpectralDensity	4	Vehicle odometry scale factor random walk spectral density. • Type: 32-bit float • Units: (1/seconds)/Hz ^{0.5} • Range: Approximately 0.0001 to 0.001 • Default: 0.001 (actual calibration recommended)
Type	0x1B			1	Vehicle Odometry Variance
Length	4			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	float	vehicleOdometryVariance	4	Vehicle odometry variance of each odometry sample (coarseness of measurement). <ul style="list-style-type: none"> • Type: 32-bit float • Units: Meters² • Valid values: Positive values • Default: 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

3.61.3 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. It is safe for multiple clients to use this command.

3.62 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

3.62.1 Request - QMI_LOC_SET_SENSOR_PERFORMANCE_CONTROL_- CONFIGURATION_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

None

Optional TLVs

Name	Version introduced	Version last modified
Sensor Performance Control Mode	2.2	2.2
Accelerometer Sampling Specification	2.2	2.2
Gyroscope Sampling Specification	2.2	2.2
Algorithm Configuration	2.7	2.7
High Data Rate Filter Accelerometer Sampling Specification	2.12	2.12
High Data Rate Filter Gyroscope Sampling Specification	2.12	2.12

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

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	enum	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: <ul style="list-style-type: none"> • eQMI_LOC_SENSOR_PERFORMANCE_CONTROL_MODE_AUTO (0) – Sensors usage is to be determined by the GNSS location engine. This mode can optimize power consumption and give a power-balanced positioning and heading enhancement using inertial sensors • eQMI_LOC_SENSOR_PERFORMANCE_CONTROL_MODE_FORCED (1) – Sensors usage is to be forced ON. This mode can be requested by the control point when power consumption is not a restriction to the use of inertial sensors.
Type	0x11			1	Accelerometer Sampling Specification Sets the nominal rate at which the GNSS location engine is to request acceleration data to be used by the low data rate filter. 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. Default: 10 Hz sampling rate and 2 Hz batching rate.
Length	4			2	
Value	→	uint16	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: $\text{samplingFrequency} = \text{samplesPerBatch} * \text{batchesPerSecond}$ samplesPerBatch must be a nonzero positive value.

Field	Field value	Field type	Parameter	Size (byte)	Description
		uint16	batchesPerSecond	2	Number of sensor-data batches the GNSS location engine is to receive per second. The rate is specified in an integral number of batches per second (Hz). batchesPerSecond must be a nonzero positive value.
Type	0x12			1	Gyroscope Sampling Specification Sets the nominal rate at which the GNSS location engine is to request gyro data to be used by the high data rate filter. 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. Default: 10 Hz sampling rate and 2 Hz batching rate.
Length	4			2	
Value	→	uint16	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 nonzero positive value.
		uint16	batchesPerSecond	2	Number of sensor-data batches the GNSS location engine is to receive per second. The rate is specified in an integral number of batches per second (Hz). batchesPerSecond must be a nonzero positive value.
Type	0x13			1	Algorithm Configuration
Length	4			2	
Value	→	mask32	algorithmConfig	4	Sets which sensor algorithms are to be used when processing sensor data. Valid bitmasks: • 0x00000001 – DISABLE_INS_POSITIONING_FILTER

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x14			1	High Data Rate Filter Accelerometer Sampling Specification Sets the nominal rate at which the GNSS location engine is to request acceleration data to be used by the high data rate filter. 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. Default: 100 Hz sampling rate and 4 Hz batching rate.
Length	4			2	
Value	→	uint16	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: $\text{samplingFrequency} = \text{samplesPerBatch} * \text{batchesPerSecond}$ samplesPerBatch must be a nonzero positive value.
		uint16	batchesPerSecond	2	Number of sensor-data batches the GNSS location engine is to receive per second. The rate is specified in an integral number of batches per second (Hz). batchesPerSecond must be a nonzero positive value.
Type	0x15			1	High Data Rate Filter Gyroscope Sampling Specification Sets the nominal rate at which the GNSS location engine is to request gyro data to be used by the high data rate filter. 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. Default: 100 Hz sampling rate and 4 Hz batching rate.
Length	4			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	uint16	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 nonzero positive value.
		uint16	batchesPerSecond	2	Number of sensor-data batches the GNSS location engine is to receive per second. The rate is specified in an integral number of batches per second (Hz). batchesPerSecond must be a nonzero positive value.

3.62.2 Indication - QMI_LOC_SET_SENSOR_PERFORMANCE - CONTROL_CONFIGURATION_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version introduced	Version last modified
Set Sensor Perf Control Config Status	2.2	2.28

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Set Sensor Perf Control Config Status
Length	4			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	enum	status	4	<p>Status of the Set Sensor Performance Control Configuration request.</p> <p>Valid values:</p> <ul style="list-style-type: none"> • eQMI_LOC_SUCCESS (0) – Request was completed successfully • eQMI_LOC_GENERAL_FAILURE (1) – Request failed because of a general failure • eQMI_LOC_UNSUPPORTED (2) – Request failed because it is not supported • eQMI_LOC_INVALID_PARAMETER (3) – Request failed because it contained invalid parameters • eQMI_LOC_ENGINE_BUSY (4) – Request failed because the engine is busy • eQMI_LOC_PHONE_OFFLINE (5) – Request failed because the phone is offline • eQMI_LOC_TIMEOUT (6) – Request failed because it timed out • eQMI_LOC_CONFIG_NOT_SUPPORTED (7) – Request failed because an undefined configuration was requested • eQMI_LOC_INSUFFICIENT_MEMORY (8) – Request failed because the engine could not allocate sufficient memory for the request • eQMI_LOC_MAX_GEOFENCE_PROGRAMMED (9) – Request failed because the maximum number of Geofences are already programmed • eQMI_LOC_XTRA_VERSION_CHECK_FAILURE (10) – Location service failed because of an XTRA version-based file format check failure

Optional TLVs

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

Name	Version introduced	Version last modified
Failed Configuration	2.2	2.12

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Failed Configuration
Length	4			2	
Value	→	mask32	failedConfiguration	4	Identifies parameters that were not configured successfully. This field is sent only if the status is not a success. Valid bitmasks: <ul style="list-style-type: none"> • 0x00000001 – PERFORMANCE_MODE • 0x00000002 – ACCEL_SAMPLING_SPEC • 0x00000004 – GYRO_SAMPLING_SPEC • 0x00000008 – ALGORITHM_CONFIG • 0x00000010 – ACCEL_SAMPLING_SPEC_HIGH • 0x00000020 – GYRO_SAMPLING_SPEC_HIGH

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_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. 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.

3.63 QMI_LOC_GET_SENSOR_PERFORMANCE_CONTROL_- CONFIGURATION

Retrieves the current sensor performance control configuration.

LOC message ID

0x005B

Version introduced

Major - 2, Minor - 2

3.63.1 Request - QMI_LOC_GET_SENSOR_PERFORMANCE_CONTROL_- CONFIGURATION_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

None

Optional TLVs

None

3.63.2 Indication - QMI_LOC_GET_SENSOR_PERFORMANCE_- CONTROL_CONFIGURATION_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version introduced	Version last modified
Get Sensor Perf Control Config Status	2.2	2.28

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Get Sensor Perf Control Config Status
Length	4			2	
Value	→	enum	status	4	<p>Status of the Get Sensor Performance Control Configuration request.</p> <p>Valid values:</p> <ul style="list-style-type: none"> • eQMI_LOC_SUCCESS (0) – Request was completed successfully • eQMI_LOC_GENERAL_FAILURE (1) – Request failed because of a general failure • eQMI_LOC_UNSUPPORTED (2) – Request failed because it is not supported • eQMI_LOC_INVALID_PARAMETER (3) – Request failed because it contained invalid parameters • eQMI_LOC_ENGINE_BUSY (4) – Request failed because the engine is busy • eQMI_LOC_PHONE_OFFLINE (5) – Request failed because the phone is offline • eQMI_LOC_TIMEOUT (6) – Request failed because it timed out • eQMI_LOC_CONFIG_NOT_SUPPORTED (7) – Request failed because an undefined configuration was requested • eQMI_LOC_INSUFFICIENT_MEMORY (8) – Request failed because the engine could not allocate sufficient memory for the request • eQMI_LOC_MAX_GEOFENCE_PROGRAMMED (9) – Request failed because the maximum number of Geofences are already programmed • eQMI_LOC_XTRA_VERSION_CHECK_FAILURE (10) – Location service failed because of an XTRA version-based file format check failure

Optional TLVs

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

Name	Version introduced	Version last modified
Performance Control Mode	2.2	2.2
Accelerometer Sampling Specification	2.2	2.2
Gyroscope Sampling Specification	2.2	2.2

Name	Version introduced	Version last modified
Algorithm Configuration	2.7	2.7
High Data Rate Filter Accelerometer Sampling Specification	2.12	2.12
High Data Rate Filter Gyroscope Sampling Specification	2.12	2.12

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Performance Control Mode
Length	4			2	
Value	→	enum	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: • eQMI_LOC_SENSOR_PERFORMANCE_CONTROL_MODE_AUTO (0) – Sensors usage is to be determined by the GNSS location engine. This mode can optimize power consumption and give a power-balanced positioning and heading enhancement using inertial sensors • eQMI_LOC_SENSOR_PERFORMANCE_CONTROL_MODE_FORCED (1) – Sensors usage is to be forced ON. This mode can be requested by the control point when power consumption is not a restriction to the use of inertial sensors.
Type	0x11			1	Accelerometer Sampling Specification Sets the nominal rate at which the GNSS location engine is to request acceleration data to be used by the high data rate filter. 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. Default: 10 Hz sampling rate and 2 Hz batching rate.
Length	4			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	uint16	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 nonzero positive value.
		uint16	batchesPerSecond	2	Number of sensor-data batches the GNSS location engine is to receive per second. The rate is specified in an integral number of batches per second (Hz). batchesPerSecond must be a nonzero positive value.
Type	0x12			1	Gyroscope Sampling Specification Sets the nominal rate at which the GNSS location engine is to request gyro data to be used by the high data rate filter. 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. Default: 10 Hz sampling rate and 2 Hz batching rate.
Length	4			2	
Value	→	uint16	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 nonzero positive value.
		uint16	batchesPerSecond	2	Number of sensor-data batches the GNSS location engine is to receive per second. The rate is specified in an integral number of batches per second (Hz). batchesPerSecond must be a nonzero positive value.

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x13			1	Algorithm Configuration
Length	4			2	
Value	→	mask32	algorithmConfig	4	<p>Informs which sensor algorithms are currently set.</p> <p>Valid bitmasks:</p> <ul style="list-style-type: none"> • 0x00000001 – DISABLE_INS_POSITIONING_FILTER
Type	0x14			1	<p>High Data Rate Filter Accelerometer Sampling Specification</p> <p>Sets the nominal rate at which the GNSS location engine is to request acceleration data to be used by the high data rate filter. 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.</p> <p>Default: 100 Hz sampling rate and 4 Hz batching rate.</p>
Length	4			2	
Value	→	uint16	samplesPerBatch	2	<p>Specifies the number of samples per batch the GNSS location engine is to receive. The sensor sampling frequency can be computed as follows:</p> $\text{samplingFrequency} = \text{samplesPerBatch} * \text{batchesPerSecond}$ <p>samplesPerBatch must be a nonzero positive value.</p>
		uint16	batchesPerSecond	2	<p>Number of sensor-data batches the GNSS location engine is to receive per second. The rate is specified in an integral number of batches per second (Hz).</p> <p>batchesPerSecond must be a nonzero positive value.</p>

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x15			1	High Data Rate Filter Gyroscope Sampling Specification Sets the nominal rate at which the GNSS location engine is to request gyro data to be used by the high data rate filter. 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. Default: 100 Hz sampling rate and 4 Hz batching rate.
Length	4			2	
Value	→	uint16	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 nonzero positive value.
		uint16	batchesPerSecond	2	Number of sensor-data batches the GNSS location engine is to receive per second. The rate is specified in an integral number of batches per second (Hz). batchesPerSecond must be a nonzero 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

3.63.3 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. It is safe for multiple clients to use this command.

QUALCOMM®
2016-05-16 00:17:18 PDT
deon_zhang@askey.com.tw

3.64 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

3.64.1 Request - QMI_LOC_INJECT_SUPL_CERTIFICATE_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

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

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	SUPL Certificate ID
Length	1			2	
Value	→	uint8	suplCertId	1	Certificate ID of the SUPL certificate. • Units: Bytes • Range: 0 to 9
Type	0x02			1	SUPL Certificate Data
Length	Var			2	
Value	→	uint16	suplCertData_len	2	Number of sets of the following elements: • suplCertData
		uint8	suplCertData	Var	SUPL certificate contents. • Type: Array of bytes • Maximum certificate size: 2000 bytes

Optional TLVs

None

3.64.2 Indication - QMI_LOC_INJECT_SUPL_CERTIFICATE_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version introduced	Version last modified
SUPL Certificate Injection Status	2.3	2.28

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	SUPL Certificate Injection Status
Length	4			2	
Value	→	enum	status	4	<p>Status of the Inject SUPL Certificate request.</p> <p>Valid values:</p> <ul style="list-style-type: none"> • eQMI_LOC_SUCCESS (0) – Request was completed successfully • eQMI_LOC_GENERAL_FAILURE (1) – Request failed because of a general failure • eQMI_LOC_UNSUPPORTED (2) – Request failed because it is not supported • eQMI_LOC_INVALID_PARAMETER (3) – Request failed because it contained invalid parameters • eQMI_LOC_ENGINE_BUSY (4) – Request failed because the engine is busy • eQMI_LOC_PHONE_OFFLINE (5) – Request failed because the phone is offline • eQMI_LOC_TIMEOUT (6) – Request failed because it timed out • eQMI_LOC_CONFIG_NOT_SUPPORTED (7) – Request failed because an undefined configuration was requested • eQMI_LOC_INSUFFICIENT_MEMORY (8) – Request failed because the engine could not allocate sufficient memory for the request

Field	Field value	Field type	Parameter	Size (byte)	Description
			status (cont.)		<ul style="list-style-type: none"> • eQMI_LOC_MAX_GEOFENCE_PROGRAMMED (9) – Request failed because the maximum number of Geofences are already programmed • eQMI_LOC_XTRA_VERSION_CHECK_FAILURE (10) – Location service failed because of an XTRA version-based file format check failure

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

3.64.3 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. The SUPL certificates are used in the AGNSS sessions for all clients, therefore it is recommended that only one client control the injection of SUPL certificates.

3.65 QMI_LOC_DELETE_SUPL_CERTIFICATE

Deletes a SUPL certificate.

LOC message ID

0x005D

Version introduced

Major - 2, Minor - 3

3.65.1 Request - QMI_LOC_DELETE_SUPL_CERTIFICATE_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

None

Optional TLVs

Name	Version introduced	Version last modified
SUPL Certificate ID	2.3	2.3

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	SUPL Certificate ID
Length	1			2	
Value	→	uint8	suplCertId	1	Certificate ID of the SUPL certificate to be deleted. • Units: Bytes • Range: 0 to 9 If suplCertId is not specified, all SUPL certificates are deleted.

3.65.2 Indication - QMI_LOC_DELETE_SUPL_CERTIFICATE_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version introduced	Version last modified
SUPL Certificate Deletion Status	2.3	2.28

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	SUPL Certificate Deletion Status
Length	4			2	
Value	→	enum	status	4	<p>Status of the Delete SUPL Certificate request.</p> <p>Valid values:</p> <ul style="list-style-type: none"> • eQMI_LOC_SUCCESS (0) – Request was completed successfully • eQMI_LOC_GENERAL_FAILURE (1) – Request failed because of a general failure • eQMI_LOC_UNSUPPORTED (2) – Request failed because it is not supported • eQMI_LOC_INVALID_PARAMETER (3) – Request failed because it contained invalid parameters • eQMI_LOC_ENGINE_BUSY (4) – Request failed because the engine is busy • eQMI_LOC_PHONE_OFFLINE (5) – Request failed because the phone is offline • eQMI_LOC_TIMEOUT (6) – Request failed because it timed out • eQMI_LOC_CONFIG_NOT_SUPPORTED (7) – Request failed because an undefined configuration was requested • eQMI_LOC_INSUFFICIENT_MEMORY (8) – Request failed because the engine could not allocate sufficient memory for the request • eQMI_LOC_MAX_GEOFENCE_PROGRAMMED (9) – Request failed because the maximum number of Geofences are already programmed • eQMI_LOC_XTRA_VERSION_CHECK_FAILURE (10) – Location service failed because of an XTRA version-based file format check failure

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

3.65.3 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. The SUPL certificates are used in the AGNSS sessions for all clients, therefore it is recommended that only one client control the deletion of SUPL certificates.

3.66 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

3.66.1 Request - QMI_LOC_SET_POSITION_ENGINE_CONFIG - PARAMETERS_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

None

Optional TLVs

Name	Version introduced	Version last modified
Injected Position Control	2.3	2.3
Filter SV Usage	2.3	2.3
Store Assist Data	2.3	2.3
Enable Faster TTFF	2.20	2.20

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Injected Position Control
Length	1			2	
Value	→	boolean	injectedPositionControl	1	Controls how the injected position is used in the position engine. Valid values: <ul style="list-style-type: none"> • 0x01 (TRUE) – Use the injected position in a direct position calculation • 0x00 (FALSE) – Do not use the injected position in a direct position calculation The default value is TRUE.

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x11			1	Filter SV Usage
Length	1			2	
Value	→	boolean	filterSvUsage	1	Controls whether SV usage is filtered in a position fix. Valid values: <ul style="list-style-type: none"> • 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	→	boolean	storeAssistData	1	Controls whether assistance data is to be stored in persistent memory. Valid values: <ul style="list-style-type: none"> • 0x01 (TRUE) – Store assistance data in persistent memory • 0x00 (FALSE) – Do not store assistance data in persistent memory The default value is TRUE.
Type	0x13			1	Enable Faster TTFF
Length	1			2	
Value	→	boolean	enableFasterTTFF	1	Allows the receiver to stay on after a position session in order to collect information that will help reduce the Time To First Fix (TTFF) when the next position request is made. The receiver will stay on only if the engine determines that it needs to collect some information. The receiver will stay on for the duration needed to collect the information. If enabled, the clients may see a delay in receiving the Engine Off event after the position session ends. Valid values: <ul style="list-style-type: none"> • 0x01 (TRUE) – Allow the engine to stay on for reduced TTFF • 0x00 (FALSE) – Do not allow the engine to stay on for reduced TTFF The default value is TRUE.

3.66.2 Indication - QMI_LOC_SET_POSITION_ENGINE_CONFIG_-PARAMETERS_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version introduced	Version last modified
Set Position Engine Configuration Status	2.3	2.28

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Set Position Engine Configuration Status
Length	4			2	
Value	→	enum	status	4	<p>Status of the Set Configuration Parameters request.</p> <p>Valid values:</p> <ul style="list-style-type: none"> • eQMI_LOC_SUCCESS (0) – Request was completed successfully • eQMI_LOC_GENERAL_FAILURE (1) – Request failed because of a general failure • eQMI_LOC_UNSUPPORTED (2) – Request failed because it is not supported • eQMI_LOC_INVALID_PARAMETER (3) – Request failed because it contained invalid parameters • eQMI_LOC_ENGINE_BUSY (4) – Request failed because the engine is busy • eQMI_LOC_PHONE_OFFLINE (5) – Request failed because the phone is offline • eQMI_LOC_TIMEOUT (6) – Request failed because it timed out • eQMI_LOC_CONFIG_NOT_SUPPORTED (7) – Request failed because an undefined configuration was requested • eQMI_LOC_INSUFFICIENT_MEMORY (8) – Request failed because the engine could not allocate sufficient memory for the request

Field	Field value	Field type	Parameter	Size (byte)	Description
			status (cont.)		<ul style="list-style-type: none"> • eQMI_LOC_MAX_GEOFENCE_PROGRAMMED (9) – Request failed because the maximum number of Geofences are already programmed • eQMI_LOC_XTRA_VERSION_CHECK_FAILURE (10) – Location service failed because of an XTRA version-based file format check failure

Optional TLVs

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

Name	Version introduced	Version last modified
Failed Parameters	2.3	2.20

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Failed Parameters
Length	4			2	
Value	→	mask32	failedPositionEngineConfigParamMask	4	<p>Identifies the parameters that were not set successfully. This field is sent only if the status is other than SUCCESS.</p> <p>Valid bitmasks:</p> <ul style="list-style-type: none"> • QMI_LOC_POSITION_ENGINE_CONFIG_PARAM_MASK_INJECTED_POSITION_CONTROL (0x00000001) – Denotes whether the position engine uses the injected position in a direct position calculation. • QMI_LOC_POSITION_ENGINE_CONFIG_PARAM_MASK_FILTER_SV_USAGE (0x00000002) – Denotes whether the position engine filters the SV usage in the fix. • QMI_LOC_POSITION_ENGINE_CONFIG_PARAM_MASK_STORE_ASSIST_DATA (0x00000004) – Denotes whether the position engine stores assistance data in persistent memory. • QMI_LOC_POSITION_ENGINE_CONFIG_PARAM_MASK_ENABLE_FASTER_TTF (0x00000008) – Denotes whether the position engine stays on to optimize the TTF for the subsequent position fix.

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

3.66.3 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. These parameters control the global state of the engine, hence it is recommended that multiple clients do not set conflicting parameters.

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

3.67 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

3.67.1 Request - QMI_LOC_GET_POSITION_ENGINE_CONFIG - PARAMETERS_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

Name	Version introduced	Version last modified
Config Parameters	2.3	2.3

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Config Parameters
Length	4			2	
Value	→	mask32	getPositionEngineConfigParamMask	4	Mask denoting the configuration parameters to be retrieved. Valid bitmasks: <ul style="list-style-type: none"> • QMI_LOC_POSITION_ENGINE_CONFIG_PARAM_MASK_INJECTED_POSITION_CONTROL (0x00000001) – Denotes whether the position engine uses the injected position in a direct position calculation. • QMI_LOC_POSITION_ENGINE_CONFIG_PARAM_MASK_FILTER_SV_USAGE (0x00000002) – Denotes whether the position engine filters the SV usage in the fix.

Field	Field value	Field type	Parameter	Size (byte)	Description
			getPositionEngineConfigParamMask (cont.)		<ul style="list-style-type: none"> • QMI_LOC_POSITION_ENGINE_CONFIG_PARAM_MASK_STORE_ASSIST_DATA (0x00000004) – Denotes whether the position engine stores assistance data in persistent memory. • QMI_LOC_POSITION_ENGINE_CONFIG_PARAM_MASK_ENABLE_FASTER_TTF (0x00000008) – Denotes whether the position engine stays on to optimize the TTF for the subsequent position fix.

Optional TLVs

None

3.67.2 Indication - QMI_LOC_GET_POSITION_ENGINE_CONFIG_PARAMETERS_IND**Message type**

Indication

Sender

Service

Mandatory TLVs

Name	Version introduced	Version last modified
Get Position Engine Configuration Status	2.3	2.28

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Get Position Engine Configuration Status
Length	4			2	
Value	→	enum	status	4	Status of the Get Configuration Parameters request. Valid values: <ul style="list-style-type: none"> • eQMI_LOC_SUCCESS (0) – Request was completed successfully • eQMI_LOC_GENERAL_FAILURE (1) – Request failed because of a general failure • eQMI_LOC_UNSUPPORTED (2) – Request failed because it is not supported

Field	Field value	Field type	Parameter	Size (byte)	Description
			status (cont.)		<ul style="list-style-type: none"> • eQMI_LOC_INVALID_PARAMETER (3) – Request failed because it contained invalid parameters • eQMI_LOC_ENGINE_BUSY (4) – Request failed because the engine is busy • eQMI_LOC_PHONE_OFFLINE (5) – Request failed because the phone is offline • eQMI_LOC_TIMEOUT (6) – Request failed because it timed out • eQMI_LOC_CONFIG_NOT_SUPPORTED (7) – Request failed because an undefined configuration was requested • eQMI_LOC_INSUFFICIENT_MEMORY (8) – Request failed because the engine could not allocate sufficient memory for the request • eQMI_LOC_MAX_GEOFENCE_PROGRAMMED (9) – Request failed because the maximum number of Geofences are already programmed • eQMI_LOC_XTRA_VERSION_CHECK_FAILURE (10) – Location service failed because of an XTRA version-based file format check failure

Optional TLVs

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

Name	Version introduced	Version last modified
Injected Position Control	2.3	2.3
Filter SV Usage	2.3	2.3
Store Assist Data	2.3	2.3
Enable Faster TTFF	2.20	2.20

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

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	boolean	injectedPositionControl	1	Specifies whether the injected position is used for a direct calculation in the position engine. Valid values: <ul style="list-style-type: none"> • 0x01 (TRUE) – The injected position is used in a direct position calculation • 0x00 (FALSE) – The injected position is not used in a direct position calculation The default value is TRUE.
Type	0x11			1	Filter SV Usage
Length	1			2	
Value	→	boolean	filterSvUsage	1	Specifies whether SV usage is filtered in a position fix. Valid values: <ul style="list-style-type: none"> • 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	→	boolean	storeAssistData	1	Specifies whether assistance data is stored in persistent memory. Valid values: <ul style="list-style-type: none"> • 0x01 (TRUE) – Assistance data is stored in persistent memory • 0x00 (FALSE) – Assistance data is not stored in persistent memory The default value is TRUE.
Type	0x13			1	Enable Faster TTFF
Length	1			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	boolean	enableFasterTTFF	1	<p>Allows the receiver to stay on after a position session in order to collect information that will help reduce the TTFF when the next position request is made. The receiver will stay on only if the engine determines that it needs to collect some information. The receiver will stay on for the duration needed to collect the information. If enabled, the clients may see a delay in receiving the Engine Off event after the position session ends.</p> <p>Valid values:</p> <ul style="list-style-type: none"> • 0x01 (TRUE) – Allow the engine to stay on for reduced TTFF • 0x00 (FALSE) – Do not allow the engine to stay on for reduced TTFF <p>The default value is TRUE.</p>

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

3.67.3 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 is returned and no action is taken. It is safe for multiple clients to use this command.

3.68 QMI_LOC_EVENT_NI_GEOFENCE_NOTIFICATION

Informs the control point about network-initiated Geofences.

LOC message ID

0x0060

Version introduced

Major - 2, Minor - 8

3.68.1 Indication - QMI_LOC_EVENT_NI_GEOFENCE_NOTIFICATION_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version introduced	Version last modified
Geofence ID	2.8	2.8
Operation Type	2.8	2.8

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Geofence ID
Length	4			2	
Value	→	uint32	geofenceId	4	ID of the Geofence for which this notification was generated.
Type	0x02			1	Operation Type
Length	4			2	
Value	→	enum	operationType	4	Operation for which this notification was generated. Valid values: <ul style="list-style-type: none"> • eQMI_LOC_NI_GEOFENCE_ADDED (1) – An NI Geofence was added • eQMI_LOC_NI_GEOFENCE_DELETED (2) – An NI Geofence was deleted • eQMI_LOC_NI_GEOFENCE_EDITED (3) – An NI Geofence was edited; the control point can query the Geofence to find the its current state

Optional TLVs

None

3.68.2 Description of QMI_LOC_EVENT_NI_GEOFENCE_NOTIFICATION

This command notifies the client about network-initiated Geofences. The event is generated when a network-initiated Geofence is added, deleted, or edited by a Geofence client on the network side. The control point can use this information to query or delete an NI Geofence.

This request is generated only if the control point has indicated its interest by setting the appropriate mask in QMI_LOC_REG_EVENTS_REQ.

QUALCOMM
2016-05-16 00:17:18 PDT
deon_zhang@askey.com.tw

3.69 QMI_LOC_EVENT_GEOFENCE_GEN_ALERT

Notifies the control point of the Geofence status.

LOC message ID

0x0061

Version introduced

Major - 2, Minor - 8

3.69.1 Indication - QMI_LOC_EVENT_GEOFENCE_GEN_ALERT_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version introduced	Version last modified
Geofence General Alert	2.8	2.8

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Geofence General Alert
Length	4			2	
Value	→	enum	geofenceAlert	4	Specifies the Geofence general alert type. Valid values: <ul style="list-style-type: none"> • eQMI_LOC_GEOFENCE_GEN_ALERT_GNSS_UNAVAILABLE (1) – GNSS is unavailable and GNSS position fixes cannot be used to monitor Geofences • eQMI_LOC_GEOFENCE_GEN_ALERT_GNSS_AVAILABLE (2) – GNSS is now available and GNSS position fixes can be used to monitor Geofences • eQMI_LOC_GEOFENCE_GEN_ALERT_OOS (3) – The engine is out of service and no cell ID coverage information is available • eQMI_LOC_GEOFENCE_GEN_ALERT_TIME_INVALID (4) – The engine has an invalid time

Optional TLVs

None

3.69.2 Description of QMI_LOC_EVENT_GEOFENCE_GEN_ALERT

This command alerts the control point of an event that may affect the engine's ability to monitor Geofences. This request is generated only if the control point has indicated its interest by setting the appropriate mask in QMI_LOC_REG_EVENTS_REQ.

QUALCOMM®
2016-05-16 00:17:18 PDT
deon_zhang@askey.com.tw

3.70 QMI_LOC_EVENT_GEOFENCE_BREACH_NOTIFICATION

Notifies the control point of a Geofence breach event.

LOC message ID

0x0062

Version introduced

Major - 2, Minor - 8

3.70.1 Indication - QMI_LOC_EVENT_GEOFENCE_BREACH_NOTIFICATION_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version introduced	Version last modified
Geofence ID	2.8	2.8
Geofence Breach Type	2.8	2.8

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Geofence ID
Length	4			2	
Value	→	uint32	geofenceId	4	ID of the Geofence for which this notification was generated.
Type	0x02			1	Geofence Breach Type
Length	4			2	
Value	→	enum	breachType	4	The type of breach that generated this event. Valid values: • eQMI_LOC_GEOFENCE_BREACH_TYPE_ENTERING (1) – Denotes that a client entered the Geofence • eQMI_LOC_GEOFENCE_BREACH_TYPE_LEAVING (2) – Denotes that a client left the Geofence

Optional TLVs

Name	Version introduced	Version last modified
Geofence Position	2.8	2.8
Geofence Breach Confidence	2.23	2.23

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Geofence Position Position of the client when it breached the Geofence. This TLV is included if the client configures the Geofence to report position. The position is reported at the same confidence level that was specified in the Add Circular Geofence request.
Length	61			2	
Value	→	uint64	timestampUtc	8	UTC timestamp. • Units: Milliseconds since Jan. 1, 1970
		double	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
		double	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
		float	horUncEllipseSemiMinor	4	Semi-minor axis of horizontal elliptical uncertainty. • Units: Meters
		float	horUncEllipseSemiMajor	4	Semi-major axis of horizontal elliptical uncertainty. • Units: Meters
		float	horUncEllipseOrient Azimuth	4	Elliptical horizontal uncertainty azimuth of orientation. • Units: Decimal degrees • Range: 0 to 180

Field	Field value	Field type	Parameter	Size (byte)	Description
		boolean	speedHorizontal_valid	1	Indicates whether the Horizontal speed field contains valid information. <ul style="list-style-type: none"> • 0x01 (TRUE) – Horizontal speed is valid • 0x00 (FALSE) – Horizontal speed is invalid and is to be ignored
		float	speedHorizontal	4	Horizontal speed. <ul style="list-style-type: none"> • Units: Meters/second
		boolean	altitudeWrtEllipsoid_valid	1	Indicates whether the altitude field contains valid information. <ul style="list-style-type: none"> • 0x01 (TRUE) – Altitude field is valid • 0x00 (FALSE) – Altitude field is invalid and is to be ignored
		float	altitudeWrtEllipsoid	4	Altitude with respect to the WGS84 ellipsoid. <ul style="list-style-type: none"> • Units: Meters • Range: -500 to 15883
		boolean	vertUnc_valid	1	Indicates whether the Vertical Uncertainty field contains valid information. <ul style="list-style-type: none"> • 0x01 (TRUE) – Vertical Uncertainty field is valid • 0x00 (FALSE) – Vertical Uncertainty field is invalid and is to be ignored
		float	vertUnc	4	Vertical uncertainty. <ul style="list-style-type: none"> • Units: Meters
		boolean	speedVertical_valid	1	Indicates whether the Vertical Speed field contains valid information. <ul style="list-style-type: none"> • 0x01 (TRUE) – Vertical Speed field is valid • 0x00 (FALSE) – Vertical Speed field is invalid and is to be ignored
		float	speedVertical	4	Vertical speed. <ul style="list-style-type: none"> • Units: Meters/second
		boolean	heading_valid	1	Indicates whether the Heading field contains valid information. <ul style="list-style-type: none"> • 0x01 (TRUE) – Heading field is valid • 0x00 (FALSE) – Heading field is invalid and is to be ignored
		float	heading	4	Heading. <ul style="list-style-type: none"> • Units: Degrees • Range: 0 to 359.999
Type	0x11			1	Geofence Breach Confidence
Length	4			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	enum	breachConfidence	4	<p>Given a breach event, the confidence determines the probability that the breach happened at the Geofence boundary. Valid values:</p> <ul style="list-style-type: none"> • eQMI_LOC_GEOFENCE_CONFIDENCE_LOW (0x01) – Geofence engine indicates a breach with low confidence; this setting results in lower power usage, and it can impact the yield because incorrect breach events may be sent • eQMI_LOC_GEOFENCE_CONFIDENCE_MED (0x02) – (Default) Geofence engine indicates a breach with medium confidence • eQMI_LOC_GEOFENCE_CONFIDENCE_HIGH (0x03) – Geofence engine indicates a breach with high confidence; this setting results in higher power usage

3.70.2 Description of QMI_LOC_EVENT_GEOFENCE_BREACH_-NOTIFICATION

This command notifies the control point when a Geofence is breached. This request is generated only if the control point has indicated its interest by setting the appropriate mask in QMI_LOC_REG_EVENTS_REQ.

3.71 QMI_LOC_ADD_CIRCULAR_GEOFENCE

Used by the control point to add a circular Geofence.

LOC message ID

0x0063

Version introduced

Major - 2, Minor - 8

3.71.1 Request - QMI_LOC_ADD_CIRCULAR_GEOFENCE_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

Name	Version introduced	Version last modified
Transaction ID	2.8	2.8
Circular Geofence Arguments	2.8	2.8
Breach Event Mask	2.8	2.8
Include Position in Breach Event	2.8	2.8

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Transaction ID
Length	4			2	
Value	→	uint32	transactionId	4	Identifies the transaction. The transaction ID is returned in the Add Circular Geofence indication.
Type	0x02			1	Circular Geofence Arguments
Length	20			2	
Value	→	double	latitude	8	Latitude of the center of the Geofence.
		double	longitude	8	Longitude of the center of the Geofence.
		uint32	radius	4	Radius of the circular Geofence in meters.
Type	0x03			1	Breach Event Mask
Length	1			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	mask8	breachMask	1	Specifies the breach events in which the client is interested. Valid values: <ul style="list-style-type: none"> • 0x01 – GEOFENCE_BREACH_ENTERING_MASK • 0x02 – GEOFENCE_BREACH_LEAVING_MASK
Type	0x04			1	Include Position in Breach Event
Length	1			2	
Value	→	boolean	includePosition	1	Specifies whether the Geofence engine is to include the position in a breach event. Valid values: <ul style="list-style-type: none"> • 0x01 (TRUE) – Position will be reported with the breach event • 0x00 (FALSE) – Position will not be reported with the breach event

Optional TLVs

Name	Version introduced	Version last modified
Responsiveness	2.8	2.28
Confidence	2.8	2.8

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Responsiveness
Length	4			2	
Value	→	enum	responsiveness	4	Specifies the rate of detection for a Geofence breach. This may impact the time lag between the actual breach event and when it is reported. This parameter has power implications and is to be fine-tuned to optimize power savings. Valid values: <ul style="list-style-type: none"> • eQMI_LOC_GEOFENCE_RESPONSIVENESS_LOW (0x01) – The Geofence is monitored for a breach at a low rate of 15 minutes. The gap between the actual breach and the time it is reported is higher. This setting results in lower power usage. • eQMI_LOC_GEOFENCE_RESPONSIVENESS_MED (0x02) – The Geofence is monitored for a breach at a medium rate of 2 minutes. This is the default setting.

Field	Field value	Field type	Parameter	Size (byte)	Description
			responsiveness (cont.)		<ul style="list-style-type: none"> • eQMI_LOC_GEOFENCE_RESPONSIVENESS_HIGH (0x03) – The Geofence is monitored for a breach at a high rate of 10 seconds. The gap between the actual breach and the time it is reported is low. This results in higher power usage. • eQMI_LOC_GEOFENCE_RESPONSIVENESS_ULTRA_HIGH (0x04) – The Geofence is monitored for a breach at a very high rate of 1 second. The gap between the actual breach and the time it is reported is very low. This results in very high power usage. This setting must be avoided whenever possible because of the drastic power implications.
Type	0x11			1	Confidence
Length	4			2	
Value	→	enum	confidence	4	<p>Given a breach event, the confidence determines the probability that the breach happened at the Geofence boundary. This parameter has power implications and is to be fine-tuned to optimize power savings.</p> <p>Valid values:</p> <ul style="list-style-type: none"> • eQMI_LOC_GEOFENCE_CONFIDENCE_LOW (0x01) – Geofence engine indicates a breach with low confidence; this setting results in lower power usage, and it can impact the yield because incorrect breach events may be sent • eQMI_LOC_GEOFENCE_CONFIDENCE_MED (0x02) – (Default) Geofence engine indicates a breach with medium confidence • eQMI_LOC_GEOFENCE_CONFIDENCE_HIGH (0x03) – Geofence engine indicates a breach with high confidence; this setting results in higher power usage

3.71.2 Indication - QMI_LOC_ADD_CIRCULAR_GEOFENCE_IND

Message type

Indication

Sender

Control point

Mandatory TLVs

Name	Version introduced	Version last modified
Add Circular Geofence Status	2.8	2.28

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Add Circular Geofence Status
Length	4			2	
Value	→	enum	status	4	<p>Status of the Add Circular Geofence request.</p> <p>Valid values:</p> <ul style="list-style-type: none"> • eQMI_LOC_SUCCESS (0) – Request was completed successfully • eQMI_LOC_GENERAL_FAILURE (1) – Request failed because of a general failure • eQMI_LOC_UNSUPPORTED (2) – Request failed because it is not supported • eQMI_LOC_INVALID_PARAMETER (3) – Request failed because it contained invalid parameters • eQMI_LOC_ENGINE_BUSY (4) – Request failed because the engine is busy • eQMI_LOC_PHONE_OFFLINE (5) – Request failed because the phone is offline • eQMI_LOC_TIMEOUT (6) – Request failed because it timed out • eQMI_LOC_CONFIG_NOT_SUPPORTED (7) – Request failed because an undefined configuration was requested • eQMI_LOC_INSUFFICIENT_MEMORY (8) – Request failed because the engine could not allocate sufficient memory for the request

Field	Field value	Field type	Parameter	Size (byte)	Description
			status (cont.)		<ul style="list-style-type: none"> • eQMI_LOC_MAX_GEOFENCE_PROGRAMMED (9) – Request failed because the maximum number of Geofences are already programmed • eQMI_LOC_XTRA_VERSION_CHECK_FAILURE (10) – Location service failed because of an XTRA version-based file format check failure

Optional TLVs

Name	Version introduced	Version last modified
Transaction ID	2.8	2.8
Geofence ID	2.8	2.8

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Transaction ID
Length	4			2	
Value	→	uint32	transactionId	4	Transaction ID that was specified in the Add Circular Geofence request. This parameter will always be present if the status field is set to SUCCESS.
Type	0x11			1	Geofence ID
Length	4			2	
Value	→	uint32	geofenceId	4	Geofence identifier allocated by the engine. The client must include this identifier in all transactions pertaining to this Geofence.

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

3.71.3 Description of QMI_LOC_ADD_CIRCULAR_GEOFENCE

This command is used by the control point to add a circular Geofence. The client must specify a transaction ID with each add request to identify the transaction. The same transaction ID is returned in the QMI_LOC_ADD_CIRCULAR_GEOFENCE_IND indication. If the request is successful, a Geofence ID is returned. This ID is to be specified in all further requests that correspond to the Geofence that was added. Multiple clients can send this request, however there can only be one request outstanding at any time. The breach indications for all Geofences are sent to all clients that registered for the breach events.

QUALCOMM®
2016-05-16 00:17:18 PDT
deon_zhang@askey.com.tw

3.72 QMI_LOC_DELETE_GEOFENCE

Used by the control point to delete a Geofence.

LOC message ID

0x0064

Version introduced

Major - 2, Minor - 8

3.72.1 Request - QMI_LOC_DELETE_GEOFENCE_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

Name	Version introduced	Version last modified
Geofence ID	2.8	2.8
Transaction ID	2.8	2.8

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Geofence ID
Length	4			2	
Value	→	uint32	geofenceId	4	Identifier for the Geofence that is to be deleted.
Type	0x02			1	Transaction ID
Length	4			2	
Value	→	uint32	transactionId	4	Identifies the transaction. The transaction ID is returned in the Delete Geofence indication.

Optional TLVs

None

3.72.2 Indication - QMI_LOC_DELETE_GEOFENCE_IND

Message type

Indication

Sender

Control point

Mandatory TLVs

Name	Version introduced	Version last modified
Delete Geofence Status	2.8	2.28

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Delete Geofence Status
Length	4			2	
Value	→	enum	status	4	<p>Status of the Delete Geofence request.</p> <p>Valid values:</p> <ul style="list-style-type: none"> • eQMI_LOC_SUCCESS (0) – Request was completed successfully • eQMI_LOC_GENERAL_FAILURE (1) – Request failed because of a general failure • eQMI_LOC_UNSUPPORTED (2) – Request failed because it is not supported • eQMI_LOC_INVALID_PARAMETER (3) – Request failed because it contained invalid parameters • eQMI_LOC_ENGINE_BUSY (4) – Request failed because the engine is busy • eQMI_LOC_PHONE_OFFLINE (5) – Request failed because the phone is offline • eQMI_LOC_TIMEOUT (6) – Request failed because it timed out • eQMI_LOC_CONFIG_NOT_SUPPORTED (7) – Request failed because an undefined configuration was requested • eQMI_LOC_INSUFFICIENT_MEMORY (8) – Request failed because the engine could not allocate sufficient memory for the request

Field	Field value	Field type	Parameter	Size (byte)	Description
			status (cont.)		<ul style="list-style-type: none"> • eQMI_LOC_MAX_GEOFENCE_PROGRAMMED (9) – Request failed because the maximum number of Geofences are already programmed • eQMI_LOC_XTRA_VERSION_CHECK_FAILURE (10) – Location service failed because of an XTRA version-based file format check failure

Optional TLVs

Name	Version introduced	Version last modified
Geofence ID	2.8	2.8
Transaction ID	2.8	2.8

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Geofence ID
Length	4			2	
Value	→	uint32	geofenceId	4	Identifier for the Geofence that was deleted.
Type	0x11			1	Transaction ID
Length	4			2	
Value	→	uint32	transactionId	4	Transaction ID that was specified in the Delete Geofence request. This parameter will always be present if the status field is set to SUCCESS.

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

3.72.3 Description of QMI_LOC_DELETE_GEOFENCE

This command is used by the client to delete a specific Geofence. Multiple clients must ensure that they do not delete Geofences for which they are not responsible.

3.73 QMI_LOC_QUERY_GEOFENCE

Used by the control point to query a Geofence.

LOC message ID

0x0065

Version introduced

Major - 2, Minor - 8

3.73.1 Request - QMI_LOC_QUERY_GEOFENCE_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

Name	Version introduced	Version last modified
Geofence ID	2.8	2.8
Transaction ID	2.8	2.8

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Geofence ID
Length	4			2	
Value	→	uint32	geofenceId	4	Identifier for the Geofence that is to be queried.
Type	0x02			1	Transaction ID
Length	4			2	
Value	→	uint32	transactionId	4	Identifies the transaction. The transaction ID is returned with the Query Geofence indication.

Optional TLVs

None

3.73.2 Indication - QMI_LOC_QUERY_GEOFENCE_IND

Message type

Indication

Sender

Control point

Mandatory TLVs

Name	Version introduced	Version last modified
Query Geofence Status	2.8	2.28

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Query Geofence Status
Length	4			2	
Value	→	enum	status	4	<p>Status of the Query Geofence request.</p> <p>Valid values:</p> <ul style="list-style-type: none"> • eQMI_LOC_SUCCESS (0) – Request was completed successfully • eQMI_LOC_GENERAL_FAILURE (1) – Request failed because of a general failure • eQMI_LOC_UNSUPPORTED (2) – Request failed because it is not supported • eQMI_LOC_INVALID_PARAMETER (3) – Request failed because it contained invalid parameters • eQMI_LOC_ENGINE_BUSY (4) – Request failed because the engine is busy • eQMI_LOC_PHONE_OFFLINE (5) – Request failed because the phone is offline • eQMI_LOC_TIMEOUT (6) – Request failed because it timed out • eQMI_LOC_CONFIG_NOT_SUPPORTED (7) – Request failed because an undefined configuration was requested • eQMI_LOC_INSUFFICIENT_MEMORY (8) – Request failed because the engine could not allocate sufficient memory for the request

Field	Field value	Field type	Parameter	Size (byte)	Description
			status (cont.)		<ul style="list-style-type: none"> • eQMI_LOC_MAX_GEOFENCE_PROGRAMMED (9) – Request failed because the maximum number of Geofences are already programmed • eQMI_LOC_XTRA_VERSION_CHECK_FAILURE (10) – Location service failed because of an XTRA version-based file format check failure

Optional TLVs

Name	Version introduced	Version last modified
Geofence ID	2.8	2.8
Transaction ID	2.8	2.8
Geofence Origin	2.8	2.8
Position with Respect to Geofence	2.8	2.8
Circular Geofence Parameters	2.8	2.8
Geofence State	2.8	2.8

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Geofence ID
Length	4			2	
Value	→	uint32	geofenceId	4	Identifier for the Geofence that was queried.
Type	0x11			1	Transaction ID
Length	4			2	
Value	→	uint32	transactionId	4	Transaction ID that was specified in the Query Geofence request. This parameter will always be present if the status field is set to SUCCESS.
Type	0x12			1	Geofence Origin
Length	4			2	
Value	→	enum	geofenceOrigin	4	Originator of the Geofence. Valid values: <ul style="list-style-type: none"> • eQMI_LOC_GEOFENCE_ORIGIN_NETWORK (1) – Geofence was initiated by a network-initiated client • eQMI_LOC_GEOFENCE_ORIGIN_DEVICE (2) – Geofence was initiated by the device
Type	0x13			1	Position with Respect to Geofence
Length	4			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	enum	posWrtGeofence	4	Indicates if the client is currently inside or outside the Geofence. Valid values: • eQMI_LOC_GEOFENCE_POSITION_INSIDE (0x01) – Position is inside a Geofence • eQMI_LOC_GEOFENCE_POSITION_OUTSIDE (0x02) – Position is outside a Geofence
Type	0x14			1	Circular Geofence Parameters
Length	20			2	
Value	→	double	latitude	8	Latitude of the center of the Geofence.
		double	longitude	8	Longitude of the center of the Geofence.
		uint32	radius	4	Radius of the circular Geofence in meters.
Type	0x15			1	Geofence State
Length	4			2	
Value	→	enum	geofenceState	4	Specifies whether the Geofence is to be actively monitored. Valid values: • eQMI_LOC_GEOFENCE_STATE_ACTIVE (1) – Geofence is being actively monitored • eQMI_LOC_GEOFENCE_STATE_SUSPEND (2) – Geofence monitoring is suspended

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

3.73.3 Description of QMI_LOC_QUERY_GEOFENCE

This command is used by the client to obtain information regarding a specific Geofence. It is safe for multiple clients to use this command, however only one request can be outstanding at any time.

3.74 QMI_LOC_EDIT_GEOFENCE

Used by the control point to edit a Geofence.

LOC message ID

0x0066

Version introduced

Major - 2, Minor - 8

3.74.1 Request - QMI_LOC_EDIT_GEOFENCE_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

Name	Version introduced	Version last modified
Geofence ID	2.8	2.8
Transaction ID	2.8	2.8

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Geofence ID
Length	4			2	
Value	→	uint32	geofenceId	4	Identifier for the Geofence to be edited.
Type	0x02			1	Transaction ID
Length	4			2	
Value	→	uint32	transactionId	4	Transaction ID that was specified in the Edit Geofence request. This parameter will always be present if the status field is set to SUCCESS.

Optional TLVs

Name	Version introduced	Version last modified
Geofence State	2.8	2.8
Breach Event Mask	2.8	2.8
Responsiveness	2.23	2.23

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Geofence State
Length	4			2	
Value	→	enum	geofenceState	4	Specifies whether the Geofence is to be actively monitored. Valid values: <ul style="list-style-type: none"> • eQMI_LOC_GEOFENCE_STATE_ACTIVE (1) – Geofence is being actively monitored • eQMI_LOC_GEOFENCE_STATE_SUSPEND (2) – Geofence monitoring is suspended
Type	0x11			1	Breach Event Mask
Length	1			2	
Value	→	mask8	breachMask	1	Specifies the breach events in which the client is interested. Valid values: <ul style="list-style-type: none"> • 0x01 – GEOFENCE_BREACH_ENTERING_MASK • 0x02 – GEOFENCE_BREACH_LEAVING_MASK
Type	0x12			1	Responsiveness
Length	4			2	
Value	→	enum	responsiveness	4	Specifies the rate of detection for a Geofence breach. This may impact the time lag between the actual breach event and when it is reported. This parameter has power implications and is to be fine-tuned to optimize power savings. Valid values: <ul style="list-style-type: none"> • eQMI_LOC_GEOFENCE_RESPONSIVENESS_LOW (0x01) – The Geofence is monitored for a breach at a low rate of 15 minutes. The gap between the actual breach and the time it is reported is higher. This setting results in lower power usage. • eQMI_LOC_GEOFENCE_RESPONSIVENESS_MED (0x02) – The Geofence is monitored for a breach at a medium rate of 2 minutes. This is the default setting.

Field	Field value	Field type	Parameter	Size (byte)	Description
			responsiveness (cont.)		<ul style="list-style-type: none"> • eQMI_LOC_GEOFENCE_RESPONSIVENESS_HIGH (0x03) – The Geofence is monitored for a breach at a high rate of 10 seconds. The gap between the actual breach and the time it is reported is low. This results in higher power usage. • eQMI_LOC_GEOFENCE_RESPONSIVENESS_ULTRA_HIGH (0x04) – The Geofence is monitored for a breach at a very high rate of 1 second. The gap between the actual breach and the time it is reported is very low. This results in very high power usage. This setting must be avoided whenever possible because of the drastic power implications.

3.74.2 Indication - QMI_LOC_EDIT_GEOFENCE_IND

Message type

Indication

Sender

Control point

Mandatory TLVs

Name	Version introduced	Version last modified
Edit Geofence Status	2.8	2.28

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Edit Geofence Status
Length	4			2	
Value	→	enum	status	4	Status of the Edit Geofence request. Valid values: <ul style="list-style-type: none"> • eQMI_LOC_SUCCESS (0) – Request was completed successfully • eQMI_LOC_GENERAL_FAILURE (1) – Request failed because of a general failure • eQMI_LOC_UNSUPPORTED (2) – Request failed because it is not supported

Field	Field value	Field type	Parameter	Size (byte)	Description
			status (cont.)		<ul style="list-style-type: none"> • eQMI_LOC_INVALID_PARAMETER (3) – Request failed because it contained invalid parameters • eQMI_LOC_ENGINE_BUSY (4) – Request failed because the engine is busy • eQMI_LOC_PHONE_OFFLINE (5) – Request failed because the phone is offline • eQMI_LOC_TIMEOUT (6) – Request failed because it timed out • eQMI_LOC_CONFIG_NOT_SUPPORTED (7) – Request failed because an undefined configuration was requested • eQMI_LOC_INSUFFICIENT_MEMORY (8) – Request failed because the engine could not allocate sufficient memory for the request • eQMI_LOC_MAX_GEOFENCE_PROGRAMMED (9) – Request failed because the maximum number of Geofences are already programmed • eQMI_LOC_XTRA_VERSION_CHECK_FAILURE (10) – Location service failed because of an XTRA version-based file format check failure

Optional TLVs

Name	Version introduced	Version last modified
Geofence ID	2.8	2.8
Transaction ID	2.8	2.8
Failed Parameters	2.8	2.8

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Geofence ID
Length	4			2	
Value	→	uint32	geofenceId	4	Identifier for the Geofence that was edited.
Type	0x11			1	Transaction ID
Length	4			2	
Value	→	uint32	transactionId	4	Identifies the transaction. The transaction ID is specified in the Edit Geofence request.

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x12			1	Failed Parameters
Length	4			2	
Value	→	mask32	failedParams	4	Specified only when the status is not set to SUCCESS. If the mask corresponding to a field is set, it indicates that the Geofence parameter could not be edited. Valid values: <ul style="list-style-type: none"> • 0x00000001 – GEOFENCE_PARAM_MASK_GEOFENCE_STATE • 0x00000002 – GEOFENCE_PARAM_MASK_BREACH_MASK

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

3.74.3 Description of QMI_LOC_EDIT_GEOFENCE

This command is used to edit an already existing Geofence. Not all configuration parameters of a Geofence can be changed. If a client wants to change other parameters, it must delete the current Geofence and add a new one. Multiple clients must not edit Geofences for which they are not responsible. Only one outstanding request is allowed at any time.

3.75 QMI_LOC_GET_BEST_AVAILABLE_POSITION

Used by the control point to get the best available position estimate from the location engine.

LOC message ID

0x0067

Version introduced

Major - 2, Minor - 10

3.75.1 Request - QMI_LOC_GET_BEST_AVAILABLE_POSITION_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

Name	Version introduced	Version last modified
Transaction ID	2.10	2.10

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Transaction ID
Length	4			2	
Value	→	uint32	transactionId	4	Identifies the transaction. The transaction ID is returned in the Get Best Available Position indication.

Optional TLVs

None

3.75.2 Indication - QMI_LOC_GET_BEST_AVAILABLE_POSITION_IND

Message type

Indication

Sender

Control point

Mandatory TLVs

Name	Version introduced	Version last modified
Get Best Available Position Status	2.10	2.28

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Get Best Available Position Status
Length	4			2	
Value	→	enum	status	4	<p>Status of the Get Best Available Position request.</p> <p>Valid values:</p> <ul style="list-style-type: none"> • eQMI_LOC_SUCCESS (0) – Request was completed successfully • eQMI_LOC_GENERAL_FAILURE (1) – Request failed because of a general failure • eQMI_LOC_UNSUPPORTED (2) – Request failed because it is not supported • eQMI_LOC_INVALID_PARAMETER (3) – Request failed because it contained invalid parameters • eQMI_LOC_ENGINE_BUSY (4) – Request failed because the engine is busy • eQMI_LOC_PHONE_OFFLINE (5) – Request failed because the phone is offline • eQMI_LOC_TIMEOUT (6) – Request failed because it timed out • eQMI_LOC_CONFIG_NOT_SUPPORTED (7) – Request failed because an undefined configuration was requested • eQMI_LOC_INSUFFICIENT_MEMORY (8) – Request failed because the engine could not allocate sufficient memory for the request • eQMI_LOC_MAX_GEOFENCE_PROGRAMMED (9) – Request failed because the maximum number of Geofences are already programmed • eQMI_LOC_XTRA_VERSION_CHECK_FAILURE (10) – Location service failed because of an XTRA version-based file format check failure

Optional TLVs

Name	Version introduced	Version last modified
Transaction ID	2.10	2.10
Latitude	2.10	2.10
Longitude	2.10	2.10
Circular Horizontal Position Uncertainty	2.10	2.10
Altitude With Respect to Ellipsoid	2.10	2.10
Vertical Uncertainty	2.10	2.10
UTC Timestamp	2.10	2.10
Time Uncertainty	2.10	2.10
Horizontal Elliptical Uncertainty Semi-Minor Axis	2.11	2.11
Horizontal Elliptical Uncertainty Semi-Major Axis	2.11	2.11
Horizontal Elliptical Uncertainty Azimuth	2.11	2.11
Horizontal Circular Confidence	2.11	2.11
Horizontal Elliptical Confidence	2.11	2.11
Horizontal Reliability	2.11	2.11
Horizontal Speed	2.11	2.11
Horizontal Speed Uncertainty	2.11	2.11
Altitude With Respect to Sea Level	2.11	2.11
Vertical Confidence	2.11	2.11
Vertical Reliability	2.11	2.11
Vertical Speed	2.11	2.11
Vertical Speed Uncertainty	2.11	2.11
Heading	2.11	2.11
Heading Uncertainty	2.11	2.11
Magnetic Deviation	2.11	2.11
Technology Used Mask	2.11	2.20
Dilution of Precision	2.11	2.11
GPS Time	2.11	2.11
Time Source	2.11	2.22
Sensor Data Usage	2.11	2.11
SVs Used to Calculate the Fix	2.11	2.22

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Transaction ID
Length	4			2	
Value	→	uint32	transactionId	4	Transaction ID that was specified in the Get Best Available Position request. This parameter will always be present if the status field is set to SUCCESS.
Type	0x11			1	Latitude
Length	8			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	double	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	0x12			1	Longitude
Length	8			2	
Value	→	double	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	0x13			1	Circular Horizontal Position Uncertainty
Length	4			2	
Value	→	float	horUncCircular	4	Horizontal position uncertainty (circular). • Units: Meters
Type	0x14			1	Altitude With Respect to Ellipsoid
Length	4			2	
Value	→	float	altitudeWrtEllipsoid	4	Altitude with respect to the WGS84 ellipsoid. • Units: Meters • Range: -500 to 15883
Type	0x15			1	Vertical Uncertainty
Length	4			2	
Value	→	float	vertUnc	4	Vertical uncertainty. • Units: Meters
Type	0x16			1	UTC Timestamp
Length	8			2	
Value	→	uint64	timestampUtc	8	UTC timestamp. • Units: Milliseconds since Jan. 1, 1970
Type	0x17			1	Time Uncertainty
Length	4			2	
Value	→	float	timeUnc	4	Time uncertainty. • Units: Milliseconds
Type	0x18			1	Horizontal Elliptical Uncertainty Semi-Minor Axis
Length	4			2	
Value	→	float	horUncEllipseSemiMinor	4	Semi-minor axis of horizontal elliptical uncertainty. • Units: Meters

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x19			1	Horizontal Elliptical Uncertainty Semi-Major Axis
Length	4			2	
Value	→	float	horUncEllipseSemiMajor	4	Semi-major axis of horizontal elliptical uncertainty. • Units: Meters
Type	0x1A			1	Horizontal Elliptical Uncertainty Azimuth
Length	4			2	
Value	→	float	horUncEllipseOrient Azimuth	4	Elliptical horizontal uncertainty azimuth of orientation. • Units: Decimal degrees • Range: 0 to 180
Type	0x1B			1	Horizontal Circular Confidence
Length	1			2	
Value	→	uint8	horCircularConfidence	1	Horizontal circular uncertainty confidence. • Units: Percent • Range: 0 to 99
Type	0x1C			1	Horizontal Elliptical Confidence
Length	1			2	
Value	→	uint8	horEllipticalConfidence	1	Horizontal elliptical uncertainty confidence. • Units: Percent • Range: 0 to 99
Type	0x1D			1	Horizontal Reliability
Length	4			2	
Value	→	enum	horReliability	4	Specifies the reliability of the horizontal position. Valid values: • eQMI_LOC_RELIABILITY_NOT_SET (0) – Location reliability is not set • eQMI_LOC_RELIABILITY_VERY_LOW (1) – Location reliability is very low; use it at your own risk • eQMI_LOC_RELIABILITY_LOW (2) – Location reliability is low; little or no cross-checking is possible • eQMI_LOC_RELIABILITY_MEDIUM (3) – Location reliability is medium; limited cross-check passed • eQMI_LOC_RELIABILITY_HIGH (4) – Location reliability is high; strong cross-check passed
Type	0x1E			1	Horizontal Speed
Length	4			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	float	horSpeed	4	Horizontal speed. • Units: Meters/second
Type	0x1F			1	Horizontal Speed Uncertainty
Length	4			2	
Value	→	float	horSpeedUnc	4	Horizontal speed uncertainty. • Units: Meters/second
Type	0x20			1	Altitude With Respect to Sea Level
Length	4			2	
Value	→	float	altitudeWrtMeanSeaLevel	4	Altitude with respect to mean sea level. • Units: Meters
Type	0x21			1	Vertical Confidence
Length	1			2	
Value	→	uint8	vertConfidence	1	Vertical uncertainty confidence. • Units: Percent • Range: 0 to 99
Type	0x22			1	Vertical Reliability
Length	4			2	
Value	→	enum	vertReliability	4	Specifies the reliability of the vertical position. Valid values: • eQMI_LOC_RELIABILITY_NOT_SET (0) – Location reliability is not set • eQMI_LOC_RELIABILITY_VERY_LOW (1) – Location reliability is very low; use it at your own risk • eQMI_LOC_RELIABILITY_LOW (2) – Location reliability is low; little or no cross-checking is possible • eQMI_LOC_RELIABILITY_MEDIUM (3) – Location reliability is medium; limited cross-check passed • eQMI_LOC_RELIABILITY_HIGH (4) – Location reliability is high; strong cross-check passed
Type	0x23			1	Vertical Speed
Length	4			2	
Value	→	float	vertSpeed	4	Vertical speed. • Units: Meters/second
Type	0x24			1	Vertical Speed Uncertainty
Length	4			2	
Value	→	float	vertSpeedUnc	4	Vertical speed uncertainty. • Units: Meters/second
Type	0x25			1	Heading
Length	4			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	float	heading	4	Heading. • Units: Degrees • Range: 0 to 359.999
Type	0x26			1	Heading Uncertainty
Length	4			2	
Value	→	float	headingUnc	4	Heading uncertainty. • Type: Floating point • Range: 0 to 359.999
Type	0x27			1	Magnetic Deviation
Length	4			2	
Value	→	float	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	0x28			1	Technology Used Mask
Length	4			2	
Value	→	mask32	technologyMask	4	Technology used in computing this fix. Valid bitmasks: • QMI_LOC_POS_TECH_MASK_SATELLITE (0x00000001) – Satellites were used to generate the fix • QMI_LOC_POS_TECH_MASK_CELLID (0x00000002) – Cell towers were used to generate the fix • QMI_LOC_POS_TECH_MASK_WIFI (0x00000004) – Wi-Fi access points were used to generate the fix • QMI_LOC_POS_TECH_MASK_SENSORS (0x00000008) – Sensors were used to generate the fix • QMI_LOC_POS_TECH_MASK_REFERENCE_LOCATION (0x00000010) – Reference Location was used to generate the fix • QMI_LOC_POS_TECH_MASK_INJECTED_COARSE_POSITION (0x00000020) – Coarse position injected into the location engine was used to generate the fix • QMI_LOC_POS_TECH_MASK_AFLT (0x00000040) – AFLT was used to generate the fix • QMI_LOC_POS_TECH_MASK_HYBRID (0x00000080) – GNSS and network-provided measurements were used to generate the fix

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x29			1	Dilution of Precision
Length	12			2	
Value	→	float	PDOP	4	Position dilution of precision. <ul style="list-style-type: none"> Range: 1 (highest accuracy) to 50 (lowest accuracy) PDOP = square root of (HDOP² + VDOP²)
		float	HDOP	4	Horizontal dilution of precision. <ul style="list-style-type: none"> Range: 1 (highest accuracy) to 50 (lowest accuracy)
		float	VDOP	4	Vertical dilution of precision. <ul style="list-style-type: none"> Range: 1 (highest accuracy) to 50 (lowest accuracy)
Type	0x2A			1	GPS Time
Length	6			2	
Value	→	uint16	gpsWeek	2	Current GPS week as calculated from midnight, Jan. 6, 1980. <ul style="list-style-type: none"> Units: Weeks
		uint32	gpsTimeOfWeekMs	4	Amount of time into the current GPS week. <ul style="list-style-type: none"> Units: Milliseconds
Type	0x2B			1	Time Source
Length	4			2	
Value	→	enum	timeSrc	4	Time source. Valid values: <ul style="list-style-type: none"> eQMI_LOC_TIME_SRC_INVALID (0) – Invalid time. eQMI_LOC_TIME_SRC_NETWORK_TIME_TRANSFER (1) – Time is set by the 1X system eQMI_LOC_TIME_SRC_NETWORK_TIME_TAGGING (2) – Time is set by WCDMA/GSM time tagging (i.e., associating network time with GPS time) eQMI_LOC_TIME_SRC_EXTERNAL_INPUT (3) – Time is set by an external injection eQMI_LOC_TIME_SRC_TOW_DECODE (4) – Time is set after decoding over-the-air GPS navigation data from one GPS satellite eQMI_LOC_TIME_SRC_TOW_CONFIRMED (5) – Time is set after decoding over-the-air GPS navigation data from multiple satellites

Field	Field value	Field type	Parameter	Size (byte)	Description
			timeSrc (cont.)		<ul style="list-style-type: none"> • eQMI_LOC_TIME_SRC_TOW_AND_WEEK_CONFIRMED (6) – Both time of the week and the GPS week number are known • eQMI_LOC_TIME_SRC_NAV_SOLUTION (7) – Time is set by the position engine after the fix is obtained • eQMI_LOC_TIME_SRC_SOLVE_FOR_TIME (8) – Time is set by the position engine after performing SFT; this is done when the clock time uncertainty is large • eQMI_LOC_TIME_SRC_GLO_TOW_DECODE (9) – Time is set after decoding GLO satellites • eQMI_LOC_TIME_SRC_TIME_TRANSFORM (10) – Time is set after transforming the GPS to GLO time • eQMI_LOC_TIME_SRC_WCDMA_SLEEP_TIME_TAGGING (11) – Time is set by the sleep time tag provided by the WCDMA network • eQMI_LOC_TIME_SRC_GSM_SLEEP_TIME_TAGGING (12) – Time is set by the sleep time tag provided by the GSM network • eQMI_LOC_TIME_SRC_UNKNOWN (13) – Source of the time is unknown • eQMI_LOC_TIME_SRC_SYSTEM_TIMETICK (14) – Time is derived from the system clock (better known as the slow clock); GNSS time is maintained irrespective of the GNSS receiver state • eQMI_LOC_TIME_SRC_QZSS_TOW_DECODE (15) – Time is set after decoding QZSS satellites • eQMI_LOC_TIME_SRC_BDS_TOW_DECODE (16) – Time is set after decoding BDS satellites
Type	0x2C			1	Sensor Data Usage
Length	8			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	mask32	usageMask	4	Specifies which sensors were used in calculating the position in the position report. Valid bitmasks: <ul style="list-style-type: none"> • 0x00000001 – SENSOR_USED_ACCEL • 0x00000002 – SENSOR_USED_GYRO
		mask32	aidingIndicatorMask	4	Specifies which results were aided by sensors. Valid bitmasks: <ul style="list-style-type: none"> • 0x00000001 – AIDED_HEADING • 0x00000002 – AIDED_SPEED • 0x00000004 – AIDED_POSITION • 0x00000008 – AIDED_VELOCITY
Type	0x2D			1	SVs Used to Calculate the Fix
Length	Var			2	
Value	→	uint8	gnssSvUsedList_len	1	Number of sets of the following elements: <ul style="list-style-type: none"> • gnssSvUsedList
		uint16	gnssSvUsedList	Var	Each entry in the list contains the SV ID of a satellite used for calculating this position report. The following information is associated with each SV ID: Range: <ul style="list-style-type: none"> • For GPS: 1 to 32 • For SBAS: 33 to 64 • For GLONASS: 65 to 96 • For QZSS: 193 to 197 • For BDS: 201 to 237

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

3.75.3 Description of QMI_LOC_GET_BEST_AVAILABLE_POSITION

This command can be used to get the best available position estimate from the location engine. The engine shall not consume any additional power to obtain a position for this request. For example the GNSS engine will not be turned on if it was off previously. The engine will send the best position estimate it currently has through the indication for this request. It is safe for multiple clients to use this command. However, there can only be one outstanding request at any time.

QUALCOMM®
2016-05-16 00:17:18 PDT
deon_zhang@askey.com.tw

3.76 QMI_LOC_INJECT_MOTION_DATA

Injects motion data for MSM GPS service use.

LOC message ID

0x0068

Version introduced

Major - 2, Minor - 12

3.76.1 Request - QMI_LOC_INJECT_MOTION_DATA_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

Name	Version introduced	Version last modified
Motion Data	2.12	2.12

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Motion Data
Length	16			2	
Value	→	enum	motion_state	4	Current motion state of the user. Valid values: <ul style="list-style-type: none"> • eQMI_LOC_MOTION_STATE_UNKNOWN (0) – Device state is not known • eQMI_LOC_MOTION_STATE_STATIONARY (1) – Device state is Stationary • eQMI_LOC_MOTION_STATE_IN_MOTION (2) – Device state is In Motion

Field	Field value	Field type	Parameter	Size (byte)	Description
		enum	motion_mode	4	<p>Modes of user motion.</p> <p>Valid values:</p> <ul style="list-style-type: none"> • eQMI_LOC_MOTION_MODE_UNKNOWN (0) – Device movement is not known • eQMI_LOC_MOTION_MODE_STATIONARY (1) – Device is not moving • eQMI_LOC_MOTION_MODE_PEDESTRIAN_UNKNOWN (200) – Device movement is in Pedestrian mode; nothing else is known about the movement • eQMI_LOC_MOTION_MODE_PEDESTRIAN_WALKING (201) – Device movement is in pedestrian Walking mode • eQMI_LOC_MOTION_MODE_PEDESTRIAN_RUNNING (202) – Device movement is in pedestrian Running mode • eQMI_LOC_MOTION_MODE_VEHICLE_UNKNOWN (300) – Device movement is in Vehicular mode; nothing else is known about the movement

Field	Field value	Field type	Parameter	Size (byte)	Description
		float	probability_of_state	4	<p>Probability that the device is actually undergoing the motion state specified by the combination of the values of motion_state, motion_mode, and motion_sub_mode.</p> <p>This value is a floating point number in the range of 0 to 100, in units of percent probability. Any value greater than 99.9999 is applied as 99.9999.</p> <p>It is recommended that if a particular combination of motion_state and motion_mode cannot be determined with more than 50 percent confidence, that a more general statement of user motion be made. For example, if the mode of In-Motion + Pedestrian-Running can only be determined with 50 percent probability, and the simpler statement of In-Motion can be determined with 90 percent probability, it is recommended that this field be used to simply state In-Motion with 90 percent probability.</p> <p>If the motion_state is not known, the value in this field is not used.</p>
		uint16	age	2	Age of the motion data in milliseconds at the time of injection.

Field	Field value	Field type	Parameter	Size (byte)	Description
		uint16	timeout	2	<p>If the age of the motion data input exceeds the timeout value, the data will no longer be used. The timeout value is in units of milliseconds. Values in the range of 0 to 10000 are accepted. If 65535 is provided, the motion data input is applied until the next input is received.</p> <p>If the determination of motion data is an instantaneous observation and no notice is guaranteed to be given via the QMI on a change in the state of the motion data, it is recommended that this field be set to 0.</p> <p>If the determination of motion data is continuously monitored external to the QMI and an update is always applied to the QMI upon any change in state, a value of 65535 is used for this field. Note that in this case, if a certain mode is set and is not later unset (e.g., by sending in the request message with a user motion state of Unknown), the value is applied indefinitely.</p>

Optional TLVs

None

3.76.2 Indication - QMI_LOC_INJECT_MOTION_DATA_IND**Message type**

Indication

Sender

Service

Mandatory TLVs

Name	Version introduced	Version last modified
Inject Motion Data Request Status	2.12	2.28

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Inject Motion Data Request Status
Length	4			2	
Value	→	enum	status	4	<p>Status of the Inject Motion Data request. Valid values:</p> <ul style="list-style-type: none"> • eQMI_LOC_SUCCESS (0) – Request was completed successfully • eQMI_LOC_GENERAL_FAILURE (1) – Request failed because of a general failure • eQMI_LOC_UNSUPPORTED (2) – Request failed because it is not supported • eQMI_LOC_INVALID_PARAMETER (3) – Request failed because it contained invalid parameters • eQMI_LOC_ENGINE_BUSY (4) – Request failed because the engine is busy • eQMI_LOC_PHONE_OFFLINE (5) – Request failed because the phone is offline • eQMI_LOC_TIMEOUT (6) – Request failed because it timed out • eQMI_LOC_CONFIG_NOT_SUPPORTED (7) – Request failed because an undefined configuration was requested • eQMI_LOC_INSUFFICIENT_MEMORY (8) – Request failed because the engine could not allocate sufficient memory for the request • eQMI_LOC_MAX_GEOFENCE_PROGRAMMED (9) – Request failed because the maximum number of Geofences are already programmed • eQMI_LOC_XTRA_VERSION_CHECK_FAILURE (10) – Location service failed because of an XTRA version-based file format check failure

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

3.76.3 Description of QMI_LOC_INJECT_MOTION_DATA

This command enables the injection of data to describe the motion of the user.

Motion data is an indication of user motion that is applied within the navigation engine to improve navigation performance. Motion data consists primarily of a stationary/nonstationary indication, a motion mode (i.e., pedestrian and walking, or vehicle), and an associated confidence. The source of motion data is typically inertial sensor data that has undergone some level of external processing. Alternatively, certain aspects of the motion data may be predefined depending on the specific application.

Motion data information is used by all clients, and it is imperative that multiple clients do not inject conflicting information into the engine.

3.77 QMI_LOC_GET_NI_GEOFENCE_ID_LIST

Used by the control point to retrieve the list of network initiated Geofence IDs.

LOC message ID

0x0069

Version introduced

Major - 2, Minor - 13

3.77.1 Request - QMI_LOC_GET_NI_GEOFENCE_ID_LIST_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

Name	Version introduced	Version last modified
Transaction ID	2.13	2.13

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Transaction ID
Length	4			2	
Value	→	uint32	transactionId	4	Identifies the transaction. The same transaction ID will be returned in the Get NI Geofence ID List indication.

Optional TLVs

None

3.77.2 Indication - QMI_LOC_GET_NI_GEOFENCE_ID_LIST_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version introduced	Version last modified
Get NI Geofence ID List Status	2.13	2.28

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Get NI Geofence ID List Status
Length	4			2	
Value	→	enum	status	4	<p>Status of the Get NI Geofence ID List request.</p> <p>Valid values:</p> <ul style="list-style-type: none"> • eQMI_LOC_SUCCESS (0) – Request was completed successfully • eQMI_LOC_GENERAL_FAILURE (1) – Request failed because of a general failure • eQMI_LOC_UNSUPPORTED (2) – Request failed because it is not supported • eQMI_LOC_INVALID_PARAMETER (3) – Request failed because it contained invalid parameters • eQMI_LOC_ENGINE_BUSY (4) – Request failed because the engine is busy • eQMI_LOC_PHONE_OFFLINE (5) – Request failed because the phone is offline • eQMI_LOC_TIMEOUT (6) – Request failed because it timed out • eQMI_LOC_CONFIG_NOT_SUPPORTED (7) – Request failed because an undefined configuration was requested • eQMI_LOC_INSUFFICIENT_MEMORY (8) – Request failed because the engine could not allocate sufficient memory for the request • eQMI_LOC_MAX_GEOFENCE_PROGRAMMED (9) – Request failed because the maximum number of Geofences are already programmed • eQMI_LOC_XTRA_VERSION_CHECK_FAILURE (10) – Location service failed because of an XTRA version-based file format check failure

Optional TLVs

Name	Version introduced	Version last modified
Transaction ID	2.13	2.13
NI Geofence ID List	2.13	2.13

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Transaction ID
Length	4			2	
Value	→	uint32	transactionId	4	Transaction ID that was specified in the Get NI Geofence ID List request.
Type	0x11			1	NI Geofence ID List
Length	Var			2	
Value	→	uint8	niGeofenceIdList_len	1	Number of sets of the following elements: • niGeofenceIdList
		uint32	niGeofenceIdList	Var	List containing the NI Geofence IDs. • Type: Array of unsigned 32-bit integers • Maximum NI Geofence ID List length: 16

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

3.77.3 Description of QMI_LOC_GET_NI_GEOFENCE_ID_LIST

This command is used to retrieve the list of network-initiated Geofence IDs. It is safe for multiple clients to use this command.

3.78 QMI_LOC_INJECT_GSM_CELL_INFO

Injects GSM cell information into the location engine.

LOC message ID

0x006A

Version introduced

Major - 2, Minor - 15

3.78.1 Request - QMI_LOC_INJECT_GSM_CELL_INFO_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

Name	Version introduced	Version last modified
GSM Cell ID	2.15	2.15
Roaming Status	2.15	2.15

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	GSM Cell ID Identifies the GSM cell on which the device is currently camped.
Length	16			2	
Value	→	uint32	MCC	4	GSM mobile country code. Refer to ITU-T E.212 [S6].
		uint32	MNC	4	GSM mobile network code. Refer to [S6].
		uint32	LAC	4	GSM location area code. Refer to [S6].
		uint32	CID	4	GSM cell identification. Refer to [S6].
Type	0x02			1	Roaming Status
Length	1			2	
Value	→	boolean	roamingStatus	1	Indicates whether the device is roaming. • 0x01 (TRUE) – Device is roaming • 0x00 (FALSE) – Device is not roaming

Optional TLVs

Name	Version introduced	Version last modified
Timing Advance	2.18	2.18

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Timing Advance
Length	4			2	
Value	→	uint32	timingAdvance	4	Round trip delay between the MS and the BS, in units of 3.69 microseconds. Refer to 3GPP TS 05.10 and TS 45.010.

3.78.2 Indication - QMI_LOC_INJECT_GSM_CELL_INFO_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version introduced	Version last modified
Inject GSM Cell Info Status	2.15	2.28

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Inject GSM Cell Info Status
Length	4			2	
Value	→	enum	status	4	Status of the Inject GSM Cell Info request. Valid values: <ul style="list-style-type: none"> • eQMI_LOC_SUCCESS (0) – Request was completed successfully • eQMI_LOC_GENERAL_FAILURE (1) – Request failed because of a general failure • eQMI_LOC_UNSUPPORTED (2) – Request failed because it is not supported • eQMI_LOC_INVALID_PARAMETER (3) – Request failed because it contained invalid parameters • eQMI_LOC_ENGINE_BUSY (4) – Request failed because the engine is busy

Field	Field value	Field type	Parameter	Size (byte)	Description
			status (cont.)		<ul style="list-style-type: none"> • eQMI_LOC_PHONE_OFFLINE (5) – Request failed because the phone is offline • eQMI_LOC_TIMEOUT (6) – Request failed because it timed out • eQMI_LOC_CONFIG_NOT_SUPPORTED (7) – Request failed because an undefined configuration was requested • eQMI_LOC_INSUFFICIENT_MEMORY (8) – Request failed because the engine could not allocate sufficient memory for the request • eQMI_LOC_MAX_GEOFENCE_PROGRAMMED (9) – Request failed because the maximum number of Geofences are already programmed • eQMI_LOC_XTRA_VERSION_CHECK_FAILURE (10) – Location service failed because of an XTRA version-based file format check failure

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

3.78.3 Description of QMI_LOC_INJECT_GSM_CELL_INFO

This command is used to inject GSM cell information into the location engine. This command is to be called by the client when the cell information or the roaming status of the device changes for a GSM network. This information is used by the location engine for assisted GNSS. Depending on the modem configuration, the roaming status of the modem will be used to decide whether the engine operates in AGNSS mode or Stand Alone mode.

If the device supports more than one service at a time, e.g., a Dual Service - Dual Standby (DS-DS) type of device, only the cell information of the user-designated preferred data service is injected.

3.79 QMI_LOC_INJECT_NETWORK_INITIATED_MESSAGE

Injects a network-initiated message into the location engine.

LOC message ID

0x006B

Version introduced

Major - 2, Minor - 15

3.79.1 Request - QMI_LOC_INJECT_NETWORK_INITIATED_MESSAGE_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

Name	Version introduced	Version last modified
Injected Network Initiated Message Type	2.15	2.15
Injected Network Initiated Message	2.15	2.15

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Injected Network Initiated Message Type
Length	4			2	
Value	→	enum	injectedNIMessageType	4	Type of the network-initiated message being injected. Valid values: • eQMI_LOC_INJECTED_NETWORK_INITIATED_MESSAGE_TYPE_SUPL (0) – SUPL network-initiated message is being injected.
Type	0x02			1	Injected Network Initiated Message
Length	Var			2	
Value	→	uint16	injectedNIMessage_len	2	Number of sets of the following elements: • injectedNIMessage

Field	Field value	Field type	Parameter	Size (byte)	Description
		uint8	injectedNIMessage	Var	Network-initiated message body. If the inject NI message type is TYPE_SUPL, the message contains a SUPL INIT message as defined in OMA-TS-ULP-V2_0-20110527-C [S5].

Optional TLVs

None

3.79.2 Indication - QMI_LOC_INJECT_NETWORK_INITIATED_MESSAGE_IND**Message type**

Indication

Sender

Service

Mandatory TLVs

Name	Version introduced	Version last modified
Inject Network Initiated Message Status	2.15	2.28

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Inject Network Initiated Message Status
Length	4			2	
Value	→	enum	status	4	Status of the Inject Network Initiated Message request. Valid values: <ul style="list-style-type: none"> • eQMI_LOC_SUCCESS (0) – Request was completed successfully • eQMI_LOC_GENERAL_FAILURE (1) – Request failed because of a general failure • eQMI_LOC_UNSUPPORTED (2) – Request failed because it is not supported • eQMI_LOC_INVALID_PARAMETER (3) – Request failed because it contained invalid parameters • eQMI_LOC_ENGINE_BUSY (4) – Request failed because the engine is busy

Field	Field value	Field type	Parameter	Size (byte)	Description
			status (cont.)		<ul style="list-style-type: none"> • eQMI_LOC_PHONE_OFFLINE (5) – Request failed because the phone is offline • eQMI_LOC_TIMEOUT (6) – Request failed because it timed out • eQMI_LOC_CONFIG_NOT_SUPPORTED (7) – Request failed because an undefined configuration was requested • eQMI_LOC_INSUFFICIENT_MEMORY (8) – Request failed because the engine could not allocate sufficient memory for the request • eQMI_LOC_MAX_GEOFENCE_PROGRAMMED (9) – Request failed because the maximum number of Geofences are already programmed • eQMI_LOC_XTRA_VERSION_CHECK_FAILURE (10) – Location service failed because of an XTRA version-based file format check failure

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

3.79.3 Description of QMI_LOC_INJECT_NETWORK_INITIATED_-MESSAGE

This command is used to inject a network-initiated message that was delivered to the client through WAP or SMS. This message is used by the location engine for assisted GNSS.

3.80 QMI_LOC_WWAN_OUT_OF_SERVICE_NOTIFICATION

Notifies the location engine that the device is out of service.

LOC message ID

0x006C

Version introduced

Major - 2, Minor - 15

3.80.1 Request - QMI_LOC_WWAN_OUT_OF_SERVICE_NOTIFICATION_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

None

Optional TLVs

None

3.80.2 Indication - QMI_LOC_WWAN_OUT_OF_SERVICE_NOTIFICATION_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version introduced	Version last modified
Notify WWAN Out of Service Status	2.15	2.28

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Notify WWAN Out of Service Status
Length	4			2	
Value	→	enum	status	4	<p>Status of the Notify WWAN Out of Service request.</p> <p>Valid values:</p> <ul style="list-style-type: none"> • eQMI_LOC_SUCCESS (0) – Request was completed successfully • eQMI_LOC_GENERAL_FAILURE (1) – Request failed because of a general failure • eQMI_LOC_UNSUPPORTED (2) – Request failed because it is not supported • eQMI_LOC_INVALID_PARAMETER (3) – Request failed because it contained invalid parameters • eQMI_LOC_ENGINE_BUSY (4) – Request failed because the engine is busy • eQMI_LOC_PHONE_OFFLINE (5) – Request failed because the phone is offline • eQMI_LOC_TIMEOUT (6) – Request failed because it timed out • eQMI_LOC_CONFIG_NOT_SUPPORTED (7) – Request failed because an undefined configuration was requested • eQMI_LOC_INSUFFICIENT_MEMORY (8) – Request failed because the engine could not allocate sufficient memory for the request • eQMI_LOC_MAX_GEOFENCE_PROGRAMMED (9) – Request failed because the maximum number of Geofences are already programmed • eQMI_LOC_XTRA_VERSION_CHECK_FAILURE (10) – Location service failed because of an XTRA version-based file format check failure

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

3.80.3 Description of QMI_LOC_WWAN_OUT_OF_SERVICE_-NOTIFICATION

This command is used to notify the location engine that the device is out of service (no longer camped on a network). This information is used by the location engine for assisted GNSS.

3.81 QMI_LOC_EVENT_PEDOMETER_CONTROL

Recommends how pedometer reports are to be sent to the location engine.

LOC message ID

0x006D

Version introduced

Major - 2, Minor - 17

3.81.1 Indication - QMI_LOC_EVENT_PEDOMETER_CONTROL_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version introduced	Version last modified
Request Pedometer Data	2.17	2.17

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Request Pedometer Data
Length	1			2	
Value	→	boolean	requestPedometerData	1	Indicates whether the GNSS location engine is requesting the client to send pedometer data. <ul style="list-style-type: none"> 0x01 (TRUE) – GNSS location engine is requesting pedometer data 0x00 (FALSE) – GNSS location engine is not requesting pedometer data

Optional TLVs

Name	Version introduced	Version last modified
Reset Step Count	2.17	2.17
Step Count Threshold	2.17	2.17

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Reset Step Count
Length	1			2	
Value	→	boolean	resetStepCount	1	Indicates whether the location engine is to reset the step count. <ul style="list-style-type: none"> • 0x01 (TRUE) – Pedometer step count is to be reset • 0x00 (FALSE) – Pedometer step count is not to be reset
Type	0x11			1	Step Count Threshold
Length	4			2	
Value	→	uint32	stepCountThreshold	4	Specifies the number of steps to be sampled in a pedometer report, as recommended by the the location engine. If the threshold is set to 0, the location engine wants a pedometer report at every step event.

3.81.2 Description of QMI_LOC_EVENT_PEDOMETER_CONTROL

This command is used to recommend how pedometer reports are to be sent to the location engine for optimal performance. This request is generated only if the control point has indicated its interest by setting the appropriate mask in QMI_LOC_REG_EVENTS_REQ. The client must use the QMI_LOC_PEDOMETER_REPORT_REQ message to send the pedometer data to the service.

3.82 QMI_LOC_EVENT_MOTION_DATA_CONTROL

Recommends how motion data reports are to be sent to the location engine.

LOC message ID

0x006E

Version introduced

Major - 2, Minor - 17

3.82.1 Indication - QMI_LOC_EVENT_MOTION_DATA_CONTROL_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version introduced	Version last modified
Request Motion Data	2.17	2.17

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Request Motion Data
Length	1			2	
Value	→	boolean	requestMotionData	1	Indicates whether the GNSS location engine is requesting the client to send motion data. <ul style="list-style-type: none"> • 0x01 (TRUE) – GNSS location engine is requesting motion data • 0x00 (FALSE) – GNSS location engine is not requesting motion data

Optional TLVs

None

3.82.2 Description of QMI_LOC_EVENT_MOTION_DATA_CONTROL

This command recommends how motion data is to be sent to the location engine for optimal performance. This request is generated only if the control point has indicated its interest by setting the appropriate mask in QMI_LOC_REG_EVENTS_REQ. The client must use the QMI_LOC_INJECT_MOTION_DATA_REQ message to send the motion data to the service.

3.83 QMI_LOC_PEDOMETER_REPORT

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

LOC message ID

0x006F

Version introduced

Major - 2, Minor - 17

3.83.1 Request - QMI_LOC_PEDOMETER_REPORT_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

Name	Version introduced	Version last modified
Time Source	2.17	2.17
Pedometer Report Timestamp	2.17	2.17
Time Interval	2.17	2.17
Step Count	2.17	2.17

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Time Source
Length	4			2	
Value	→	enum	timeSource	4	Time source for the pedometer. The location service uses this field to identify the time reference used in the pedometer data time stamp. Valid values: • eQMI_LOC_SENSOR_TIME_SOURCE_UNSPECIFIED (0) – Sensor time source is unspecified • eQMI_LOC_SENSOR_TIME_SOURCE_COMMON (1) – Time source is common between the sensors and the location engine
Type	0x02			1	Pedometer Report Timestamp
Length	4			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	uint32	timestamp	4	Time stamp of the last step event in this report, i.e., the time stamp of the step event that caused this report to be generated. The time stamp is in the time reference scale that is used by the pedometer time source. • Units: Milliseconds
Type	0x03			1	Time Interval
Length	4			2	
Value	→	uint32	timeInterval	4	Time interval during which the step count was calculated. Subtracting timeInterval from the timestamp field yields the time when the step detection for the first step in this report started. • Units: Milliseconds
Type	0x04			1	Step Count
Length	4			2	
Value	→	uint32	stepCount	4	Number of steps counted during the time interval.

Optional TLVs

Name	Version introduced	Version last modified
Step Confidence	2.17	2.17
Step Count Uncertainty	2.17	2.17
Step Rate	2.17	2.17

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Step Confidence
Length	1			2	
Value	→	uint8	stepConfidence	1	Confidence associated with the step. This field is only applicable for a single step report, i.e., if the step count is one. • Range: 0 to 100 Note: The report is ignored if confidence is 0.
Type	0x11			1	Step Count Uncertainty
Length	4			2	
Value	→	float	stepCountUncertainty	4	Uncertainty (in steps) associated with the step count.
Type	0x12			1	Step Rate
Length	4			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	float	stepRate	4	Current estimate for the rate of steps per second. • Units: steps/second • Range: >= 0.0

3.83.2 Indication - QMI_LOC_PEDOMETER_REPORT_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version introduced	Version last modified
Status of Pedometer Report Request	2.17	2.28

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Status of Pedometer Report Request
Length	4			2	
Value	→	enum	status	4	Status of the Pedometer Report request. Valid values: • eQMI_LOC_SUCCESS (0) – Request was completed successfully • eQMI_LOC_GENERAL_FAILURE (1) – Request failed because of a general failure • eQMI_LOC_UNSUPPORTED (2) – Request failed because it is not supported • eQMI_LOC_INVALID_PARAMETER (3) – Request failed because it contained invalid parameters • eQMI_LOC_ENGINE_BUSY (4) – Request failed because the engine is busy • eQMI_LOC_PHONE_OFFLINE (5) – Request failed because the phone is offline • eQMI_LOC_TIMEOUT (6) – Request failed because it timed out • eQMI_LOC_CONFIG_NOT_SUPPORTED (7) – Request failed because an undefined configuration was requested

Field	Field value	Field type	Parameter	Size (byte)	Description
			status (cont.)		<ul style="list-style-type: none"> • eQMI_LOC_INSUFFICIENT_MEMORY (8) – Request failed because the engine could not allocate sufficient memory for the request • eQMI_LOC_MAX_GEOFENCE_PROGRAMMED (9) – Request failed because the maximum number of Geofences are already programmed • eQMI_LOC_XTRA_VERSION_CHECK_FAILURE (10) – Location service failed because of an XTRA version-based file format check failure

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

3.83.3 Description of QMI_LOC_PEDOMETER_REPORT

This command is used to send pedometer data to the location engine. The pedometer data must only be sent if the location service indicates its readiness to accept pedomter data in the QMI_LOC_EVENT_PEDOMETER_CONTROL_IND_V02 indication.

3.84 QMI_LOC_INJECT_WCDMA_CELL_INFO

Injects WCDMA cell information into the location engine.

LOC message ID

0x0070

Version introduced

Major - 2, Minor - 18

3.84.1 Request - QMI_LOC_INJECT_WCDMA_CELL_INFO_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

Name	Version introduced	Version last modified
WCDMA Cell ID	2.18	2.18
Roaming Status	2.18	2.18

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	WCDMA Cell ID Identifies the WCDMA cell on which the device is currently camped.
Length	12			2	
Value	→	uint32	mcc	4	WCDMA mobile country code. Refer to ITU-T E.212 [S6].
		uint32	mnc	4	WCDMA mobile network code. Refer to [S6].
		uint32	cid	4	WCDMA cell identity. Refer to [S6].
Type	0x02			1	Roaming Status
Length	4			2	
Value	→	enum	roamingStatus	4	Indicates whether the device is roaming. Valid values: • eQMI_LOC_PHONE_NOT_ROAMING (1) – Modem is camped on a home network • eQMI_LOC_PHONE_ROAMING (2) – Modem is camped on a roaming network

Optional TLVs

Name	Version introduced	Version last modified
Cell Frequency	2.18	2.18
Primary Scrambling Code	2.18	2.18

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Cell Frequency
Length	4			2	
Value	→	uint32	freq	4	Frequency information of the serving cell. Valid range: 0 to 16383 Refer to TS 25.331 [S7].
Type	0x11			1	Primary Scrambling Code
Length	4			2	
Value	→	uint32	psc	4	Primary scrambling code of the serving cell. Valid range: 0 to 511 Refer to [S7].

3.84.2 Indication - QMI_LOC_INJECT_WCDMA_CELL_INFO_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version introduced	Version last modified
Inject WCDMA Cell Info Status	2.18	2.28

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Inject WCDMA Cell Info Status
Length	4			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	enum	status	4	<p>Status of the Inject WCDMA Cell Info request.</p> <p>Valid values:</p> <ul style="list-style-type: none"> • eQMI_LOC_SUCCESS (0) – Request was completed successfully • eQMI_LOC_GENERAL_FAILURE (1) – Request failed because of a general failure • eQMI_LOC_UNSUPPORTED (2) – Request failed because it is not supported • eQMI_LOC_INVALID_PARAMETER (3) – Request failed because it contained invalid parameters • eQMI_LOC_ENGINE_BUSY (4) – Request failed because the engine is busy • eQMI_LOC_PHONE_OFFLINE (5) – Request failed because the phone is offline • eQMI_LOC_TIMEOUT (6) – Request failed because it timed out • eQMI_LOC_CONFIG_NOT_SUPPORTED (7) – Request failed because an undefined configuration was requested • eQMI_LOC_INSUFFICIENT_MEMORY (8) – Request failed because the engine could not allocate sufficient memory for the request • eQMI_LOC_MAX_GEOFENCE_PROGRAMMED (9) – Request failed because the maximum number of Geofences are already programmed • eQMI_LOC_XTRA_VERSION_CHECK_FAILURE (10) – Location service failed because of an XTRA version-based file format check failure

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

3.84.3 Description of QMI_LOC_INJECT_WCDMA_CELL_INFO

This command is used to inject WCDMA cell information into the location engine. This command is to be called by the client when the cell information or the roaming status of the device changes for a WCDMA network. The information is used by the location engine for assisted GNSS. Depending on the modem configuration, the roaming status of the modem is used to decide whether the engine operates in AGNSS mode or Stand Alone mode.

If the device supports more than one service at a time, e.g., a Dual Service - Dual Standby (DS-DS) type of device, only the cell information of the user-designated preferred data service is injected.

3.85 QMI_LOC_INJECT_TDSCDMA_CELL_INFO

Injects TDSCDMA cell information into the location engine.

LOC message ID

0x0071

Version introduced

Major - 2, Minor - 18

3.85.1 Request - QMI_LOC_INJECT_TDSCDMA_CELL_INFO_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

Name	Version introduced	Version last modified
TDSCDMA Cell ID	2.18	2.18
Roaming Status	2.18	2.18

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	TDSCDMA Cell ID Identifies the TDSCDMA cell on which the device is currently camped.
Length	16			2	
Value	→	uint32	mcc	4	TDSCDMA mobile country code. Refer to ITU-T E.212 [S6].
		uint32	mnc	4	TDSCDMA mobile network code. Refer to [S6].
		uint32	cid	4	TDSCDMA cell identity. Refer to TS 25.331 [S7].
		uint32	lac	4	TDSCDMA location area code. Refer to [S6].
Type	0x02			1	Roaming Status
Length	4			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	enum	roamingStatus	4	Indicates whether the device is roaming. Valid values: • eQMI_LOC_PHONE_NOT_ROAMING (1) – Modem is camped on a home network • eQMI_LOC_PHONE_ROAMING (2) – Modem is camped on a roaming network

Optional TLVs

Name	Version introduced	Version last modified
Cell Frequency	2.18	2.18

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Cell Frequency
Length	4			2	
Value	→	uint32	freq	4	Frequency information of the serving cell. Valid range: 0 to 16383 Refer to [S7].

3.85.2 Indication - QMI_LOC_INJECT_TDSCDMA_CELL_INFO_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version introduced	Version last modified
Inject TDSCDMA Cell Info Status	2.18	2.28

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Inject TDSCDMA Cell Info Status
Length	4			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	enum	status	4	<p>Status of the Inject TDSCDMA Cell Info request.</p> <p>Valid values:</p> <ul style="list-style-type: none"> • eQMI_LOC_SUCCESS (0) – Request was completed successfully • eQMI_LOC_GENERAL_FAILURE (1) – Request failed because of a general failure • eQMI_LOC_UNSUPPORTED (2) – Request failed because it is not supported • eQMI_LOC_INVALID_PARAMETER (3) – Request failed because it contained invalid parameters • eQMI_LOC_ENGINE_BUSY (4) – Request failed because the engine is busy • eQMI_LOC_PHONE_OFFLINE (5) – Request failed because the phone is offline • eQMI_LOC_TIMEOUT (6) – Request failed because it timed out • eQMI_LOC_CONFIG_NOT_SUPPORTED (7) – Request failed because an undefined configuration was requested • eQMI_LOC_INSUFFICIENT_MEMORY (8) – Request failed because the engine could not allocate sufficient memory for the request • eQMI_LOC_MAX_GEOFENCE_PROGRAMMED (9) – Request failed because the maximum number of Geofences are already programmed • eQMI_LOC_XTRA_VERSION_CHECK_FAILURE (10) – Location service failed because of an XTRA version-based file format check failure

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

3.85.3 Description of QMI_LOC_INJECT_TDSCDMA_CELL_INFO

This command is used to inject TDSCDMA cell information into the location engine. This command is to be called by the client when the cell information or the roaming status of the device changes for a TDSCDMA network. The information is used by the location engine for assisted GNSS. Depending on the modem configuration, the roaming status of the modem is used to decide whether the engine operates in AGNSS mode or Stand Alone mode.

If the device supports more than one service at a time, e.g., a DS-DS type of device, only the cell information of the user-designated preferred data service is injected.

3.86 QMI_LOC_INJECT_SUBSCRIBER_ID

Injects the phone's subscriber ID into the location engine.

LOC message ID

0x0072

Version introduced

Major - 2, Minor - 18

3.86.1 Request - QMI_LOC_INJECT_SUBSCRIBER_ID

Message type

Request

Sender

Control point

Mandatory TLVs

None

Optional TLVs

Name	Version introduced	Version last modified
Preferred IMSI	2.18	2.18
Preferred MSISDN	2.18	2.18

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Preferred IMSI
Length	8			2	
Value	→	uint64	preferredIMSI	8	IMSI number of the preferred RAT. Refer to [S6].
Type	0x11			1	Preferred MSISDN
Length	8			2	
Value	→	uint64	preferredMSISDN	8	MSISDN number of the preferred RAT. Refer to [S6].

3.86.2 Indication - QMI_LOC_INJECT_SUBSCRIBER_ID_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version introduced	Version last modified
Inject Subscriber ID Status	2.18	2.28

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Inject Subscriber ID Status
Length	4			2	
Value	→	enum	status	4	<p>Status of the Inject Subscriber ID request.</p> <p>Valid values:</p> <ul style="list-style-type: none"> • eQMI_LOC_SUCCESS (0) – Request was completed successfully • eQMI_LOC_GENERAL_FAILURE (1) – Request failed because of a general failure • eQMI_LOC_UNSUPPORTED (2) – Request failed because it is not supported • eQMI_LOC_INVALID_PARAMETER (3) – Request failed because it contained invalid parameters • eQMI_LOC_ENGINE_BUSY (4) – Request failed because the engine is busy • eQMI_LOC_PHONE_OFFLINE (5) – Request failed because the phone is offline • eQMI_LOC_TIMEOUT (6) – Request failed because it timed out • eQMI_LOC_CONFIG_NOT_SUPPORTED (7) – Request failed because an undefined configuration was requested • eQMI_LOC_INSUFFICIENT_MEMORY (8) – Request failed because the engine could not allocate sufficient memory for the request • eQMI_LOC_MAX_GEOFENCE_PROGRAMMED (9) – Request failed because the maximum number of Geofences are already programmed • eQMI_LOC_XTRA_VERSION_CHECK_FAILURE (10) – Location service failed because of an XTRA version-based file format check failure

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

3.86.3 Description of QMI_LOC_INJECT_SUBSCRIBER_ID

This command is used to inject the subscriber ID of the preferred RAT into the location engine. This command is to be called by the client when the preferred RAT changes or the subscriber ID itself changes. This information is used by the location engine for assisted TDSCDMA. If the device is a DS-DS type of device, only the subscriber ID information of the user-designated preferred data service is injected.

3.87 QMI_LOC_SET_GEOFENCE_ENGINE_CONFIG

Used by the control point to set the Geofence engine configuration.

LOC message ID

0x0073

Version introduced

Major - 2, Minor - 23

3.87.1 Request - QMI_LOC_SET_GEOFENCE_ENGINE_CONFIG_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

Name	Version introduced	Version last modified
Transaction ID	2.23	2.23

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Transaction ID
Length	4			2	
Value	→	uint32	transactionId	4	Identifies the transaction. The transaction ID is returned with the Set Geofence Configuration indication.

Optional TLVs

Name	Version introduced	Version last modified
GNSS Unavailable Indication Timeout	2.23	2.23
Max Geofences	2.23	2.23
Enable Motion Detection Sources	2.23	2.23
Enable Coarse Position Injection Usage	2.23	2.23
GNSS Position QOS Session Timeout	2.23	2.23
GNSS Position Maximum Position Uncertainty Acceptable	2.30	2.30
Medium Responsiveness Value	2.30	2.30
Challenging GNSS Environment Minimum CPI Wait Interval	2.30	2.30
Geofence Motion State Information	2.30	2.30

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	GNSS Unavailable Indication Timeout
Length	4			2	
Value	→	uint32	gnssUnavailableIndicationTimeout	4	<p>In a bad GNSS environment, this is the timeout after which the Geofence engine sends out a GNSS Unavailable indication. The GNSS Unavailable indication is sent under the following conditions:</p> <ul style="list-style-type: none"> • If gnssUnavailableIndicationTimeout is less than gnssPositionSessionTimeout, the GNSS Unavailable timeout indication is sent after gnssPositionSessionTimeout expires • If gnssPositionSessionTimeout is less than gnssUnavailableIndicationTimeout, the GNSS Unavailable timeout indication is sent after gnssUnavailableIndicationTimeout expires
Type	0x11			1	Max Geofences
Length	4			2	
Value	→	uint32	maxGeofences	4	<p>Identifies the maximum number of Geofences that can be supported by the Geofence engine. If this number is less than the currently deployed Geofences, this command fails.</p> <p>If the command succeeds, the engine supports the maximum number of Geofences requested, provided there is enough memory to support that many Geofences. Increasing this value to a very large number in a constrained memory environment might affect other modules negatively. This value is determined by phone manufacturers. The default value is 200.</p>
Type	0x12			1	Enable Motion Detection Sources
Length	4			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	mask32	enableMotionDetectionSources	4	Identifies the sources that can be enabled for motion detection by the Geofence engine. The sources of motion detection that are enabled by the Geofence engine are dependent on the platform. These sources can only be set once at boot time and they are not expected to be changed after that. Any attempt to set the value of the motion detection sources at runtime results in an undefined behavior. Valid values: <ul style="list-style-type: none"> • QMI_LOC_MOTION_DETECTION_SOURCE_SENSORS (0x00000001) – Sensors are used for motion detection • QMI_LOC_MOTION_DETECTION_SOURCE_WIFI (0x00000002) – Wi-Fi is used for motion detection • QMI_LOC_MOTION_DETECTION_SOURCE_WWAN (0x00000004) – Wireless WAN is used for motion detection
Type	0x13			1	Enable Coarse Position Injection Usage
Length	1			2	
Value	→	boolean	enableCpiUsage	1	Indicates whether external Coarse Position Injection (CPI) is used by the Geofence engine. <ul style="list-style-type: none"> • 0x01 (TRUE) – CPI is enabled (default) • 0x00 (FALSE) – CPI is disabled
Type	0x14			1	GNSS Position QOS Session Timeout
Length	4			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	uint32	gnssPositionSessionTimeout	4	Identifies the session timeout value (in seconds) for requesting a position in a bad GNSS environment. Valid values: <ul style="list-style-type: none"> If the gnssUnavailableIndicationTimeout value is less than gnssPositionSessionTimeout, in a bad GNSS environment, the GNSS Unavailable timeout indication is sent after gnssPositionSessionTimeout expires. If gnssPositionSessionTimeout is less than gnssUnavailableIndicationTimeout, in a bad GNSS environment, the GNSS Unavailable timeout indication is sent after gnssUnavailableIndicationTimeout expires.
Type	0x15			1	GNSS Position Maximum Position Uncertainty Acceptable
Length	4			2	
Value	→	uint32	gnssPositionMaxPuncAcceptable	4	GNSS maximum position uncertainty in meters acceptable by the Geofence engine. Valid values: <ul style="list-style-type: none"> All positive values
Type	0x16			1	Medium Responsiveness Value
Length	4			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	uint32	mediumResponsiveness Value	4	Medium responsiveness value in seconds that the Geofence engine uses for all medium responsiveness Geofences in the Geofence engine. Valid values: <ul style="list-style-type: none"> • Positive values (in seconds) • If the value is configured for less than 30 sec, the value is set at 30 sec • If the value is configured for more than 600 sec, the value is set at 600 sec • Default – The Geofence engine uses 120 sec as the medium responsiveness value If the medium responsiveness value is changed, the responsiveness of the existing medium responsiveness Geofence does not change until the next position fix, which is based on the previous medium responsiveness setting.
Type	0x17			1	Challenging GNSS Environment Minimum CPI Wait Interval
Length	4			2	
Value	→	uint32	chalGnssEnvMinCpiWait Interval	4	Number of seconds that the Geofence engine is to wait between CPI requests in challenging a GNSS environment. Valid values: <ul style="list-style-type: none"> • Positive values (in seconds)
Type	0x18			1	Geofence Motion State Information Motion state informatino (e.g., motion state speed) that the Geofence engine is to use.
Length	Var			2	
Value	→	uint8	motionStateInfo_len	1	Number of sets of the following elements: <ul style="list-style-type: none"> • motionState • motionStateSpeed
		enum	motionState	4	Motion state for which information is being configured.

Field	Field value	Field type	Parameter	Size (byte)	Description
		float	motionStateSpeed	4	<p>Motion state speed in milliseconds. These are positive floating values. The state speed must be configured carefully. Very low speed configuration for a state may result in missing Geofence breaches in some scenarios.</p> <p>Typical motion state speeds:</p> <ul style="list-style-type: none"> • Stationary speed – 0 meters/sec • Fiddle speed – 0 meters/sec • Walk speed – 3 meters/sec • Run speed – 8 meters/sec • Drive speed – 56 meters/sec

3.87.2 Indication - QMI_LOC_SET_GEOFENCE_ENGINE_CONFIG_IND

Message type

Indication

Sender

Control point

Mandatory TLVs

Name	Version introduced	Version last modified
Set Geofence Engine Configuration Status.	2.23	2.28

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Set Geofence Engine Configuration Status.
Length	4			2	
Value	→	enum	status	4	<p>Status of the Set Geofence Engine Configuration request.</p> <p>Valid values:</p> <ul style="list-style-type: none"> • eQMI_LOC_SUCCESS (0) – Request was completed successfully • eQMI_LOC_GENERAL_FAILURE (1) – Request failed because of a general failure • eQMI_LOC_UNSUPPORTED (2) – Request failed because it is not supported • eQMI_LOC_INVALID_PARAMETER (3) – Request failed because it contained invalid parameters

Field	Field value	Field type	Parameter	Size (byte)	Description
			status (cont.)		<ul style="list-style-type: none"> • eQMI_LOC_ENGINE_BUSY (4) – Request failed because the engine is busy • eQMI_LOC_PHONE_OFFLINE (5) – Request failed because the phone is offline • eQMI_LOC_TIMEOUT (6) – Request failed because it timed out • eQMI_LOC_CONFIG_NOT_SUPPORTED (7) – Request failed because an undefined configuration was requested • eQMI_LOC_INSUFFICIENT_MEMORY (8) – Request failed because the engine could not allocate sufficient memory for the request • eQMI_LOC_MAX_GEOFENCE_PROGRAMMED (9) – Request failed because the maximum number of Geofences are already programmed • eQMI_LOC_XTRA_VERSION_CHECK_FAILURE (10) – Location service failed because of an XTRA version-based file format check failure

Optional TLVs

Name	Version introduced	Version last modified
Transaction ID	2.23	2.23

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Transaction ID
Length	4			2	
Value	→	uint32	transactionId	4	Transaction ID that was specified in the Set Geofence Configuration request. This parameter is always present if the status field is set to SUCCESS.

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

3.87.3 Description of QMI_LOC_SET_GEOFENCE_ENGINE_CONFIG

This command is used by the client to set the configuration of the Geofence engine. The command can be used to set only one configuration parameter at a time. Any attempt to set more than one parameter results in an error. Multiple clients must not set configurations that conflict with each other, since these impact the global state of the Geofence engine.

3.88 QMI_LOC_GET_GEOFENCE_ENGINE_CONFIG

Used by the control point to get the Geofence engine configuration.

LOC message ID

0x0074

Version introduced

Major - 2, Minor - 23

3.88.1 Request - QMI_LOC_GET_GEOFENCE_ENGINE_CONFIG_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

Name	Version introduced	Version last modified
Transaction ID	2.23	2.23

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Transaction ID
Length	4			2	
Value	→	uint32	transactionId	4	Identifies the transaction. The transaction ID is returned with the Get Geofence Engine Configuration indication.

Optional TLVs

None

3.88.2 Indication - QMI_LOC_GET_GEOFENCE_ENGINE_CONFIG_IND

Message type

Indication

Sender

Control point

Mandatory TLVs

Name	Version introduced	Version last modified
Get Geofence Engine Configuration Status	2.23	2.28

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Get Geofence Engine Configuration Status
Length	4			2	
Value	→	enum	status	4	<p>Status of the Get Geofence Engine Configuration request.</p> <p>Valid values:</p> <ul style="list-style-type: none"> • eQMI_LOC_SUCCESS (0) – Request was completed successfully • eQMI_LOC_GENERAL_FAILURE (1) – Request failed because of a general failure • eQMI_LOC_UNSUPPORTED (2) – Request failed because it is not supported • eQMI_LOC_INVALID_PARAMETER (3) – Request failed because it contained invalid parameters • eQMI_LOC_ENGINE_BUSY (4) – Request failed because the engine is busy • eQMI_LOC_PHONE_OFFLINE (5) – Request failed because the phone is offline • eQMI_LOC_TIMEOUT (6) – Request failed because it timed out • eQMI_LOC_CONFIG_NOT_SUPPORTED (7) – Request failed because an undefined configuration was requested • eQMI_LOC_INSUFFICIENT_MEMORY (8) – Request failed because the engine could not allocate sufficient memory for the request • eQMI_LOC_MAX_GEOFENCE_PROGRAMMED (9) – Request failed because the maximum number of Geofences are already programmed • eQMI_LOC_XTRA_VERSION_CHECK_FAILURE (10) – Location service failed because of an XTRA version-based file format check failure

Optional TLVs

Name	Version introduced	Version last modified
Transaction ID	2.23	2.23
GPS Unavailable Indication Timeout	2.23	2.23
Max Geofences	2.23	2.23
Enabled Motion Detection Sources	2.23	2.23
Enabled for CPI Position Injection Usage	2.23	2.23

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Transaction ID
Length	4			2	
Value	→	uint32	transactionId	4	Transaction ID that was specified in the Get Geofence Engine Configuration request. This parameter is always present if the status field is set to SUCCESS.
Type	0x11			1	GPS Unavailable Indication Timeout
Length	4			2	
Value	→	uint32	gnssUnavailableIndicationTimeout	4	In a bad GNSS environment, the timeout after which the Geofence engine sends out a GNSS unavailable indication.
Type	0x12			1	Max Geofences
Length	4			2	
Value	→	uint32	maxGeofences	4	Identifies the maximum number of Geofences that are currently supported in the Geofence engine.
Type	0x13			1	Enabled Motion Detection Sources
Length	4			2	
Value	→	mask32	enabledMotionDetectionSources	4	Identifies the sources that are currently enabled for motion detection by the Geofence engine. Valid values: <ul style="list-style-type: none"> • QMI_LOC_MOTION_DETECTION_SOURCE_SENSORS (0x00000001) – Sensors are used for motion detection • QMI_LOC_MOTION_DETECTION_SOURCE_WIFI (0x00000002) – Wi-Fi is used for motion detection • QMI_LOC_MOTION_DETECTION_SOURCE_WWAN (0x00000004) – Wireless WAN is used for motion detection
Type	0x14			1	Enabled for CPI Position Injection Usage
Length	1			2	
Value	→	boolean	enabledCpiUsage	1	Indicates whether CPI usage is enabled. <ul style="list-style-type: none"> • 0x01 (TRUE) – CPI usage is enabled • 0x00 (FALSE) – CPI usage 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

3.88.3 Description of QMI_LOC_GET_GEOFENCE_ENGINE_CONFIG

This command is used by the client to get the configuration of the Geofence engine. It is safe for multiple clients to use this command, however, only one request can be outstanding at any time.

3.89 QMI_LOC_GET_BATCH_SIZE

Used by the control point to get the batching size.

LOC message ID

0x0075

Version introduced

Major - 2, Minor - 24

3.89.1 Request - QMI_LOC_GET_BATCH_SIZE_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

Name	Version introduced	Version last modified
Transaction ID	2.24	2.24
Requested Batch Size	2.24	2.24

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Transaction ID
Length	4			2	
Value	→	uint32	transactionId	4	Identifies the transaction. The transaction ID is returned in the Get Batch Size indication.
Type	0x02			1	Requested Batch Size
Length	4			2	
Value	→	uint32	batchSize	4	Request the service with the number of location fixes to be batched.

Optional TLVs

None

3.89.2 Indication - QMI_LOC_GET_BATCH_SIZE_IND

Message type

Indication

Sender

Control point

Mandatory TLVs

Name	Version introduced	Version last modified
Get Batch Size Status	2.24	2.28
Transaction ID	2.24	2.24
Batch Size Supported	2.24	2.24

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Get Batch Size Status
Length	4			2	
Value	→	enum	status	4	<p>Status of the Get Batch Size request.</p> <p>Valid values:</p> <ul style="list-style-type: none"> • eQMI_LOC_SUCCESS (0) – Request was completed successfully • eQMI_LOC_GENERAL_FAILURE (1) – Request failed because of a general failure • eQMI_LOC_UNSUPPORTED (2) – Request failed because it is not supported • eQMI_LOC_INVALID_PARAMETER (3) – Request failed because it contained invalid parameters • eQMI_LOC_ENGINE_BUSY (4) – Request failed because the engine is busy • eQMI_LOC_PHONE_OFFLINE (5) – Request failed because the phone is offline • eQMI_LOC_TIMEOUT (6) – Request failed because it timed out • eQMI_LOC_CONFIG_NOT_SUPPORTED (7) – Request failed because an undefined configuration was requested • eQMI_LOC_INSUFFICIENT_MEMORY (8) – Request failed because the engine could not allocate sufficient memory for the request

Field	Field value	Field type	Parameter	Size (byte)	Description
			status (cont.)		<ul style="list-style-type: none"> • eQMI_LOC_MAX_GEOFENCE_PROGRAMMED (9) – Request failed because the maximum number of Geofences are already programmed • eQMI_LOC_XTRA_VERSION_CHECK_FAILURE (10) – Location service failed because of an XTRA version-based file format check failure
Type	0x02			1	Transaction ID
Length	4			2	
Value	→	uint32	transactionId	4	Transaction ID that was specified in the Get Batch Size request.
Type	0x03			1	Batch Size Supported
Length	4			2	
Value	→	uint32	batchSize	4	Number of location fixes that the service is able to batch. The batch size value is returned as 0 in the case of a failure status.

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

3.89.3 Description of QMI_LOC_GET_BATCH_SIZE

This command is used by the control point to allocate the buffer at the service side that is required for the batching operation. The batch size conveys the number of location fixes to be batched at the service. The service allocates the memory for the requested batch size, or a lesser value based on the memory availability. The batch size in the indication conveys the number of location fixes that the service is capable of handling.

It is the responsibility of the control point to configure the appropriate value of the batch size depending on the memory configuration on the product and the reporting interval that is being requested. If a control point chooses a large batch size value, any other control point requesting the batching operation, as well as overall system performance, may be affected.

It is the responsibility of the control point to release the allocated memory at the service using QMI_LOC_RELEASE_BATCH.

It is safe for multiple clients to use this command. The service can handle simultaneous batching operation for multiple clients.

3.90 QMI_LOC_START_BATCHING

Used by the control point to initiate a batching session.

LOC message ID

0x0076

Version introduced

Major - 2, Minor - 24

3.90.1 Request - QMI_LOC_START_BATCHING_REQ

Message type

Request

Sender

Control Point

Mandatory TLVs

None

Optional TLVs

Name	Version introduced	Version last modified
Minimum Interval Between Position Reports	2.24	2.24
Horizontal Accuracy Level	2.24	2.24
Fix Session Timeout Period	2.26	2.26

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Minimum Interval Between Position Reports
Length	4			2	
Value	→	uint32	minInterval	4	Minimum time interval, specified by the control point, that must elapse between position reports. • Units: milliseconds • Default: 60000 ms
Type	0x11			1	Horizontal Accuracy Level
Length	4			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	enum	horizontalAccuracyLevel	4	Specifies the horizontal accuracy level required by the control point. If not specified, accuracy defaults to LOW. Valid values: • eQMI_LOC_ACCURACY_LOW (1) – Low accuracy • eQMI_LOC_ACCURACY_MED (2) – Medium accuracy • eQMI_LOC_ACCURACY_HIGH (3) – High accuracy
Type	0x12			1	Fix Session Timeout Period
Length	4			2	
Value	→	uint32	fixSessionTimeout	4	Configures the fix session timeout duration. • Units: Milliseconds • Default: 20,000 ms

3.90.2 Indication - QMI_LOC_START_BATCHING_IND

Message type

Indication

Sender

Control point

Mandatory TLVs

Name	Version introduced	Version last modified
Start Batching Status	2.24	2.28

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Start Batching Status
Length	4			2	
Value	→	enum	status	4	Status of the Start Batching request. Valid values: • eQMI_LOC_SUCCESS (0) – Request was completed successfully • eQMI_LOC_GENERAL_FAILURE (1) – Request failed because of a general failure • eQMI_LOC_UNSUPPORTED (2) – Request failed because it is not supported

Field	Field value	Field type	Parameter	Size (byte)	Description
			status (cont.)		<ul style="list-style-type: none"> • eQMI_LOC_INVALID_PARAMETER (3) – Request failed because it contained invalid parameters • eQMI_LOC_ENGINE_BUSY (4) – Request failed because the engine is busy • eQMI_LOC_PHONE_OFFLINE (5) – Request failed because the phone is offline • eQMI_LOC_TIMEOUT (6) – Request failed because it timed out • eQMI_LOC_CONFIG_NOT_SUPPORTED (7) – Request failed because an undefined configuration was requested • eQMI_LOC_INSUFFICIENT_MEMORY (8) – Request failed because the engine could not allocate sufficient memory for the request • eQMI_LOC_MAX_GEOFENCE_PROGRAMMED (9) – Request failed because the maximum number of Geofences are already programmed • eQMI_LOC_XTRA_VERSION_CHECK_FAILURE (10) – Location service failed because of an XTRA version-based file format check failure

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

3.90.3 Description of QMI_LOC_START_BATCHING

This message starts a batching session at the service. The control point sends the batching configuration in this request.

Upon receiving this request, the service starts generating the position fixes and stores each position fix in the batching buffer. This is called the batching operation, or batching session.

The service notifies the control point with a Batch Full indication event if the control point registers. The control point must register for the QMI_LOC_EVENT_MASK_BATCH_FULL_NOTIFICATION event mask to receive notifications when the batching buffer is full. The service overwrites the oldest batched position report entries if the batch is full.

The service notifies the control point with live position report indications if the control point registers. The control point must register for the QMI_LOC_EVENT_MASK_LIVE_BATCHED_POSITION_REPORT event mask to receive live batched position reports. The service also stores these fix reports in the batching buffer.

The control point can also resend this message to the service to restart the ongoing batching session with an updated configuration. The service discards the previous configuration and starts to batch with the new configuration.

It is safe for multiple clients to use this command. The service can handle simultaneous batching operation for multiple clients.

3.91 QMI_LOC_EVENT_BATCH_FULL_NOTIFICATION

Used to notify the control point that the batched buffer is full.

LOC message ID

0x0077

Version introduced

Major - 2, Minor - 24

3.91.1 Indication - QMI_LOC_EVENT_BATCH_FULL_NOTIFICATION_IND

Message type

Indication

Sender

Control Point

Mandatory TLVs

Name	Version introduced	Version last modified
Number of Entries in the Batch During Full Event	2.24	2.24

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Number of Entries in the Batch During Full Event
Length	4			2	
Value	→	uint32	batchCount	4	Number of entries in the batch during a full event.

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

3.91.2 Description of QMI_LOC_EVENT_BATCH_FULL_NOTIFICATION

This event is used to notify the control point that the batching buffer is full for the ongoing batching session. The batch count field informs of the number of fixes available at the service in the batching buffer.

This notification is generated only if the control point has indicated its interest by setting the QMI_LOC_EVENT_MASK_BATCH_FULL_NOTIFICATION bitmask using QMI_LOC_REG_EVENTS_REQ.

A client can handle this notification from the service by retrieving the location fixes from the batch using the QMI_LOC_READ_FROM_BATCH_REQ message.

QUALCOMM
2016-05-16 00:17:18 PDT
deon_zhang@askey.com.tw

3.92 QMI_LOC_EVENT_LIVE_BATCHED_POSITION_REPORT

Used to notify the control point with the live batched position report.

LOC message ID

0x0078

Version introduced

Major - 2, Minor - 24

3.92.1 Indication - QMI_LOC_EVENT_LIVE_BATCHED_POSITION_REPORT_IND

Message type

Indication

Sender

Control Point

Mandatory TLVs

Name	Version introduced	Version last modified
Batched Position Report	2.24	2.24

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Batched Position Report Live position report that is also batched.
Length	87			2	
Value	→	uint32	fixId	4	Fix count for the session. The count starts at 0 and increments by one for each successive batched position report for a particular session.
		mask	validFields	8	Mask of all valid fields for this fix. Valid bitmasks: <ul style="list-style-type: none"> • QMI_LOC_BATCHED_REPORT_MASK_VALID_LATITUDE (0x00000001) – Latitude field is valid for this fix • QMI_LOC_BATCHED_REPORT_MASK_VALID_LONGITUDE (0x00000002) – Longitude field is valid for this fix

Field	Field value	Field type	Parameter	Size (byte)	Description
			validFields (cont.)		<ul style="list-style-type: none"> • QMI_LOC_BATCHED_REPORT_MASK_VALID_HOR_CIR_UNC (0x00000004) – Horizontal circular uncertainty field is valid for this fix • QMI_LOC_BATCHED_REPORT_MASK_VALID_SPEED_HOR (0x00000008) – Horizontal speed field is valid for this fix • QMI_LOC_BATCHED_REPORT_MASK_VALID_SPEED_UNC (0x00000010) – Speed uncertainty field is valid for this fix • QMI_LOC_BATCHED_REPORT_MASK_VALID_ALT_WRT_ELP (0x00000020) – Altitude with respect to ellipsoid field is valid for this fix • QMI_LOC_BATCHED_REPORT_MASK_VALID_SPEED_VER (0x00000040) – Vertical speed field is valid for this fix • QMI_LOC_BATCHED_REPORT_MASK_VALID_HEADING (0x00000080) – Heading field is valid for this fix • QMI_LOC_BATCHED_REPORT_MASK_VALID_HEADING_UNC (0x00000100) – Heading uncertainty field is valid for this fix • QMI_LOC_BATCHED_REPORT_MASK_VALID_TECH_MASK (0x00000200) – Technology source mask field is valid for this fix • QMI_LOC_BATCHED_REPORT_MASK_VALID_TIMESTAMP.UTC (0x00000400) – UTC timestamp field is valid for this fix • QMI_LOC_BATCHED_REPORT_MASK_VALID_TIME_UNC (0x00000800) – Time uncertainty field is valid for this fix • QMI_LOC_BATCHED_REPORT_MASK_VALID_MAGNETIC_DEV (0x00001000) – Magnetic deviation field is valid for this fix • QMI_LOC_BATCHED_REPORT_MASK_VALID_VERT_UNC (0x00002000) – Vertical uncertainty field is valid for this fix

Field	Field value	Field type	Parameter	Size (byte)	Description
			validFields (cont.)		<ul style="list-style-type: none"> • QMI_LOC_BATCHED_REPORT_MASK_VALID_HOR_CONF (0x00004000) – Horizontal confidence field is valid for this fix • QMI_LOC_BATCHED_REPORT_MASK_VALID_TIMESTAMP_GPS (0x00008000) – GPS timestamp field is valid for this fix
		double	latitude	8	Latitude (specified in WGS84 datum). <ul style="list-style-type: none"> • Type: Floating point • Units: Degrees • Range: -90.0 to 90.0 <ul style="list-style-type: none"> – Positive values indicate northern latitude – Negative values indicate southern latitude
		double	longitude	8	Longitude (specified in WGS84 datum). <ul style="list-style-type: none"> • Type: Floating point • Units: Degrees • Range: -180.0 to 180.0 <ul style="list-style-type: none"> – Positive values indicate eastern longitude – Negative values indicate western longitude
		float	horUncCircular	4	Horizontal position uncertainty (circular). <ul style="list-style-type: none"> • Units: Meters
		float	speedHorizontal	4	Horizontal speed. <ul style="list-style-type: none"> • Units: Meters/second
		float	speedUnc	4	3-D Speed uncertainty. <ul style="list-style-type: none"> • Units: Meters/second
		float	altitudeWrtEllipsoid	4	Altitude with respect to the WGS84 ellipsoid. <ul style="list-style-type: none"> • Units: Meters • Range: -500 to 15883
		float	speedVertical	4	Vertical speed. <ul style="list-style-type: none"> • Units: Meters/second
		float	heading	4	Heading. <ul style="list-style-type: none"> • Units: Degrees • Range: 0 to 359.999
		float	headingUnc	4	Heading uncertainty. <ul style="list-style-type: none"> • Units: Degrees • Range: 0 to 359.999

Field	Field value	Field type	Parameter	Size (byte)	Description
		mask32	technologyMask	4	Technology used in computing this fix. Valid bitmasks: <ul style="list-style-type: none"> • QMI_LOC_POS_TECH_MASK_SATELLITE (0x00000001) – Satellites were used to generate the fix • QMI_LOC_POS_TECH_MASK_CELLID (0x00000002) – Cell towers were used to generate the fix • QMI_LOC_POS_TECH_MASK_WIFI (0x00000004) – Wi-Fi access points were used to generate the fix • QMI_LOC_POS_TECH_MASK_SENSORS (0x00000008) – Sensors were used to generate the fix • QMI_LOC_POS_TECH_MASK_REFERENCE_LOCATION (0x00000010) – Reference Location was used to generate the fix • QMI_LOC_POS_TECH_MASK_INJECTED_COARSE_POSITION (0x00000020) – Coarse position injected into the location engine was used to generate the fix • QMI_LOC_POS_TECH_MASK_AFLT (0x00000040) – AFLT was used to generate the fix • QMI_LOC_POS_TECH_MASK_HYBRID (0x00000080) – GNSS and network-provided measurements were used to generate the fix
		uint64	timestampUtc	8	UTC timestamp. <ul style="list-style-type: none"> • Units: Milliseconds since Jan. 1, 1970
		float	timeUnc	4	Time uncertainty. <ul style="list-style-type: none"> • Units: Milliseconds
		float	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.
		float	vertUnc	4	Vertical uncertainty. <ul style="list-style-type: none"> • Units: Meters
		uint8	horConfidence	1	Horizontal confidence. <ul style="list-style-type: none"> • Units: Percent • Range: 0 to 99
		uint16	gpsWeek	2	Current GPS week as calculated from midnight, Jan. 6, 1980. <ul style="list-style-type: none"> • Units: Weeks

Field	Field value	Field type	Parameter	Size (byte)	Description
		uint32	gpsTimeOfWeekMs	4	Amount of time into the current GPS week. • Units: Milliseconds

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

3.92.2 Description of QMI_LOC_EVENT_LIVE_BATCHED_POSITION_REPORT

This event is used to notify the control point with the position report that is added to the ongoing batching session.

This notification is generated only if the control point has indicated its interest by setting the QMI_LOC_EVENT_MASK_LIVE_BATCHED_POSITION_REPORT bitmask using QMI_LOC_REG_EVENTS_REQ.

3.93 QMI_LOC_READ_FROM_BATCH

Used by the control point to retrieve fixes from the batch.

LOC message ID

0x0079

Version introduced

Major - 2, Minor - 24

3.93.1 Request - QMI_LOC_READ_FROM_BATCH_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

Name	Version introduced	Version last modified
Number of Fix Entries to be Retrieved from the Batch	2.24	2.24
Transaction ID	2.24	2.24

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Number of Fix Entries to be Retrieved from the Batch
Length	4			2	
Value	→	uint32	numberOfEntries	4	Number of fix entries to be retrieved from the batch. Maximum limit – QMI_LOC_READ_FROM_BATCH_MAX_SIZE.
Type	0x02			1	Transaction ID
Length	4			2	
Value	→	uint32	transactionId	4	Identifies the transaction. The transaction ID is returned in the Read from Batch indication.

Optional TLVs

None

3.93.2 Indication - QMI_LOC_READ_FROM_BATCH_IND

Message type

Indication

Sender

Control point

Mandatory TLVs

Name	Version introduced	Version last modified
Read from Batch Status	2.24	2.28
Transaction ID	2.24	2.24

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Read from Batch Status
Length	4			2	
Value	→	enum	status	4	<p>Status of the Read from Batch request.</p> <p>Valid values:</p> <ul style="list-style-type: none"> • eQMI_LOC_SUCCESS (0) – Request was completed successfully • eQMI_LOC_GENERAL_FAILURE (1) – Request failed because of a general failure • eQMI_LOC_UNSUPPORTED (2) – Request failed because it is not supported • eQMI_LOC_INVALID_PARAMETER (3) – Request failed because it contained invalid parameters • eQMI_LOC_ENGINE_BUSY (4) – Request failed because the engine is busy • eQMI_LOC_PHONE_OFFLINE (5) – Request failed because the phone is offline • eQMI_LOC_TIMEOUT (6) – Request failed because it timed out • eQMI_LOC_CONFIG_NOT_SUPPORTED (7) – Request failed because an undefined configuration was requested • eQMI_LOC_INSUFFICIENT_MEMORY (8) – Request failed because the engine could not allocate sufficient memory for the request

Field	Field value	Field type	Parameter	Size (byte)	Description
			status (cont.)		<ul style="list-style-type: none"> • eQMI_LOC_MAX_GEOFENCE_PROGRAMMED (9) – Request failed because the maximum number of Geofences are already programmed • eQMI_LOC_XTRA_VERSION_CHECK_FAILURE (10) – Location service failed because of an XTRA version-based file format check failure
Type	0x02			1	Transaction ID
Length	4			2	
Value	→	uint32	transactionId	4	Transaction ID that was specified in the Read from Batch request.

Optional TLVs

Name	Version introduced	Version last modified
Number of Fix Entries Returned from the Batch	2.24	2.24
List of Batched Position Reports Returned	2.24	2.24

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Number of Fix Entries Returned from the Batch
Length	4			2	
Value	→	uint32	numberOfEntries	4	Number of fix entries returned from the batch.
Type	0x11			1	List of Batched Position Reports Returned List of fix reports returned from the batch.
Length	Var			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	uint8	batchedReportList_len	1	Number of sets of the following elements: <ul style="list-style-type: none"> • fixId • validFields • latitude • longitude • horUncCircular • speedHorizontal • speedUnc • altitudeWrtEllipsoid • speedVertical • heading • headingUnc • technologyMask • timestampUtc • timeUnc • magneticDeviation • vertUnc • horConfidence • gpsWeek • gpsTimeOfWeekMs
		uint32	fixId	4	Fix count for the session. The count starts at 0 and increments by one for each successive batched position report for a particular session.
		mask	validFields	8	Mask of all valid fields for this fix. Valid bitmasks: <ul style="list-style-type: none"> • QMI_LOC_BATCHED_REPORT_MASK_VALID_LATITUDE (0x00000001) – Latitude field is valid for this fix • QMI_LOC_BATCHED_REPORT_MASK_VALID_LONGITUDE (0x00000002) – Longitude field is valid for this fix • QMI_LOC_BATCHED_REPORT_MASK_VALID_HOR_CIR_UNC (0x00000004) – Horizontal circular uncertainty field is valid for this fix • QMI_LOC_BATCHED_REPORT_MASK_VALID_SPEED_HOR (0x00000008) – Horizontal speed field is valid for this fix • QMI_LOC_BATCHED_REPORT_MASK_VALID_SPEED_UNC (0x00000010) – Speed uncertainty field is valid for this fix

Field	Field value	Field type	Parameter	Size (byte)	Description
			validFields (cont.)		<ul style="list-style-type: none"> • QMI_LOC_BATCHED_REPORT_MASK_VALID_ALT_WRT_ELP (0x00000020) – Altitude with respect to ellipsoid field is valid for this fix • QMI_LOC_BATCHED_REPORT_MASK_VALID_SPEED_VER (0x00000040) – Vertical speed field is valid for this fix • QMI_LOC_BATCHED_REPORT_MASK_VALID_HEADING (0x00000080) – Heading field is valid for this fix • QMI_LOC_BATCHED_REPORT_MASK_VALID_HEADING_UNC (0x00000100) – Heading uncertainty field is valid for this fix • QMI_LOC_BATCHED_REPORT_MASK_VALID_TECH_MASK (0x00000200) – Technology source mask field is valid for this fix • QMI_LOC_BATCHED_REPORT_MASK_VALID_TIMESTAMP.UTC (0x00000400) – UTC timestamp field is valid for this fix • QMI_LOC_BATCHED_REPORT_MASK_VALID_TIME_UNC (0x00000800) – Time uncertainty field is valid for this fix • QMI_LOC_BATCHED_REPORT_MASK_VALID_MAGNETIC_DEV (0x00001000) – Magnetic deviation field is valid for this fix • QMI_LOC_BATCHED_REPORT_MASK_VALID_VERT_UNC (0x00002000) – Vertical uncertainty field is valid for this fix • QMI_LOC_BATCHED_REPORT_MASK_VALID_HOR_CONF (0x00004000) – Horizontal confidence field is valid for this fix • QMI_LOC_BATCHED_REPORT_MASK_VALID_TIMESTAMP_GPS (0x00008000) – GPS timestamp field is valid for this fix

Field	Field value	Field type	Parameter	Size (byte)	Description
		double	latitude	8	Latitude (specified in WGS84 datum). <ul style="list-style-type: none"> • Type: Floating point • Units: Degrees • Range: -90.0 to 90.0 <ul style="list-style-type: none"> – Positive values indicate northern latitude – Negative values indicate southern latitude
		double	longitude	8	Longitude (specified in WGS84 datum). <ul style="list-style-type: none"> • Type: Floating point • Units: Degrees • Range: -180.0 to 180.0 <ul style="list-style-type: none"> – Positive values indicate eastern longitude – Negative values indicate western longitude
		float	horUncCircular	4	Horizontal position uncertainty (circular). <ul style="list-style-type: none"> • Units: Meters
		float	speedHorizontal	4	Horizontal speed. <ul style="list-style-type: none"> • Units: Meters/second
		float	speedUnc	4	3-D Speed uncertainty. <ul style="list-style-type: none"> • Units: Meters/second
		float	altitudeWrtEllipsoid	4	Altitude with respect to the WGS84 ellipsoid. <ul style="list-style-type: none"> • Units: Meters • Range: -500 to 15883
		float	speedVertical	4	Vertical speed. <ul style="list-style-type: none"> • Units: Meters/second
		float	heading	4	Heading. <ul style="list-style-type: none"> • Units: Degrees • Range: 0 to 359.999
		float	headingUnc	4	Heading uncertainty. <ul style="list-style-type: none"> • Units: Degrees • Range: 0 to 359.999
		mask32	technologyMask	4	Technology used in computing this fix. Valid bitmasks: <ul style="list-style-type: none"> • QMI_LOC_POS_TECH_MASK_SATELLITE (0x00000001) – Satellites were used to generate the fix • QMI_LOC_POS_TECH_MASK_CELLID (0x00000002) – Cell towers were used to generate the fix • QMI_LOC_POS_TECH_MASK_WIFI (0x00000004) – Wi-Fi access points were used to generate the fix

Field	Field value	Field type	Parameter	Size (byte)	Description
			technologyMask (cont.)		<ul style="list-style-type: none"> • QMI_LOC_POS_TECH_MASK_SENSORS (0x00000008) – Sensors were used to generate the fix • QMI_LOC_POS_TECH_MASK_REFERENCE_LOCATION (0x00000010) – Reference Location was used to generate the fix • QMI_LOC_POS_TECH_MASK_INJECTED_COARSE_POSITION (0x00000020) – Coarse position injected into the location engine was used to generate the fix • QMI_LOC_POS_TECH_MASK_AFLT (0x00000040) – AFLT was used to generate the fix • QMI_LOC_POS_TECH_MASK_HYBRID (0x00000080) – GNSS and network-provided measurements were used to generate the fix
		uint64	timestampUtc	8	UTC timestamp. • Units: Milliseconds since Jan. 1, 1970
		float	timeUnc	4	Time uncertainty. • Units: Milliseconds
		float	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.
		float	vertUnc	4	Vertical uncertainty. • Units: Meters
		uint8	horConfidence	1	Horizontal confidence. • Units: Percent • Range: 0 to 99
		uint16	gpsWeek	2	Current GPS week as calculated from midnight, Jan. 6, 1980. • Units: Weeks
		uint32	gpsTimeOfWeekMs	4	Amount of time into the current GPS week. • 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

3.93.3 Description of QMI_LOC_READ_FROM_BATCH

The control point sends this message to the service to read the location fix entries from the batching buffer. The control point can send this request to retrieve the entries asynchronously or when the service sends the event QMI_LOC_EVENT_MASK_BATCH_FULL_NOTIFICATION_IND.

The number of entries that can be retrieved from the service in each attempt is limited to QMI_LOC_READ_FROM_BATCH_MAX_SIZE fix entries. The control point must recursively perform the retrieve operation to empty the batch. If the control point sends this request when the batch is already empty, the service returns an indication with no entries.

If the batch at the service has fewer entries than the numberOfEntries value in the request from control point, the service only sends the fix entries that are available in the batch.

QUALCOMM
2016-05-16 00:17:18 PDT
deon_zhang@askey.com.tw

3.94 QMI_LOC_STOP_BATCHING

Used by the control point to stop an ongoing batching session.

LOC message ID

0x007A

Version introduced

Major - 2, Minor - 24

3.94.1 Request - QMI_LOC_STOP_BATCHING_REQ

Message type

Request

Sender

Control Point

Mandatory TLVs

Name	Version introduced	Version last modified
Transaction ID	2.24	2.24

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Transaction ID
Length	4			2	
Value	→	uint32	transactionId	4	Transaction ID of the request.

Optional TLVs

None

3.94.2 Indication - QMI_LOC_STOP_BATCHING_IND

Message type

Indication

Sender

Control point

Mandatory TLVs

Name	Version introduced	Version last modified
Stop Batching Status	2.24	2.28
Transaction ID	2.24	2.24

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Stop Batching Status
Length	4			2	
Value	→	enum	status	4	<p>Status of the Stop Batching request.</p> <p>Valid values:</p> <ul style="list-style-type: none"> • eQMI_LOC_SUCCESS (0) – Request was completed successfully • eQMI_LOC_GENERAL_FAILURE (1) – Request failed because of a general failure • eQMI_LOC_UNSUPPORTED (2) – Request failed because it is not supported • eQMI_LOC_INVALID_PARAMETER (3) – Request failed because it contained invalid parameters • eQMI_LOC_ENGINE_BUSY (4) – Request failed because the engine is busy • eQMI_LOC_PHONE_OFFLINE (5) – Request failed because the phone is offline • eQMI_LOC_TIMEOUT (6) – Request failed because it timed out • eQMI_LOC_CONFIG_NOT_SUPPORTED (7) – Request failed because an undefined configuration was requested • eQMI_LOC_INSUFFICIENT_MEMORY (8) – Request failed because the engine could not allocate sufficient memory for the request • eQMI_LOC_MAX_GEOFENCE_PROGRAMMED (9) – Request failed because the maximum number of Geofences are already programmed • eQMI_LOC_XTRA_VERSION_CHECK_FAILURE (10) – Location service failed because of an XTRA version-based file format check failure
Type	0x02			1	Transaction ID
Length	4			2	
Value	→	uint32	transactionId	4	Transaction ID that was specified in the Stop Batching request.

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

3.94.3 Description of QMI_LOC_STOP_BATCHING

The control point sends this request to stop an ongoing batching session at the service. The service stops generating location fixes and storing them in the batch. Any existing location fixes in the batch are unaffected until QMI_LOC_RELEASE_BATCH is requested.

It is safe for multiple clients to use this command. The service can handle simultaneous batching operation for multiple clients.

3.95 QMI_LOC_RELEASE_BATCH

Used by the control point to release the batching buffer.

LOC message ID

0x007B

Version introduced

Major - 2, Minor - 24

3.95.1 Request - QMI_LOC_RELEASE_BATCH_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

Name	Version introduced	Version last modified
Transaction ID	2.24	2.24

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Transaction ID
Length	4			2	
Value	→	uint32	transactionId	4	Identifies the transaction.

Optional TLVs

None

3.95.2 Indication - QMI_LOC_RELEASE_BATCH_IND

Message type

Indication

Sender

Control point

Mandatory TLVs

Name	Version introduced	Version last modified
Release Batch Status	2.24	2.28
Transaction ID	2.24	2.24

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Release Batch Status
Length	4			2	
Value	→	enum	status	4	<p>Status of the Release Batch request.</p> <p>Valid values:</p> <ul style="list-style-type: none"> • eQMI_LOC_SUCCESS (0) – Request was completed successfully • eQMI_LOC_GENERAL_FAILURE (1) – Request failed because of a general failure • eQMI_LOC_UNSUPPORTED (2) – Request failed because it is not supported • eQMI_LOC_INVALID_PARAMETER (3) – Request failed because it contained invalid parameters • eQMI_LOC_ENGINE_BUSY (4) – Request failed because the engine is busy • eQMI_LOC_PHONE_OFFLINE (5) – Request failed because the phone is offline • eQMI_LOC_TIMEOUT (6) – Request failed because it timed out • eQMI_LOC_CONFIG_NOT_SUPPORTED (7) – Request failed because an undefined configuration was requested • eQMI_LOC_INSUFFICIENT_MEMORY (8) – Request failed because the engine could not allocate sufficient memory for the request • eQMI_LOC_MAX_GEOFENCE_PROGRAMMED (9) – Request failed because the maximum number of Geofences are already programmed • eQMI_LOC_XTRA_VERSION_CHECK_FAILURE (10) – Location service failed because of an XTRA version-based file format check failure
Type	0x02			1	Transaction ID
Length	4			2	
Value	→	uint32	transactionId	4	Transaction ID that was specified in the Release Batch request.

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

3.95.3 Description of QMI_LOC_RELEASE_BATCH

This command is used by the control point to release the batching buffer that was allocated at the service. The control point must stop the batching session before sending this release request.

It is safe for multiple clients to use this command. The service can handle simultaneous batching operation for multiple clients.

3.96 QMI_LOC_EVENT_INJECT_WIFI_AP_DATA_REQ

Requests the control point to inject Wi-Fi AP data.

LOC message ID

0x007C

Version introduced

Major - 2, Minor - 24

3.96.1 Indication - QMI_LOC_EVENT_INJECT_WIFI_AP_DATA_REQ_IND

Message type

Indication

Sender

Service

Mandatory TLVs

None

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

3.96.2 Description of QMI_LOC_EVENT_INJECT_WIFI_AP_DATA_REQ

This event is used to request the control point to inject Wi-Fi AP 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. A client can satisfy this request from the service by sending the QMI_LOC_INJECT_WIFI_AP_DATA message.

It is not safe for multiple clients to inject data into the engine, hence only one client should try to handle this request.

3.97 QMI_LOC_INJECT_WIFI_AP_DATA

Injects Wi-Fi AP data.

LOC message ID

0x007D

Version introduced

Major - 2, Minor - 24

3.97.1 Request - QMI_LOC_INJECT_WIFI_AP_DATA_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

Name	Version introduced	Version last modified
Wi-Fi AP Scan Data	2.24	2.24

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Wi-Fi AP Scan Data List of Wi-Fi AP scan information entered by the control point.
Length	Var			2	
Value	→	uint8	wifiApInfo_len	1	Number of sets of the following elements: <ul style="list-style-type: none"> wifiApDataMask macAddress apTransmitPower apAntennaGain apSignalToNoise apDeviceType apRssi apChannel apRoundTripDelay apRoundTripDelayUnit apRoundTripDelayAccuracy mobileSignalToNoise mobileRssi

Field	Field value	Field type	Parameter	Size (byte)	Description
		mask32	wifiApDataMask	4	<p>Specifies which Wi-Fi AP scan information types are being used.</p> <p>Valid values:</p> <ul style="list-style-type: none"> • QMI_LOC_WIFI_APDATA_MASK_AP_TRANSMIT_POWER (0x00000001) – AP transmit power is valid • QMI_LOC_WIFI_APDATA_MASK_AP_ANTENNA_GAIN (0x00000002) – AP antenna gain is valid • QMI_LOC_WIFI_APDATA_MASK_AP_SNR (0x00000004) – AP signal-to-noise ratio is valid • QMI_LOC_WIFI_APDATA_MASK_AP_DEVICE_TYPE (0x00000008) – AP device type is valid • QMI_LOC_WIFI_APDATA_MASK_AP_RSSI (0x00000010) – AP RSSI is valid • QMI_LOC_WIFI_APDATA_MASK_AP_CHANNEL (0x00000020) – AP channel is valid • QMI_LOC_WIFI_APDATA_MASK_AP_ROUNDTRIP_DELAY (0x00000040) – AP roundtrip delay is valid • QMI_LOC_WIFI_APDATA_MASK_AP_ROUNDTRIP_DELAY_ACCURACY (0x00000080) – AP roundtrip delay accuracy is valid • QMI_LOC_WIFI_APDATA_MASK_MOBILE_SNR (0x00000100) – Mobile signal-to-noise ratio is valid • QMI_LOC_WIFI_APDATA_MASK_MOBILE_RSSI (0x00000200) – Mobile RSSI is valid
		uint8	macAddress	6	<p>MAC address.</p> <p>Each address is of length QMI_LOC_WIFI_MAC_ADDR_LENGTH.</p>
		int32	apTransmitPower	4	AP transmit power in dBm.
		int32	apAntennaGain	4	AP antenna gain in dBI.
		int32	apSignalToNoise	4	AP SNR received at the mobile device.
		enum	apDeviceType	4	List of AP device types.
		int32	apRssi	4	AP signal strength indicator in dBm.
		uint16	apChannel	2	AP Wi-Fi channel on which a beacon was received.

Field	Field value	Field type	Parameter	Size (byte)	Description
		uint32	apRoundTripDelay	4	Round trip delay between the mobile device and the AP, in units of apRoundTripDelayUnit.
		enum	apRoundTripDelayUnit	4	Units of apRoundTripDelay and its accuracy; mandatory if apRoundTripDelay is present.
		uint8	apRoundTripDelayAccuracy	1	AP's accuracy of round trip delay apRoundTripDelay, in units of apRoundTripDelayUnit.
		int32	mobileSignalToNoise	4	Mobile SNR received at the AP.
		int32	mobileRssi	4	Mobile signal strength at the AP.

Optional TLVs

None

3.97.2 Indication - QMI_LOC_INJECT_WIFI_AP_DATA_IND**Message type**

Indication

Sender

Service

Mandatory TLVs

Name	Version introduced	Version last modified
Wi-Fi AP Scan Information Injection Status	2.24	2.28

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Wi-Fi AP Scan Information Injection Status
Length	4			2	
Value	→	enum	status	4	Status of the Inject Wi-Fi AP Scan Information request. Valid values: <ul style="list-style-type: none"> • eQMI_LOC_SUCCESS (0) – Request was completed successfully • eQMI_LOC_GENERAL_FAILURE (1) – Request failed because of a general failure • eQMI_LOC_UNSUPPORTED (2) – Request failed because it is not supported

Field	Field value	Field type	Parameter	Size (byte)	Description
			status (cont.)		<ul style="list-style-type: none"> • eQMI_LOC_INVALID_PARAMETER (3) – Request failed because it contained invalid parameters • eQMI_LOC_ENGINE_BUSY (4) – Request failed because the engine is busy • eQMI_LOC_PHONE_OFFLINE (5) – Request failed because the phone is offline • eQMI_LOC_TIMEOUT (6) – Request failed because it timed out • eQMI_LOC_CONFIG_NOT_SUPPORTED (7) – Request failed because an undefined configuration was requested • eQMI_LOC_INSUFFICIENT_MEMORY (8) – Request failed because the engine could not allocate sufficient memory for the request • eQMI_LOC_MAX_GEOFENCE_PROGRAMMED (9) – Request failed because the maximum number of Geofences are already programmed • eQMI_LOC_XTRA_VERSION_CHECK_FAILURE (10) – Location service failed because of an XTRA version-based file format check failure

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

3.97.3 Description of QMI_LOC_INJECT_WIFI_AP_DATA

This command is called to inject Wi-Fi AP information.

It is safe for multiple clients to inject data into the engine.

3.98 QMI_LOC_NOTIFY_WIFI_ATTACHMENT_STATUS

Used by the control point to inject the Wi-Fi attachment status.

LOC message ID

0x007E

Version introduced

Major - 2, Minor - 24

3.98.1 Request - QMI_LOC_NOTIFY_WIFI_ATTACHMENT_STATUS_REQ

Message type

Request

Sender

Control Point

Mandatory TLVs

Name	Version introduced	Version last modified
Attach State	2.24	2.24

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Attach State
Length	4			2	
Value	→	enum	attachState	4	Wi-Fi access point attach state. Valid values: • eQMI_LOC_WIFI_ACCESS_POINT_ATTACHED (0) – Attached to an access point • eQMI_LOC_WIFI_ACCESS_POINT_DETACHED (1) – Detached from an access point • eQMI_LOC_WIFI_ACCESS_POINT_HANDOVER (2) – Handed over to another access point

Optional TLVs

Name	Version introduced	Version last modified
Access Point MAC Address	2.24	2.24
Wi-Fi AP SSID String	2.32	2.32

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Access Point MAC Address
Length	6			2	
Value	→	uint8	accessPointMacAddress	6	MAC address of the access point to which the Wi-Fi is attached. This must always be specified if the attach state is Handover.
Type	0x11			1	Wi-Fi AP SSID String
Length	Var			2	
Value	→	string	wifiApSsid	Var	The NULL-terminated SSID of the Wi-Fi AP. Its maximum length according to the ASCII standard is 32 octets.

3.98.2 Request - QMI_LOC_NOTIFY_WIFI_ATTACHMENT_STATUS_IND

Message type

Request

Sender

Control Point

Mandatory TLVs

Name	Version introduced	Version last modified
Status of Wi-Fi Attachment Status Request	2.24	2.28

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Status of Wi-Fi Attachment Status Request
Length	4			2	
Value	→	enum	status	4	Status of Wi-Fi Attachment Status request. Valid values: <ul style="list-style-type: none"> • eQMI_LOC_SUCCESS (0) – Request was completed successfully • eQMI_LOC_GENERAL_FAILURE (1) – Request failed because of a general failure • eQMI_LOC_UNSUPPORTED (2) – Request failed because it is not supported • eQMI_LOC_INVALID_PARAMETER (3) – Request failed because it contained invalid parameters

Field	Field value	Field type	Parameter	Size (byte)	Description
			status (cont.)		<ul style="list-style-type: none"> • eQMI_LOC_ENGINE_BUSY (4) – Request failed because the engine is busy • eQMI_LOC_PHONE_OFFLINE (5) – Request failed because the phone is offline • eQMI_LOC_TIMEOUT (6) – Request failed because it timed out • eQMI_LOC_CONFIG_NOT_SUPPORTED (7) – Request failed because an undefined configuration was requested • eQMI_LOC_INSUFFICIENT_MEMORY (8) – Request failed because the engine could not allocate sufficient memory for the request • eQMI_LOC_MAX_GEOFENCE_PROGRAMMED (9) – Request failed because the maximum number of Geofences are already programmed • eQMI_LOC_XTRA_VERSION_CHECK_FAILURE (10) – Location service failed because of an XTRA version-based file format check failure

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

3.98.3 Description of QMI_LOC_NOTIFY_WIFI_ATTACHMENT_STATUS REQ/RESP

This command is used by the control point to inform the location engine when Wi-Fi attaches to or detaches from an access point.

3.99 QMI_LOC_NOTIFY_WIFI_ENABLED_STATUS

Used by the control point to inject the Wi-Fi enabled status.

LOC message ID

0x007F

Version introduced

Major - 2, Minor - 24

3.99.1 Request - QMI_LOC_NOTIFY_WIFI_ENABLED_STATUS_REQ

Message type

Request

Sender

Control Point

Mandatory TLVs

Name	Version introduced	Version last modified
Enabled Status	2.24	2.24

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Enabled Status
Length	4			2	
Value	→	enum	enabledStatus	4	Wi-Fi enabled status on the device. Valid values: • eQMI_LOC_WIFI_ENABLED_FALSE (0) – Wi-Fi is disabled on the device • eQMI_LOC_WIFI_ENABLED_TRUE (1) – Wi-Fi is enabled on the device

Optional TLVs

None

3.99.2 Indication - QMI_LOC_NOTIFY_WIFI_ENABLED_STATUS_IND

Message type

Indication

Sender

Control Point

Mandatory TLVs

Name	Version introduced	Version last modified
Status of Wi-Fi Enabled Status Request	2.24	2.28

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Status of Wi-Fi Enabled Status Request
Length	4			2	
Value	→	enum	status	4	<p>Status of the Wi-Fi Enabled Status request.</p> <p>Valid values:</p> <ul style="list-style-type: none"> • eQMI_LOC_SUCCESS (0) – Request was completed successfully • eQMI_LOC_GENERAL_FAILURE (1) – Request failed because of a general failure • eQMI_LOC_UNSUPPORTED (2) – Request failed because it is not supported • eQMI_LOC_INVALID_PARAMETER (3) – Request failed because it contained invalid parameters • eQMI_LOC_ENGINE_BUSY (4) – Request failed because the engine is busy • eQMI_LOC_PHONE_OFFLINE (5) – Request failed because the phone is offline • eQMI_LOC_TIMEOUT (6) – Request failed because it timed out • eQMI_LOC_CONFIG_NOT_SUPPORTED (7) – Request failed because an undefined configuration was requested • eQMI_LOC_INSUFFICIENT_MEMORY (8) – Request failed because the engine could not allocate sufficient memory for the request

Field	Field value	Field type	Parameter	Size (byte)	Description
			status (cont.)		<ul style="list-style-type: none"> • eQMI_LOC_MAX_GEOFENCE_PROGRAMMED (9) – Request failed because the maximum number of Geofences are already programmed • eQMI_LOC_XTRA_VERSION_CHECK_FAILURE (10) – Location service failed because of an XTRA version-based file format check failure

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

3.99.3 Description of QMI_LOC_NOTIFY_WIFI_ENABLED_STATUS

This command is used by the control point to inform the location engine when Wi-Fi is turned off or turned on the device.

3.100 QMI_LOC_EVENT_GEOFENCE_BATCHED_BREACH_NOTIFICATION

Notifies the control point of a Geofence breach event by batching all the Geofences that were breached.

LOC message ID

0x0080

Version introduced

Major - 2, Minor - 24

3.100.1 Indication - QMI_LOC_EVENT_GEOFENCE_BATCHED_BREACH_NOTIFICATION_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version introduced	Version last modified
Geofence Breach Type	2.24	2.24

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Geofence Breach Type
Length	4			2	
Value	→	enum	breachType	4	Type of breach that generated this event. Valid values: • eQMI_LOC_GEOFENCE_BREACH_TYPE_ENTERING (1) – Denotes that a client entered the Geofence • eQMI_LOC_GEOFENCE_BREACH_TYPE_LEAVING (2) – Denotes that a client left the Geofence

Optional TLVs

Name	Version introduced	Version last modified
Geofence ID Continuous	2.24	2.24
Geofence ID Discrete	2.24	2.24
Geofence Position	2.24	2.24

Name	Version introduced	Version last modified
Geofence Breach Confidence	2.24	2.24
Heading Uncertainty	2.27	2.27
Vertical Uncertainty	2.27	2.27
Speed Uncertainty	2.27	2.27
Horizontal Confidence	2.27	2.27
Vertical Confidence	2.27	2.27
Dilution of Precision	2.27	2.27
SVs Used to Calculate the Fix	2.27	2.27

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Geofence ID Continuous Each entry in the list contains the continuous range of Geofence IDs that were breached at the same position. This list does not overlap with the discrete Geofence ID list.
Length	Var			2	
Value	→	uint8	geofenceIdContinuousList_len	1	Number of sets of the following elements: • idLow • idHigh
		uint32	idLow	4	Contains the starting ID of the Geofence in the range of the continuous range of Geofences that were breached at the same position.
		uint32	idHigh	4	Contains the ending ID of the Geofence in the range of the continuous range of Geofences that were breached at the same position.
Type	0x11			1	Geofence ID Discrete
Length	Var			2	
Value	→	uint8	geofenceIdDiscreteList_len	1	Number of sets of the following elements: • geofenceIdDiscreteList
		uint32	geofenceIdDiscreteList	Var	This list contains the Geofence IDs that were breached at the same position. This list does not overlap with the continuous Geofence ID list.
Type	0x12			1	Geofence Position Position of the client when it breached the Geofence. This TLV is included if the client configures the Geofence to report its position. The position is reported at the same confidence level that was specified in the Add Circular Geofence request.

Field	Field value	Field type	Parameter	Size (byte)	Description
Length	61			2	
Value	→	uint64	timestampUtc	8	UTC timestamp. • Units: Milliseconds since Jan. 1, 1970
		double	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
		double	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
		float	horUncEllipseSemiMinor	4	Semi-minor axis of horizontal elliptical uncertainty. • Units: Meters
		float	horUncEllipseSemiMajor	4	Semi-major axis of horizontal elliptical uncertainty. • Units: Meters
		float	horUncEllipseOrient Azimuth	4	Elliptical horizontal uncertainty azimuth of orientation. • Units: Decimal degrees • Range: 0 to 180
		boolean	speedHorizontal_valid	1	Indicates whether the Horizontal speed field contains valid information. • 0x01 (TRUE) – Horizontal speed is valid • 0x00 (FALSE) – Horizontal speed is invalid and is to be ignored
		float	speedHorizontal	4	Horizontal speed. • Units: Meters/second
		boolean	altitudeWrtEllipsoid_valid	1	Indicates whether the altitude field contains valid information. • 0x01 (TRUE) – Altitude field is valid • 0x00 (FALSE) – Altitude field is invalid and is to be ignored
		float	altitudeWrtEllipsoid	4	Altitude with respect to the WGS84 ellipsoid. • Units: Meters • Range: -500 to 15883

Field	Field value	Field type	Parameter	Size (byte)	Description
		boolean	vertUnc_valid	1	Indicates whether the Vertical Uncertainty field contains valid information. <ul style="list-style-type: none"> • 0x01 (TRUE) – Vertical Uncertainty field is valid • 0x00 (FALSE) – Vertical Uncertainty field is invalid and is to be ignored
		float	vertUnc	4	Vertical uncertainty. <ul style="list-style-type: none"> • Units: Meters
		boolean	speedVertical_valid	1	Indicates whether the Vertical Speed field contains valid information. <ul style="list-style-type: none"> • 0x01 (TRUE) – Vertical Speed field is valid • 0x00 (FALSE) – Vertical Speed field is invalid and is to be ignored
		float	speedVertical	4	Vertical speed. <ul style="list-style-type: none"> • Units: Meters/second
		boolean	heading_valid	1	Indicates whether the Heading field contains valid information. <ul style="list-style-type: none"> • 0x01 (TRUE) – Heading field is valid • 0x00 (FALSE) – Heading field is invalid and is to be ignored
		float	heading	4	Heading. <ul style="list-style-type: none"> • Units: Degrees • Range: 0 to 359.999
Type	0x13			1	Geofence Breach Confidence
Length	4			2	
Value	→	enum	breachConfidence	4	Given a breach event, the confidence determines the probability that the breach happened at the Geofence boundary. Valid values: <ul style="list-style-type: none"> • eQMI_LOC_GEOFENCE_CONFIDENCE_LOW (0x01) – Geofence engine indicates a breach with low confidence; this setting results in lower power usage, and it can impact the yield because incorrect breach events may be sent • eQMI_LOC_GEOFENCE_CONFIDENCE_MED (0x02) – (Default) Geofence engine indicates a breach with medium confidence • eQMI_LOC_GEOFENCE_CONFIDENCE_HIGH (0x03) – Geofence engine indicates a breach with high confidence; this setting results in higher power usage

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x14			1	Heading Uncertainty
Length	4			2	
Value	→	float	headingUnc	4	Heading uncertainty. • Units: Degrees • Range: 0 to 359.999
Type	0x15			1	Vertical Uncertainty
Length	4			2	
Value	→	float	vertUnc	4	Vertical uncertainty. • Units: Meters
Type	0x16			1	Speed Uncertainty
Length	4			2	
Value	→	float	speedUnc	4	3-D speed uncertainty. • Units: Meters/second
Type	0x17			1	Horizontal Confidence
Length	1			2	
Value	→	uint8	horConfidence	1	Horizontal uncertainty confidence. • Units: Percent • Range: 0 to 99
Type	0x18			1	Vertical Confidence
Length	1			2	
Value	→	uint8	vertConfidence	1	Vertical uncertainty confidence. • Units: Percent • Range: 0 to 99
Type	0x19			1	Dilution of Precision Dilution of precision associated with this position.
Length	12			2	
Value	→	float	PDOP	4	Position dilution of precision. • Range: 1 (highest accuracy) to 50 (lowest accuracy) • PDOP = square root of (HDOP ² + VDOP ²)
		float	HDOP	4	Horizontal dilution of precision. • Range: 1 (highest accuracy) to 50 (lowest accuracy)
		float	VDOP	4	Vertical dilution of precision. • Range: 1 (highest accuracy) to 50 (lowest accuracy)
Type	0x1A			1	SVs Used to Calculate the Fix
Length	Var			2	
Value	→	uint8	gnssSvUsedList_len	1	Number of sets of the following elements: • gnssSvUsedList

Field	Field value	Field type	Parameter	Size (byte)	Description
		uint16	gnssSvUsedList	Var	Each entry in the list contains the SV ID of a satellite used for calculating this position report. The following information is associated with each SV ID: Range: <ul style="list-style-type: none"> • For GPS: 1 to 32 • For SBAS: 33 to 64 • For GLONASS: 65 to 96 • For QZSS: 193 to 197 • For BDS: 201 to 237

3.100.2 Description of QMI_LOC_EVENT_GEOFENCE_BATCHED - BREACH_NOTIFICATION

This command notifies the control point when a Geofence is breached. All the Geofences that were breached at the same position are batched. This request is generated only if the control point has indicated its interest by setting the appropriate mask in QMI_LOC_REG_EVENTS_REQ.

Here is a brief description of the two Geofence ID lists:

- Continuous Geofence ID list – Each entry in the list contains the continuous range of Geofence IDs that were breached at the same position. For example, if Geofence IDs from 2 to 9 and 13 to 20 were breached at the same position, the continuous list is (2, 9) (13, 20), etc.
- Discrete Geofence ID list – Each entry is a single Geofence ID that was breached at the same position. These entries do not form a continuous range of IDs, e.g., 34, 67, 78, etc.

The two lists do not overlap, meaning a Geofence ID present in one list is not present in the other.

3.101 QMI_LOC_EVENT_VEHICLE_DATA_READY_STATUS

Notifies the control point whether the GNSS location engine is ready to accept vehicle data.

LOC message ID

0x0081

Version introduced

Major - 2, Minor - 24

3.101.1 Indication - QMI_LOC_EVENT_VEHICLE_DATA_READY_STATUS_IND

Message type

Indication

Sender

Service

Mandatory TLVs

None

Optional TLVs

Name	Version introduced	Version last modified
Vehicle Accelerometer Ready Status	2.24	2.24
Vehicle Angular Rate Ready Status	2.24	2.24
Vehicle Odometry Ready Status	2.24	2.24

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Vehicle Accelerometer Ready Status
Length	1			2	
Value	→	boolean	vehicleAccelReadyStatus	1	The location service uses this TLV to let a control point know when it is ready or not ready to receive vehicle accelerometer data input. Values: <ul style="list-style-type: none"> • 0x00 – Not ready • 0x01 – Ready
Type	0x11			1	Vehicle Angular Rate Ready Status
Length	1			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	boolean	vehicleAngularRateReady Status	1	The location service uses this TLV to let a control point know when it is ready or not ready to receive vehicle angular rate data input. Values: • 0x00 – Not ready • 0x01 – Ready
Type	0x12			1	Vehicle Odometry Ready Status
Length	1			2	
Value	→	boolean	vehicleOdometryReady Status	1	The location service uses this TLV to let a control point know when it is ready or not ready to receive vehicle odometry data input. Values: • 0x00 – Not ready • 0x01 – Ready

3.101.2 Description of QMI_LOC_EVENT_VEHICLE_DATA_READY - STATUS

This command sends a Vehicle Data Ready Status event to the control point. The control point can start injecting vehicle data into the location engine after it receives this event.

This request is generated only if the control point has indicated its interest by setting the appropriate mask in QMI_LOC_REG_EVENTS_REQ.

3.102 QMI_LOC_INJECT_VEHICLE_SENSOR_DATA

Injects on-vehicle sensor data into the location engine.

LOC message ID

0x0082

Version introduced

Major - 2, Minor - 24

3.102.1 Request - QMI_LOC_INJECT_VEHICLE_SENSOR_DATA_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

None

Optional TLVs

Name	Version introduced	Version last modified
On-Vehicle Accelerometer Data	2.24	2.24
On-Vehicle Angular Rotation Data	2.24	2.24
Odometry Data	2.24	2.24
External Time Sync Information	2.24	2.24

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	On-Vehicle Accelerometer Data Vehicle accelerometer sensor samples.
Length	Var			2	
Value	→	uint32	sampleTimeBase	4	Denotes a 32-bit time tag of the reference time from which all samples in this message are offset. This time must be the same as or (slightly) earlier than the first (oldest) sample in this message. <ul style="list-style-type: none"> Units: Milliseconds Range: 4 million seconds, or almost 50 days between rollovers

Field	Field value	Field type	Parameter	Size (byte)	Description
		mask8	axesValidity	1	Identifies the axes that are valid for all sensor samples. Valid values: <ul style="list-style-type: none"> • QMI_LOC_MASK_X_AXIS (0x01) – X-axis is valid • QMI_LOC_MASK_Y_AXIS (0x02) – Y-axis is valid • QMI_LOC_MASK_Z_AXIS (0x04) – Z-axis is valid
		uint8	sensorData_len	1	Number of sets of the following elements: <ul style="list-style-type: none"> • timeOffset • axisSample_len • axisSample
		uint32	timeOffset	4	Sample time offset. This time offset must be relative to the vehicle sensor time of the first sample. <ul style="list-style-type: none"> • Units: Microseconds • Range: Up to over 4000 seconds
		uint8	axisSample_len	1	Number of sets of the following elements: <ul style="list-style-type: none"> • axisSample
		float	axisSample	Var	Sensor axis sample. <ul style="list-style-type: none"> • Type: Floating point • Units accelerometer: Meters/seconds² • Units gyroscope: Radians/seconds Note: The axes samples must be in the following order: 1. X-Axis 2. Y-Axis 3. Z-Axis
Type	0x11			1	On-Vehicle Angular Rotation Data Vehicle angular rotation data sensor samples.
Length	Var			2	
Value	→	uint32	sampleTimeBase	4	Denotes a 32-bit time tag of the reference time from which all samples in this message are offset. This time must be the same as or (slightly) earlier than the first (oldest) sample in this message. <ul style="list-style-type: none"> • Units: Milliseconds • Range: 4 million seconds, or almost 50 days between rollovers

Field	Field value	Field type	Parameter	Size (byte)	Description
		mask8	axesValidity	1	Identifies the axes that are valid for all sensor samples. Valid values: <ul style="list-style-type: none"> • QMI_LOC_MASK_X_AXIS (0x01) – X-axis is valid • QMI_LOC_MASK_Y_AXIS (0x02) – Y-axis is valid • QMI_LOC_MASK_Z_AXIS (0x04) – Z-axis is valid
		uint8	sensorData_len	1	Number of sets of the following elements: <ul style="list-style-type: none"> • timeOffset • axisSample_len • axisSample
		uint32	timeOffset	4	Sample time offset. This time offset must be relative to the vehicle sensor time of the first sample. <ul style="list-style-type: none"> • Units: Microseconds • Range: Up to over 4000 seconds
		uint8	axisSample_len	1	Number of sets of the following elements: <ul style="list-style-type: none"> • axisSample
		float	axisSample	Var	Sensor axis sample. <ul style="list-style-type: none"> • Type: Floating point • Units accelerometer: Meters/seconds² • Units gyroscope: Radians/seconds Note: The axes samples must be in the following order: 1. X-Axis 2. Y-Axis 3. Z-Axis
Type	0x12			1	Odometry Data Odometer sensor samples.
Length	Var			2	
Value	→	uint32	sampleTimeBase	4	Denotes a 32-bit time tag of a reference time, from which all samples in this message are offset. Note this time must be the same or (slightly) earlier than the first (oldest) sample in this message. <ul style="list-style-type: none"> • Units: 1 millisecond • Range: 4 million seconds, or almost 50 days between rollovers

Field	Field value	Field type	Parameter	Size (byte)	Description
		mask32	flags	4	<p>Flags to indicate any deviation from the default measurement assumptions.</p> <p>Valid bitmasks:</p> <ul style="list-style-type: none"> • QMI_LOC_MASK_VEHICLE_ODOMETRY_REVERSE_MOVEMENT (0x00000001) – Odometry data in this message includes at least some data where the vehicle may have been moving in the reverse direction; this bit must be set if odometry data may be in reverse, and should not be set if odometry data is all in the forward direction • QMI_LOC_MASK_VEHICLE_ODOMETRY_AFFECTED_BY_ERRORS (0x00000002) – Odometry data in this message includes at least some data affected by a major error source affecting distance-travelled accuracy, such as wheel slippage due to skidding, gravel, snow, or ice, as detected by the vehicle, e.g., via an ABS or other system • QMI_LOC_MASK_VEHICLE_ODOMETRY_ABSOLUTE_MEASUREMENT (0x00000004) – Odometry data in this message is an absolute amount since the vehicle began service, and is the same vehicle that is regularly used with this device (so that the offset of this value, since the last time this measurement was used by the location engine, can safely be used as a likely correct estimate of distance travelled since last use)

Field	Field value	Field type	Parameter	Size (byte)	Description
		mask32	wheelFlags	4	<p>Delineates for which wheels measurements are being provided in the following samples, where one or more of the following bits must be set, and data samples aligned with these axes must appear in groups, in this order.</p> <p>Valid bitmasks:</p> <ul style="list-style-type: none"> • QMI_LOC_MASK_VEHICLE_ODOMETRY_LEFT_AND_RIGHT_AVERAGE (0x00000001) – Average of left and right non-turning wheels • QMI_LOC_MASK_VEHICLE_ODOMETRY_LEFT (0x00000002) – Left side, non-turning wheel • QMI_LOC_MASK_VEHICLE_ODOMETRY_RIGHT (0x00000004) – Right side, non-turning wheel
		uint32	distanceTravelledBase	4	<p>Distance traveled base.</p> <ul style="list-style-type: none"> • Units of accumulated distance: Meters • Range: Over 4,000,0000 kilometers <p>Distance travelled (odometry) is to be reported in a continuously accumulating way from device power up. It may be incremental distance starting at 0, or another arbitrary point, from device power up, or the absolute distance traveled by the vehicle (and if so, set QMI_LOC_MASK_VEHICLE_ODOMETRY_ABSOLUTE_MEASUREMENT), as long as it grows incrementally from device power up.</p> <p>This distance_travelled_base is added to the distance_travelled_offset of each sample (below) to get the absolute distance of each sample point.</p> <p>Distance travelled errors are expected to be primarily due to the scale factor, with some allowance for noise due to minor slippage events (e.g., gravel.) Major wheel slippage events that affect odometry must be flagged – see the flags field.</p> <p>Note that other events, such as a vehicle travelling in reverse, may also affect the available accuracy of this information, and notification of those events must be provided – see the flags field.</p>

Field	Field value	Field type	Parameter	Size (byte)	Description
		uint8	odometryData_len	1	Number of sets of the following elements: • timeOffset • distanceTravelled_len • distanceTravelled
		uint32	timeOffset	4	Sample time offset. This time offset must be relative to the sensor time of the first sample. • Units: Microseconds • Range: Up to over 4000 seconds
		uint8	distanceTravelled_len	1	Number of sets of the following elements: • distanceTravelled
		uint32	distanceTravelled	Var	Distance travelled (odometry) sample offset. • Units of accumulated distance: Millimeters • Range: Over 4000 kilometers This measurement (with units in millimeters) is added to the distance_travelled_base measurement (in meters) to get the total distance travelled sample value. Note: The order of measurements must be as follows: 1. Left and right average 2. Left 3. Right
Type	0x13			1	External Time Sync Information
Length	4			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	int32	changeInTimeScales	4	<p>This field is to be used in conjunction with an external time-sync mechanism that is aligning the vehicle sensor time scale with the on-device sensor time scale to ensure that updates in that time offset do not appear as jumps in the relative sensor time of the samples provided in this message. If there is no such sync mechanism, e.g., if only the vehicle time is provided, this field may be left at 0.</p> <p>This field is defined as the change from the previously-sent QMI message with similar TLVs 0x10, 0x11, or 0x12 in it, to this QMI message in the amount that the sensor_time is ahead of an external vehicle time.</p> <ul style="list-style-type: none"> • Units: Microseconds • Range: Approximately -2100 seconds to + 2100 seconds, where full-scale (minimum and maximum value) is interpreted as equal to or greater than this value of an offset change (unlikely to be reached in practice, unless there is a startup, major resync, or some other rollover event).

3.102.2 Indication - QMI_LOC_INJECT_VEHICLE_SENSOR_DATA_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version introduced	Version last modified
Inject Vehicle Sensor Data Status	2.24	2.28

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

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	enum	status	4	<p>Status of the Inject Vehicle Sensor Data request.</p> <p>Valid values:</p> <ul style="list-style-type: none"> • eQMI_LOC_SUCCESS (0) – Request was completed successfully • eQMI_LOC_GENERAL_FAILURE (1) – Request failed because of a general failure • eQMI_LOC_UNSUPPORTED (2) – Request failed because it is not supported • eQMI_LOC_INVALID_PARAMETER (3) – Request failed because it contained invalid parameters • eQMI_LOC_ENGINE_BUSY (4) – Request failed because the engine is busy • eQMI_LOC_PHONE_OFFLINE (5) – Request failed because the phone is offline • eQMI_LOC_TIMEOUT (6) – Request failed because it timed out • eQMI_LOC_CONFIG_NOT_SUPPORTED (7) – Request failed because an undefined configuration was requested • eQMI_LOC_INSUFFICIENT_MEMORY (8) – Request failed because the engine could not allocate sufficient memory for the request • eQMI_LOC_MAX_GEOFENCE_PROGRAMMED (9) – Request failed because the maximum number of Geofences are already programmed • eQMI_LOC_XTRA_VERSION_CHECK_FAILURE (10) – Location service failed because of an XTRA version-based file format check failure

Optional TLVs

None

Error codes

QMI_ERR_NONE	Operation requested by the control point completed successfully
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_INVALID_ARG	Value field of one or more TLVs in the request message contains an invalid value
QMI_ERR_OP_DEVICE_UNSUPPORTED	Operation is not supported by the MSM GPS service
QMI_ERR_INVALID_OPERATION	Operation is not allowed due to the current state of the location engine
QMI_ERR_INFO_UNAVAILABLE	Samples were dropped because the message time-sanity check failed; failure is due to one of the following: sensor-to-GPS time synchronization information is not available, average sampling rate is faster than the threshold specified in the description below, or the message time has drifted too far from the expected time
QMI_ERR_NO_MEMORY	Samples were dropped because no memory is available

3.102.3 Description of QMI_LOC_INJECT_VEHICLE_SENSOR_DATA

This command injects vehicle sensor information to the location service.

This command must only be sent after it is requested by the location service. The location service requests this command input by sending a Streaming Ready status for the specific sensors in the QMI_LOC_EVENT_VEHICLE_DATA_READY_STATUS_IND indication. Commands that are sent without this request by the location service receive a QMI_ERR_INTERNAL error response.

Samples provided for each sample type must be provided in chronological order. The timescale and timestamps associated with the injected sensor samples are expected to be monotonically increasing and to increase at approximately the same rate as GPS time, as determined by the location service. The initial relationship of GPS time to sensor time is established by the QMI_LOC_INJECT_VEHICLE_TIME_SYNC_DATA command. When the location service determines that the sensor timestamps of an incoming command have drifted too far from the expected sensor timestamp, the following occurs:

- QMI_ERR_INFO_UNAVAILABLE error is returned
- Current GPS time to sensor time relationship is discarded and a request for time synchronization is sent out

These actions re-establish the GPS time to sensor time relationship and account for the clock drift.

The sensor timestamps of this command are expected to have an average time between samples of no less than 8 ms (approximately 125 Hz). If the samples span less than this threshold, a QMI_ERR_INFO_UNAVAILABLE error is returned.

The inertial and odometry sensor measurements described in this message are intended to be on-vehicle sensors, where stable alignment with respect to a wheeled land vehicle platform can be assumed. This could, for example be sensors from a vehicle, in which the device is located, being routed to the device, or where the device itself is embedded in a vehicle.

If similar sensor information is coming from on-device sensors, where stable alignment with a vehicle is only a possibility, and not a known state, the use of the QMI_LOC_INJECT_SENSOR_DATA message, instead of this message, is required.

3.103 QMI_LOC_GET_AVAILABLE_WWAN_POSITION

Used by the control point to get the first available WWAN position from the location engine.

LOC message ID

0x0083

Version introduced

Major - 2, Minor - 26

3.103.1 Request - QMI_LOC_GET_AVAILABLE_WWAN_POSITION_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

Name	Version introduced	Version last modified
Transaction ID	2.26	2.26

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Transaction ID
Length	4			2	
Value	→	uint32	transactionId	4	Identifies the transaction. The transaction ID is returned in the Get Available WWAN Position indication.

Optional TLVs

None

3.103.2 Indication - QMI_LOC_GET_AVAILABLE_WWAN_POSITION_IND

Message type

Indication

Sender

Control point

Mandatory TLVs

Name	Version introduced	Version last modified
Get Available WWAN Position Status	2.26	2.28

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Get Available WWAN Position Status
Length	4			2	
Value	→	enum	status	4	<p>Status of the Get Available WWAN Position request.</p> <p>Valid values:</p> <ul style="list-style-type: none"> • eQMI_LOC_SUCCESS (0) – Request was completed successfully • eQMI_LOC_GENERAL_FAILURE (1) – Request failed because of a general failure • eQMI_LOC_UNSUPPORTED (2) – Request failed because it is not supported • eQMI_LOC_INVALID_PARAMETER (3) – Request failed because it contained invalid parameters • eQMI_LOC_ENGINE_BUSY (4) – Request failed because the engine is busy • eQMI_LOC_PHONE_OFFLINE (5) – Request failed because the phone is offline • eQMI_LOC_TIMEOUT (6) – Request failed because it timed out • eQMI_LOC_CONFIG_NOT_SUPPORTED (7) – Request failed because an undefined configuration was requested • eQMI_LOC_INSUFFICIENT_MEMORY (8) – Request failed because the engine could not allocate sufficient memory for the request • eQMI_LOC_MAX_GEOFENCE_PROGRAMMED (9) – Request failed because the maximum number of Geofences are already programmed • eQMI_LOC_XTRA_VERSION_CHECK_FAILURE (10) – Location service failed because of an XTRA version-based file format check failure

Optional TLVs

Name	Version introduced	Version last modified
Transaction ID	2.26	2.26
Latitude	2.26	2.26
Longitude	2.26	2.26
Circular Horizontal Position Uncertainty	2.26	2.26
Altitude With Respect to Ellipsoid	2.26	2.26
Vertical Uncertainty	2.26	2.26
UTC Timestamp	2.26	2.26
Time Uncertainty	2.26	2.26
Horizontal Elliptical Uncertainty Semi-Minor Axis	2.26	2.26
Horizontal Elliptical Uncertainty Semi-Major Axis	2.26	2.26
Horizontal Elliptical Uncertainty Azimuth	2.26	2.26
Horizontal Circular Confidence	2.26	2.26
Horizontal Elliptical Confidence	2.26	2.26
Horizontal Reliability	2.26	2.26
Altitude With Respect to Sea Level	2.26	2.26
Vertical Confidence	2.26	2.26
Vertical Reliability	2.26	2.26
GPS Time	2.26	2.26
Time Source	2.26	2.26

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Transaction ID
Length	4			2	
Value	→	uint32	transactionId	4	Transaction ID that was specified in the Get Available WWAN Position request. This parameter will always be present if the status field is set to SUCCESS.
Type	0x11			1	Latitude
Length	8			2	
Value	→	double	latitude	8	Latitude (specified in WGS84 datum). <ul style="list-style-type: none"> Type: Floating point Units: Degrees Range: -90.0 to 90.0 <ul style="list-style-type: none"> Positive values indicate northern latitude Negative values indicate southern latitude
Type	0x12			1	Longitude
Length	8			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	double	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	0x13			1	Circular Horizontal Position Uncertainty
Length	4			2	
Value	→	float	horUncCircular	4	Horizontal position uncertainty (circular). • Units: Meters
Type	0x14			1	Altitude With Respect to Ellipsoid
Length	4			2	
Value	→	float	altitudeWrtEllipsoid	4	Altitude with respect to the WGS84 ellipsoid. • Units: Meters • Range: -500 to 15883
Type	0x15			1	Vertical Uncertainty
Length	4			2	
Value	→	float	vertUnc	4	Vertical uncertainty. • Units: Meters
Type	0x16			1	UTC Timestamp
Length	8			2	
Value	→	uint64	timestampUtc	8	UTC timestamp. • Units: Milliseconds since Jan. 1, 1970
Type	0x17			1	Time Uncertainty
Length	4			2	
Value	→	float	timeUnc	4	Time uncertainty. • Units: Milliseconds
Type	0x18			1	Horizontal Elliptical Uncertainty Semi-Minor Axis
Length	4			2	
Value	→	float	horUncEllipseSemiMinor	4	Semi-minor axis of horizontal elliptical uncertainty. • Units: Meters
Type	0x19			1	Horizontal Elliptical Uncertainty Semi-Major Axis
Length	4			2	
Value	→	float	horUncEllipseSemiMajor	4	Semi-major axis of horizontal elliptical uncertainty. • Units: Meters
Type	0x1A			1	Horizontal Elliptical Uncertainty Azimuth
Length	4			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	float	horUncEllipseOrient Azimuth	4	Elliptical horizontal uncertainty azimuth of orientation. • Units: Decimal degrees • Range: 0 to 180
Type	0x1B			1	Horizontal Circular Confidence
Length	1			2	
Value	→	uint8	horCircularConfidence	1	Horizontal circular uncertainty confidence. • Units: Percent • Range: 0 to 99
Type	0x1C			1	Horizontal Elliptical Confidence
Length	1			2	
Value	→	uint8	horEllipticalConfidence	1	Horizontal elliptical uncertainty confidence. • Units: Percent • Range: 0 to 99
Type	0x1D			1	Horizontal Reliability
Length	4			2	
Value	→	enum	horReliability	4	Specifies the reliability of the horizontal position. Valid values: • eQMI_LOC_RELIABILITY_NOT_SET (0) – Location reliability is not set • eQMI_LOC_RELIABILITY_VERY_LOW (1) – Location reliability is very low; use it at your own risk • eQMI_LOC_RELIABILITY_LOW (2) – Location reliability is low; little or no cross-checking is possible • eQMI_LOC_RELIABILITY_MEDIUM (3) – Location reliability is medium; limited cross-check passed • eQMI_LOC_RELIABILITY_HIGH (4) – Location reliability is high; strong cross-check passed
Type	0x1E			1	Altitude With Respect to Sea Level
Length	4			2	
Value	→	float	altitudeWrtMeanSeaLevel	4	Altitude with respect to mean sea level. • Units: Meters
Type	0x1F			1	Vertical Confidence
Length	1			2	
Value	→	uint8	vertConfidence	1	Vertical uncertainty confidence. • Units: Percent • Range: 0 to 99
Type	0x20			1	Vertical Reliability
Length	4			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	enum	vertReliability	4	Specifies the reliability of the vertical position. Valid values: <ul style="list-style-type: none"> • eQMI_LOC_RELIABILITY_NOT_SET (0) – Location reliability is not set • eQMI_LOC_RELIABILITY_VERY_LOW (1) – Location reliability is very low; use it at your own risk • eQMI_LOC_RELIABILITY_LOW (2) – Location reliability is low; little or no cross-checking is possible • eQMI_LOC_RELIABILITY_MEDIUM (3) – Location reliability is medium; limited cross-check passed • eQMI_LOC_RELIABILITY_HIGH (4) – Location reliability is high; strong cross-check passed
Type	0x21			1	GPS Time
Length	6			2	
Value	→	uint16	gpsWeek	2	Current GPS week as calculated from midnight, Jan. 6, 1980. <ul style="list-style-type: none"> • Units: Weeks
		uint32	gpsTimeOfWeekMs	4	Amount of time into the current GPS week. <ul style="list-style-type: none"> • Units: Milliseconds
Type	0x22			1	Time Source
Length	4			2	
Value	→	enum	timeSrc	4	Time source. Valid values: <ul style="list-style-type: none"> • eQMI_LOC_TIME_SRC_INVALID (0) – Invalid time. • eQMI_LOC_TIME_SRC_NETWORK_TIME_TRANSFER (1) – Time is set by the 1X system • eQMI_LOC_TIME_SRC_NETWORK_TIME_TAGGING (2) – Time is set by WCDMA/GSM time tagging (i.e., associating network time with GPS time) • eQMI_LOC_TIME_SRC_EXTERNAL_INPUT (3) – Time is set by an external injection • eQMI_LOC_TIME_SRC_TOW_DECODE (4) – Time is set after decoding over-the-air GPS navigation data from one GPS satellite

Field	Field value	Field type	Parameter	Size (byte)	Description
			timeSrc (cont.)		<ul style="list-style-type: none"> • eQMI_LOC_TIME_SRC_TOW_CONFIRMED (5) – Time is set after decoding over-the-air GPS navigation data from multiple satellites • eQMI_LOC_TIME_SRC_TOW_AND_WEEK_CONFIRMED (6) – Both time of the week and the GPS week number are known • eQMI_LOC_TIME_SRC_NAV_SOLUTION (7) – Time is set by the position engine after the fix is obtained • eQMI_LOC_TIME_SRC_SOLVE_FOR_TIME (8) – Time is set by the position engine after performing SFT; this is done when the clock time uncertainty is large • eQMI_LOC_TIME_SRC_GLO_TOW_DECODE (9) – Time is set after decoding GLO satellites • eQMI_LOC_TIME_SRC_TIME_TRANSFORM (10) – Time is set after transforming the GPS to GLO time • eQMI_LOC_TIME_SRC_WCDMA_SLEEP_TIME_TAGGING (11) – Time is set by the sleep time tag provided by the WCDMA network • eQMI_LOC_TIME_SRC_GSM_SLEEP_TIME_TAGGING (12) – Time is set by the sleep time tag provided by the GSM network • eQMI_LOC_TIME_SRC_UNKNOWN (13) – Source of the time is unknown • eQMI_LOC_TIME_SRC_SYSTEM_TIMETICK (14) – Time is derived from the system clock (better known as the slow clock); GNSS time is maintained irrespective of the GNSS receiver state • eQMI_LOC_TIME_SRC_QZSS_TOW_DECODE (15) – Time is set after decoding QZSS satellites • eQMI_LOC_TIME_SRC_BDS_TOW_DECODE (16) – Time is set after decoding BDS satellites

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

3.103.3 Description of QMI_LOC_GET_AVAILABLE_WWAN_POSITION

This command is used to get the first available WWAN position estimate from the location engine. The engine does not consume any additional power to obtain a position for this request. For example, the GNSS engine is not turned on if it was off previously. The engine sends the available WWAN position estimate it currently has through the indication for this request. It is safe for multiple clients to use this command. However, there can only be one outstanding request at any time.

3.104 QMI_LOC_SET_PREMIUM_SERVICES_CONFIG

Used by the control point to set the configuration information for all iZat premium services to the location engine.

LOC message ID

0x0084

Version introduced

Major - 2, Minor - 26

3.104.1 Request - QMI_LOC_SET_PREMIUM_SERVICES_CONFIG_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

Name	Version introduced	Version last modified
Set Premium Service Type	2.26	2.26
Set Premium Service Configuration	2.26	2.26

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Set Premium Service Type
Length	4			2	
Value	→	enum	premiumServiceType	4	Specifies the premium service to configure. Valid values: • eQMI_LOC_PREMIUM_SERVICE_GTP_CELL (0) – Premium service – Global terrestrial positioning for the cell • eQMI_LOC_PREMIUM_SERVICE_SAP (1) – Premium service – Sensor-assisted positioning
Type	0x02			1	Set Premium Service Configuration
Length	4			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	enum	premiumServiceCfg	4	Specifies the premium service configuration mode. Valid values: <ul style="list-style-type: none"> • eQMI_LOC_PREMIUM_SERVICE_DISABLED (0) – Premium service disabled • eQMI_LOC_PREMIUM_SERVICE_ENABLED_BASIC (1) – Premium service enabled for basic • eQMI_LOC_PREMIUM_SERVICE_ENABLED_PREMIUM (2) – Premium service enabled for premium

Optional TLVs

None

3.104.2 Indication - QMI_LOC_SET_PREMIUM_SERVICES_CONFIG_IND**Message type**

Indication

Sender

Control point

Mandatory TLVs

Name	Version introduced	Version last modified
Set Premium Service Configuration Status	2.26	2.28

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Set Premium Service Configuration Status
Length	4			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	enum	status	4	<p>Status of the Set Premium Services Configuration request.</p> <p>Valid values:</p> <ul style="list-style-type: none"> • eQMI_LOC_SUCCESS (0) – Request was completed successfully • eQMI_LOC_GENERAL_FAILURE (1) – Request failed because of a general failure • eQMI_LOC_UNSUPPORTED (2) – Request failed because it is not supported • eQMI_LOC_INVALID_PARAMETER (3) – Request failed because it contained invalid parameters • eQMI_LOC_ENGINE_BUSY (4) – Request failed because the engine is busy • eQMI_LOC_PHONE_OFFLINE (5) – Request failed because the phone is offline • eQMI_LOC_TIMEOUT (6) – Request failed because it timed out • eQMI_LOC_CONFIG_NOT_SUPPORTED (7) – Request failed because an undefined configuration was requested • eQMI_LOC_INSUFFICIENT_MEMORY (8) – Request failed because the engine could not allocate sufficient memory for the request • eQMI_LOC_MAX_GEOFENCE_PROGRAMMED (9) – Request failed because the maximum number of Geofences are already programmed • eQMI_LOC_XTRA_VERSION_CHECK_FAILURE (10) – Location service failed because of an XTRA version-based file format check failure

Optional TLVs

None

3.105 QMI_LOC_SET_XTRA_VERSION_CHECK

Used by the control point to enable or disable XTRA version verification.

LOC message ID

0x0085

Version introduced

Major - 2, Minor - 28

3.105.1 Request - QMI_LOC_SET_XTRA_VERSION_CHECK_REQ

Message type

Request

Sender

Control point

Mandatory TLVs

Name	Version introduced	Version last modified
Set XTRA Version Check Mode	2.28	2.28

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Set XTRA Version Check Mode
Length	4			2	
Value	→	enum	xtraVersionCheckMode	4	Specifies XTRA version check mode. Valid values: <ul style="list-style-type: none"> • eQMI_LOC_XTRA_VERSION_CHECK_DISABLE (0) – XTRA file version check is not required • eQMI_LOC_XTRA_VERSION_CHECK_AUTO (1) – XTRA file version check is required; the Location service decides the ‘expected version’ based on the preprovisioned XTRA version configuration • eQMI_LOC_XTRA_VERSION_CHECK_XTRA2 (2) – Check the XTRA file against XTRA2 format • eQMI_LOC_XTRA_VERSION_CHECK_XTRA3 (3) – Check the XTRA file against XTRA3 format

Optional TLVs

None

3.105.2 Indication - QMI_LOC_SET_XTRA_VERSION_CHECK_IND**Message type**

Indication

Sender

Service

Mandatory TLVs

Name	Version introduced	Version last modified
Set XTRA Version Check Mode Status	2.28	2.28

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Set XTRA Version Check Mode Status
Length	4			2	
Value	→	enum	status	4	<p>Status of the Set XTRA version check request.</p> <p>Valid values:</p> <ul style="list-style-type: none"> • eQMI_LOC_SUCCESS (0) – Request was completed successfully • eQMI_LOC_GENERAL_FAILURE (1) – Request failed because of a general failure • eQMI_LOC_UNSUPPORTED (2) – Request failed because it is not supported • eQMI_LOC_INVALID_PARAMETER (3) – Request failed because it contained invalid parameters • eQMI_LOC_ENGINE_BUSY (4) – Request failed because the engine is busy • eQMI_LOC_PHONE_OFFLINE (5) – Request failed because the phone is offline • eQMI_LOC_TIMEOUT (6) – Request failed because it timed out • eQMI_LOC_CONFIG_NOT_SUPPORTED (7) – Request failed because an undefined configuration was requested

Field	Field value	Field type	Parameter	Size (byte)	Description
			status (cont.)		<ul style="list-style-type: none"> • eQMI_LOC_INSUFFICIENT_MEMORY (8) – Request failed because the engine could not allocate sufficient memory for the request • eQMI_LOC_MAX_GEOFENCE_PROGRAMMED (9) – Request failed because the maximum number of Geofences are already programmed • eQMI_LOC_XTRA_VERSION_CHECK_FAILURE (10) – Location service failed because of an XTRA version-based file format check failure

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
QMI_ERR_NOT_SUPPORTED	QMI_LOC_SET_XTRA_VERSION_CHECK message is not yet supported

3.105.3 Description of QMI_LOC_SET_XTRA_VERSION_CHECK

At the time the QMI_LOC_SET_XTRA_VERSION_CHECK message is generated, XTRA2 and XTRA3 are the available XTRA versions. The XTRA file is downloaded from the network. A network attacker can replace the XTRA3 file with an XTRA2 file to bypass a security check. To protect against an XTRA file version attack, the mobile software verifies the format of the received XTRA file based on the XTRA version. A mismatch between the expected requested and the received XTRA file format checking causes an XTRA file rejection. If the HLOS XTRA client does not send the QMI command, the check is disabled by default.

3.106 QMI_LOC_EVENT_GNSS_MEASUREMENT_REPORT_IND

Sends a satellite measurement report to the control point.

LOC message ID

0x0086

Version introduced

Major - 2, Minor - 31

3.106.1 Indication - QMI_LOC_EVENT_GNSS_MEASUREMENT_REPORT_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version introduced	Version last modified
Current Message Sequence Number	2.31	2.31
Maximum Number of Messages to be Sent for Present Time Epoch	2.31	2.31
Specifies Satellite System Constellation of This Report	2.31	2.31

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Current Message Sequence Number
Length	1			2	
Value	→	uint8	seqNum	1	Current message number. Used for segmentation/assembly of measurement reports.
Type	0x02			1	Maximum Number of Messages to be Sent for Present Time Epoch
Length	1			2	
Value	→	uint8	maxMessageNum	1	Maximum number of messages that are to be sent for the present time epoch.
Type	0x03			1	Specifies Satellite System Constellation of This Report
Length	4			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	enum	system	4	Specifies the satellite system constellation of this report. Valid values: <ul style="list-style-type: none"> • eQMI_LOC_SV_SYSTEM_GPS (1) – GPS satellite • eQMI_LOC_SV_SYSTEM_GALILEO (2) – GALILEO satellite • eQMI_LOC_SV_SYSTEM_SBAS (3) – SBAS satellite • eQMI_LOC_SV_SYSTEM_COMPASS (4) – COMPASS satellite • eQMI_LOC_SV_SYSTEM_GLONASS (5) – GLONASS satellite • eQMI_LOC_SV_SYSTEM_BDS (6) – BDS satellite

Optional TLVs

Name	Version introduced	Version last modified
GNSS Receiver Clock Frequency Information	2.31	2.31
Leap Second Information	2.31	2.31
GPS to GLONASS Intersystem Time Bias	2.31	2.31
GPS to BDS Intersystem Time Bias	2.31	2.31
GPS to GALILEO Intersystem Time Bias	2.31	2.31
BDS to GLONASS Intersystem Time Bias	2.31	2.31
GAL to GLONASS Intersystem Time Bias	2.31	2.31
GAL to BDS Intersystem Time Bias	2.31	2.31
Satellite System Time Information for GPS, BDS, GAL Constellation	2.31	2.31
GLONASS System Time Information	2.31	2.31
Extended Time Information	2.31	2.31
Satellite System Measurement Report for Enabled Constellation	2.31	2.31

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	GNSS Receiver Clock Frequency Information
Length	12			2	
Value	→	float	clockDrift	4	Receiver clock drift. • Units: Meters per second
		float	clockDriftUnc	4	Receiver clock drift uncertainty. • Units: Meters per second

Field	Field value	Field type	Parameter	Size (byte)	Description
		enum	sourceOfFreq	4	Source of the clock frequency information. Valid values: <ul style="list-style-type: none"> • eQMI_LOC_FREQ_SOURCE_INVALID (0) – Source of the frequency is invalid • eQMI_LOC_FREQ_SOURCE_EXTERNAL (1) – Source of the frequency is from an external injection • eQMI_LOC_FREQ_SOURCE_PE_CLK_REPORT (2) – Source of the frequency is from the GNSS navigation engine • eQMI_LOC_FREQ_SOURCE_UNKNOWN (3) – Source of the frequency is unknown
Type	0x11			1	Leap Second Information
Length	2			2	
Value	→	uint8	leapSec	1	GPS time leap second delta to UTC time. For nonzero values of leapSecUnc, leapSec must be treated as unknown. <ul style="list-style-type: none"> • Units: Seconds
		uint8	leapSecUnc	1	Uncertainty for the GPS leap second. <ul style="list-style-type: none"> • Units: Seconds
Type	0x12			1	GPS to GLONASS Intersystem Time Bias This is reported if both GPS and GLONASS system information reporting are enabled. <ul style="list-style-type: none"> • System 1: GPS • System 2: GLONASS
Length	9			2	
Value	→	mask8	validMask	1	Fields that are valid. Valid values: <ul style="list-style-type: none"> • QMI_LOC_SYS_TIME_BIAS_VALID (0x01) – System time bias is valid • QMI_LOC_SYS_TIME_BIAS_UNC_VALID (0x02) – System time bias uncertainty is valid
		float	timeBias	4	System 1 to System 2 time bias. <ul style="list-style-type: none"> • Units: Milliseconds
		float	timeBiasUnc	4	System 1 to System 2 time bias uncertainty. <ul style="list-style-type: none"> • Units: Milliseconds

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x13			1	GPS to BDS Intersystem Time Bias This is reported if both GPS and BDS system information reporting are enabled. • System 1: GPS • System 2: BDS
Length	9			2	
Value	→	mask8	validMask	1	Fields that are valid. Valid values: • QMI_LOC_SYS_TIME_BIAS_VALID (0x01) – System time bias is valid • QMI_LOC_SYS_TIME_BIAS_UNC_VALID (0x02) – System time bias uncertainty is valid
		float	timeBias	4	System 1 to System 2 time bias. • Units: Milliseconds
		float	timeBiasUnc	4	System 1 to System 2 time bias uncertainty. • Units: Milliseconds
Type	0x14			1	GPS to GALILEO Intersystem Time Bias This is reported if both GPS and GALILEO system information reporting are enabled. • System 1: GPS • System 2: GALILEO
Length	9			2	
Value	→	mask8	validMask	1	Fields that are valid. Valid values: • QMI_LOC_SYS_TIME_BIAS_VALID (0x01) – System time bias is valid • QMI_LOC_SYS_TIME_BIAS_UNC_VALID (0x02) – System time bias uncertainty is valid
		float	timeBias	4	System 1 to System 2 time bias. • Units: Milliseconds
		float	timeBiasUnc	4	System 1 to System 2 time bias uncertainty. • Units: Milliseconds

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x15			1	BDS to GLONASS Intersystem Time Bias This is reported if both BDS and GLONASS system information reporting are enabled. <ul style="list-style-type: none"> • System 1: BDS • System 2: GLONASS
Length	9			2	
Value	→	mask8	validMask	1	Fields that are valid. Valid values: <ul style="list-style-type: none"> • QMI_LOC_SYS_TIME_BIAS_VALID (0x01) – System time bias is valid • QMI_LOC_SYS_TIME_BIAS_UNC_VALID (0x02) – System time bias uncertainty is valid
		float	timeBias	4	System 1 to System 2 time bias. <ul style="list-style-type: none"> • Units: Milliseconds
		float	timeBiasUnc	4	System 1 to System 2 time bias uncertainty. <ul style="list-style-type: none"> • Units: Milliseconds
Type	0x16			1	GAL to GLONASS Intersystem Time Bias This is reported if both GAL and GLONASS system information reporting are enabled. <ul style="list-style-type: none"> • System 1: GAL • System 2: GLONASS
Length	9			2	
Value	→	mask8	validMask	1	Fields that are valid. Valid values: <ul style="list-style-type: none"> • QMI_LOC_SYS_TIME_BIAS_VALID (0x01) – System time bias is valid • QMI_LOC_SYS_TIME_BIAS_UNC_VALID (0x02) – System time bias uncertainty is valid
		float	timeBias	4	System 1 to System 2 time bias. <ul style="list-style-type: none"> • Units: Milliseconds
		float	timeBiasUnc	4	System 1 to System 2 time bias uncertainty. <ul style="list-style-type: none"> • Units: Milliseconds

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x17			1	GAL to BDS Intersystem Time Bias This is reported if both GAL and BDS system information reporting are enabled. • System 1: GAL • System 2: BDS
Length	9			2	
Value	→	mask8	validMask	1	Fields that are valid. Valid values: • QMI_LOC_SYS_TIME_BIAS_VALID (0x01) – System time bias is valid • QMI_LOC_SYS_TIME_BIAS_UNC_VALID (0x02) – System time bias uncertainty is valid
		float	timeBias	4	System 1 to System 2 time bias. • Units: Milliseconds
		float	timeBiasUnc	4	System 1 to System 2 time bias uncertainty. • Units: Milliseconds
Type	0x18			1	Satellite System Time Information for GPS, BDS, GAL Constellation
Length	18			2	
Value	→	enum	system	4	Specifies the satellite system constellation.
		uint16	systemWeek	2	Current system week. • For GPS: Calculated from midnight, Jan. 6, 1980 • For BDS: Calculated from 00:00:00 on January 1, 2006 of Coordinated Universal Time (UTC) • For GAL: Calculated from 00:00 UT on Sunday August 22, 1999 (midnight between August 21 and August 22) If the week is unknown, set this value to 65535. • Units: Weeks
		uint32	systemMsec	4	Amount of time into the current week. • Units: Milliseconds
		float	systemClkTimeBias	4	System clock time bias (submilliseconds). • Units: Milliseconds (system time = systemMsec - systemClkTimeBias)
		float	systemClkTimeUncMs	4	Single-sided maximum time bias uncertainty. • Units: Milliseconds
Type	0x19			1	GLONASS System Time Information

Field	Field value	Field type	Parameter	Size (byte)	Description
Length	15			2	
Value	→	uint8	gloFourYear	1	GLONASS four year number from 1996. Refer to GLONASS ICD. Applicable only for GLONASS and is to be ignored for other constellations. If unknown, set this value to 255.
		uint16	gloDays	2	GLONASS day number in four years. Refer to GLONASS ICD. Applicable only for GLONASS and is to be ignored for other constellations. If unknown, set this value to 65535.
		uint32	gloMsec	4	GLONASS time of day in msec. Refer to GLONASS ICD. • Units: Milliseconds
		float	gloClkTimeBias	4	System clock time bias (submillisecond). • Units: Milliseconds (system time = systemMsec - systemClkTimeBias)
		float	gloClkTimeUncMs	4	Single-sided maximum time bias uncertainty. • Units: Milliseconds
Type	0x1A			1	Extended Time Information
Length	17			2	
Value	→	uint32	refFCount	4	Receiver frame counter value at a reference tick.
		boolean	systemRtc_valid	1	Validity indicator for the system RTC.
		uint64	systemRtcMs	8	Platform system RTC value. • Units: Milliseconds
		enum	sourceOfTime	4	Source of the time information. Valid values: • eQMI_LOC_TIME_SRC_INVALID (0) – Invalid time. • eQMI_LOC_TIME_SRC_NETWORK_TIME_TRANSFER (1) – Time is set by the 1X system • eQMI_LOC_TIME_SRC_NETWORK_TIME_TAGGING (2) – Time is set by WCDMA/GSM time tagging (i.e., associating network time with GPS time) • eQMI_LOC_TIME_SRC_EXTERNAL_INPUT (3) – Time is set by an external injection • eQMI_LOC_TIME_SRC_TOW_DECODE (4) – Time is set after decoding over-the-air GPS navigation data from one GPS satellite

Field	Field value	Field type	Parameter	Size (byte)	Description
			sourceOfTime (cont.)		<ul style="list-style-type: none"> • eQMI_LOC_TIME_SRC_TOW_CONFIRMED (5) – Time is set after decoding over-the-air GPS navigation data from multiple satellites • eQMI_LOC_TIME_SRC_TOW_AND_WEEK_CONFIRMED (6) – Both time of the week and the GPS week number are known • eQMI_LOC_TIME_SRC_NAV_SOLUTION (7) – Time is set by the position engine after the fix is obtained • eQMI_LOC_TIME_SRC_SOLVE_FOR_TIME (8) – Time is set by the position engine after performing SFT; this is done when the clock time uncertainty is large • eQMI_LOC_TIME_SRC_GLO_TOW_DECODE (9) – Time is set after decoding GLO satellites • eQMI_LOC_TIME_SRC_TIME_TRANSFORM (10) – Time is set after transforming the GPS to GLO time • eQMI_LOC_TIME_SRC_WCDMA_SLEEP_TIME_TAGGING (11) – Time is set by the sleep time tag provided by the WCDMA network • eQMI_LOC_TIME_SRC_GSM_SLEEP_TIME_TAGGING (12) – Time is set by the sleep time tag provided by the GSM network • eQMI_LOC_TIME_SRC_UNKNOWN (13) – Source of the time is unknown • eQMI_LOC_TIME_SRC_SYSTEM_TIMETICK (14) – Time is derived from the system clock (better known as the slow clock); GNSS time is maintained irrespective of the GNSS receiver state • eQMI_LOC_TIME_SRC_QZSS_TOW_DECODE (15) – Time is set after decoding QZSS satellites • eQMI_LOC_TIME_SRC_BDS_TOW_DECODE (16) – Time is set after decoding BDS satellites
Type	0x1B			1	Satellite System Measurement Report for Enabled Constellation
Length	Var			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	uint8	svMeasurement_len	1	Number of sets of the following elements: <ul style="list-style-type: none"> • gnssSvId • gloFrequency • svStatus • validMask • healthStatus • svInfoMask • validMeasStatusMask • measurementStatus • CNo • gloRfLoss • measLatency • svTimeMs • svTimeSubMs • svTimeUncMs • dopplerShift • dopplerShiftUnc • dopplerAccel_valid • dopplerAccel • lossOfLock • multipathEstimate • fineSpeed • fineSpeedUnc • carrierPhase • cycleSlipCount • svAzimuth • svElevation
		uint16	gnssSvId	2	GNSS SV ID. <ul style="list-style-type: none"> • Range: <ul style="list-style-type: none"> – For GPS: 1 to 32 – For SBAS: 33 to 64 – For GLONASS: 65 to 96. When slot-number to SV ID mapping is unknown, set as 255. – For BDS: 201 to 237
		uint8	gloFrequency	1	GLONASS frequency number + 7. Valid only for a GLONASS system and is to be ignored for all other systems. <ul style="list-style-type: none"> • Range: 1 to 14
		enum	svStatus	4	Satellite search state. Valid values: <ul style="list-style-type: none"> • eQMI_LOC_SV_STATUS_IDLE (1) – SV is not being actively processed • eQMI_LOC_SV_STATUS_SEARCH (2) – The system is searching for this SV • eQMI_LOC_SV_STATUS_TRACK (3) – SV is being tracked

Field	Field value	Field type	Parameter	Size (byte)	Description
		mask16	validMask	2	Validity mask (0 = Not valid; 1 = Valid). Valid masks: <ul style="list-style-type: none"> • QMI_LOC_SV_HEALTH_VALID (0x01) – SV health information is valid • QMI_LOC_SV_MULTIPATH_EST_VALID (0x02) – Multipath estimate for SV is valid • QMI_LOC_SV_FINE_SPEED_VALID (0x04) – Fine speed for SV is valid • QMI_LOC_SV_FINE_SPEED_UNC_VALID (0x08) – Fine speed uncertainty for SV is valid • QMI_LOC_SV_CARRIER_PHASE_VALID (0x10) – Carrier phase for SV is valid • QMI_LOC_SV_SV_DIRECTION_VALID (0x20) – SV direction information for SV is valid • QMI_LOC_SV_CYCLESリップ_COUNT_VALID (0x40) – Cycle slip count information is valid • QMI_LOC_SV_LOSSOFLOCK_VALID (0x80) – Loss of lock information is valid
		uint8	healthStatus	1	Health status. <ul style="list-style-type: none"> • Range: 0 to 1, where 0 = unhealthy, 1 = healthy
		mask8	svInfoMask	1	Indicates whether almanac and ephemeris information is available. Valid values: <ul style="list-style-type: none"> • QMI_LOC_SVINFO_MASK_HAS_EPHEMERIS (0x01) – Ephemeris is available for this SV • QMI_LOC_SVINFO_MASK_HAS_ALMANAC (0x02) – Almanac is available for this SV

Field	Field value	Field type	Parameter	Size (byte)	Description
		mask	validMeasStatusMask	8	<p>Validity mask for measurement status information.</p> <p>A set bit in validMeasStatusMask indicates that the corresponding bit in measurementStatus has valid status information:</p> <p>Valid bitmask:</p> <ul style="list-style-type: none"> • QMI_LOC_MASK_MEAS_STATUS_SM_STAT_BIT_VALID (0x00000001) – Satellite time in submilliseconds (code phase) • QMI_LOC_MASK_MEAS_STATUS_SB_STAT_BIT_VALID (0x00000002) – Satellite sub-bit time • QMI_LOC_MASK_MEAS_STATUS_MS_STAT_BIT_VALID (0x00000004) – Satellite time in milliseconds • QMI_LOC_MASK_MEAS_STATUS_BE_CONFIRM_STAT_BIT_VALID (0x00000008) – Signal bit edge is confirmed • QMI_LOC_MASK_MEAS_STATUS_VEL_STAT_BIT_VALID (0x00000010) – Satellite Doppler • QMI_LOC_MASK_MEAS_STATUS_VEL_FINE_STAT_BIT_VALID (0x00000020) – Fine/coarse Doppler measurement indicator • QMI_LOC_MASK_MEAS_STATUS_FROM_RNG_DIFF_STAT_BIT_VALID (0x00000200) – Range update from satellite differences • QMI_LOC_MASK_MEAS_STATUS_FROM_VE_DIFF_STAT_BIT_VALID (0x00000400) – Doppler update from satellite differences <p>Additionally, MSB 0xFFC0000000000000 bits indicate the validity of DONT_USE bits.</p>

Field	Field value	Field type	Parameter	Size (byte)	Description
		mask	measurementStatus	8	<p>Bitmask indicating the SV measurement status.</p> <p>Valid bitmasks:</p> <ul style="list-style-type: none"> • QMI_LOC_MASK_MEAS_STATUS_SM_VALID (0x00000001) – Satellite time in submilliseconds (code phase) is known • QMI_LOC_MASK_MEAS_STATUS_SB_VALID (0x00000002) – Satellite sub-bit time is known • QMI_LOC_MASK_MEAS_STATUS_MS_VALID (0x00000004) – Satellite time in milliseconds is known • QMI_LOC_MASK_MEAS_STATUS_BE_CONFIRM (0x00000008) – Signal bit edge is confirmed • QMI_LOC_MASK_MEAS_STATUS_VELOCITY_VALID (0x00000010) – Satellite Doppler is measured • QMI_LOC_MASK_MEAS_STATUS_VELOCITY_FINE (0x00000020) – TRUE: Fine Doppler is measured, FALSE: Coarse Doppler is measured • QMI_LOC_MASK_MEAS_STATUS_FROM_RNG_DIFF (0x00000200) – Range update from satellite differences is measured • QMI_LOC_MASK_MEAS_STATUS_FROM_VE_DIFF (0x00000400) – Doppler update from satellite differences is measured <p>If any MSB bit in 0xFFC0000000000000 DONT_USE is set, the measurement must not be used by the client.</p>
		uint16	CNo	2	<p>Carrier to noise ratio.</p> <ul style="list-style-type: none"> • Units: dBHz • Scale: 0.1
		uint16	gloRfLoss	2	<p>GLONASS RF loss reference to the antenna.</p> <ul style="list-style-type: none"> • Units: dB • Scale: 0.1
		int32	measLatency	4	<p>Age of the measurement. A positive value means the measurement precedes the reference time.</p> <ul style="list-style-type: none"> • Units: Milliseconds

Field	Field value	Field type	Parameter	Size (byte)	Description
		uint32	svTimeMs	4	Satellite time in milliseconds. • For GPS, BDS, and GAL: Range is 0 thru (604800000-1) • For GLONASS: Range is 0 thru (86400000-1) • Units: Milliseconds This is valid when the QMI_LOC_MEAS_STATUS_MS_VALID bit is set in the measurement status. Note: All SV times in the current measurement block are already propagated to a common reference time epoch.
		float	svTimeSubMs	4	Satellite time in submilliseconds. Total SV Time = svMs + svSubMs • Units: Milliseconds
		float	svTimeUncMs	4	Satellite time uncertainty. • Units: Milliseconds
		float	dopplerShift	4	Satellite Doppler. • Units: Meters per second
		float	dopplerShiftUnc	4	Satellite Doppler uncertainty. • Units: Meters per second
		boolean	dopplerAccel_valid	1	Validity for Doppler acceleration.
		float	dopplerAccel	4	Satellite Doppler acceleration. • Units: Hz/second
		boolean	lossOfLock	1	Loss of signal lock indicator. • 0: Signal is in continuous track • 1: Signal is not in track
		float	multipathEstimate	4	Estimate of multipath in a measurement. • Units: Meters
		float	fineSpeed	4	Carrier phase derived speed. • Units: Meters per second
		float	fineSpeedUnc	4	Carrier phase derived speed uncertainty. • Units: Meters per second
		double	carrierPhase	8	Carrier phase measurement (L1 cycles).
		uint8	cycleSlipCount	1	Increments when a cycle slip is detected.
		float	svAzimuth	4	Satellite azimuth. • Units: Radians • Range: 0 to 2*pi()
		float	svElevation	4	Satellite elevation. • Units: Radians • Range: 0 to pi()/2

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

3.106.2 Description of QMI_LOC_EVENT_GNSS_MEASUREMENT_-REPORT_IND

This event report is used to send the satellite measurement, system clock, and intersystem bias information to the control point. The satellite measurement 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. The satellite reports are sent only to the control point that sent the QMI_LOC_START message that generated the satellite report.

On every reporting instant/time-epoch, based on enabled satellite constellations (using QMI_LOC_SET_GNSS_CONSTELL_REPORT_CONFIG), one or more QMI_LOC_EVENT_GNSS_MEASUREMENT_REPORT_IND are sent, one per constellation. Such multiple indication reporting is communicated using seqNum (current part number) and maxMessageNum (of total parts to arrive). A client that needs multiple constellation information for its function must await arrival of relevant indications before processing.

3.107 QMI_LOC_EVENT_SV_POLYNOMIAL_REPORT_IND

Sends a satellite polynomial report to the control point.

LOC message ID

0x0087

Version introduced

Major - 2, Minor - 31

3.107.1 Indication - QMI_LOC_EVENT_SV_POLYNOMIAL_REPORT_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version introduced	Version last modified
GNSS SV Polynomial Report	2.31	2.31
Reference Time for Polynomial Calculation	2.31	2.31
SV Polynomial Validity Status	2.31	2.31
SV Polynomial Report Status	2.31	2.31

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	GNSS SV Polynomial Report
Length	2			2	
Value	→	uint16	gnssSvId	2	GNSS SV ID. <ul style="list-style-type: none"> Range: <ul style="list-style-type: none"> For GPS: 1 to 32 For SBAS: 33 to 64 For GLONASS: 65 to 96 (when the slot number to SV ID mapping is unknown, set to 255) For BDS: 201 to 237
Type	0x02			1	Reference Time for Polynomial Calculation
Length	8			2	
Value	→	double	T0	8	Reference time for polynomial calculations. <ul style="list-style-type: none"> GPS: Seconds in the week GLO: Full seconds since Jan. 1, 1996 BDS: Full seconds since Jan. 1, 2006

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x03			1	SV Polynomial Validity Status
Length	2			2	
Value	→	mask16	svPolyFlagValid	2	Validity mask for bits in svPolyFlags. A set bit in svPolyFlagValid indicates that a corresponding bit in svPolyFlags has valid status information. Valid bitmasks: <ul style="list-style-type: none"> • QMI_LOC_SV_POLY_SRC_ALM_CORR_VALID (0x01) – Validity status for QMI_LOC_SV_POLY_SRC_ALM_CORR • QMI_LOC_SV_POLY_GLO_STR4_VALID (0x02) – Validity status for QMI_LOC_SV_POLY_GLO_STR4
Type	0x04			1	SV Polynomial Report Status
Length	2			2	
Value	→	mask16	svPolyFlags	2	Flags indicating the status of a polynomial report. Valid bitmasks: <ul style="list-style-type: none"> • QMI_LOC_SV_POLY_SRC_ALM_CORR (0x01) – Polynomials based on XTRA • QMI_LOC_SV_POLY_GLO_STR4 (0x02) – GLONASS string 4 has been received

Optional TLVs

Name	Version introduced	Version last modified
Polynomial Coefficient's 0th Term for X, Y, and Z Coordinates	2.31	2.31
Polynomial Coefficient's 1st, 2nd, and 3rd Terms for X, Y, and Z Coordinates	2.31	2.31
Polynomial Coefficients for Satellite Clock Bias Correction	2.31	2.31
GLONASS Frequency Number	2.31	2.31
Ephemeris Reference Time	2.31	2.31
Enhanced Reference Time	2.31	2.31
SV Position Uncertainty	2.31	2.31
Iono Delay	2.31	2.31
Iono Delay Rate	2.31	2.31
SBAS Iono Delay	2.31	2.31
SBAS Iono Delay Rate	2.31	2.31
Tropospheric Delay	2.31	2.31
Satellite Elevation	2.31	2.31
Satellite Elevation Rate	2.31	2.31

Name	Version introduced	Version last modified
Satellite Elevation Uncertainty	2.31	2.31
Polynomial Coefficients for SV Velocity	2.31	2.31

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Polynomial Coefficient's 0th Term for X, Y, and Z Coordinates
Length	24			2	
Value	→	double	polyCoeffXYZ0	24	Polynomial coefficient's 0th term for X, Y, and Z coordinates (C0X, C0Y, C0Z). • Units: Meters
Type	0x11			1	Polynomial Coefficient's 1st, 2nd, and 3rd Terms for X, Y, and Z Coordinates
Length	72			2	
Value	→	double	polyCoefXYZN	72	Polynomial coefficients 1st, 2nd, and 3rd terms for X, Y, and Z coordinates (C1X, C2X,... C2Z, C3Z). • Units: – 1st term – Meters/seconds – 2nd term – Meters/seconds ² – 3rd term – Meters/seconds ³
Type	0x12			1	Polynomial Coefficients for Satellite Clock Bias Correction
Length	16			2	
Value	→	float	polyCoefClockBias	16	Polynomial coefficients for satellite clock bias correction (C0T, C1T, C2T, C3T). • Units: – 0th term – Milliseconds/seconds – 1st term – Milliseconds/seconds ² – 2nd term – Milliseconds/seconds ³ – 3rd term – Milliseconds/seconds ⁴
Type	0x13			1	GLONASS Frequency Number
Length	1			2	
Value	→	uint8	gloFrequency	1	GLONASS frequency number + 7. Valid only for GLONASS systems and must be ignored for all other systems. • Range: 1 to 14
Type	0x14			1	Ephemeris Reference Time
Length	2			2	
Value	→	uint16	IODE	2	Ephemeris reference time. • GPS – Issue of data ephemeris used (unitless) • GLONASS – Tb 7-bit
Type	0x15			1	Enhanced Reference Time
Length	4			2	
Value	→	uint32	enhancedIOD	4	For BDS ephemeris, this is TOE.

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x16			1	SV Position Uncertainty
Length	4			2	
Value	→	float	svPosUnc	4	SV position uncertainty. • Units: Meters
Type	0x17			1	Iono Delay
Length	4			2	
Value	→	float	ionoDelay	4	Ionospheric delay at T0. • Units: Meters
Type	0x18			1	Iono Delay Rate
Length	4			2	
Value	→	float	ionoDot	4	Ionospheric delay rate. • Units: Meters/second
Type	0x19			1	SBAS Iono Delay
Length	4			2	
Value	→	float	sbasIonoDelay	4	SBAS ionospheric delay at T0. • Units: Meters
Type	0x1A			1	SBAS Iono Delay Rate
Length	4			2	
Value	→	float	sbasIonoDot	4	SBAS ionospheric delay rate. • Units: Meters/second
Type	0x1B			1	Tropospheric Delay
Length	4			2	
Value	→	float	tropoDelay	4	Tropospheric delay. • Units: Meters
Type	0x1C			1	Satellite Elevation
Length	4			2	
Value	→	float	elevation	4	Satellite elevation at T0. • Units: Radians
Type	0x1D			1	Satellite Elevation Rate
Length	4			2	
Value	→	float	elevationDot	4	Satellite elevation rate. • Units: Radians/second
Type	0x1E			1	Satellite Elevation Uncertainty
Length	4			2	
Value	→	float	elevationUnc	4	SV elevation uncertainty. • Units: Radians
Type	0x1F			1	Polynomial Coefficients for SV Velocity
Length	96			2	
Value	→	double	velCoef	96	Polynomial coefficients for SV velocity (C0X, C1X, C2X, C3X,... C2Z, C3Z). • Units: – 0th term – Meters/seconds – 1st term – Meters/seconds ² – 2nd term – Meters/seconds ³ – 3rd term – Meters/seconds 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

3.107.2 Description of QMI_LOC_EVENT_SV_POLYNOMIAL_REPORT_IND

This event report is used to send the satellite position report in the form of a polynomial. The satellite polynomial 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. The satellite reports are sent only to the control point that sent the QMI_LOC_START message that generated the satellite report.

3.108 QMI_LOC_SET_GNSS_CONSTELL_REPORT_CONFIG

Sets satellite constellations of interest for reporting.

LOC message ID

0x0088

Version introduced

Major - 2, Minor - 31

3.108.1 Request - QMI_LOC_SET_GNSS_CONSTELL_REPORT_CONFIG

Message type

Request

Sender

Control point

Mandatory TLVs

None

Optional TLVs

Name	Version introduced	Version last modified
GNSS Measurement Report Constellation Control	2.31	2.31
SV Polynomial Report Constellation Control	2.31	2.31

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	GNSS Measurement Report Constellation Control
Length	8			2	
Value	→	mask	measReportConfig	8	GNSS measurement report constellation control. Valid values: <ul style="list-style-type: none"> • eQMI_SYSTEM_GPS (0x01) – Enable GPS • eQMI_SYSTEM_GLO (0x02) – Enable GLONASS • eQMI_SYSTEM_BDS (0x04) – Enable BDS • eQMI_SYSTEM_GAL (0x08) – Enable Galileo
Type	0x11			1	SV Polynomial Report Constellation Control

Field	Field value	Field type	Parameter	Size (byte)	Description
Length	8			2	
Value	→	mask	svPolyReportConfig	8	SV polynomial report constellation control. Valid values: <ul style="list-style-type: none"> • eQMI_SYSTEM_GPS (0x01) – Enable GPS • eQMI_SYSTEM_GLO (0x02) – Enable GLONASS • eQMI_SYSTEM_BDS (0x04) – Enable BDS • eQMI_SYSTEM_GAL (0x08) – Enable Galileo

3.108.2 Indication - QMI_LOC_SET_GNSS_CONSTELL_REPORT_CONFIG_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version introduced	Version last modified
Set GNSS Constellation Status	2.31	2.31

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Set GNSS Constellation Status
Length	4			2	
Value	→	enum	status	4	Status of the GNSS constellation. Valid values: <ul style="list-style-type: none"> • eQMI_LOC_SUCCESS (0) – Request was completed successfully • eQMI_LOC_GENERAL_FAILURE (1) – Request failed because of a general failure • eQMI_LOC_UNSUPPORTED (2) – Request failed because it is not supported • eQMI_LOC_INVALID_PARAMETER (3) – Request failed because it contained invalid parameters

Field	Field value	Field type	Parameter	Size (byte)	Description
			status (cont.)		<ul style="list-style-type: none"> • eQMI_LOC_ENGINE_BUSY (4) – Request failed because the engine is busy • eQMI_LOC_PHONE_OFFLINE (5) – Request failed because the phone is offline • eQMI_LOC_TIMEOUT (6) – Request failed because it timed out • eQMI_LOC_CONFIG_NOT_SUPPORTED (7) – Request failed because an undefined configuration was requested • eQMI_LOC_INSUFFICIENT_MEMORY (8) – Request failed because the engine could not allocate sufficient memory for the request • eQMI_LOC_MAX_GEOFENCE_PROGRAMMED (9) – Request failed because the maximum number of Geofences are already programmed • eQMI_LOC_XTRA_VERSION_CHECK_FAILURE (10) – Location service failed because of an XTRA version-based file format check failure

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

3.108.3 Description of QMI_LOC_SET_GNSS_CONSTELL_REPORT_CONFIG

This command is used to set GNSS constellations of interest for reporting purposes. The request is acknowledged through the response, but the SUCCESS/FAILURE status is sent through the indication QMI_LOC_SET_GNSS_CONFIG_IND. Only one client may control a constellation, since the configuration significantly impacts the operation of all clients.

3.109 QMI_LOC_ADD_GEOFENCE_CONTEXT

Used by the control point to inject the Geofence context.

LOC message ID

0x0089

Version introduced

Major - 2, Minor - 32

3.109.1 QMI_LOC_EVENT_SV_POLYNOMIAL_REPORT_IND

Message type

Sender

Mandatory TLVs

Name	Version introduced	Version last modified
Transaction ID	2.32	2.32

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Transaction ID
Length	4			2	
Value	→	uint32	transactionId	4	Identifies the transaction. The transaction ID is returned in the Add Geofence context indication.

Optional TLVs

Name	Version introduced	Version last modified
Geofence ID	2.32	2.32
Wi-Fi AP SSID String	2.32	2.32
Wi-Fi AP MAC Address List for the Geofence	2.32	2.32
TDSCDMA Cell ID List for the Geofence	2.32	2.32
WCDMA Cell ID List for the Geofence	2.32	2.32
GSM Cell ID List for the Geofence	2.32	2.32
IBeacon List of the Geofence	2.32	2.32

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Geofence ID
Length	4			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	uint32	geofenceId	4	Geofence identifier allocated by the engine. If the Geofence ID is not provided, a Geofence is created with an Area ID list only (e.g., Wi-Fi only list Geofence). If the Geofence ID is provided, the added list is used as assistance data to the existing Geofence.
Type	0x11			1	Wi-Fi AP SSID String The ordering of the Wi-Fi AP SSID list should match the Wi-Fi AP MAC address list when both are provided, i.e., the first element of the Wi-Fi AP SSID list must be the SSID of the AP whose MAC address is in the first element in the Wi-Fi AP MAC address, etc.
Length	Var			2	
Value	→	uint8	wifiApSsidInfo_len	1	Number of sets of the following elements: • ssid_len • ssid
		uint8	ssid_len	1	Number of sets of the following elements: • ssid
		string	ssid	Var	NULL-terminated SSID string of the Wi-Fi AP. Its maximum length according to the ASCII standard is 32 octets.
Type	0x12			1	Wi-Fi AP MAC Address List for the Geofence The ordering of the Wi-Fi AP SSID list should match the Wi-Fi AP MAC address list when both are provided, i.e., the first element of the Wi-Fi AP SSID list must be the SSID of the AP whose MAC address is in the first element in the Wi-Fi AP MAC address, etc.
Length	Var			2	
Value	→	uint8	wifiApMacAddressList_len	1	Number of sets of the following elements: • wifiApMacAddress
		uint8	wifiApMacAddress	6	MAC address of the Wi-Fi AP.
Type	0x13			1	TDSCDMA Cell ID List for the Geofence Identifies the TDSCDMA cell on which the device is currently camped.
Length	320			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	uint32	mcc	4	TDSCDMA mobile country code. Refer to ITU-T E.212 [S6].
		uint32	mnc	4	TDSCDMA mobile network code. Refer to [S6].
		uint32	cid	4	TDSCDMA cell identity. Refer to TS 25.331 [S7].
		uint32	lac	4	TDSCDMA location area code. Refer to [S6].
Type	0x14			1	WCDMA Cell ID List for the Geofence Identifies the WCDMA cell on which the device is currently camped.
Length	Var			2	
Value	→	uint8	wcdmaCellIDList_len	1	Number of sets of the following elements: • mcc • mnc • cid
		uint32	mcc	4	WCDMA mobile country code. Refer to ITU-T E.212 [S6].
		uint32	mnc	4	WCDMA mobile network code. Refer to [S6].
		uint32	cid	4	WCDMA cell identity. Refer to [S6].
Type	0x15			1	GSM Cell ID List for the Geofence Identifies the GSM cell on which the device is currently camped.
Length	Var			2	
Value	→	uint8	gsmCellIDList_len	1	Number of sets of the following elements: • MCC • MNC • LAC • CID
		uint32	MCC	4	GSM mobile country code. Refer to ITU-T E.212 [S6].
		uint32	MNC	4	GSM mobile network code. Refer to [S6].
		uint32	LAC	4	GSM location area code. Refer to [S6].
		uint32	CID	4	GSM cell identification. Refer to [S6].
Type	0x16			1	iBeacon List of the Geofence
Length	Var			2	
Value	→	uint8	iBeaconList_len	1	Number of sets of the following elements: • uuid_len • uuid • majorNumber • minorNumber

Field	Field value	Field type	Parameter	Size (byte)	Description
		uint8	uuid_len	1	Number of sets of the following elements: • uuid
		string	uuid	Var	NULL-terminated IBeacon identifier string; a 128-bit value.
		uint32	majorNumber	4	IBeacon major number.
		uint32	minorNumber	4	IBeacon minor number.

3.109.2 Indication - QMI_LOC_ADD_GEOFENCE_CONTEXT_IND

Message type

Indication

Sender

Control point

Mandatory TLVs

Name	Version introduced	Version last modified
Status of the Add Geofence Context Request	2.32	2.32

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Status of the Add Geofence Context Request
Length	4			2	
Value	→	enum	status	4	Status of the Add Geofence Context request. Valid values: • eQMI_LOC_SUCCESS (0) – Request was completed successfully • eQMI_LOC_GENERAL_FAILURE (1) – Request failed because of a general failure • eQMI_LOC_UNSUPPORTED (2) – Request failed because it is not supported • eQMI_LOC_INVALID_PARAMETER (3) – Request failed because it contained invalid parameters • eQMI_LOC_ENGINE_BUSY (4) – Request failed because the engine is busy • eQMI_LOC_PHONE_OFFLINE (5) – Request failed because the phone is offline

Field	Field value	Field type	Parameter	Size (byte)	Description
			status (cont.)		<ul style="list-style-type: none"> • eQMI_LOC_TIMEOUT (6) – Request failed because it timed out • eQMI_LOC_CONFIG_NOT_SUPPORTED (7) – Request failed because an undefined configuration was requested • eQMI_LOC_INSUFFICIENT_MEMORY (8) – Request failed because the engine could not allocate sufficient memory for the request • eQMI_LOC_MAX_GEOFENCE_PROGRAMMED (9) – Request failed because the maximum number of Geofences are already programmed • eQMI_LOC_XTRA_VERSION_CHECK_FAILURE (10) – Location service failed because of an XTRA version-based file format check failure

Optional TLVs

Name	Version introduced	Version last modified
Transaction ID	2.32	2.32
Geofence ID	2.32	2.32
Context ID	2.32	2.32

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Transaction ID
Length	4			2	
Value	→	uint32	transactionId	4	Transaction ID that was specified in the Add Geofence Context request. This parameter is always present if the status field is set to SUCCESS.
Type	0x11			1	Geofence ID
Length	4			2	
Value	→	uint32	geofenceId	4	Geofence identifier allocated by the engine. If the client specifies the Geofence ID during the Add Geofence Context request, the same ID is returned. If the client does not specify the Geofence ID during the Add Geofence Context request, a new Geofence ID is created by the Geofence engine and returned.

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x12			1	Context ID
Length	4			2	
Value	→	uint32	contextId	4	Geofence context ID allocated by the engine. The context ID is generated by the Geofence engine to identify the context for a particular Geofence ID. The same Geofence ID may be associated with multiple contexts.

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

3.109.3 Description of QMI_LOC_ADD_GEOFENCE_CONTEXT

This command is used to add Geofence context to an existing Geofence or to create a new Geofence with associated context data.

If the Geofence ID is not provided in the request message, a Geofence is created with the area ID list only (e.g., Wi-Fi only list Geofence).

If the Geofence ID is provided in the request message, the added list is used as context data to the existing Geofence.

This command can be safely used by multiple clients. However, there can only be one request outstanding at any time.

3.110 QMI_LOC_SET_GEOFENCE_ENGINE_CONTEXT

Used by the control point to inject the Geofence engine context.

LOC message ID

0x008A

Version introduced

Major - 2, Minor - 32

3.110.1 - QMI_LOC_ADD_GEOFENCE_CONTEXT_IND

Message type

Sender

Mandatory TLVs

Name	Version introduced	Version last modified
Transaction ID	2.32	2.32

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Transaction ID
Length	4			2	
Value	→	uint32	transactionId	4	Identifies the transaction. The transaction ID is returned in the Set Geofence Engine Context indication.

Optional TLVs

Name	Version introduced	Version last modified
UTC Timestamp of the Day	2.32	2.32
Temperature of the Day in Fahrenheit	2.32	2.32

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	UTC Timestamp of the Day
Length	8			2	
Value	→	uint64	utcTimeOfDay	8	The UTC time of the day.
Type	0x11			1	Temperature of the Day in Fahrenheit
Length	4			2	
Value	→	int32	temperature	4	The temperature of the day in degrees Fahrenheit.

3.110.2 Indication - QMI_LOC_SET_GEOFENCE_ENGINE_CONTEXT_IND

Message type

Indication

Sender

Control point

Mandatory TLVs

Name	Version introduced	Version last modified
Status of the Set Geofence Engine Context Request	2.32	2.32

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Status of the Set Geofence Engine Context Request
Length	4			2	
Value	→	enum	status	4	<p>Status of the Set Geofence Engine Context request.</p> <p>Valid values:</p> <ul style="list-style-type: none"> • eQMI_LOC_SUCCESS (0) – Request was completed successfully • eQMI_LOC_GENERAL_FAILURE (1) – Request failed because of a general failure • eQMI_LOC_UNSUPPORTED (2) – Request failed because it is not supported • eQMI_LOC_INVALID_PARAMETER (3) – Request failed because it contained invalid parameters • eQMI_LOC_ENGINE_BUSY (4) – Request failed because the engine is busy • eQMI_LOC_PHONE_OFFLINE (5) – Request failed because the phone is offline • eQMI_LOC_TIMEOUT (6) – Request failed because it timed out • eQMI_LOC_CONFIG_NOT_SUPPORTED (7) – Request failed because an undefined configuration was requested • eQMI_LOC_INSUFFICIENT_MEMORY (8) – Request failed because the engine could not allocate sufficient memory for the request

Field	Field value	Field type	Parameter	Size (byte)	Description
			status (cont.)		<ul style="list-style-type: none"> • eQMI_LOC_MAX_GEOFENCE_PROGRAMMED (9) – Request failed because the maximum number of Geofences are already programmed • eQMI_LOC_XTRA_VERSION_CHECK_FAILURE (10) – Location service failed because of an XTRA version-based file format check failure

Optional TLVs

Name	Version introduced	Version last modified
Transaction ID	2.32	2.32

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Transaction ID
Length	4			2	
Value	→	uint32	transactionId	4	Transaction ID that was specified in the Set Geofence Engine Context request. This parameter will always be present if the status field is set to SUCCESS.

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

3.110.3 Description of QMI_LOC_SET_GEOFENCE_ENGINE_CONTEXT

This command is used to set the Geofence engine context. This context is applicable to all Geofences that are present in the engine. Multiple clients must not set contexts that conflict with each other, since these apply to all Geofences in the system.

3.111 QMI_LOC_DELETE_GEOFENCE_CONTEXT

Used by the control point to Delete the geofence context.

LOC message ID

0x008B

Version introduced

Major - 2, Minor - 32

3.111.1 QMI_LOC_DELETE_GEOFENCE_CONTEXT_IND

Message type

Sender

Mandatory TLVs

Name	Version introduced	Version last modified
Transaction ID	2.32	2.32
Geofence ID	2.32	2.32

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Transaction ID
Length	4			2	
Value	→	uint32	transactionId	4	Identifies the transaction. The transaction ID is returned in the Delete Geofence Context indication.
Type	0x02			1	Geofence ID
Length	4			2	
Value	→	uint32	geofenceId	4	Identifies the Geofence whose context is to be deleted.

Optional TLVs

Name	Version introduced	Version last modified
Context ID	2.32	2.32

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Context ID
Length	4			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	uint32	contextId	4	Identifies the context associated with the Geofence to be deleted. If not specified, all contexts associated with this Geofence are deleted.

3.111.2 Indication - QMI_LOC_DELETE_GEOFENCE_CONTEXT_IND

Message type

Indication

Sender

Control point

Mandatory TLVs

Name	Version introduced	Version last modified
Status of the Delete Geofence Context Request	2.32	2.32

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Status of the Delete Geofence Context Request
Length	4			2	
Value	→	enum	status	4	Status of the Delete Geofence Context request. Valid values: <ul style="list-style-type: none"> • eQMI_LOC_SUCCESS (0) – Request was completed successfully • eQMI_LOC_GENERAL_FAILURE (1) – Request failed because of a general failure • eQMI_LOC_UNSUPPORTED (2) – Request failed because it is not supported • eQMI_LOC_INVALID_PARAMETER (3) – Request failed because it contained invalid parameters • eQMI_LOC_ENGINE_BUSY (4) – Request failed because the engine is busy • eQMI_LOC_PHONE_OFFLINE (5) – Request failed because the phone is offline • eQMI_LOC_TIMEOUT (6) – Request failed because it timed out

Field	Field value	Field type	Parameter	Size (byte)	Description
			status (cont.)		<ul style="list-style-type: none"> • eQMI_LOC_CONFIG_NOT_SUPPORTED (7) – Request failed because an undefined configuration was requested • eQMI_LOC_INSUFFICIENT_MEMORY (8) – Request failed because the engine could not allocate sufficient memory for the request • eQMI_LOC_MAX_GEOFENCE_PROGRAMMED (9) – Request failed because the maximum number of Geofences are already programmed • eQMI_LOC_XTRA_VERSION_CHECK_FAILURE (10) – Location service failed because of an XTRA version-based file format check failure

Optional TLVs

Name	Version introduced	Version last modified
Transaction ID	2.32	2.32
Geofence ID	2.32	2.32
Context ID	2.32	2.32

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Transaction ID
Length	4			2	
Value	→	uint32	transactionId	4	Transaction ID that was specified in the Delete Geofence Context request. This parameter will always be present if the status field is set to SUCCESS.
Type	0x11			1	Geofence ID
Length	4			2	
Value	→	uint32	geofenceId	4	Identifier for the Geofence whose context was deleted.
Type	0x12			1	Context ID
Length	4			2	
Value	→	uint32	contextId	4	Identifier for the context of the Geofence that was deleted.

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

3.111.3 Description of QMI_LOC_DELETE_GEOFENCE_CONTEXT

This command is used to delete the Geofence context of an existing Geofence. The same Geofence may be associated with multiple contexts and each context may be deleted. Multiple clients must ensure that they do not delete Geofence context for which they are not responsible.

3.112 QMI_LOC_EVENT_GEOFENCE_PROXIMITY_-NOTIFICATION

Notifies the control point of a Geofence proximity event.

LOC message ID

0x008C

Version introduced

Major - 2, Minor - 32

3.112.1 Indication - QMI_LOC_EVENT_GEOFENCE_PROXIMITY_-NOTIFICATION_IND

Message type

Indication

Sender

Service

Mandatory TLVs

Name	Version introduced	Version last modified
Geofence Breach Type	2.32	2.32
Geofence ID	2.32	2.32

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Geofence Breach Type
Length	4			2	
Value	→	enum	proximityType	4	Valid values: <ul style="list-style-type: none"> • eQMI_LOC_GEOFENCE_PROXIMITY_TYPE_IN (1) – Denotes that a client is in proximity of the Geofence • eQMI_LOC_GEOFENCE_PROXIMITY_TYPE_OUT (2) – Denotes that a client is out of proximity of the Geofence
Type	0x02			1	Geofence ID
Length	4			2	
Value	→	uint32	geofenceId	4	Identifier of the Geofence that is in proximity to the handset.

Optional TLVs

Name	Version introduced	Version last modified
Geofence Context ID	2.32	2.32

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Geofence Context ID
Length	4			2	
Value	→	uint32	contextId	4	Identifier for the context of the Geofence to which the handset is in proximity. A single Geofence may be associated with different contexts.

3.112.2 Description of QMI_LOC_EVENT_GEOFENCE_PROXIMITY_NOTIFICATION

This command notifies the control point when a Geofence proximity is entered and exited. The proximity of a Geofence may be due to different contexts. These contexts are identified using the context ID in this indication. The context of a Geofence may contain Wi-Fi area ID lists, iBeacon lists, Cell ID list, etc.