



EC2x QMI Command

Reference Guide

LTE Standard Module Series

Rev: EC2x_QMI_Command_Reference_Guide

Date: 2020-02-18

Status: Preliminary

Our aim is to provide customers with timely and comprehensive service. For any assistance, please contact our company headquarters:

Quectel Wireless Solutions Co., Ltd.

Building 5, Shanghai Business Park Phase III (Area B), No.1016 Tianlin Road, Minhang District, Shanghai, China 200233

Tel: +86 21 5108 6236

Email: info@quectel.com

Or our local office. For more information, please visit:

<http://www.quectel.com/support/sales.htm>

For technical support, or to report documentation errors, please visit:

<http://www.quectel.com/support/technical.htm>

Or email to: support@quectel.com

GENERAL NOTES

QUECTEL OFFERS THE INFORMATION AS A SERVICE TO ITS CUSTOMERS. THE INFORMATION PROVIDED IS BASED UPON CUSTOMERS' REQUIREMENTS. QUECTEL MAKES EVERY EFFORT TO ENSURE THE QUALITY OF THE INFORMATION IT MAKES AVAILABLE. QUECTEL DOES NOT MAKE ANY WARRANTY AS TO THE INFORMATION CONTAINED HEREIN, AND DOES NOT ACCEPT ANY LIABILITY FOR ANY INJURY, LOSS OR DAMAGE OF ANY KIND INCURRED BY USE OF OR RELIANCE UPON THE INFORMATION. ALL INFORMATION SUPPLIED HEREIN IS SUBJECT TO CHANGE WITHOUT PRIOR NOTICE.

COPYRIGHT

THE INFORMATION CONTAINED HERE IS PROPRIETARY TECHNICAL INFORMATION OF QUECTEL WIRELESS SOLUTIONS CO., LTD. TRANSMITTING, REPRODUCTION, DISSEMINATION AND EDITING OF THIS DOCUMENT AS WELL AS UTILIZATION OF THE CONTENT ARE FORBIDDEN WITHOUT PERMISSION. OFFENDERS WILL BE HELD LIABLE FOR PAYMENT OF DAMAGES. ALL RIGHTS ARE RESERVED IN THE EVENT OF A PATENT GRANT OR REGISTRATION OF A UTILITY MODEL OR DESIGN.

Copyright © Quectel Wireless Solutions Co., Ltd. 2020. All rights reserved.

About the Document

Revision History

Revision	Date	Author	Description
1.0	2020-02-18	Robin	Initial

Contents

About the Document	3
--------------------------	---

1	Introduction	17
2	QMI.....	18
2.1.	QMI Framework	18
2.1.1.	MSM-TE Interconnection	19
2.1.2.	Logical device enumeration	21
2.1.3.	Control channel messaging protocol	21
2.1.3.1.	Endpoint model	22
2.1.3.2.	Usage	22
2.2.	QMI Multiplexing Protocol (QMUX).....	23
2.2.1.	Byte ordering	23
2.2.1.1.	QMUX version	24
2.2.1.2.	Multiplexing of QMI Services	24
2.2.2.	Rules for QMUX message handling	25
2.2.2.1.	Generating QMUX messages	25
2.2.2.2.	Receiving QMUX messages	25
2.2.3.	Client ID management	25
2.3.	QMI Generalized Service Message Protocol.....	25
2.3.1.	Service PDU Message Format	25
2.3.1.1.	Byte Ordering	26
2.3.1.2.	QMI transaction Structure	26
2.3.1.3.	QMI message structure.....	27
2.3.1.4.	QMI message parameter structure	28
2.3.2.	QMI message types	28
2.3.2.1.	Response	29
2.3.3.	Rules for transaction handling	29
2.3.4.	Rules for message handling	30
2.3.4.1.	General rules.....	30
2.3.4.2.	Generating requests.....	30
2.3.4.3.	Receiving requests.....	30
2.3.4.4.	Generating responses.....	30
2.3.4.5.	Receiving responses.....	31
2.3.4.6.	Generating indications	31
2.3.4.7.	Receiving indications	31
2.3.5.	Rules for parameter handling.....	31
2.3.6.	State variables	32
2.3.7.	Control Point arbitration	32
2.3.8.	QMI Service versioning	32
2.3.8.1.	Version format	32
2.3.8.2.	Learning QMI Service versions	32
2.3.8.3.	Service versioning rules	32

3 ofono-1.30 Contains QMI Service Type Values	34
4 Control Service (QMI_CTL)	35
4.1. Theory of Operation	35
4.1.1. Generalized QMI Service Compliance.....	35
4.1.2. CTL Service Type.....	35
4.1.3. Message Definition Template	35
4.1.3.1. Byte Ordering	36
4.1.3.2. QMI_CTL PDU	36
4.1.3.3. Response Message Result TLV.....	37
4.1.4. QMI_CTL Fundamental Concepts	37
4.1.4.1. QMI_CTL Control Point.....	37
4.1.4.2. QMI_CTL Service Version.....	37
4.1.4.3. QMI Link ID	37
4.1.4.4. Client ID Management	38
4.1.5. Service State Variables.....	38
4.1.5.1. State Variables Per Control Point	38
4.2. QMI_CTL Messages	38
4.2.1. QMI_CTL_SET_INSTANCE_ID.....	39
4.2.1.1. Request - QMI_CTL_SET_INSTANCE_ID_REQ	39
4.2.1.2. Response - QMI_CTL_SET_INSTANCE_ID_RESP	40
4.2.1.3. Description of QMI_CTL_SET_INSTANCE_ID REQ/RESP	41
4.2.2. QMI_CTL_GET_VERSION_INFO	41
4.2.2.1. Request-QMI_CTL_GET_VERSION_INFO_REQ.....	41
4.2.2.2. Response – QMI_CTL_GET_VERSION_INFO_RESP	42
4.2.2.3. Description of QMI_CTL_GET_VERSION_INFO REQ/RESP	43
4.2.3. QMI_CTL_GET_CLIENT_ID.....	44
4.2.3.1. Request-QMI_CTL_GET_CLIENT_ID_REQ.....	44
4.2.3.2. Response-QMI_CTL_GET_CLIENT_ID_RESP	44
4.2.3.3. Description of QMI_CTL_GET_CLIENT_ID REQ/RESP	45
4.2.4. QMI_CTL_RELEASE_CLIENT_ID	46
4.2.4.1. Request-QMI_CTL_RELEASE_CLIENT_ID_REQ.....	46
4.2.4.2. Response-QMI_CTL_RELEASE_CLIENT_ID_RESP	46
4.2.4.3. Description of QMI_CTL_RELEASE_CLIENT_ID REQ/RESP	47
4.2.5. QMI_CTL_REVOKE_CLIENT_ID_IND.....	47
4.2.5.1. Indication - QMI_CTL_REVOKE_CLIENT_ID_IND	48
4.2.5.2. Description of QMI_CTL_REVOKE_CLIENT_ID_IND	48
4.2.6. QMI_CTL_INVALID_CLIENT_ID_IND	49
4.2.6.1. Indication - QMI_CTL_INVALID_CLIENT_ID_IND	49
4.2.6.2. Description of QMI_CTL_INVALID_CLIENT_ID_IND	50
4.2.7. QMI_CTL_SET_DATA_FORMAT	50
4.2.7.1. Request - QMI_CTL_SET_DATA_FORMAT_REQ.....	50
4.2.7.2. Response - QMI_CTL_SET_DATA_FORMAT_RESP.....	51
4.2.7.3. Description of QMI_CTL_SET_DATA_FORMAT REQ/RESP	53
4.2.8. QMI_CTL_SYNC.....	54

4.2.8.1.	Request - QMI_CTL_SYNC_REQ	54
4.2.8.2.	Response - QMI_CTL_SYNC_RESP	54
4.2.8.3.	Indication - QMI_CTL_SYNC_IND.....	55
4.2.8.4.	Description of QMI_CTL_SYNC_IND	56
4.2.9.	QMI_CTL_REG_PWR_SAVE_MODE	56
4.2.9.1.	Request - QMI_CTL_REG_PWR_SAVE_MODE_REQ	56
4.2.9.2.	Response - QMI_CTL_REG_PWR_SAVE_MODE_RESP	57
4.2.9.3.	Description of QMI_CTL_REG_PWR_SAVE_MODE REQ/RESP	57
4.2.9.4.	Indication - QMI_CTL_PWR_SAVE_MODE_IND.....	57
4.2.9.5.	Description of QMI_CTL_PWR_SAVE_MODE_IND	58
4.2.10.	QMI_CTL_CONFIG_PWR_SAVE_SETTINGS	58
4.2.10.1.	Request - QMI_CTL_CONFIG_PWR_SAVE_SETTINGS_REQ.....	59
4.2.10.2.	Response - QMI_CTL_CONFIG_PWR_SAVE_SETTINGS_RESP.....	59
4.2.10.3.	Description of QMI_CTL_CONFIG_PWR_SAVE_SETTINGS REQ/RESP	60
4.2.11.	QMI_CTL_SET_PWR_SAVE_MODE	61
4.2.11.1.	Request - QMI_CTL_SET_PWR_SAVE_MODE_REQ	61
4.2.11.2.	Response - QMI_CTL_SET_PWR_SAVE_MODE_RESP	62
4.2.11.3.	Description of QMI_CTL_SET_PWR_SAVE_MODE REQ/RESP	62
4.2.12.	QMI_CTL_GET_PWR_SAVE_MODE	63
4.2.12.1.	Request - QMI_CTL_GET_PWR_SAVE_MODE_REQ	63
4.2.12.2.	Response - QMI_CTL_GET_PWR_SAVE_MODE_RESP	63
4.2.12.3.	Description of QMI_CTL_GET_PWR_SAVE_MODE REQ/RESP	64
5	Wireless Data Service (QMI_WDS).....	65
5.1.	Theory of Operation	65
5.1.1.	Generalized QMI Service Compliance.....	65
5.1.2.	WDS Service Type	65
5.1.3.	Message Definition Template	65
5.1.3.1.	Response Message Result TLV.....	66
5.1.4.	QMI_WDS Fundamental Concepts	66
5.1.4.1.	Data Session	66
5.1.4.2.	Data Session Handle	66
5.1.4.3.	Data Connection Status	66
5.1.4.4.	QMI_WDS Profile	67
5.1.5.	Service State Variables	67
5.1.5.1.	Shared State Variables	67
5.1.5.2.	State Variables Per Control Point	68
5.2.	QMI_WDS Messages	69
5.2.1.	QMI_WDS_START_NET (QMI_WDS_START_NETWORK_INTERFACE).....	70
5.2.1.1.	Request – QMI_WDS_START_NETWORK_INTERFACE_REQ	70
5.2.1.2.	Response – QMI_WDS_START_NETWORK_INTERFACE_RESP	76
5.2.1.3.	Description of QMI_WDS_START_NETWORK_INTERFACE REQ/RESP	79
5.2.2.	QMI_WDS_STOP_NETWORK_INTERFACE	81
5.2.2.1.	Request – QMI_WDS_STOP_NETWORK_INTERFACE_REQ	81
5.2.2.2.	Response – QMI_WDS_STOP_NETWORK_INTERFACE_RESP	82

5.2.2.3.	Description of QMI_WDS_STOP_NETWORK_INTERFACE_REQ/RESP	83
5.2.3.	QMI_WDS_GET_PKT_SRVC_STATUS.....	83
5.2.3.1.	Request - QMI_WDS_GET_PKT_SRVC_STATUS_REQ.....	83
5.2.3.2.	Response - QMI_WDS_GET_PKT_SRVC_STATUS_RESP	84
5.2.3.3.	Description of QMI_WDS_GET_PKT_SRVC_STATUS REQ/RESP	85
5.2.4.	QMI_WDS_PKT_STATUS_IND (QMI_WDS_GET_PKT_SRVC_STATUS_IND)	85
5.2.4.1.	Indication – QMI_WDS_PKT_SRVC_STATUS_IND	86
5.2.4.2.	Description of QMI_WDS_GET_PKT_SRVC_STATUS_IND	88
5.2.5.	QMI_WDS_GET_SETTINGS (QMI_WDS_GET_RUNTIME_SETTINGS)	89
5.2.5.1.	Request – QMI_WDS_GET_RUNTIME_SETTINGS_REQ	89
5.2.5.2.	Response – QMI_WDS_GET_RUNTIME_SETTINGS_RESP	91
5.2.5.3.	Description of QMI_WDS_GET_RUNTIME_SETTINGS REQ/RESP	99
6	Device Management Service (QMI_DMS).....	100
6.1.	Theory of Operation	100
6.1.1.	Generalized QMI Service Compliance.....	100
6.1.2.	DMS Service Type	100
6.1.3.	Message Definition Template	100
6.1.3.1.	Response Message Result TLV.....	100
6.1.4.	QMI_DMS Fundamental Concepts.....	101
6.1.5.	Service State Variables	102
6.1.5.1.	Shared State Variables	102
6.1.5.2.	State Variables Per Control Point	102
6.2.	QMI_DMS Messages	102
6.2.1.	QMI_DMS_RESET	104
6.2.1.1.	Request - QMI_DMS_RESET_REQ.....	104
6.2.1.2.	Response - QMI_DMS_RESET_RESP	104
6.2.1.3.	Description of QMI_DMS_RESET REQ/RESP	105
6.2.2.	QMI_DMS_SET_EVENT_REPORT	105
6.2.2.1.	Request - QMI_DMS_SET_EVENT_REPORT_REQ	105
6.2.2.2.	Response - QMI_DMS_SET_EVENT_REPORT_RESP	106
6.2.2.3.	Indication - QMI_DMS_EVENT_REPORT_IND	107
6.2.2.4.	Description of QMI_DMS_SET_EVENT_REPORT	110
6.2.3.	QMI_DMS_GET_DEVICE_CAP	111
6.2.3.1.	Request - QMI_DMS_GET_DEVICE_CAP_REQ	111
6.2.3.2.	Response - QMI_DMS_GET_DEVICE_CAP_RESP	112
6.2.3.3.	Description of QMI_DMS_GET_DEVICE_CAP REQ/RESP	115
6.2.4.	QMI_DMS_GET_DEVICE_MFR.....	116
6.2.4.1.	Request - QMI_DMS_GET_DEVICE_MFR_REQ	116
6.2.4.2.	Response - QMI_DMS_GET_DEVICE_MFR_RESP	116
6.2.4.3.	Description of QMI_DMS_GET_DEVICE_MFR REQ/RESP	117
6.2.5.	QMI_DMS_GET_DEVICE_MODEL_ID	117
6.2.5.1.	Request - QMI_DMS_GET_DEVICE_MODEL_ID_REQ	117
6.2.5.2.	Response - QMI_DMS_GET_DEVICE_MODEL_ID_RESP	118
6.2.5.3.	Description of QMI_DMS_GET_DEVICE_MODEL_ID REQ/RESP	119

6.2.6.	QMI_DMS_GET_DEVICE_REV_ID	119
6.2.6.1.	Request - QMI_DMS_GET_DEVICE_REV_ID_REQ.....	119
6.2.6.2.	Response - QMI_DMS_GET_DEVICE_REV_ID_RESP	119
6.2.6.3.	Description of QMI_DMS_GET_DEVICE_REV_ID REQ/RESP	120
6.2.7.	QMI_DMS_GET_MSISDN.....	121
6.2.7.1.	Request - QMI_DMS_GET_MSISDN_REQ.....	121
6.2.7.2.	Response - QMI_DMS_GET_MSISDN_RESP	121
6.2.7.3.	Description of QMI_DMS_GET_MSISDN REQ/RESP	122
6.2.8.	QMI_DMS_GET_DEVICE_SERIAL_NUMBERS	122
6.2.8.1.	Request - QMI_DMS_GET_DEVICE_SERIAL_NUMBERS_REQ.....	123
6.2.8.2.	Response - QMI_DMS_GET_DEVICE_SERIAL_NUMBERS_RESP	123
6.2.8.3.	Description of QMI_DMS_GET_DEVICE_SERIAL_NUMBERS REQ/RESP .	124
6.2.9.	QMI_DMS_GET_POWER_STATE	124
6.2.9.1.	Request - QMI_DMS_GET_POWER_STATE_REQ	125
6.2.9.2.	Response - QMI_DMS_GET_POWER_STATE_RESP	125
6.2.9.3.	Description of QMI_DMS_GET_POWER_STATE REQ/RESP	126
6.2.10.	QMI_DMS_UIM_SET_PIN_PROTECTION.....	127
6.2.10.1.	Request - QMI_DMS_UIM_SET_PIN_PROTECTION_REQ	127
6.2.10.2.	Response - QMI_DMS_UIM_SET_PIN_PROTECTION_RESP	128
6.2.10.3.	Description of QMI_DMS_UIM_SET_PIN_PROTECTION REQ/RESP	129
6.2.11.	QMI_DMS_UIM_VERIFY_PIN.....	129
6.2.11.1.	Request - QMI_DMS_UIM_VERIFY_PIN_REQ	129
6.2.11.2.	Response - QMI_DMS_UIM_VERIFY_PIN_RESP	130
6.2.11.3.	Description of QMI_DMS_UIM_VERIFY_PIN REQ/RESP	131
6.2.12.	QMI_DMS_UIM_UNBLOCK_PIN	131
6.2.12.1.	Request - QMI_DMS_UIM_UNBLOCK_PIN_REQ	132
6.2.12.2.	Response - QMI_DMS_UIM_UNBLOCK_PIN_RESP.....	132
6.2.12.3.	Description of QMI_DMS_UIM_UNBLOCK_PIN REQ/RESP	134
6.2.13.	QMI_DMS_UIM_CHANGE_PIN	134
6.2.13.1.	Request - QMI_DMS_UIM_CHANGE_PIN_REQ	134
6.2.13.2.	Response - QMI_DMS_UIM_CHANGE_PIN_RESP.....	135
6.2.13.3.	Description of QMI_DMS_UIM_CHANGE_PIN REQ/RESP	136
6.2.14.	QMI_DMS_UIM_GET_PIN_STATUS	136
6.2.14.1.	Request - QMI_DMS_UIM_GET_PIN_STATUS_REQ.....	137
6.2.14.2.	Response - QMI_DMS_UIM_GET_PIN_STATUS_RESP	137
6.2.14.3.	Description of QMI_DMS_UIM_GET_PIN_STATUS REQ/RESP	139
6.2.15.	QMI_DMS_GET_DEVICE_HARDWARE_REV	139
6.2.15.1.	Request - QMI_DMS_GET_DEVICE_HARDWARE_REV_REQ	139
6.2.15.2.	Response - QMI_DMS_GET_DEVICE_HARDWARE_REV_RESP	139
6.2.15.3.	Description of QMI_DMS_GET_DEVICE_HARDWARE_REVREQ/RESP....	140
6.2.16.	QMI_DMS_GET_OPERATING_MODE	140
6.2.16.1.	Request - QMI_DMS_GET_OPERATING_MODE_REQ	141
6.2.16.2.	Response - QMI_DMS_GET_OPERATING_MODE_RESP	141
6.2.16.3.	Description of QMI_DMS_GET_OPERATING_MODE REQ/RESP	142

6.2.17.	QMI_DMS_SET_OPERATING_MODE	143
6.2.17.1.	Request - QMI_DMS_SET_OPERATING_MODE_REQ.....	143
6.2.17.2.	Response - QMI_DMS_SET_OPERATING_MODE_RESP.....	144
6.2.17.3.	Description of QMI_DMS_SET_OPERATING_MODE REQ/RESP	145
6.2.18.	QMI_DMS_GET_TIME	146
6.2.18.1.	Request - QMI_DMS_GET_TIME_REQ.....	146
6.2.18.2.	Response - QMI_DMS_GET_TIME_RESP.....	146
6.2.18.3.	Description of QMI_DMS_GET_TIME REQ/RESP	148
6.2.19.	QMI_DMS_GET_PRL_VER	148
6.2.19.1.	Request - QMI_DMS_GET_PRL_VER_REQ.....	148
6.2.19.2.	Response - QMI_DMS_GET_PRL_VER_RESP.....	149
6.2.19.3.	Description of QMI_DMS_GET_PRL_VER REQ/RESP	150
6.2.20.	QMI_DMS_UIM_GET_ICCID	150
6.2.20.1.	Request - QMI_DMS_UIM_GET_ICCID_REQ.....	151
6.2.20.2.	Response - QMI_DMS_UIM_GET_ICCID_RESP.....	151
6.2.20.3.	Description of QMI_DMS_UIM_GET_ICCID REQ/RESP	152
6.2.21.	QMI_DMS_UIM_GET_IMSI.....	153
6.2.21.1.	Request - QMI_DMS_UIM_GET_IMSI_REQ	153
6.2.21.2.	Response - QMI_DMS_UIM_GET_IMSI_RESP	153
6.2.21.3.	Description of QMI_DMS_UIM_GET_IMSI REQ/RESP	154
6.2.22.	QMI_DMS_UIM_GET_STATE	154
6.2.22.1.	Request - QMI_DMS_UIM_GET_STATE_REQ	155
6.2.22.2.	Response - QMI_DMS_UIM_GET_STATE_RESP	155
6.2.22.3.	Description of QMI_DMS_UIM_GET_STATE REQ/RESP	156
6.2.23.	QMI_DMS_GET_BAND_CAPABILITY	156
6.2.23.1.	Request - QMI_DMS_GET_BAND_CAPABILITY_REQ	157
6.2.23.2.	Response - QMI_DMS_GET_BAND_CAPABILITY_RESP.....	157
6.2.23.3.	Description of QMI_DMS_GET_BAND_CAPABILITY REQ/RESP	158
6.2.24.	QMI_DMS_GET_FACTORY_SKU	159
6.2.24.1.	Request - QMI_DMS_GET_FACTORY_SKU_REQ.....	159
6.2.24.2.	Response - QMI_DMS_GET_FACTORY_SKU_RESP	159
6.2.24.3.	Description of QMI_DMS_GET_FACTORY_SKU REQ/RESP	160
6.2.25.	QMI_DMS_GET_SW_VERSION.....	160
6.2.25.1.	Request - QMI_DMS_GET_SW_VERSION_REQ	161
6.2.25.2.	Response - QMI_DMS_GET_SW_VERSION_RESP	161
6.2.25.3.	Description of QMI_DMS_GET_SW_VERSION REQ/RESP	162
7	Network Access Service (QMI_NAS)	163
7.1.	Theory of Operation	163
7.1.1.	Generalized QMI Service Compliance.....	163
7.1.2.	NAS Service Type	163
7.1.3.	Message Definition Template	163
7.1.3.1.	Response Message Result TLV.....	163
7.1.4.	QMI_NAS Fundamental Concepts.....	164
7.1.5.	Service State Variables	165

7.1.5.1.	Shared State Variables	165
7.1.5.2.	State Variables Per Control Point	166
7.2.	QMI_NAS Messages	167
7.2.1.1.	QMI_NAS_RESET	168
7.2.1.2.	Request - QMI_NAS_RESET_REQ_MSG	168
7.2.1.3.	Response - QMI_NAS_RESET_RESP_MSG	168
7.2.1.4.	Description of QMI_NAS_RESET REQ/RESP	169
7.2.2.	QMI_NAS_ABORT	169
7.2.2.1.	Request - QMI_NAS_ABORT_REQ_MSG	169
7.2.2.2.	Response - QMI_NAS_ABORT_RESP_MSG	170
7.2.2.3.	Description of QMI_NAS_ABORT REQ/RESP	171
7.2.3.	QMI_NAS_SET_EVENT_REPORT	171
7.2.3.1.	Request - QMI_NAS_SET_EVENT_REPORT_REQ_MSG	171
7.2.3.2.	Response - QMI_NAS_SET_EVENT_REPORT_RESP_MSG	175
7.2.3.3.	Description of QMI_NAS_SET_EVENT_REPORT REQ/RESP	176
7.2.4.	QMI_NAS_EVENT_REPORT_IND	176
7.2.4.1.	Indication - QMI_NAS_EVENT_REPORT_IND_MSG	177
7.2.4.2.	Description of QMI_NAS_EVENT_REPORT_IND	181
7.2.5.	QMI_NAS_INDICATION_REGISTER	182
7.2.5.1.	Request - QMI_NAS_INDICATION_REGISTER_REQ_MSG	183
7.2.5.2.	Response - QMI_NAS_INDICATION_REGISTER_RESP_MSG	188
7.2.5.3.	Description of QMI_NAS_INDICATION_REGISTER REQ/RESP	189
7.2.6.	QMI_NAS_GET_SIGNAL_STRENGTH	190
7.2.6.1.	Request - QMI_NAS_GET_SIGNAL_STRENGTH_REQ_MSG	190
7.2.6.2.	Response - QMI_NAS_GET_SIGNAL_STRENGTH_RESP_MSG	191
7.2.6.3.	Description of QMI_NAS_GET_SIGNAL_STRENGTH REQ/RESP	196
7.2.7.	QMI_NAS_SCAN_NETS (QMI_NAS_PERFORM_NETWORK_SCAN)	197
7.2.7.1.	Request - QMI_NAS_PERFORM_NETWORK_SCAN_REQ_MSG	197
7.2.7.2.	Response - QMI_NAS_PERFORM_NETWORK_SCAN_RESP_MSG	198
7.2.7.3.	Description of QMI_NAS_PERFORM_NETWORK_SCAN REQ/RESP	204
7.2.8.	QMI_NAS_REGISTER_NET (QMI_NAS_INITIATE_NETWORK_REGISTER) ...	204
7.2.8.1.	Request – QMI_NAS_INITIATE_NETWORK_REGISTER_REQ_MSG	204
7.2.8.2.	Response – QMI_NAS_INITIATE_NETWORK_REGISTER_RESP_MSG ...	206
7.2.8.3.	Description of QMI_NAS_INITIATE_NETWORK_REGISTER REQ/RESP ...	207
7.2.9.	QMI_NAS_INITIATE_ATTACH	207
7.2.9.1.	Request - QMI_NAS_INITIATE_ATTACH_REQ_MSG	207
7.2.9.2.	Response - QMI_NAS_INITIATE_ATTACH_RESP_MSG	208
7.2.9.3.	Description of QMI_NAS_INITIATE_ATTACH REQ/RESP	209
7.2.10.	QMI_NAS_GET_SS_INFO (QMI_NAS_GET_SERVING_SYSTEM)	209
7.2.10.1.	Request – QMI_NAS_GET_SERVING_SYSTEM_REQ_MSG	210
7.2.10.2.	Response – QMI_NAS_GET_SERVING_SYSTEM_RESP_MSG	210
7.2.10.3.	Description of QMI_NAS_GET_SERVING_SYSTEM REQ/RESP	218
7.2.11.	QMI_NAS_SS_INFO_IND (QMI_NAS_SERVING_SYSTEM_IND)	218
7.2.11.1.	7.2.5.1. Indication – QMI_NAS_SERVING_SYSTEM_IND_MSG	219

7.2.11.2. Description of QMI_NAS_SERVING_SYSTEM_IND	227
7.2.12. QMI_NAS_SET_SYSTEM_SELECTION_PREF (QMI_NAS_SET_SYSTEM_SELECTION)	228
7.2.12.1. Request	-
QMI_NAS_SET_SYSTEM_SELECTION_PREFERENCE_REQ_MSG	228
7.2.12.2. Response	-
QMI_NAS_SET_SYSTEM_SELECTION_PREFERENCE_RESP_MSG	235
7.2.12.3. Description of QMI_NAS_SET_SYSTEM_SELECTION_PREFERENCE REQ/RESP	236
7.2.13. QMI_NAS_GET_SYSTEM_SELECTION_PREFERENCE	237
7.2.13.1. Request-QMI_NAS_GET_SYSTEM_SELECTION_PREFERENCE_REQ_MSG	237
7.2.13.2. Response	-
QMI_NAS_GET_SYSTEM_SELECTION_PREFERENCE_RESP_MSG	237
7.2.13.3. Indication - QMI_NAS_SYSTEM_SELECTION_PREFERENCE_IND_MSG .	245
7.2.13.4. Description of QMI_NAS_GET_SYSTEM_SELECTION_PREFERENCE	252
8 Wireless Messaging Service (QMI_WMS)	254
8.1. Theory of Operation	254
8.1.1. Generalized QMI Service Compliance.....	254
8.1.2. WMS Service Type.....	254
8.1.3. Message Definition Template.....	254
8.1.3.1. Response Message Result TLV.....	254
8.1.4. QMI_WMS Fundamental Concepts	255
8.1.4.1. Wireless Message Network Architecture	255
8.1.4.2. Wireless Message Types	256
8.1.4.3. WMS Client/Service Architecture	256
8.1.4.4. Incoming Message Indication	256
8.1.4.5. WMS Message Layers	257
8.1.4.6. Raw Message Parameters.....	257
8.1.4.7. Routes	257
8.1.4.8. Device Memory Storage.....	258
8.1.5. Service State Variables	258
8.1.5.1. Shared State Variables	258
8.1.5.2. State Variables Per Control Point	258
8.2. QMI_WMS Messages	259
8.2.1. QMI_WMS_RESET.....	260
8.2.1.1. Request - QMI_WMS_RESET_REQ	260
8.2.1.2. Response - QMI_WMS_RESET_RESP	261
8.2.1.3. Description of QMI_WMS_RESET REQ/RESP	261
8.2.2. QMI_WMS_SET_EVENT (QMI_WMS_SET_EVENT_REPORT)	261
8.2.2.1. Request - QMI_WMS_SET_EVENT_REPORT_REQ.....	262
8.2.2.2. Response - QMI_WMS_SET_EVENT_REPORT_RESP	263
8.2.2.3. Description of QMI_WMS_SET_EVENT_REPORT REQ/RESP	263
8.2.3. Indication - QMI_WMS_SET_EVENT (QMI_WMS_EVENT_REPORT_IND).....	264
8.2.3.1. Description of QMI_WMS_EVENT_REPORT_IND	264

8.2.4.	QMI_WMS_RAW_SEND	265
8.2.4.1.	Request - QMI_WMS_RAW_SEND_REQ.....	265
8.2.4.2.	Response - QMI_WMS_RAW_SEND_RESP	267
8.2.4.3.	Description of QMI_WMS_RAW_SEND REQ/RESP	270
8.2.5.	QMI_WMS_RAW_READ	272
8.2.5.1.	Request - QMI_WMS_RAW_READ_REQ.....	272
8.2.5.2.	Response - QMI_WMS_RAW_SEND_RESP	275
8.2.5.3.	Description of QMI_WMS_RAW_READ REQ/RESP	278
8.2.6.	QMI_WMS MODIFY_TAG	278
8.2.6.1.	Request - QMI_WMS MODIFY_TAG_REQ.....	279
8.2.6.2.	Response - QMI_WMS MODIFY_TAG_RESP	280
8.2.6.3.	Description of QMI_WMS MODIFY_TAG REQ/RESP	280
8.2.7.	QMI_WMS_DELETE.....	281
8.2.7.1.	Request - QMI_WMS_DELETE_REQ.....	281
8.2.7.2.	Response - QMI_WMS_DELETE_RESP	282
8.2.7.3.	Description of QMI_WMS_DELETE REQ/RESP	283
8.2.8.	QMI_WMS_GET_MESSAGE_PROTOCOL.....	284
8.2.8.1.	Request - QMI_WMS_GET_MESSAGE_PROTOCOL_REQ	284
8.2.8.2.	Response - QMI_WMS_GET_MESSAGE_PROTOCOL_RESP	285
8.2.8.3.	Description of QMI_WMS_GET_MESSAGE_PROTOCOL REQ/RESP	285
8.2.9.	QMI_WMS_LIST_MESSAGES.....	286
8.2.9.1.	3.11.1 Request - QMI_WMS_LIST_MESSAGES_REQ	286
8.2.9.2.	Response - QMI_WMS_LIST_MESSAGES_RESP	287
8.2.9.3.	Description of QMI_WMS_LIST_MESSAGES REQ/RESP	288
8.2.10.	QMI_WMS_SET_ROUTES	289
8.2.11.	Request - QMI_WMS_SET_ROUTES_REQ.....	289
8.2.11.1.	Response - QMI_WMS_SET_ROUTES_RESP	292
8.2.11.2.	Description of QMI_WMS_SET_ROUTES REQ/RESP	293
8.2.12.	QMI_WMS_GET_ROUTES	294
8.2.13.	Request - QMI_WMS_GET_ROUTES_REQ	294
8.2.13.1.	Response - QMI_WMS_GET_ROUTES_RESP	294
8.2.13.2.	Description of QMI_WMS_GET_ROUTES REQ/RESP	297
8.2.14.	QMI_WMS_GET_SMSC_ADDRESS	297
8.2.14.1.	Request - QMI_WMS_GET_SMSC_ADDRESS_REQ	297
8.2.14.2.	Response - QMI_WMS_GET_SMSC_ADDRESS_RESP	298
8.2.14.3.	Description of QMI_WMS_GET_SMSC_ADDRESS REQ/RESP	299
8.2.15.	QMI_WMS_SET_SMSC_ADDRESS	300
8.2.15.1.	Request - QMI_WMS_SET_SMSC_ADDRESS_REQ	300
8.2.15.2.	Response - QMI_WMS_SET_SMSC_ADDRESS_RESP	301
8.2.15.3.	Description of QMI_WMS_SET_SMSC_ADDRESS REQ/RESP	302
8.2.16.	QMI_WMS_GET_STORE_MAX_SIZE.....	302
8.2.16.1.	Request - QMI_WMS_GET_STORE_MAX_SIZE_REQ	303
8.2.16.2.	Response - QMI_WMS_GET_STORE_MAX_SIZE_RESP	304
8.2.16.3.	Description of QMI_WMS_GET_STORE_MAX_SIZE REQ/RESP	305

8.2.17.	QMI_WMS_GET_DOMAIN_PREF	305
8.2.17.1.	Request - QMI_WMS_GET_DOMAIN_PREF_REQ	305
8.2.17.2.	Response - QMI_WMS_GET_DOMAIN_PREF_RESP	305
8.2.17.3.	Description of QMI_WMS_GET_DOMAIN_PREF REQ/RESP	306
8.2.18.	QMI_WMS_SET_DOMAIN_PREF	307
8.2.18.1.	Request - QMI_WMS_SET_DOMAIN_PREF_REQ	307
8.2.18.2.	Response - QMI_WMS_SET_DOMAIN_PREF_RESP	308
8.2.18.3.	Description of QMI_WMS_SET_DOMAIN_PREF REQ/RESP	308
9	Position Determination Service (QMI_PDS).....	309
9.1.	Theory of Operation	309
9.1.1.	Generalized QMI Service Compliance	309
9.1.2.	PDS Service Type	309
9.1.3.	Message Definition Template	310
9.1.3.1.	Response Message Result TLV	310
9.1.4.	QMI_PDS Fundamental Concepts	310
9.1.4.1.	GPS	310
9.1.4.2.	Position Determination Methods	310
9.1.4.3.	MS-assisted PD	311
9.1.4.4.	Client Request	312
9.1.4.5.	MS-based PD	312
9.1.4.6.	MS-based Call Flow	312
9.1.4.7.	Tracking Sessions	314
9.1.4.8.	Manual Tracking Sessions	315
9.1.4.9.	Single-Shot PD	315
9.1.4.10.	NMEA Sentence Data	316
9.1.4.11.	External Information Injection	316
9.1.4.11.1.	External Time Injection	316
9.1.4.11.2.	Coarse Position Injection	316
9.1.4.11.3.	Wi-Fi Position Injection	316
9.1.4.12.	gpsOneXTRA Satellite Database Information	317
9.1.4.13.	Satellite-Based Augmentation System Configuration	317
9.1.4.14.	External Sensor Data Input	318
Device Accelerometer Measurements	320	
9.1.4.15.	320	320
9.1.4.16.	Device Gyroscope Measurements	321
9.1.4.17.	Vehicle Sensor Measurements	321
9.1.4.18.	Accelerometer and Gyroscope Measurements	321
9.1.4.19.	Odometry Measurement	322
9.1.4.20.	External Time Synchronization	323
9.1.4.21.	Emergency Positioning Session	323
9.1.5.	Service State Variables	327
9.1.5.1.	Shared State Variables	327
9.1.5.2.	State Variables Per Control Point	328
9.2.	QMI_PDS Messages	329

9.2.1.	QMI_PDS_RESET	330
9.2.1.1.	Request - QMI_PDS_RESET_REQ	330
9.2.1.2.	Response - QMI_PDS_RESET_RESP	330
9.2.1.3.	Description of QMI_PDS_RESET REQ/RESP	331
9.2.2.	QMI_PDS_SET_EVENT_REPORT.....	331
9.2.3.	Request - QMI_PDS_SET_EVENT_REPORT_REQ	331
9.2.3.1.	Response - QMI_PDS_SET_EVENT_REPORT_RESP	339
9.2.3.2.	Indication - QMI_PDS_EVENT_REPORT_IND.....	339
9.2.3.3.	Description of QMI_PDS_SET_EVENT_REPORT	367
9.2.4.	QMI_PDS_GET_AUTO_TRACKING_STATE	370
9.2.4.1.	Request - QMI_PDS_GET_AUTO_TRACKING_STATE_REQ.....	370
9.2.4.2.	Response - QMI_PDS_GET_AUTO_TRACKING_STATE_RESP	371
9.2.4.3.	Description of QMI_PDS_GET_AUTO_TRACKING_STATE REQ/RESP	372
9.2.5.	QMI_PDS_SET_AUTO_TRACKING_STATE	372
9.2.5.1.	Request - QMI_PDS_SET_AUTO_TRACKING_STATE_REQ	372
9.2.5.2.	Response - QMI_PDS_SET_AUTO_TRACKING_STATE_RESP	373
9.2.5.3.	Description of QMI_PDS_SET_AUTO_TRACKING_STATE REQ/RESP	374
10	Voice Service (QMI_VOICE)	375
10.1.	Theory of Operation	375
10.1.1.	Generalized QMI Service Compliance.....	375
10.1.2.	VOICE Service Type	375
10.1.3.	Message Definition Template.....	375
10.1.3.1.	Response Message Result TLV	375
10.1.4.	QMI_VOICE Fundamental Concepts.....	376
10.1.4.1.	Dual SIM.....	376
10.1.5.	Service State Variables	377
10.1.5.1.	Shared State Variables	377
10.1.5.2.	State Variables Per Control Point	377
10.2.	3 QMI_VOICE Messages.....	378
10.2.1.	QMI_VOICE_CANCEL_USSD	379
10.2.1.1.	Request - QMI_VOICE_CANCEL_USSD_REQ.....	379
10.2.1.2.	Response - QMI_VOICE_CANCEL_USSD_RESP	379
10.2.1.3.	Description of QMI_VOICE_CANCEL_USSD REQ/RESP	380
10.3.	QMI_VOICE_USSD_RELEASE_IND	380
10.3.1.	Indication - QMI_VOICE_USSD_RELEASE_IND	380
10.3.1.1.	Description of QMI_VOICE_USSD_RELEASE_IND	381
10.4.	QMI_VOICE_USSD_IND	381
10.4.1.1.	Description of QMI_VOICE_USSD_IND	381
10.4.1.2.	Indication - QMI_VOICE_USSD_IND	382
10.4.2.	QMI_VOICE_ORIG_USSD_NO_WAIT_IND	383
10.4.2.1.	Indication - QMI_VOICE_ORIG_USSD_NO_WAIT_IND	383
10.4.2.2.	Description of QMI_VOICE_ORIG_USSD_NO_WAIT_IND	384
11	User Identity Module (QMI_UIM).....	386

11.1. Theory of Operation	386
11.1.1. Generalized QMI Service Compliance.....	386
11.1.2. UIM Service Type.....	386
11.1.3. Message Definition Template.....	386
11.1.3.1. Response Message Result TLV.....	386
11.1.4. QMI_UIM Fundamental Concepts	387
11.1.4.1. UIM Overview.....	387
11.1.4.2. Commands Supported via QMI_UIM	387
11.1.4.3. Multiple Applications on the Same Card	387
11.1.4.4. Sessions.....	387
11.1.4.5. Commands Supported via QMI_UIM	388
11.1.4.6. Refresh Procedure via QMI_UIM.....	388
11.1.5. Service State Variables	389
11.1.5.1. Shared State Variables	389
11.2. QMI_UIM Messages	389
11.2.1. QMI_UIM_READ_TRANSPARENT	389
11.2.1.1. Request - QMI_UIM_READ_TRANSPARENT_REQ	390
11.2.1.2. Response - QMI_UIM_READ_TRANSPARENT_RESP	391
11.2.1.3. Description of QMI_UIM_READ_TRANSPARENT REQ/RESP	394
11.2.2. QMI_UIM_READ_RECORD	395
11.2.2.1. Request - QMI_UIM_READ_RECORD_REQ	395
11.2.2.2. Response - QMI_UIM_READ_RECORD_RESP	397
11.2.2.3. Description of QMI_UIM_READ_RECORD REQ/RESP	399
11.2.3. QMI_UIM_WRITE_TRANSPARENT	399
11.2.3.1. Request - QMI_UIM_WRITE_TRANSPARENT_REQ	400
11.2.3.2. Response - QMI_UIM_WRITE_TRANSPARENT_RESP	401
11.2.3.3. Description of QMI_UIM_WRITE_TRANSPARENT REQ/RESP	402
11.2.4. QMI_UIM_WRITE_RECORD	403
11.2.4.1. Request - QMI_UIM_WRITE_RECORD_REQ	403
11.2.4.2. Response - QMI_UIM_WRITE_RECORD_RESP	405
11.2.4.3. Description of QMI_UIM_WRITE_RECORD REQ/RESP	406
11.2.5. QMI_UIM_GET_FILE_ATTRIBUTES	407
11.2.5.1. Request - QMI_UIM_GET_FILE_ATTRIBUTES_REQ	407
11.2.5.2. Response - QMI_UIM_GET_FILE_ATTRIBUTES_RESP	409
11.2.5.3. Description of QMI_UIM_GET_FILE_ATTRIBUTES REQ/RESP	413
11.2.6. QMI_UIM_VERIFY_PIN.....	414
11.2.6.1. Request - QMI_UIM_VERIFY_PIN_REQ	414
11.2.6.2. Response - QMI_UIM_VERIFY_PIN_RESP	417
11.2.6.3. Description of QMI_UIM_VERIFY_PIN REQ/RESP	419
11.2.7. QMI_UIM_EVENT_REG	420
11.2.7.1. Request - QMI_UIM_EVENT_REG_REQ	420
11.2.7.2. Response - QMI_UIM_EVENT_REG_RESP	421
11.2.7.3. Description of QMI_UIM_EVENT_REG REQ/RESP	423
11.2.8. QMI_UIM_GET_CARD_STATUS	423

11.2.8.1. Request - QMI_UIM_GET_CARD_STATUS_REQ.....	423
11.2.8.2. Response - QMI_UIM_GET_CARD_STATUS_RESP.....	424
11.2.8.3. Description of QMI_UIM_GET_CARD_STATUS REQ/RESP	440
11.2.9. QMI_UIM_STATUS_CHANGE_IND	441
11.2.9.1. Indication - QMI_UIM_STATUS_CHANGE_IND	441
11.2.9.2. Description of QMI_UIM_STATUS_CHANGE_IND	456
12 Wireless Data Administrative Service (QMI_WDA).....	457
12.1. Theory of Operation	457
12.1.1. Generalized QMI Service Compliance.....	457
12.1.2. WDA Service Type	457
12.1.3. Message Definition Template.....	457
12.1.3.1. Response Message Result TLV.....	457
12.1.4. QMI_WDA Fundamental Concepts.....	458
12.1.5. Service State Variables	458
12.1.5.1. Shared State Variables	458
12.2. QMI_WDA Messages.....	458
12.2.1. QMI_WDA_SET_DATA_FORMAT	459
12.2.1.1. Request - QMI_WDA_SET_DATA_FORMAT_REQ	459
12.2.1.2. Response - QMI_WDA_SET_DATA_FORMAT_RESP	463
12.2.1.3. Description of QMI_WDA_SET_DATA_FORMAT REQ/RESP	466
12.2.2. QMI_WDA_GET_DATA_FORMAT	467
12.2.2.1. Request - QMI_WDA_GET_DATA_FORMAT_REQ.....	467
12.2.2.2. Response - QMI_WDA_GET_DATA_FORMAT_RESP	468
12.2.2.3. Description of QMI_WDA_GET_DATA_FORMAT REQ/RESP	471
13 Appendix A References.....	472

1 Introduction

The scope of this document is to provide the Qualcomm Messaging Interface (QMI)service messages of Quectel EC2x module.

2 QMI

It is expected that applications on a tethered computing device, e.g., connection manager applications and device drivers, will interface to the QMI-enabled MSM device using the QMI Multiplexing Protocol (QMUX) and QMI Service protocols as described herein.

2.1. QMI Framework

The QMI framework defines an interface between the TE and a processor running AMSS, enabling applications on the tethered processor to make use of functionality on the AMSS processor.

The QMI framework is composed of:

Properties of the interconnection between an MSM chipset and the TE, including orthogonal control and data channels
An enumeration of logical devices emulated by the MSM device over the interconnection
A messaging protocol for messaging on the control channels of each logic device that allows applications running on the TE to access MSM-based Services

The following figure illustrates the layering of the QMI between the applications executing on a TE device and the MSM device.

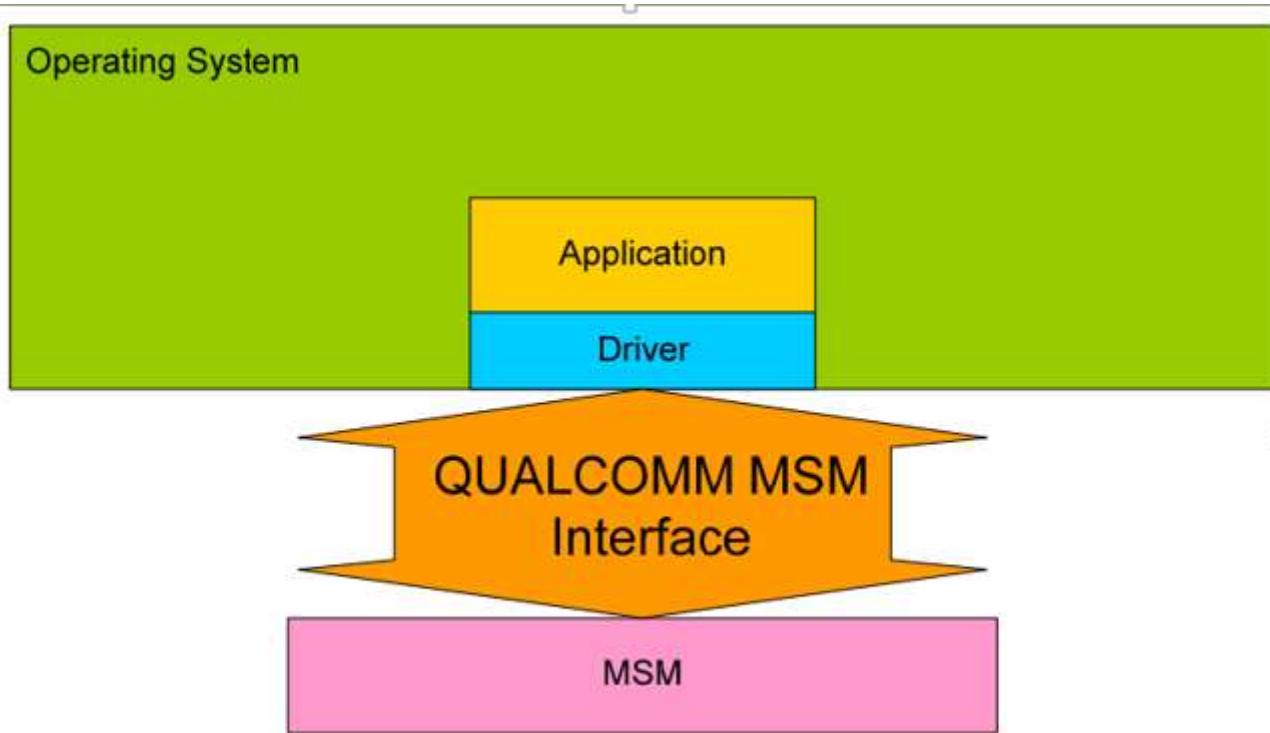


Figure 1: QUALCOMM MSM Interface

2.1.1. MSM-TE Interconnection

QMI connects an MSM device to the TE (see **Figure 2**). The term TE is inclusive of all form factors, including devices such as PCs, notebooks, PDAs, and smartphones. The TE consists of an application environment (and possibly an operating system) executing on a separate processor, which is connected to the MSM processor via some form of interconnect.

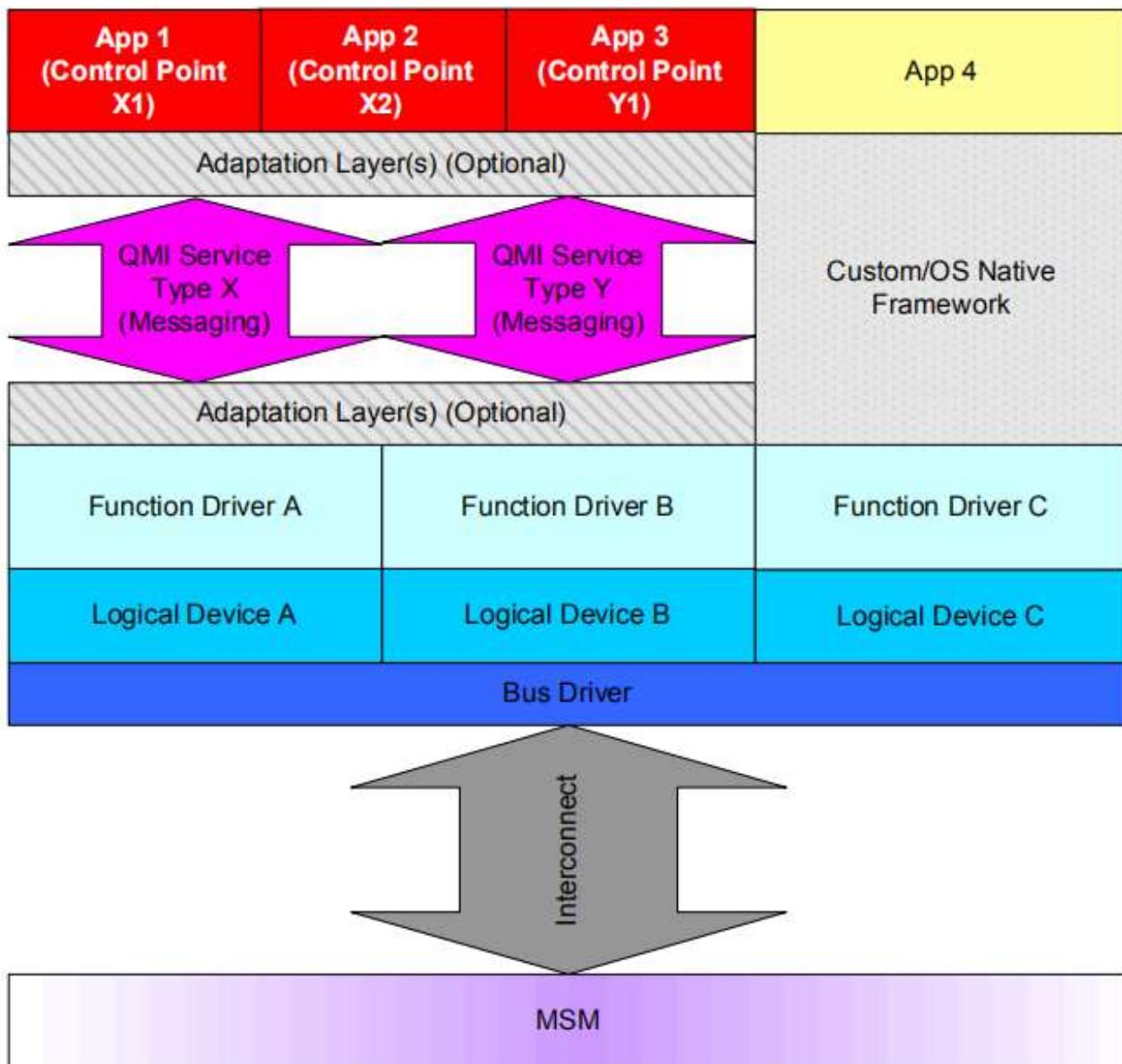


Figure 2: QMI Architecture

The TE can be attached to the MSM over various bus interconnects, e.g., serial buses like USB, RS-232, PCI, or PCMCIA; wireless links like Bluetooth® or 802.11; shared memory interfaces, etc. Regardless of which interconnect is used, QMI enumerates a number of logical devices. The interconnection must provide a mechanism for multiplexing multiple logical devices over a single physical connection. Each logical device consists of at least one communication channel, and the underlying interconnect must provide for independent data and control communication channels for each logical device. Channel independence implies that each channel must act as if there were no physical coupling between the communication channels, including (but not limited to) separate Tx and Rx path queuing, independent flow control mechanisms, and independent data transmission scheduling. A logical device uses at least one communication channel but need not have both (see Figure 3). For example, the existing MSM diagnostic interface consists of a data channel only. For both QMI control and data channels, the interconnection must provide for framing of messages exchanged, i.e., delineating packet boundaries to

the transport protocol (e.g., 802.3).

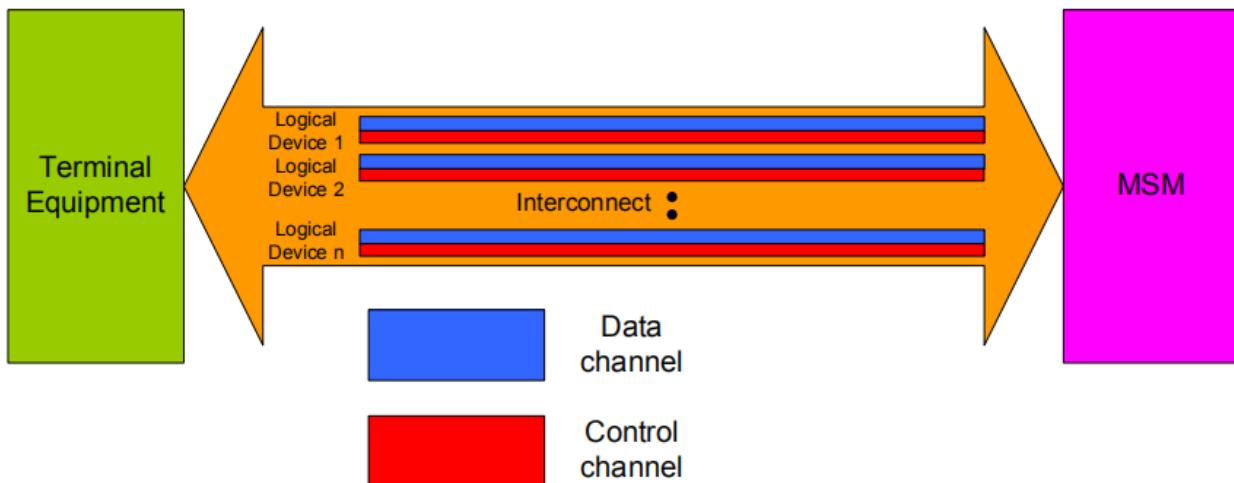


Figure 3: QMI MSM-TE Interconnection

2.1.2. Logical device enumeration

Logical devices include both those that leverage QMI messaging protocols, such as an Rm network (RmNet) device.

Existing non-QMI devices are enumerated as well, such as:

- Legacy modem device
- Diagnostic interface
- NMEA device

Each logical device that is capable of exchanging QMI messages must provide orthogonal data and control channels. QMI messages are exchanged on the control channel. The RmNet device presents an IP network interface to the TE provided by the wireless data-enabled QMI device.

2.1.3. Control channel messaging protocol

The QMI defines the protocol for communication over the control channel of a QMI logical device, consisting of: The QMUX transport protocol, which carries all control channel messages, described in **Chapter 3** of this document A communication reference model defining communication endpoints known as Control Points and Services, described below.

A generalized QMI Service protocol, including protocols to be observed and rules for message definition, as described in Chapter 4; all QMI Service interfaces, including Services that conform to this generalized Service protocol and also custom QMI Services, are outside the scope of this document and are

described in detail in their own specification document

A special QMI_CTL Service that is used by the QMI drivers on both the TE and MSM devices to negotiate client IDs and special control Services.

2.1.3.1. Endpoint model

Applications and device drivers on the TE communicate with a QMI-enabled MSM device by exchanging QMI Service messages over the QMUX transport protocol (see Chapter 3). These control messages are sent on the control channel of a QMI logical device (see **Figure 4**).

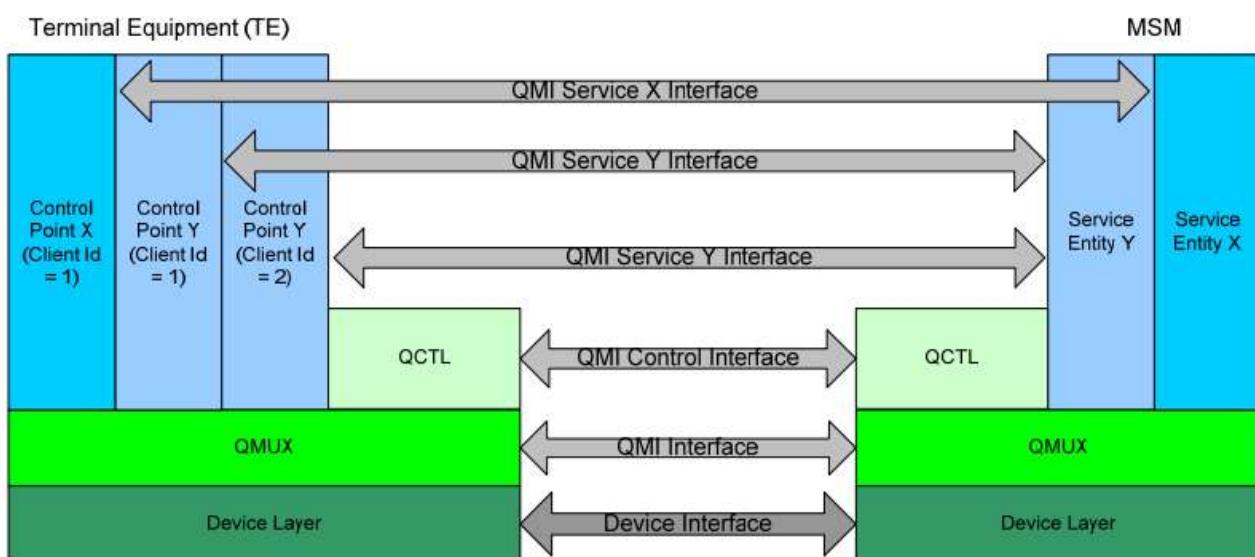


Figure 4: QMI Control Channel Messaging Endpoint Model

All controlling applications are referred to as Control Points. A Control Point is a client of a particular QMI Service. The software module that receives the QMI Service message and performs the function is referred to as a QMI Service. A Control Point is to the Service as a client is to a server in the standard software engineering client/server model. If an application makes use of several QMI Services, it will comprise a Control Point for each of the utilized Services.

2.1.3.2. Usage

Connection manager applications and device drivers on the TE are expected to interface to the QMI-enabled MSM device using QMI Service protocols. Other applications on the TE may also be capable of using QMI Services.

2.2. QMI Multiplexing Protocol (QMUX)

A QMUX implementation provides transport for QMI Service messages between QMI Control Point(s) and QMI Service(s) over the control channel of a QMI logical device.

2.2.1. Byte ordering

QMUX messages shall be transmitted in little-endian format. The convention for bit numbering is that bit 0 is the Least Significant Bit (LSB). All QMUX messages employ a common header format, regardless of the direction in which the message is sent (Control Point to Service, or vice versa). The QMUX message format is shown in **Figure 5**. Note that all messages sent on the control channel have a preamble byte (I/F type), which identifies the transport format (i.e., QMUX) (see Table 1)

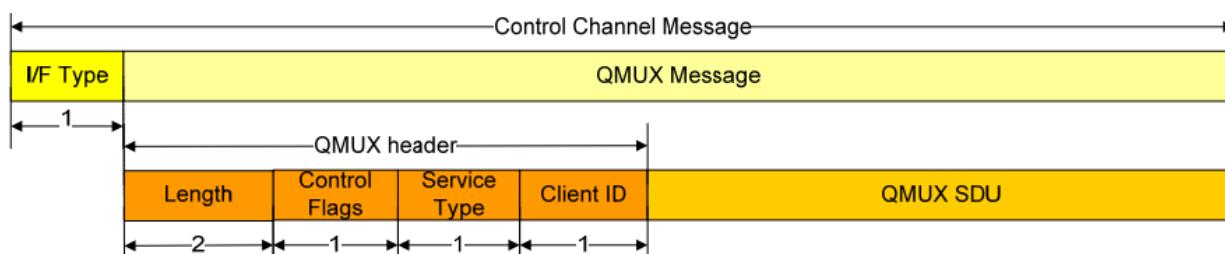


Figure 5: Control Channel Message Format QMI

Table 1: Control Channel Message Preamble

Control channel message preamble	Description
I/F type	<ul style="list-style-type: none"> ■ Interface type ■ 0x01 indicates that a QMUX message follows. ■ Other I/F type values are reserved for future transport protocol specifications.

The QMUX message consists of a header followed by an SDU containing the transported Service layer message(s). The fields of the QMUX header are described in **Table 2**.

Table 2: QMI Message Fields Control Channel Message Preamble

QMUX header field	Description
Length	Length of the QMUX message; includes the QMUX header itself, but not the preamble I/F type
Control flags	<p>bit 7 – Sender type</p> <ul style="list-style-type: none"> ■ 1 = Service ■ 0 = Control Point <p>The remaining bits are reserved for future use and must be set to 0.</p>
Service type	<p>The QMI Service type of the message(s) present in the SDU. QMI Service types are defined in [Q1].</p> <p>Individual licensees may define vendor-specific Services particular to their device(s) and assign them a Service type within the vendor-specific range [Q1].</p>
Client ID	<p>Identifies the Control Point to which the message pertains. Both the Service and Control Point should set this to the relevant Control Point client ID.</p> <p>A client ID value of 0xFF indicates a broadcast message that pertains to all Control Points of the preceding QMI Service type.¹</p>
QMUX SDU	Payload of the QMUX message; contains the QMI Service message(s)

A Control Point shall not send a message with a client ID value of 0xff.

2.2.1.1. QMUX version

The QMUX protocol version will be distinguished by I/F type preamble field value.

2.2.1.2. Multiplexing of QMI Services

The QMUX multiplexes messages pertaining to different QMI Services by calling out the Service to which the transported message pertains. The QMUX enables multiplexing of messages corresponding to different Control Points by identifying the corresponding client ID in each message sent over the control channel. The QMUX also identifies the sending endpoint type (Control Point or Service). Each QMI Service is assigned a unique QMI Service type identifier.

2.2.2. Rules for QMUX message handling

2.2.2.1. Generating QMUX messages

A QMUX protocol implementation: Appends a preamble and QMUX header to outgoing QMI Service messages, including the originating or destination Control Point's allocated client ID and the QMI Service type to which the message corresponds Sends the QMUX message over the control channel of the associated logical device

2.2.2.2. Receiving QMUX messages

A QMUX protocol implementation: Removes the preamble and QMUX header from incoming QMI Service messages Delivers the messages to the appropriate Service or Control Point based on the QMI Service type and current client ID mappings Provides an application adaptation layer (e.g., application API) to allow the application to send and receive Service messages and obtain/release client IDs. If the QMI Service type does not correspond to a Service implementation on the receiving device, the QMUX layer will discard the message.

2.2.3. Client ID management

Applications obtain/release client IDs from the QMI device driver on the host via the application adaptation layer. The details of this API are particular to the QMUX implementation and are thus outside the scope of this document. The reference implementation provided by QUALCOMM. The QMUX layer implementations in both the device and host driver implement the QMI control messages, which provide client ID management functions. These are used to implement the application client ID API.

2.3. QMI Generalized Service Message Protocol

This chapter describes the generalized message format and procedures that QMI Services should follow to ease implementation. If a particular QMI Service diverges from this protocol, the corresponding QMI Service specification will document the superseding message format and/or procedures for that particular Service.

2.3.1. Service PDU Message Format

This section describes the message format generally followed by QMI Services.

2.3.1.1. Byte Ordering

Numeric data in QMI Service messages shall be transmitted in little-endian format.

2.3.1.2. QMI transaction Structure

A QMI Service sends and receives QMI transactions. A QMI transaction contains one or more QMI Service messages. **Figure 6** describes the format of a single QMI Service transaction.

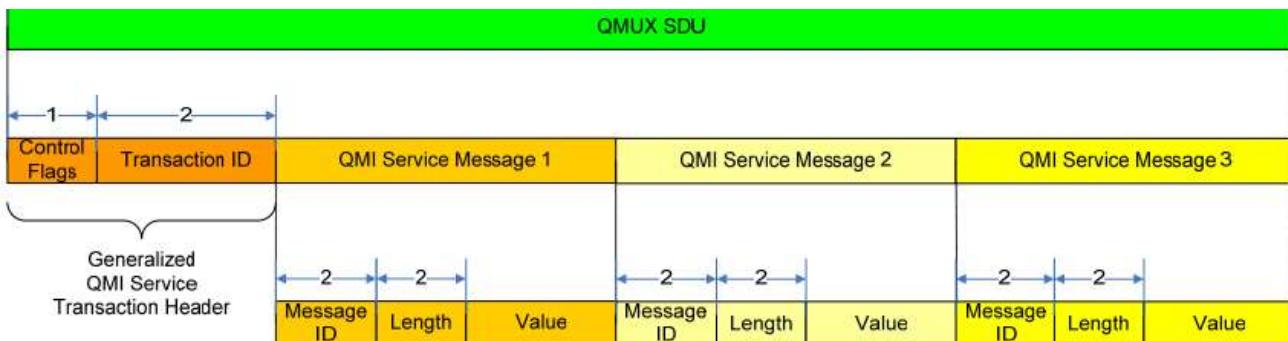


Figure 6: Generalized QMI Service Transaction Format

A QMI Service transaction starts with the general Service transaction header followed by one or more Service messages of the same QMI Service type. The structure allows multiple Service messages of the same QMI Service type to be “bundled” together in the same transaction and dispatched in the same QMUX SDU. The lower layer providing transport for the transaction (QMUX) delivers the transaction as a whole to the QMI peer (Control Point or Service). The format of the generalized QMI Service message header is described in table Below.

Table 3: QMI Service Transaction Header

QMI Service header field		Offset (bytes)	Length	Description/values														
Control Flags	Reserved	0	Bit 0	Must be 0														
	Message type	0	Bits 1-2	Message type for QMI Service message(s) present in the transaction: <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>Bit 2</th> <th>Bit 1</th> <th>Message type</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>Request</td> </tr> <tr> <td>0</td> <td>1</td> <td>Response</td> </tr> <tr> <td>1</td> <td>0</td> <td>Indication¹</td> </tr> <tr> <td>1</td> <td>1</td> <td>Reserved</td> </tr> </tbody> </table> Message types are described in Section 4.2	Bit 2	Bit 1	Message type	0	0	Request	0	1	Response	1	0	Indication ¹	1	1
Bit 2	Bit 1	Message type																
0	0	Request																
0	1	Response																
1	0	Indication ¹																
1	1	Reserved																
Reserved	0	Bits 3-7	Must be 0															
Transaction ID	1	2 bytes	The transaction ID is an identifier used to distinguish transactions. Transaction IDs are unique among transactions for a particular Control Point. The Control Point must increment the transaction ID each time a new transaction is sent. The transaction containing the response will use the same transaction ID. Each Control Point maintains its own transaction ID counter. This must always be a nonzero value.															

Indications cannot be bundled.

2.3.1.3. QMI message structure

A single QMI Service message is formatted as described in *Figure 7*.

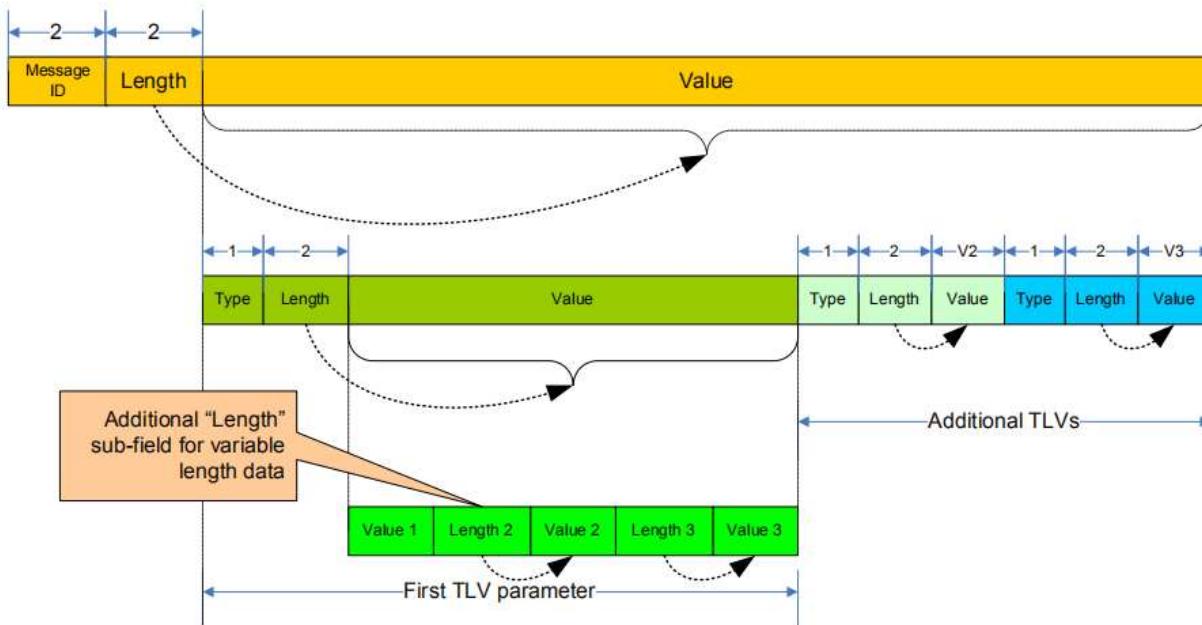


Figure 7: Generalized QMI Service Message and Parameter Formats

The QMI Service messages are distinguished by QMI message ID. Each QMI Service type has its own set of QMI messages, defined in the QMI Service specification document. The same message ID value is used in corresponding request and response messages. If an indication message is defined corresponding to the request and response, it will share the same message ID value. The length field following the message ID indicates the total number of bytes in the message following the length field, i.e., the total length of all parameters included in the message. The value portion of the message consists of zero or more parameters associated with the message. The value typically contains the information required to execute the requested action or results of the action.

2.3.1.4. QMI message parameter structure

Figure 7 illustrates parameters within the value portion of the QMI message. Message parameters are defined separately for each request, response, or indication message. Message parameters are formatted with three sections, type, length, and value. Because of this, message parameters are sometimes referred to as TLVs.

(1) Parameter types

The parameter type field indicates which parameter is being specified. A unique TLV parameter type is defined for each parameter that may be specified within a given message type. The same parameter type may have a different meaning in the context of other messages.

(2) Parameter length

The parameter length indicates the length in bytes of the following value field. The expected length will be documented per parameter in the QMI Service specification. This will be a fixed value when the value field is a fixed structure. If the parameter contains a string or other variable-length data, this will be defined as a calculated value. For example, if the value section includes a variable length string, the length field will tell the receiver how many bytes are in that string.

(3) Parameter value

The value of a parameter contains the actual information communicated by including the parameter in the message. The entire parameter, as defined in the Service specification, must be present. Any flexibility in format of the value portion of the parameter will be described in the parameter value description. All numeric data are positive (unsigned) binary values unless stated otherwise in the parameter description.

2.3.2. QMI message types

The generalized QMI Service transaction format defines three basic message types. All three message types follow the generalized QMI Service message format described in **Section 4.1.3**. A request message may be used to set parameters, query parameter values, or configure the generation of indications. The request message is issued by the Control Point. A valid request always generates a response from the Service.

2.3.2.1. Response

A response message is issued by the Service, in response to a received request. Each response contains at least the result parameter indicating that the request succeeded or failed, and the error status, indicating the result of the operation requested. Additional parameters may be present to communicate data associated with the operation. An indication is sent by a QMI Service to inform Control Point(s) of changes in state. The indication message is issued by the Service without any solicitation by a Control Point.

(1) Unicast vs. broadcast indications

Indications from the Service are either broadcast to all Control Points or unicast to a specific Control Point. Indication type is indicated by the value of the client ID field in the QMUX header. The definition of the indication message (in the associated QMI Service specification document) specifies whether it shall be unicast or broadcast.

2.3.3. Rules for transaction handling

A generalized QMI Service receives a QMUX SDU from the QMUX layer when the QMUX header identifies the Service type corresponding to that Service. This QMUX SDU contains a single QMI Service transaction. When a request transaction is received, the QMI Service must: Remember the transaction ID and use that same value in the transaction ID of the corresponding response message sent back to the Control Point Process all requests in a transaction, in the order specified; all requests in the bundle will be processed, regardless of the result of preceding bundled request(s). Return all corresponding responses in the same order as the requests in a single response transaction (i.e., sent in the same QMUX SDU); no additional messages may be added to this response transaction

Send the response transaction only when all requests are complete. If any message length field (per the message header) in the transaction indicates data past the end of the transaction (per the transaction header), the entire transaction will be considered malformed and none of the contained requests will be processed. This is done since it is ambiguous to the receiver whether the message indicating the invalid data length is corrupt or if a previous message was corrupt. This may lead the receiver to misinterpret data as a subsequent message header. If the Service cannot process the transaction because too many QMI messages were contained, none of the requests will be processed. This is done to avoid partially executing the Control Point's requests. Unless otherwise specified in the QMI Service specification, a generalized QMI Service will support up to five request messages in a transaction. This may be further limited by outstanding message limitations described in Section 4.4. Note that when multiple requests are contained in a transaction, the response will be delayed until all commands have been processed. Hence, if the processing/execution of one request contained in such a transaction is much slower than the other requests, the Control Point will not get any results until the slowest message has executed. If a Control Point wishes to execute a request conditional to the result of a preceding request, it should issue each in a separate transaction.

2.3.4. Rules for message handling

2.3.4.1. General rules

All messages generated by that QMI Service and its Control Points must set the Service type field in the QMUX header to the QMI Service type assigned to that Service. The receiver shall verify that the message and any parameters are formatted correctly before processing the message. Unless otherwise specified in the message definition, messages apply to all QMI devices, independent of the wireless technology employed by the device. All QMI Service messaging is asynchronous in that the receiver must not expect a response to be returned immediately following the issuing of the request. Further, a Control Point may issue a subsequent request before the response to the first is received. Indications and responses generated by the Service may be sent in any order.

2.3.4.2. Generating requests

Control Points will formulate requests according to the general Service message protocol or any superseding protocols defined in the QMI Service specification document.

2.3.4.3. Receiving requests

If the length of a parameter specified in the received request is incorrect, the Service will be unable to process the request. In this case, the Service will fail the request with QMI_ERR_MALFORMED_MSG. Unless otherwise specified in the QMI Service specification, a particular client is limited to five outstanding requests at a given time. An outstanding request is one for which a QMI response has not been received by the Control Point (not including discarded requests, e.g., requests in malformed transactions). Typically, the QMI Service will generate and send the response immediately while processing the request message and executing the command. However, some actions (such as a network scan) take some time in which case the sending of the response will be delayed until the action is complete.

2.3.4.4. Generating responses

For each QMI request received, the response shall include a result parameter that indicates success or failure and an error code. If failure is returned, an error status value is returned in the result parameter and none of the associated mandatory parameters will be included in the response, since there is no valid information to return. The set of possible error values returnable for each command is defined in the Service specification document. Additional TLVs may be included on a per-message basis to communicate required information, e.g., queried device parameter. If an unrecognized request is received by the Service, it may return a response using the unrecognized message type, and a single result TLV indicating result failure and error code QMI_ERR_UNRECOGNIZED_REQUEST.

2.3.4.5. Receiving responses

A QMI Control Point must be prepared to receive a response some time after the request is sent. The Control Point must be prepared to receive responses to requests in a different order than the requests are issued. Further, the Control Point must be able to process indication messages at any time; i.e., it must not expect a response to be the next message received after the request message.

2.3.4.6. Generating indications

A Service may broadcast a single indication message across all registered Control Points by setting the QMUX header client ID field to 0xFF. Broadcast indications are defined for Service status that is of interest to all Control Points, reducing the QMI link bandwidth needed to deliver that information to all. When the QMUX layer on the receive side receives a broadcast indication, it must generate and deliver one copy of the received broadcast indication to each Control Point registered to that Service. Unicast indications are addressed to a particular Control Point by setting the QMUX client ID field to the intended Control Point's client ID value. Unicast indications convey status particular to a single Control Point.

2.3.4.7. Receiving indications

The Control Point may take action based on the contents and/or update any associated user interfaces. The Control Point shall ignore any received indications that are unrecognized or unsupported. The transaction ID field of a transaction containing an indication message should be ignored by the Control Point.

2.3.5. Rules for parameter handling

Messages may have mandatory and/or optional parameters associated with them. Unless explicitly specified in the definition of the message, there is no particular ordering requirement for parameters.

The parameters are specified in the general format of [type|length|value] and hence are sometimes referred to as TLVs.

Mandatory parameters (TLVs): Must be included in the message by the sender in all cases Are assigned type field values less than 0x10

Optional parameters (TLVs): Are generated and/or processed at the Service or Control Point at their discretion; i.e., need not be implemented unless otherwise specified

Are assigned type field values greater than 0x0F

Must be ignored by the receiver if unrecognized/not implemented

Must not be relied on by the receiver; i.e., if optional parameters are not specified in a response or indication message, this should not cause an error or disrupt proper operation of the Control Point

May be conditionally required to be included on a per-message basis, as described in the message definition in the appropriate QMI Service specification

2.3.6. State variables

QMI Services may keep track of state related to the internal functionality accessed through that Service in Service global state variables. The Service may also keep track of Control Point settings and state in Control Point state variables. When a Control Point is allocated a new client ID, and when that client ID is released, that client ID's state variables are set to the default settings. Upon powerup, and when the QMI link is disconnected, Service global state variables are reset to their default settings. The handling of state variables and their impact on the system is described in the QMI Service specification document.

2.3.7. Control Point arbitration

It is possible to have multiple Control Points interact with a single Service on the QMI device. In cases where multiple Control Points issue messages related to a common resource, the default policy is that the actions will be executed in the order received; hence, the “last request wins.” In some cases, more careful arbitration of a common resource is managed by keeping track of Control Point requests via state variables. In such cases, the message definition may describe any the arbitration policy for the common resource.

2.3.8. QMI Service versioning

QMI Control Points and QMI Services are written to a particular version of a QMI Service specification document. Since the Service specifications are compiled to over time, Control Points may want to know the Service version implemented on a device, to know whether specific functionality within the Service is supported.

2.3.8.1. Version format

Each QMI Service has its own version number that is independent of other QMI Services. A QMI Service version is represented as M.n where:

M=major version, 2 bytes

n=minor version, 2 bytes

2.3.8.2. Learning QMI Service versions

The QMI driver on the TE provides an API to learn the Service version.

2.3.8.3. Service versioning rules

Major versions

As the major version of the Service is incremented, the Service specification is changed in a way that breaks backward compatibility with the previous version. A QMI Service is required only to support one major version of a QMI Service. A QMI Service may implement multiple major versions of a QMI Service.

Minor versions

The minor version of the Service is incremented when the Service specification is modified without breaking backward compatibility with previous versions sharing the same major revision number. Control Points may assume interoperability with a Service that has a different minor version. Each message definition will indicate the QMI Service version in which it was first defined. The Control Point should consider this the minimum required Service version to carry out the operation associated with that request. Each parameter definition will indicate the QMI Service version in which it was last modified. Since the backward compatibility requirement implicit to QMI ensures that parameters will not be changed in a way that renders an older minor revision incompatible, it is not critical for the application to take action based on the last modified version. This is provided as a quick means for the application writer to identify updated fields in a newer Service specification that might be handled by the application; however, the application will work without implementing any of these changes.

3 ofono-1.30 Contains QMI Service Type Values

Table 4 lists QMI service type values that are currently defined. These values are used to specify to which QMI service the messages are routed.

Table 4: QMI Service and Values

QMI Service	QMI service type value
QMI_CTL (Control Service)	0x00
QMI_WDS (Wireless Data Service)	0x01
QMI_DMS (Device Management Service)	0x02
QMI_NAS (Network Access Service)	0x03
QMI_WMS (Wireless Messaging Service)	0x05
QMI_PDS (Position Determination Service)	0x06
QMI_VOICE (Voice Service)	0x09
QMI_UIM (User Identity Module)	0x0B
QMI_WDA (Wireless Data Administrative Service)	0x1A

4 Control Service (QMI_CTL)

QMI_CTL is a QMI service within the QMI framework defined in 80-VB816-1. QMI_CTL messages are transported over the QMUX Control Message Transport Protocol.

QMI_CTL provides the QMUX layer on the Terminal Equipment (TE), e.g., the host driver, commands related to the QMUX link, and client management:

QMUX link identification

QMI service version identification

QMI service client ID allocation and deallocation/revocation

It is expected that the QMI function-level driver and/or QMUX layer implementation on the TE will use QMI_CTL to access this functionality on the MSM™ device.

4.1. Theory of Operation

4.1.1. Generalized QMI Service Compliance

The QMI_CTL 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 80-VB816-1. Compliance exceptions include:

The QMI_CTL PDU format differs from the Generalized QMI Service PDU format in that transaction ID is a single byte in length.

Multiple QMI_CTL messages (SDUs) cannot be transmitted (bundled) in a single QMUX PDU.

Extensions to the generalized QMI service theory of operation are noted in subsequent sections of this chapter.

4.1.2. CTL Service Type

CTL is assigned QMI service type 0x00.

4.1.3. Message Definition Template

4.1.3.1. Byte Ordering

Numeric values in QMI_CTL messages are encoded in little-endian format. String values in QMI_CTL messages are sent from the first to the last character (i.e., the same order that is stored in memory for most architectures).

4.1.3.2. QMI_CTL PDU

QMI_CTL messages consist of a short PDU header that is followed by the QMI_CTL message, as illustrated in Figure 8.

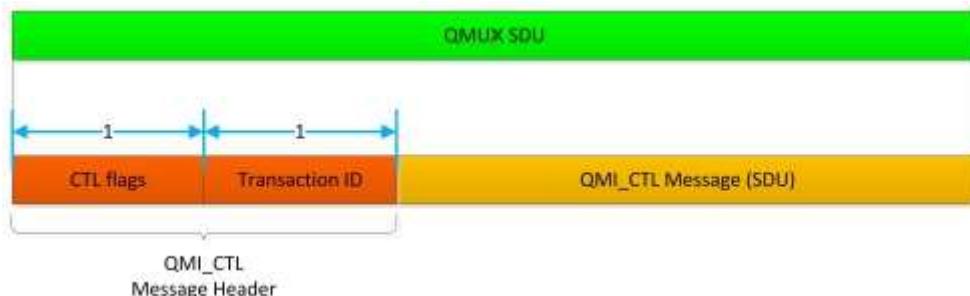


Figure 8: QMI_CTL PDU Format

NOTE

The QMI_CTL PDU must contain only one QMI_CTL message.

The QMI_CTL message (SDU) conforms to the QMI Generalized Service Message (SDU) format described in 80-VB816-1.

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

4.1.3.3. Response Message Result TLV

This Type-Length-Value (TLV) is present in all Response messages defined in this document. It is not present in the Indication messages.

4.1.4. QMI_CTL Fundamental Concepts

4.1.4.1. QMI_CTL Control Point

The QMUX endpoints on the TE (e.g., host driver) and the device use QMI_CTL to exchange information related to QMI services and to QMI client management within those services. Client ID 0x00 is implicitly assigned to the host driver for the purpose of exchanging QMI_CTL messages; therefore, the client_id field of all QMI_CTL messages it sends and receives is 0x00. This is required, since QMI_CTL provides the client ID management function and cannot assign a client ID to itself.

4.1.4.2. QMI_CTL Service Version

QMI_CTL provides a means to learn the versions of each QMI service supported by the MSM device. The QMI_CTL control point on the TE (e.g., driver) should first verify that the QMI_CTL service version is compatible before it performs client ID management operations.

4.1.4.3. QMI Link ID

A QMI-enabled MSM device can support multiple logical QMI connections to a TE. Each is capable of exchanging QMI messages and is referred to as a QMI link. In cases where the QMI-enabled devices need to be distinguished on the host (e.g., connecting two such devices to the same TE), a QMI link ID can be assigned to each QMI link using primitives provided by this service. It is good practice to do so, in case this scenario arises.

4.1.4.4. Client ID Management

QMI_CTL defines procedures for assigning unique client IDs to allow other QMI services (besides QMI_CTL) on the MSM device to serve multiple control points. For example, the TE driver can use QMI_CTL to request multiple client IDs from a QMI service on the MSM, and can assign these client IDs to control points on the TE on request. Unique client IDs enable a resource manager on the MSM to enforce arbitration policies when messages are processed from different QMI control points. When applications request a client ID for any QMI service, the QMI_CTL control point may provide the service version supported by the device. This enables the application to identify the extent of service-specific functionality that is supported.

4.1.5. Service State Variables

4.1.5.1. State Variables Per Control Point

Name	Description	Possible values	Default value
report_svc_available	Indicates whether available QMI services are to be reported.	<ul style="list-style-type: none"> · 0 – Do not report · 1 – Report available QMI services 	0

4.2. QMI_CTL Messages

Table 5: QMI_CTL Messages

Command	ID	Description
QMI_CTL_SET_INSTANCE_ID	0x0020	Generates a unique ID to distinguish the QMI link over which the message is sent.
QMI_CTL_GET_VERSION_INFO	0x0021	Queries the versions of all QMI services supported by the device.
QMI_CTL_GET_CLIENT_ID	0x0022	Requests a client ID for the specified QMI service type.
QMI_CTL_RELEASE_CLIENT_ID	0x0023	Releases a previously assigned client ID.
QMI_CTL_REVOKE_CLIENT_ID	0x0024	Indicates that a client ID has been revoked by the service.
QMI_CTL_INVALID_CLIENT_ID_IND		
QMI_CTL_INVALID_CLIENT_ID	0x0025	Indicates that a client ID/service type pair specified in a QMUX header is invalid.
QMI_CTL_INVALID_CLIENT_ID_IND		

QMI_CTL_SET_DATA_FORMAT	0x0026	Indicates the MSM device of the data format used by the driver.
QMI_CTL_SYNC	0x0027	Synchronizes the service provider and service consumer.
QMI_CTL_SYNC QMI_CTL_SYNC		
QMI_CTL_SET_EVENT	0x0028	Synchronizes the service provider and service consumer.
QMI_CTL_REG_PWR_SAVE_MODE		
QMI_CTL_SET_POWER_SAVE_CONFI		
G	0x0029	Configures the event indication filter by the Power Save state for each QMI service.
QMI_CTL_CONFIG_PWR_SAVE_SETTING		
QMI_CTL_SET_POWER_SAVE_MODE	0x002A	Sets the QMI framework Power Save mode.
QMI_CTL_SET_PWR_SAVE_MODE		
QMI_CTL_GET_POWER_SAVE_MODE	0x002B	Gets the current QMI framework Power Save mode.
QMI_CTL_GET_PWR_SAVE_MODE		

4.2.1. QMI_CTL_SET_INSTANCE_ID

Generates a unique ID to distinguish the QMI link over which the message is sent.

- **CTL message ID**

0x0020

- **Version introduced**

Major-1, Minor-0

4.2.1.1. Request - QMI_CTL_SET_INSTANCE_ID_REQ

- **Message type**

Request

- **Sender**

Control point

- **Mandatory TLVs**

Name	Version introduced	Version last modified
Host Driver Instance	1.0	1.0

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Host Driver Instance
Length	1			2	
Value	→	uint8	host_driver_instance	1	Host-unique QMI instance for this device driver.

- Optional TLVs

None

4.2.1.2. Response - QMI_CTL_SET_INSTANCE_ID_RESP

- Message type

Response

- Sender

Service

- Mandatory TLVs

The Result Code TLV (defined in Section 4.1.3.3) is always present in the response. This TLV is present if the result code is QMI_RESULT_SUCCESS.

Name	Version introduced	Version last modified
QMI Link ID	1.0	1.0
Result Code	1.0	1.0

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	QMI Link ID
Length	2			2	
Value	→	uint16	qmi_id	2	Unique QMI link ID assigned to the link over which the message is exchanged. The upper byte is assigned by the QMI_CTL service and the lower byte is assigned by the host (the value passed in the 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_MISSING_ARG	One or more required TLVs were missing in the request

4.2.1.3. Description of QMI_CTL_SET_INSTANCE_ID REQ/RESP

This command facilitates the assignment of a unique QMI link ID to the physical channel carrying the QMUX messages.

The QMI_CTL control point on the TE is required to send this message when there is a need to distinguish QMI links. This occurs in the following interconnection configurations:

Multiple QMI-enabled MSM devices connected to a single TE

Multiple TEs connected to a single QMI-enabled MSM device

The returned QMI link ID is the concatenation of the host identifier byte with the device identifier byte.

4.2.2. QMI_CTL_GET_VERSION_INFO

Queries the versions of all QMI services supported by the device.

- CTL message ID

0x0021

- Version introduced

Major – 1, Minor – 0

4.2.2.1. Request-QMI_CTL_GET_VERSION_INFO_REQ

- Message type

Request

- Sender

Control point

- **Mandatory TLVs**

None

- **Optional TLVs**

None

4.2.2.2. Response – QMI_CTL_GET_VERSION_INFO_RESP

- **Message type**

Response

- **Sender**

Service

- **Mandatory TLVs**

The Result Code TLV (defined in Section 4.1.3.3) is always present in the response. This TLV is present if the result code is QMI_RESULT_SUCCESS.

Name	Version introduced	Version last modified
QMUX Service Version List	1.0	1.0
Result Code	1.0	1.0

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	QMUX Service Version List
Length	Var			2	
Value	→	uint8	service_version_list_len	1	Number of sets of the following elements: <ul style="list-style-type: none">• qmi_svc_type• major_ver• minor_ver
		uint8	qmi_svc_type	1	QMI service type, as defined in [Q2].
		uint16	major_ver	2	Major version number of the QMI service specified by qmi_svc_type.
		uint16	minor_ver	2	Minor version number of the QMI service specified by qmi_svc_type.

- Optional TLVs

Name	Version introduced	Version last modified
Addendum Version List	1.2	1.2

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Addendum Version List
Length	Var			2	
Value	→	uint8	addendum_label_len	1	Number of sets of the following elements: <ul style="list-style-type: none">• addendum_label
		string	addendum_label	Var	Label describing the addendum.
	→	uint8	addendum_version_list_len	1	Number of sets of the following elements: <ul style="list-style-type: none">• qmi_svc_type• addendum_major_ver• addendum_minor_ver
		uint8	qmi_svc_type	1	QMI service type, as defined in [Q2] .
		uint16	addendum_major_ver	2	Addendum major version number of the QMI service specified by qmi_svc_type.
		uint16	addendum_minor_ver	2	Addendum minor version number of the QMI service specified by qmi_svc_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

4.2.2.3. Description of QMI_CTL_GET_VERSION_INFO REQ/RESP

This command queries the major and minor version numbers of all QMI services, including QMI_CTL, supported by the QMI-enabled device.

Each QMI service version can be represented as a base version (Addendum version).

Every QMI service has a base version that is represented by major and minor version numbers given by the QMI service version list mandatory TLV, described in Section 4.2.2.2.

When a QMI service wants to advertise additional functionality supported on top of the base version, the service has an addendum version given by the Addendum Version optional TLV.

The addendum label name is a text string that is a label for the overall QMI addendum. An addendum

major and minor version is present for each QMI service that wants to advertise additional functionality.

4.2.3. QMI_CTL_GET_CLIENT_ID

Requests a client ID for the specified QMI service type.

- **CTL message ID**

0x0022

- **Version introduced**

Major-1, Minor-0

4.2.3.1. Request-QMI_CTL_GET_CLIENT_ID_REQ

- **Message type**

Request

- **Sender**

Control point

- **Mandatory TLVs**

Name	Version introduced	Version last modified
QMI Service Type	1.0	1.0

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	QMI Service Type
Length	1			2	
Value	→	uint8	qmi_svc_type	1	QMI service type for which a client ID is requested.

- **Optional TLVs**

None

4.2.3.2. Response-QMI_CTL_GET_CLIENT_ID_RESP

- **Message type**

Response

- **Sender**

Service

- **Mandatory TLVs**

The Result Code TLV (defined in Section 4.1.3.3) is always present in the response. This TLV is present if the result code is QMI_RESULT_SUCCESS.

Name	Version introduced	Version last modified
Assigned Client ID	1.0	1.0
Result Code	1.0	1.0

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Assigned Client ID
Length	2			2	
Value	→	uint8	qmi_svc_type	1	QMI Service type.
		uint8	client_id	1	Client ID.

- **Optional TLVs**

None

- **Error codes**

QMI_ERR_NONE	No error in 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_MISSING_ARG	One or more required TLVs were missing in the request
QMI_ERR_CLIENT_IDS_EXHAUSTED	Maximum number of concurrently assignable client IDs have already been allocated by the service
QMI_ERR_INVALID_SERVICE_TYPE	QMI service type is not supported by the device, or the QMI service does not assign client IDs dynamically

4.2.3.3. Description of QMI_CTL_GET_CLIENT_ID REQ/RESP

This command obtains a client ID from the specified QMI service. The client ID is assigned by the issuer of this request to a specific control point (application).

The service type cannot be QMI_CTL. There is only a single control point (the QMI driver on the TE) for

the QMI_CTL service.

4.2.4. QMI_CTL_RELEASE_CLIENT_ID

Releases a previously assigned client ID.

- **CTL message ID**

0x0023

- **Version introduced**

Major - 1, Minor - 0

4.2.4.1. Request-QMI_CTL_RELEASE_CLIENT_ID_REQ

- **Message type**

Request

- **Sender**

Control point

- **Mandatory TLVs**

Name	Version introduced	Version last modified
Client ID to Release	1.0	1.0

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Client ID to Release
Length	2			2	
Value	→	uint8	qmi_svc_type	1	QMI Service type.
		uint8	client_id	1	Client ID.

- **Optional TLVs**

None

4.2.4.2. Response-QMI_CTL_RELEASE_CLIENT_ID_RESP

- **Message type**

Response

- **Sender**

Service

- **Mandatory TLVs**

The Result Code TLV (defined in Section 4.1.3.3) is always present in the response. This TLV is present if the result code is QMI_RESULT_SUCCESS.

Name	Version introduced	Version last modified
Released Client ID	1.0	1.0
Result Code	1.0	1.0

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Released Client ID
Length	2			2	
Value	→	uint8	qmi_svc_type	1	QMI Service type.
		uint8	client_id	1	Client ID.

- **Optional TLVs**

None

- **Error codes**

QMI_ERR_NONE	No error in 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_MISSING_ARG	One or more required TLVs were missing in the request
QMI_ERR_INVALID_CLIENT_ID	Client ID to be released was not allocated by the specified QMI service
QMI_ERR_INVALID_SERVICE_TYPE	QMI service type is not supported by the device, or the QMI service does not assign client IDs dynamically

4.2.4.3. Description of QMI_CTL_RELEASE_CLIENT_ID REQ/RESP

This command releases a client ID that was previously assigned by the specified QMI service.

4.2.5. QMI_CTL_REVOKED_CLIENT_ID_IND

Indicates that a client ID has been revoked by the service.

- CTL message ID

0x0024

- Version introduced

Major - 1, Minor - 0

4.2.5.1. Indication - QMI_CTL_REVOKE_CLIENT_ID_IND

- Message type

Indication

- Sender

Service

- Indication scope

Unicast (per control point)

- Mandatory TLVs

Name	Version introduced	Version last modified
Revoked Client ID	1.0	1.0

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Revoked Client ID
Length	2			2	
Value	→	uint8	qmi_svc_type	1	QMI Service type.
		uint8	client_id	1	Client ID.

- Optional TLVs

None

4.2.5.2. Description of QMI_CTL_REVOKE_CLIENT_ID_IND

This indication is sent if the service determines that it needs to revoke an assigned client ID.

The receiver should ensure that no further messages are sent using the revoked client ID, unless it has been subsequently reassigned via a QMI_CTL_GET_CLIENT_ID request.

4.2.6. QMI_CTL_INVALID_CLIENT_ID_IND

Indicates that a client ID/service type pair specified in a QMUX header is invalid.

- **CTL message ID**

0x0025

- **Version introduced**

Major - 1, Minor - 0

4.2.6.1. Indication - QMI_CTL_INVALID_CLIENT_ID_IND

- **Message type**

Indication

- **Sender**

Service

- **Indication scope**

Unicast (per control point)

- **Mandatory TLVs**

Name	Version introduced	Version last modified
Invalid Client ID	1.0	1.0

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Invalid Client ID
Length	2			2	
Value	→	uint8	qmi_svc_type	1	QMI Service type.
		uint8	client_id	1	Client ID.

- **Optional TLVs**

None

4.2.6.2. Description of QMI_CTL_INVALID_CLIENT_ID_IND

This indication may be generated when a QMI service message is received with an invalid client ID specified.

NOTE

There is no guarantee that this message will be generated. The QMI_CTL service may limit the generation rate of this indication to avoid overflowing the control path with frequent indications.

4.2.7. QMI_CTL_SET_DATA_FORMAT

Indicates the MSM device of the data format used by the driver.

- **CTL message ID**

0x0026

- **Version introduced**

Major - 1, Minor - 1

4.2.7.1. Request - QMI_CTL_SET_DATA_FORMAT_REQ

- **Message type**

Request

- **Sender**

Control point

- **Mandatory TLVs**

Name	Version introduced	Version last modified
Data Format	Unknown	1.3

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Data Format
Length	1			2	
Value	→	enum8	data_format	1	Data format used by the driver. Values: • 0 – No QOS flow header • 1 – QOS flow header present

- Optional TLVs

Name	Version introduced	Version last modified
Underlying Link Layer Protocol	Unknown	1.3
Uplink Data Aggregation Protocol	Unknown	1.6

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Underlying Link Layer Protocol
Length	2			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	mask16	link_prot	2	Bitmask of the link protocols supported by the driver. If multiple protocols are supported, they are OR'ed together as a mask. Values: • 0x1 – 802.3 • 0x2 – IP
Type	0x11			1	Uplink Data Aggregation Protocol
Length	1			2	
Value	→	enum8	ul_data_agg_setting	1	Data aggregation protocol to be used for uplink data transfer. Values: • 0x0 – Disable data aggregation • 0x1 – TLP (Thin Layer Protocol)

4.2.7.2. Response - QMI_CTL_SET_DATA_FORMAT_RESP

- Message type

Response

- Sender

Service

- **Mandatory TLVs**

The Result Code TLV (defined in Section 4.1.3.3) is always present in the response.

Name	Version introduced	Version last modified
Result Code	1.0	1.3

- **Optional TLVs**

Name	Version introduced	Version last modified
Underlying Link Layer Protocol	Unknown	1.3
Configured Uplink Data Aggregation Protocol	Unknown	1.6

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Underlying Link Layer Protocol
Length	2			2	
Value	→	mask16	link_prot	2	Link protocol used by the driver. Only one protocol in the response indicates the mode to be used. Values: <ul style="list-style-type: none">• 0x1 – 802.3• 0x2 – IP

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x11			1	Configured Uplink Data Aggregation Protocol
Length	1			2	
Value	→	enum8	ul_data_agg_setting	1	Data aggregation protocol configured on the device. Values: <ul style="list-style-type: none">• 0x0 – Disabled• 0x1 – TLP (Thin Layer Protocol)

- **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_MISSING_ARG	One or more required TLVs were missing in the request
QMI_ERR_INVALID_DATA_FORMAT	Value of the data format was incorrect

4.2.7.3. Description of QMI_CTL_SET_DATA_FORMAT REQ/RESP

Any change in data format should be performed only when the driver is initializing. The format should not be changed dynamically.

When the data format in the mandatory Data Format TLV is set to 1, the driver must add a 6-byte QOS flow header to the start of the data packet. This header is useful only if you want to access QOS when using a device that supports the QMI_QOS service. In the absence of QMI_QOS, the driver should not use this message. Instead, the driver should use the default data format (i.e., the QOS flow header is not present) or should set the data format to 0 if the driver needs to use this message for other TLVs.

If the driver does not receive a response or it receives an error response, the driver should assume that this data format is not supported. The default data format should be used (i.e., send the data packets without the QOS flow header).

Figure 9 illustrates the fields of the QOS flow header.



Field	Value
Version	1
Resvd	0 (for future use)
Flow_id	4-byte flow identifier indicating the flow to which the packet belongs; must be set to 0 if the packet belongs to the default (best effort) flow

Figure 9: QOS flow header

NOTES

1. The QOS flow header is present only in the up (reverse) link direction. It is not present in the down (forward) link direction.

If QOS is not used, the data format in the Mandatory Data Format TLV in the request must be set to a value of 0.

The default underlying link layer protocol is 802.3. To change the protocol to use another mode, such as Raw IP, the underlying link layer protocol optional TLV in the request must be used to specify the modes supported by the driver. The device then chooses the protocol and uses the underlying link

layer protocol-optional TLV in the response to indicate the mode which is to be used by the driver.

The default data aggregation protocol setting is disabled by default on the device. To change the setting, the Uplink Data Aggregation Protocol optional TLV must be included in the request. The device then uses the Configured Uplink Data Aggregation Protocol optional TLV in the response to reflect whether the setting took effect.

2. In the future, this interface will be deprecated and will be replaced by a new API.

4.2.8. QMI_CTL_SYNC

Synchronizes the service provider and service consumer.

- **CTL message ID**

0x0027

- **Version introduced**

Major - 1, Minor - 4

4.2.8.1. Request - QMI_CTL_SYNC_REQ

- **Message type**

Request

- **Sender**

Control point

- **Mandatory TLVs**

None

- **Optional TLVs**

None

4.2.8.2. Response - QMI_CTL_SYNC_RESP

- **Message type**

Response

- **Sender**

Service

- **Mandatory TLVs**

The Result Code TLV (defined in Section 4.1.3.3) is always present in the response.

Name	Version introduced	Version last modified
Result Code	1.4	1.4

- **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

4.2.8.3. Indication - QMI_CTL_SYNC_IND

- **Message type**

Indication

- **Sender**

Service

- **Indication scope**

Unicast (per control point)

- **Mandatory TLVs**

None

- **Optional TLVs**

None

4.2.8.4. Description of QMI_CTL_SYNC_IND

This indication synchronizes service and control points. The QMI_CTL_SYNC_IND command is only sent at modem bootup, including modem restarts. The indication is delivered using a backoff schedule of 0s, 1s, 2s, 4s, 8s, 16s, 32s, and 64s, after which the service stops sending indications. See Appendix B for state diagrams that explain the behavior of service and control points.

4.2.9. QMI_CTL_REG_PWR_SAVE_MODE

Sets the Control Power Save mode state reporting conditions for the requesting control point.

- **CTL message ID**

0x0028

- **Version introduced**

Major - 1, Minor - 4

4.2.9.1. Request - QMI_CTL_REG_PWR_SAVE_MODE_REQ

- **Message type**

Request

- **Sender**

Control point

- **Mandatory TLVs**

Name	Version introduced	Version last modified
Power Save Normal State Report	1.4	1.4

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Power Save Normal State Report
Length	1			2	
Value	→	boolean	report_pwrsnormal_state	1	Reports when the Power Save state changes to NORMAL. Values: • 0 – Do not report • 1 – Report when Power Save state changes to NORMAL

- **Optional TLVs**

None

4.2.9.2. Response - QMI_CTL_REG_PWR_SAVE_MODE_RESP

- **Message type**

Response

- **Sender**

Service

- **Mandatory TLVs**

The Result Code TLV (defined in Section 4.1.3.3) is always present in the response.

Name	Version introduced	Version last modified
Result Code	1.4	1.4

- **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_MISSING_ARG	One or more required TLVs were missing in the request

4.2.9.3. Description of QMI_CTL_REG_PWR_SAVE_MODE REQ/RESP

The QMI_CTL_PWR_SAVE_MODE_IND indication is sent on return to NORMAL/Full Power mode if the control point is registered for the Power Save Normal State Report TLV.

4.2.9.4. Indication - QMI_CTL_PWR_SAVE_MODE_IND

- **Message type**

Indication

- **Sender**

Service

- **Indication scope**

Unicast (per control point)

- **Mandatory TLVs**

None

- **Optional TLVs**

Name	Version introduced	Version last modified
Power Save State Report	1.4	1.4

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Power Save State Report
Length	8			2	
Value	→	uint32	current_state	4	Handle specified by the control point representing the Power Save state; indicates the state after transition.
		uint32	previous_state	4	Handle specified by the control point representing the Power Save state; indicates the state prior to transition.

4.2.9.5. Description of QMI_CTL_PWR_SAVE_MODE_IND

Interested control points must previously register for the reporting using the QMI_CTL_REG_PWR_SAVE_MODE_REQ message.

4.2.10. QMI_CTL_CONFIG_PWR_SAVE_SETTINGS

Configures the event indication filter by the Power Save state for each QMI service.

- **CTL message ID**

0x0029

- **Version introduced**

Major - 1, Minor - 4

4.2.10.1. Request - QMI_CTL_CONFIG_PWR_SAVE_SETTINGS_REQ

- Message type

Request

- Sender

Control point

- Mandatory TLVs

Name	Version introduced	Version last modified
Power Save Descriptor	1.4	1.4

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Power Save Descriptor
Length	5			2	
Value	→	uint32	pwrsave_state	4	Handle specified by the control point representing the Power Save state. The client-specified value is to match the value used in QMI_CTL_SET_PWR_SAVE_MODE_REQ. Reserved values: 0x00 through 0xFF.
		uint8	qmi_service	1	Valid QMI service identifier.

- Optional TLVs

Name	Version introduced	Version last modified
Permitted Indication Set	1.4	1.8

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x11			1	Permitted Indication Set
Length	Var			2	
Value	→	uint16	indication_set	Var	Sequence of indication message identifiers (2 bytes each).

4.2.10.2. Response - QMI_CTL_CONFIG_PWR_SAVE_SETTINGS_RESP

- Message type

Response

- **Sender**

Service

- **Mandatory TLVs**

The Result Code TLV (defined in Section 4.1.3.3) is always present in the response.

Name	Version introduced	Version last modified
Result Code	1.4	1.4

- **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_MISSING_ARG	One or more required TLVs were missing in the request
QMI_ERR_INVALID_ARG	Value exceeds the allowed range

4.2.10.3. Description of QMI_CTL_CONFIG_PWR_SAVE_SETTINGS REQ/RESP

The control point's Power Save indication filter internal state variables are modified according to the settings specified in the TLVs included in the request message.

The command configures the settings for a specific Power state handle for a specified QMI service. The device allows configuration for up to four power state handles. Any internal or vendor-specific QMI service can be specified for a given Power Save state handle. Predefined handle values are:

- 0x00 – NORMAL – Full-power mode; no indication filtering performed
- 0x01 – SUSPEND – Low-power mode
- 0x02 – POWERDOWN – Low-power mode
- 0x03 – Reserved for future use

Modem event indications to be forwarded to a control point during the specified Power Save descriptor are to be specified in the indication_set field. Any indication not present is filtered and dropped. The absence of the indication_set value dictates all indications are filtered for the power state descriptor.

NOTE

This command is used only for configuring the device with the allowable indication set for a specific power state. The filter is set only when the specified power state takes effect.

4.2.11. QMI_CTL_SET_PWR_SAVE_MODE

Sets the QMI framework Power Save mode.

- **CTL message ID**

0x002A

- **Version introduced**

Major - 1, Minor - 4

4.2.11.1. Request - QMI_CTL_SET_PWR_SAVE_MODE_REQ

- **Message type**

Request

- **Sender**

Control point

- **Mandatory TLVs**

Name	Version introduced	Version last modified
Power Save State	1.4	1.4

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Power Save State
Length	4			2	
Value	→	uint32	pwrsave_state	4	Handle specified by the control point representing the Power Save state. The client-specified value is to match the value used in QMI_CTL_CONFIG_PWR_SAVE_SAVINGS.

- **Optional TLVs**

None

4.2.11.2. Response - QMI_CTL_SET_PWR_SAVE_MODE_RESP

- **Message type**

Response

- **Sender**

Service

- **Mandatory TLVs**

The Result Code TLV (defined in Section 4.1.3.3) is always present in the response.

Name	Version introduced	Version last modified
Result Code	1.4	1.4

- **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_MISSING_ARG	One or more required TLVs were missing in the request
QMI_ERR_INVALID_ARG	Value exceeds the allowed range

4.2.11.3. Description of QMI_CTL_SET_PWR_SAVE_MODE REQ/RESP

This command sets the QMI framework Power Save mode and is used for direct power state notification applications.

The control point's Power Save indication filter that matches the specified Power Save state is installed for each configured service. Only those indications specified in the Indication Set TLV are forwarded to the control point. Any pending transaction response is discarded upon entering the Power Save state to avoid waking up the client processor. The QMI_CTL_CONFIGURE_RESPONSE_FILTERING_IN_PWR_SAVE_REQ message can override this response suppression behavior while in Power Save mode and enable the control point to receive response messages.

Control points are responsible to ensure all transactions are complete before setting the Power Save mode to avoid loss of pending responses.

If the received Power Save state is not previously configured through QMI_CTL_CONFIG_PWR_SAVE_SETTINGS, it is created with default settings and becomes one of the four device-configurable Power Save handles. When four Power Save handles have been previously configured, the request is rejected and the device continues to operate in the previously set Power Save state.

To return to Full Power mode, the control point must specify the NORMAL predefined handle value described in Section 4.2.10.3.

4.2.12. QMI_CTL_GET_PWR_SAVE_MODE

Gets the current QMI framework Power Save mode.

- **CTL message ID**

0x002B

- **Version introduced**

Major - 1, Minor - 4

4.2.12.1. Request - QMI_CTL_GET_PWR_SAVE_MODE_REQ

- **Message type**

Request

- **Sender**

Control point

- **Mandatory TLVs**

None

- **Optional TLVs**

None

4.2.12.2. Response - QMI_CTL_GET_PWR_SAVE_MODE_RESP

- **Message type**

Response

- **Sender**

Service

- **Mandatory TLVs**

The Result Code TLV (defined in Section 4.1.3.3) is always present in the response. This TLV is present if the result code is QMI_RESULT_SUCCESS.

- **Optional TLVs**

This TLV is present if the result code is QMI_RESULT_SUCCESS.

Name	Version introduced	Version last modified
Power Save State	1.4	1.4
Result Code	1.4	1.4

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Power Save State
Length	4			2	
Value	→	uint32	pwrsave_state	4	Handle specified by the control point representing the Power Save state.

- **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

4.2.12.3. Description of QMI_CTL_GET_PWR_SAVE_MODE REQ/RESP

This command queries the control point's current Power Save state. The predefined handle value NORMAL, described in Section 4.2.10.3, indicates that the Power Save mode has not been set. See Appendix A for details about indication filtering.

5 Wireless Data Service (QMI_WDS)

The QMI_WDS provides a command set to interface to a wireless mobile station, providing IP connectivity and related value-added services.

The QMI_WDS provides the following applications running on a host PC with commands related to IP data service over wireless radio networks:

- Data call setup and teardown
- Network registration and attach
- Packet transmission statistics
- Data bearer rate
- Data session profile management

It is expected that user-level applications, for example, connection managers and device drivers on the Terminal Equipment (TE), use QMI_WDS to access this functionality on the MSM™ device.

QMI_WDS is a QMI native service that conforms to the generalized behavior defined for QMI services, as defined in 80-VB816-1.

5.1. Theory of Operation

5.1.1. Generalized QMI Service Compliance

The QMI_WDS 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 80-VB816-1. Extensions to the generalized QMI service theory of operation are noted in subsequent sections of this chapter.

5.1.2. WDS Service Type

WDS is assigned QMI service type 0x01.

5.1.3. Message Definition Template

5.1.3.1. Response Message Result TLV

This Type-Length-Value (TLV) is present in all Response messages defined in this document. It is not present in the Indication messages.

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

5.1.4. QMI_WDS Fundamental Concepts

All data session related messages apply to RmNet only unless explicitly specified for DUN.

5.1.4.1. Data Session

A wireless MSM device supporting QMI_WDS provides packet data (Internet Protocol) service through a wireless network. This service provides APIs to start and end the wireless data session. Multiple control points might need to use the packet data session. If at least one control point has requested it, the wireless device attempts to establish the packet data session. If multiple control points request a wireless data session, the device maintains the session until all requesting control points release the data session.

5.1.4.2. Data Session Handle

A packet data handle is an opaque identifier that represents an active wireless data connection. When the control point starts a data session, the service assigns a pkt_data_handle to the control point. It is provided back to the service in the message issued by the control point to release its use of IP services.

5.1.4.3. Data Connection Status

The wireless data service can report various state information about the wireless data connection. The

fundamental status reported to all control points is the connectivity status, or `Packet_data_connection_state`. This is a primary sequencing signal for the TE to begin using, that is, start Ipv4 address configuration, or discontinue use of Ipv4 service. Other state information that is exposed by the WDS service includes packet statistics, channel rate, and radio technology serving the data session. The control point can obtain this information via a polling interface (request/response messages) or by configuring the device to asynchronously report changes in other state information via indication messages.

5.1.4.4. QMI_WDS Profile

A QMI_WDS profile is a collection of configurable data session-related settings stored on the MSM device in persistent storage. When a data session is established using QMI_WDS, a profile might be referenced as the basis of the data session-related settings negotiated with the serving network. When a configured profile is referenced in this case, the device attempts to negotiate the preferred settings defined in the profile. The network might assign different settings to the device, however. The device might support storage of one or more QMI_WDS profiles. Each profile is uniquely identified by a profile index. A control point might add, modify, or delete a profile, and might refer to the profile when starting a data session. As of WDS version 1.1, profile parameters are defined only for 3GPP devices. The meanings of these parameters are further explained in 80-VB816-1. To date, only primary Packet Data Protocol (PDP) profiles are supported.

5.1.5. Service State Variables

5.1.5.1. Shared State Variables

Name	Description	Possible values	Default value	Arbitration
<code>packet_data_connection_state</code>	<ul style="list-style-type: none"> Indicates whether a network connection has been established Value of authenticating indicates that authentication started but not connected Value of suspended indicates when the radio interface is in use by other services, for example, voice and data transfer are suspended temporarily 	<ul style="list-style-type: none"> Connected Not connected Authenticating Suspended 	Not connected when the device is initialized unless autoconnect is enabled and proper state conditions are met	<ul style="list-style-type: none"> Connectivity attempted when at least one control point requests data service or enables autoconnect Disconnected when all control points no longer require data service and autoconnect is disabled

5.1.5.2. State Variables Per Control Point

Name	Description	Possible values	Default value
report_channel_rate	Whether change in data channel Rx or Tx rate is reported to control point	• FALSE • TRUE	FALSE
pkt_stats_report_period	Period in seconds between transfer statistic reports	• 0 – Do not report • 1 to 255 (sec)	0
pkt_stats_report_mask	Which packet statistics to be reported (bitmask)	0x00 to 0x3F	0x00
report_data_bearer_tech	Whether change in data bearer technology is reported to control point	• FALSE • TRUE	FALSE
report_dormancy_status	Whether change in traffic-channel state is reported to control point	• FALSE • TRUE	FALSE
report_mip_status	Whether change in MIP status is reported to control point	• FALSE • TRUE	FALSE
report_current_data_bearer_tech	Whether change in current data bearer technology is reported to control point	• FALSE • TRUE	FALSE

Name	Description	Possible values	Default value
report_evdo_page_monitor_period_change	Whether EV-DO page monitor period change event is reported to control point	• FALSE • TRUE	FALSE
report_data_call_status	Whether change in data call status is reported to control point	• FALSE • TRUE	FALSE
report_preferred_data_system	Whether change in preferred data system is reported to control point	• FALSE • TRUE	FALSE
report_data_system_status	Whether change in data system status is reported to control point	• FALSE • TRUE	FALSE
report_data_bearer_tech_ex	Whether change in data bearer technology extended is reported to control point	• FALSE • TRUE	FALSE
report_embms_tmgi_list	Whether to report the eMBMS TMGI list	• FALSE • TRUE	FALSE
suppress_pkt_srvc_ind	Whether to suppress the packet service status indication	• FALSE • TRUE	FALSE
report_extended_ip_config_change	Whether change in extended IP configuration is reported to control point	• FALSE • TRUE	FALSE
report_lte_attach_pdn_list_change	Whether change in LTE attach PDN list is reported to control point	• FALSE • TRUE	FALSE
report_reverse_ip_transport_filter_setup	Whether to report a reverse IP transport filter setup	• FALSE • TRUE	FALSE
report_handoff_information	Whether to report handoff information	• FALSE • TRUE	FALSE

5.2. QMI_WDS Messages

Table 6: QMI_WDS messages

Command	ID	Description
QMI_WDS_START_NET	0x0020	Activates a packet data session (if not already started) on behalf of the requesting control point.
QMI_WDS_STOP_NET	0x0021	Deactivates a packet data session (unless in use by other control points) on behalf of the requesting control point
QMI_WDS_GET_PKT_STATUS QMI_WDS_GET_PKT_SRVC_STATUS	0x0022	Queries the current packet data connection status.

QMI_WDS_PKT_STATUS_IND	0x0023	Indicates a change in the current packet data connection status.
QMI_WDS_GET_SETTINGS	0x0024	Retrieves the packet data session settings currently in use.

5.2.1. QMI_WDS_START_NET (QMI_WDS_START_NETWORK_INTERFACE)

Activates a packet data session (if not already started) on behalf of the requesting control point.

- **WDS message ID**

0x0020

- **Version introduced**

Major – 1, Minor – 0

5.2.1.1. Request – QMI_WDS_START_NETWORK_INTERFACE_REQ

- **Message type**

Request

- **Sender**

Control point

- **Mandatory TLVs**

None

- **Optional TLVs**

Name	Version introduced	Version last modified
Primary DNS Address Preference	Unknown	1.1
Secondary DNS Address Preference	Unknown	1.1
Primary NetBIOS Name Server Address Preference	Unknown	1.1
Secondary NBNS Address Preference	Unknown	1.1
Context Access Point Node Name	Unknown	1.1
IP Address Preference	Unknown	1.1
Authentication Preference	Unknown	1.1
Username	Unknown	1.1
Password	Unknown	1.1
IP Family Preference	Unknown	1.7
Technology Preference	Unknown	1.1
3GPP Configured Profile Identifier	Unknown	1.1
3GPP2 Configured Profile Identifier	Unknown	1.6
Enable Autoconnect	Unknown	1.12
Extended Technology Preference	Unknown	1.25
Call Type Identifier	Unknown	1.8
Handoff Context	1.44	1.44
IP Stream ID	1.45	1.45
APN Type Enum	1.84	1.84

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Primary DNS Address Preference
Length	4			2	
Value	→	uint32	primary_DNS_IPv4_address_preference	4	Used as a preference during negotiation with the network; if not specified, the wireless device attempts to obtain the DNS address automatically from the network. The negotiated value is provided to the host via DHCP.
Type	0x11			1	Secondary DNS Address Preference
Length	4			2	
Value	→	uint32	secondary_DNS_IPv4_address_preference	4	Used as a preference during negotiation with the network; if not specified, the wireless device attempts to obtain the DNS address automatically from the network. The negotiated value is provided to the host via DHCP.
Type	0x12			1	Primary NetBIOS Name Server (NBNS) Address Preference
Length	4			2	
Value	→	uint32	primary_nbns_address_pref	4	Primary NBNS address. The specified IPv4 address is requested as the primary NBNS server during data session establishment. If it is not provided, the primary NBNS server address is obtained automatically from the network. The result of negotiation (the assigned address) is provided to the host via DHCP
Type	0x13			1	Secondary NBNS Address Preference
Length	4			2	
Value	→	uint32	secondary_nbns_address_pref	4	Secondary NetBIOS name server address. The specified IPv4 address is requested as the secondary NBNS server during data session establishment. If not provided, the secondary NBNS server address is obtained automatically from the network. The result of negotiation (the assigned address) is provided to the host via DHCP.
Type	0x14			1	Context Access Point Node (APN) Name
Length	Var			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	string	apn_name	Var	<p>String parameter that is a logical name used to select the GGSN and external packet data network. If the value is NULL or omitted, the subscription default value is requested.</p> <p>QMI_ERR_ARG_TOO_LONG is returned if the APN name is too long. This TLV is ignored if the 3GPP-configured profile TLV is present, that is, the APN name cannot be overridden.</p>
Type	0x15			1	IP Address Preference
Length	4			2	
Value	→	uint32	ipv4_address_pref	4	<p>The preferred IPv4 address to be assigned to the TE. The actual assigned address is negotiated with the network and might differ from this value. If not specified, the IPv4 Address is obtained automatically from the network. The assigned value is provided to the host via DHCP.</p>
Type	0x16			1	Authentication Preference
Length	1			2	
Value	→	mask8	authentication_preference	1	<p>Bitmap that indicates the authentication algorithm preference. Values:</p> <ul style="list-style-type: none"> Bit 0 – PAP preference: <ul style="list-style-type: none"> • 0 – PAP is never performed • 1 – PAP might be performed Bit 1 – CHAP preference: <ul style="list-style-type: none"> • 0 – CHAP is never performed • 1 – CHAP might be performed <p>All other bits are reserved and ignored even if they are set in the request. If more than one bit is set, the device decides which authentication procedure is performed while setting up the data session. For example, the device might have a policy to select the most secure authentication mechanism.</p>
Type	0x17			1	Username
Length	Var			2	
Value	→	string	username	Var	<p>Username to use during data network authentication.</p> <p>QMI_ERR_ARG_TOO_LONG is returned if the storage on the wireless device is insufficient in size to hold the value.</p>

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x18			1	Password
Length	Var			2	
Value	→	string	password	Var	<p>Password used during data network authentication.</p> <p>QMI_ERR_ARG_TOO_LONG is returned if the storage on the wireless device is insufficient in size to hold the value.</p>
Type	0x19			1	IP Family Preference
Length	1			2	
Value	→	enum8	ip_family_preference	1	<p>If this TLV is absent, the device attempts to bring up a call on default IP preference (currently IPv4, to maintain current behavioral backward compatibility). Values:</p> <ul style="list-style-type: none"> • 4 – IPv4 • 6 – IPv6 • 8 – Unspecified
Type	0x30			1	Technology Preference
Length	1			2	
Value	→	mask8	technology_preference	1	<p>Bitmap that indicates the technology preference. A single connection is attempted using the following specified technology preferences:</p> <ul style="list-style-type: none"> • Bit 0 – 3GPP • Bit 1 – 3GPP2 <p>All other bits are reserved and ignored even if they are set in the request. If a single value of the technology preference bitmask is set, the device attempts to use that technology. If two or more bits in the technology preference bitmask are set, the device determines which technology to use from the bits specified. If this TLV is absent, the device assumes that all supported technologies are acceptable.</p>
Type	0x31			1	3GPP Configured Profile Identifier
Length	1			2	
Value	→	uint8	profile_index	1	Index of the configured profile on which data call parameters are based (other TLVs present override the profile settings). If this TLV is not present, the data call parameters are based on device default settings for each parameter.
Type	0x32			1	3GPP2 Configured Profile Identifier
Length	1			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	uint8	profile_index_3gpp2	1	Index of the configured profile on which data call parameters are based (other TLVs present override the profile settings). If this TLV is not present, data call parameters are based on device default settings for each parameter.
Type	0x33			1	Enable Autoconnect
Length	1			2	
Value	→	boolean	enable_autoconnect	1	<ul style="list-style-type: none"> • 1 – TRUE – Device attempts to bring up a call automatically • 0 – FALSE – Default Note: When this TLV is used, the override parameters passed in other TLVs in this message are ignored by the device.
Type	0x34			1	Extended Technology Preference
Length	2			2	
Value	→	enum16	ext_technology_preference	2	Technology preference used while attempting a packet data connection. Values: <ul style="list-style-type: none"> • -32767 – CDMA • -32764 – UMTS • -30590 – eMBMS • -30584 – Modem Link Local Modem Link Local is an interface for transferring data between entities on the AP and modem.
Type	0x35			1	Call Type Identifier
Length	1			2	
Value	→	enum8	call_type	1	Type of call to be originated. Values: <ul style="list-style-type: none"> • WDS_CALL_TYPE_LAPTOP_CALL (0x00) – Laptop call • WDS_CALL_TYPE_EMBEDDED_CALL (0x01) – Embedded call If this TLV is not present, by default the call is considered to be a laptop call.
Type	0x36			1	Handoff Context Context information needed if the TE is handing off a call to the modem.
Length	21			2	
Value	→	uint32	ipv4_addr	4	IPv4 address of the PDN.
		uint8	ipv6_address	16	IPv6 address of the PDN.

Field	Field value	Field type	Parameter	Size (byte)	Description
		enum8	bearer_ip_type	1	Type of bearer IP. Values: • WDS_IP_SUPPORT_TYPE_IPV4 (0x00) – IPv4 • WDS_IP_SUPPORT_TYPE_IPV6 (0x01) – IPv6 • WDS_IP_SUPPORT_TYPE_IPV4V6 (0x02) – IPv4 and IPv6
Type	0x37			1	IP Stream ID
Length	1			2	
Value	→	uint8	ips_id	1	IP stream ID associated with the data call.
Type	0x38			1	APN Type Enum
Length	4			2	
Value	→	enum	apn_type	4	Values: • WDS_APN_TYPE_UNSPECIFIED (0) – APN type unspecified • WDS_APN_TYPE_INTERNET (1) – APN type for internet traffic • WDS_APN_TYPE_IMS (2) – APN type for IMS

5.2.1.2. Response – QMI_WDS_START_NETWORK_INTERFACE_RESP

- Message type

Response

- Sender

Service

- Mandatory TLVs

Name	Version introduced	Version last modified
Packet Data Handle	Unknown	1.0

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

Name	Version introduced	Version last modified
Packet Data Handle	Unknown	1.0

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

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	uint32	pkt_data_handle	4	The handle identifying the call instance providing packet service. The packet data handle must be retained by the control point and specified in the STOP_NETWORK_INTERFACE message issued when the control point is finished with the packet data session.

Optional TLVs

Name	Version introduced	Version last modified
Call End Reason	Unknown	1.3
Verbose Call End Reason	1.8	1.113
Peripheral End Point ID	1.54	1.77
Mux ID	1.54	1.54

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Call End Reason
Length	2			2	
Value	→	enum16	call_end_reason	2	Reason the call ended; see Appendix A for the definition of these values.
Type	0x11			1	Verbose Call End Reason
Length	4			2	
Value	→	enum16	call_end_reason_type	2	<p>Call end reason type. Values:</p> <ul style="list-style-type: none"> • WDS_VCER_TYPE_UNSPECIFIED (0x00) – Unspecified • WDS_VCER_TYPE_MOBILE_IP (0x01) – Mobile IP • WDS_VCER_TYPE_INTERNAL (0x02) – Internal • WDS_VCER_TYPE_CALL_MANAGER_DEFINED (0x03) – Call manager defined • WDS_VCER_TYPE_3GPP_SPEC_DEFINED (0x06) – 3GPP specification defined • WDS_VCER_TYPE_PPP (0x07) – PPP • WDS_VCER_TYPE_EHRPD (0x08) – EHRPD • WDS_VCER_TYPE_IPV6 (0x09) – IPv6 • WDS_VCER_TYPE_HANDOFF (0x0c) – Handoff
		uint16	call_end_reason	2	Reason the call ended (verbose); see Appendix B for the definition of these values.
Type	0x12			1	Peripheral End Point ID Peripheral end point of the RmNet instance where a data call is already present.
Length	8			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	enum	ep_type	4	Peripheral endpoint type. Values: • DATA_EP_TYPE_RESERVED (0x00) – Reserved • DATA_EP_TYPE_HSIC (0x01) – High-speed inter-chip interface • DATA_EP_TYPE_HSUSB (0x02) – High-speed universal serial bus • DATA_EP_TYPE_PCIE (0x03) – Peripheral component interconnect express • DATA_EP_TYPE_EMBEDDED (0x04) – Embedded • DATA_EP_TYPE_BAM_DMUX (0x05) – Bus access manager data multiplexer All other values are reserved and are ignored.
		uint32	iface_id	4	Peripheral interface number.
Type	0x13			1	Mux ID
Length	1			2	
Value	→	uint8	mux_id	1	Mux ID of the RmNet instance where a data call is already present.

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_ARG_TOO_LONG	Argument passed in a TLV is larger than the available storage in the device
QMI_ERR_INVALID_PROFILE	Specified configured profile index does not exist
QMI_ERR_NO_EFFECT	Control point has already started the network interface
QMI_ERR_CALL_FAILED	Data call failed
QMI_ERR_INVALID_TECH_PREF	Invalid technology preference
QMI_ERR_INVALID_PDP_TYPE	Invalid PDP type
QMI_ERR_ACCESS_DENIED	Autoconnect feature is unavailable at this time
QMI_ERR_INVALID_IP_FAMILY_PREF	Invalid IP family preference

5.2.1.3. Description of QMI_WDS_START_NETWORK_INTERFACE REQ/RESP

This command is used by a control point to request packet data service. The wireless device starts a packet data session if one is not already in progress. By issuing this command, the control point registers its interest in (binds itself to) the WWAN data connection. The data session remains connected while at least one control point is bound to the WWAN data connection.

The call is established either using the default call parameters (if a configured profile TLV is not present in the request) or using parameters from a stored profile (if a configured profile TLV is present in the request). The default call parameters are defined outside the scope of this document.

The optional Autoconnect TLV causes the session to automatically reconnect if the packet data session is disconnected and persists over device power cycles. This support has been deprecated. Clients must useQMI_WDS_SET_AUTOCONNECT_SETTING to modify autoconnect settings. Optional TLVs 0x10 through 0x18, included in the START_NETWORK_INTERFACE request command,supersede (override) the call parameters (default or configured profile) selected.

The technology preference value included in the optional Extended Technology Preference TLV 0x34 in the START_NETWORK_INTERFACE request command supersedes the value in the technology preference optional TLV 0x30. Qualcomm recommends that all clients use the newer Extended Technology Preference TLV, as the older TLV is planned to be deprecated over time.

The optional Handoff Context TLV is included if the TE wants to hand off an existing call to the modem and must convey some context information. For a dual IP PDN, the TLV must include both the Ipv4 and Ipv6 address, with bearer_ip_type set to WDS_IP_SUPPORT_TYPE_IPV4V6. Also, the same TLV is to be present on both the QMI_WDS_START_NETWORK_INTERFACE_REQ messages from the Ipv4 client and Ipv6 client.

The optional APN Type Enum TLV specifies the type of APN on which the call is being attempted. In the absence of this TLV, the APN type default is WDS_APN_TYPE_UNSPECIFIED.

The QMI_WDS_START_NETWORK_INTERFACE_RESP command is returned only when the packet data session is established, or sooner if an error occurs. After the response is sent, the tethered device can perform IP address configuration.

A successful QMI_WDS_START_NETWORK_INTERFACE_REQ modifies the packet_data_connection_state shared state variable.

If the Result TLV indicates failure and the qmi_error field is set to QMI_ERR_CALL_FAILED, the Call End Reason and Verbose Call End Reason optional TLVs are included with the response conveying the additional call failure reason.

If the Verbose Call End Reason TLV indicates that a data call with the same policy is already present on another RmNet instance (internal CALL_ALREADY_PRESENT error), the optional Peripheral End Point ID and Mux ID TLVs are included to identify the RmNet instance where the data call is present.

The AT command equivalents of this command are ATD and AT+CGACT defined in 3GPP TS 27.007, 3GPP2 C.S0017-003-A, and TIA/EIA/IS-131.

5.2.2. QMI_WDS_STOP_NETWORK_INTERFACE

Deactivates a packet data session (unless in use by other control points) on behalf of the requesting control point.

- **WDS message ID**

0x0021

- **Version introduced**

Major – 1, Minor – 0

5.2.2.1. Request – QMI_WDS_STOP_NETWORK_INTERFACE_REQ

- **Message type**

Request

- **Sender**

Control point

- **Mandatory TLVs**

Name	Version introduced	Version last modified
Packet Data Handle	Unknown	1.0

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Packet Data Handle
Length	4			2	
Value	→	uint32	pkt_data_handle	4	Handle identifying the call instance from which to unbind the control point. The value must be the handle previously returned by QMI_WDS_START_NETWORK_INTERFACE_REQ.

- Optional TLVs

Name	Version introduced	Version last modified
Disable Autoconnect	Unknown	1.12
Local Abort	1.117	1.117

Field	Field value	Field type	Parameter	Size (byte)	Description

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Disable Autoconnect
Length	1			2	
Value	→	boolean	disable_autoconnect	1	<ul style="list-style-type: none"> • 1 – TRUE – Device disables autoconnect; the calls must be made manually until the setting is enabled again • 0 – FALSE – Default. Note: When this TLV is present, the client must use a global handle (0xFFFFFFFF) in the Packet Data Handle TLV.
Type	0x11			1	Local Abort
Length	1			2	
Value	→	boolean	local_abort	1	<ul style="list-style-type: none"> • 1 – TRUE – OTA teardown request is not triggered by the modem when the data call is torn down and only results in local clean up of the PDN on the UE • 0 – FALSE – Triggers an OTA teardown and local clean up of the PDN (default)

5.2.2.2. Response – QMI_WDS_STOP_NETWORK_INTERFACE_RESP

- Message type

Response

- Sender

Service

- Mandatory TLVs

The Result Code TLV (defined in Section 5.1.3.1) 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_MISSING_ARG	One or more required TLVs were missing in the request
QMI_ERR_INVALID_HANDLE	Packet_data_handle provided in the request is not valid, that is, it is not assigned to the control point

5.2.2.3. Description of QMI_WDS_STOP_NETWORK_INTERFACE REQ/RESP

This command is used by a control point to end packet data service. By issuing this command, the control point releases its interest in (unbinds itself from) the WWAN data connection. The wireless device ends the current packet data session when all control points release their binding using this message.

The control point considers that the packet_data_connection_state is unchanged until notified of the state change via the QMI_WDS_PKT_SRVC_STATUS_IND indication.

Requests using the global packet data handle (0xFFFFFFFF) and a nonzero value for the optional Disable Autoconnect TLV disables the autoconnect of the device. This support is deprecated. Clients must use QMI_WDS_SET_AUTOCONNECT_SETTING to modify autoconnect settings.

The AT command equivalents of this command are ATD and AT+CGACT defined in 3GPP TS 27.007, 3GPP2 C.S0017-003-A, and TIA/EIA/IS-131.

5.2.3. QMI_WDS_GET_PKT_SRVC_STATUS

Queries the current packet data connection status.

- WDS message ID

0x0022

- Version introduced

Major - 1, Minor - 0

5.2.3.1. Request - QMI_WDS_GET_PKT_SRVC_STATUS_REQ

- Message type

Request

- **Sender**

Control point

- **Mandatory TLVs**

None

- **Optional TLVs**

None

5.2.3.2. Response - QMI_WDS_GET_PKT_SRVC_STATUS_RESP

- **Message type**

Response

- **Sender**

Service

- **Mandatory TLVs**

The Result Code TLV (defined in Section 5.1.3.1) is always present in the response. This TLV is present if the result code is QMI_RESULT_SUCCESS.

Name	Version introduced	Version last modified
Connection status.	Unknown	1.0

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Connection status.
Length	1			2	
Value	→	enum8	connection_status	1	Current link status. Values: <ul style="list-style-type: none">• WDS_CONNECTION_STATUS_DISCONNECTED (0x01) – Disconnected• WDS_CONNECTION_STATUS_CONNECTED (0x02) – Connected• WDS_CONNECTION_STATUS_SUSPENDED (0x03) – Suspended• WDS_CONNECTION_STATUS_AUTHENTICATING (0x04) – Autheniciating

- **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_INCOMPATIBLE_STATE	Request from a client whose subscription does not match the subscription of the current data session (incompatible subscription)

5.2.3.3. Description of QMI_WDS_GET_PKT_SRVC_STATUS REQ/RESP

This command queries the state of the packet data connection provided by the wireless device. It returns the current value of Packet_data_connection_state value, as described in Section 5.1.5.1.

A data connection being established does not imply that the IP address has been assigned to the host. This is only an indication that address configuration can commence.

The QMI_WDS_PKT_DATA_AUTHENTICATING connection status is not always supported. In such cases, the device directly transitions to the connected state without entering the authenticating state.

The AT command equivalents of this command are ATD and AT+CGACT, defined in 3GPP TS 27.007, 3GPP2 C.S0017-003-A, and TIA/EIA/IS-131.

5.2.4. QMI_WDS_PKT_STATUS_IND (QMI_WDS_GET_PKT_SRVC_STATUS_IND)

Indicates a change in the current packet data connection status.

- **WDS message ID**

0x0022

- **Version introduced**

Major – 1, Minor – 0

5.2.4.1. Indication – QMI_WDS_PKT_SRVC_STATUS_IND

- Message type

Indication

- Sender

Service

- Scope

Unicast

- Mandatory TLVs

Name	Version introduced	Version last modified
Packet Service Status	Unknown	1.0

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Packet Service Status
Length	2			2	
Value	→	enum8	connection_status	1	Current link status. Values: • WDS_CONNECTION_STATUS_DISCONNECTED (0x01) – Disconnected • WDS_CONNECTION_STATUS_CONNECTED (0x02) – Connected • WDS_CONNECTION_STATUS_SUSPENDED (0x03) – Suspended • WDS_CONNECTION_STATUS_AUTHENTICATING (0x04) – Autheniciating
		boolean	reconfiguration_required	1	Indicates whether the network interface on the host must be reconfigured. Values: • 0 – Not necessary to reconfigure • 1 – Reconfiguration required

Optional TLVs

Name	Version introduced	Version last modified
Call End Reason	Unknown	1.3
Verbose Call End Reason	1.8	1.113
IP Family	Unknown	1.9
Technology Name	Unknown	1.25
Bearer ID	1.50	1.50
XLAT Capability	1.87	1.87

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Call End Reason
Length	2			2	
Value	→	enum16	call_end_reason	2	See Appendix A for the definition of these values.
Type	0x11			1	Verbose Call End Reason
Length	4			2	
Value	→	enum16	call_end_reason_type	2	Call end reason type. Values: • WDS_VCER_TYPE_UNSPECIFIED (0x00) – Unspecified • WDS_VCER_TYPE_MOBILE_IP (0x01) – Mobile IP • WDS_VCER_TYPE_INTERNAL (0x02) – Internal • WDS_VCER_TYPE_CALL_MANAGER_DEFINED (0x03) – Call manager defined • WDS_VCER_TYPE_3GPP_SPEC_DEFINED (0x06) – 3GPP specification defined • WDS_VCER_TYPE_PPP (0x07) – PPP • WDS_VCER_TYPE_EHRPD (0x08) – EHRPD • WDS_VCER_TYPE_IPV6 (0x09) – IPv6 • WDS_VCER_TYPE_HANDOFF (0x0c) – Handoff
		uint16	call_end_reason	2	Reason the call ended (verbose); see Appendix B for the definition of these values.
Type	0x12			1	IP Family
Length	1			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	enum8	ip_family	1	IP family of the packet data connection. Values: • WDS_IP_FAMILY_IPV4 (0x04) – IPv4 • WDS_IP_FAMILY_IPV6 (0x06) – IPv6
Type	0x13			1	Technology Name
Length	2			2	
Value	→	enum16	tech_name	2	Technology name of the packet data connection. Values: • WDS TECHNOLOGY_NAME_CDMA (-32767) – 0x8001 – CDMA • WDS TECHNOLOGY_NAME_UMTS (-32764) – 0x8004 – UMTS • WDS TECHNOLOGY_NAME_WLAN_LOCAL_BRKOUT (-32736) – 0x8020 – WLAN_LOCAL_BRKOUT • WDS TECHNOLOGY_NAME_IWLAN_S2B (-32735) – 0x8021 – IWLAN_S2B • WDS TECHNOLOGY_NAME_EPC (-30592) – 0x8880 – EPC • WDS TECHNOLOGY_NAME_EMBMS (-30590) – 0x8882 – EMBMS • WDS TECHNOLOGY_NAME_MODEM_LINK_LOCAL (-30584) – 0x8888 – Modem link local EPC is a logical interface to support LTE/eHRPD handoff; it is returned if the device supports IP session continuity. Modem Link Local is an interface for transferring data between entities on the AP and modem.
Type	0x14			1	Bearer ID
Length	1			2	
Value	→	uint8	bearer_id	1	Bearer ID (3GPP) or RLP ID (3GPP2) of the packet data connection.
Type	0x15			1	XLAT Capability
Length	1			2	
Value	→	boolean	xlat_capable	1	Indicates XLAT capability of the data session. Values: • 0 – XLAT not capable • 1 – XLAT capable

5.2.4.2. Description of QMI_WDS_GET_PKT_SRVC_STATUS_IND

This indication communicates changes in the Packet_data_connection_state value.

When the IP address assigned to the host is no longer valid, the reconfiguration required value is set to

one.

If the indication is sent because of a disconnected state change, the Call End Reason and Verbose Call End Reason optional TLVs are included and contain the reason the call was terminated. These include network and user-generated reasons. The Call End Reason TLV is kept for backwards compatibility, and all new QMI clients must use the newer Verbose Call End Reason TLV. Any new Call End Reason is added only to the new TLV.

The QMI_WDS_PKT_DATA_AUTHENTICATING connection status is not always supported. In such cases, the device directly transitions to the connected state without entering the authenticating state.

The optional IP Family TLV indicates the IP type of the packet data connection.

The optional XLAT Capability TLV indicates whether the packet data session is XLAT capable.

For QMI_WDS revision 1.35 and newer, this indication has been changed from broadcast to unicast. By default, the indication is sent to all control points on the QMI link that have a packet data connection status change. If a control point is bound to an IP family type using QMI_WDS_SET_CLIENT_IP_FAMILY_PREF, it does not receive the packet data connection status indication for a different IP type. Control points can also suppress the indication by using the QMI_WDS_INDICATION_REGISTER command.

5.2.5. QMI_WDS_GET_SETTINGS (QMI_WDS_GET_RUNTIME_SETTINGS)

Retrieves the packet data session settings currently in use.

- **WDS message ID**

0x002D

- **Version introduced**

Major – 1, Minor – 2

5.2.5.1. Request – QMI_WDS_GET_RUNTIME_SETTINGS_REQ

- **Message type**

Request

- **Sender**

Control point

- Mandatory TLVs

None

- Optional TLVs

Name	Version introduced	Version last modified
Requested Settings	Unknown	1.37

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

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	mask32	requested_settings	4	<p>Set bits to 1, corresponding to requested information. All other bits must be set to 0.</p> <p>If the values are not available, the corresponding TLVs are not returned in the response.</p> <p>Absence of this mask TLV results in the device returning all of the available information corresponding to bits 0 through 12. In cases where the information from bit 13 or greater is required, this TLV with all the necessary bits set must be present in the request.</p> <p>Values:</p> <ul style="list-style-type: none"> • Bit 0 – Profile identifier • Bit 1 – Profile name • Bit 2 – PDP type • Bit 3 – APN name • Bit 4 – DNS address • Bit 5 – UMTS/GPRS granted QoS • Bit 6 – Username • Bit 7 – Authentication Protocol • Bit 8 – IP address • Bit 9 – Gateway information (address and subnet mask) • Bit 10 – PCSCF address using a PCO flag • Bit 11 – PCSCF server address list • Bit 12 – PCSCF domain name list • Bit 13 – MTU • Bit 14 – Domain name list • Bit 15 – IP family • Bit 16 – IM_CM flag • Bit 17 – Technology name • Bit 18 – Operator reserved PCO

5.2.5.2. Response – QMI_WDS_GET_RUNTIME_SETTINGS_RESP

- **Message type**

Response

- **Sender**

Service

- **Mandatory TLVs**

The Result Code TLV (defined in Section 5.1.3.3) is always present in the response.

- **Optional TLVs**

Name	Version introduced	Version last modified
Profile Name **	Unknown	1.2
PDP Type **	Unknown	1.2
Context APN Name **	Unknown	1.2
Primary DNS Address Preference * ***	Unknown	1.2
Secondary DNS Address Preference * ***	Unknown	1.2
UMTS Requested QoS **	Unknown	1.2
GPRS Requested QoS **	Unknown	1.2
Username **	Unknown	1.2
Authentication Preference **	Unknown	1.2
IPv4 Address Preference * ***	Unknown	1.2
Profile Identifier **	Unknown	1.2
IPv4 Gateway Address * ***	Unknown	1.2
IPv4 Subnet Mask * ***	Unknown	1.2
PCSCF Address Using PCO Flag **	Unknown	1.3
PCSCF IPv4 Server Address List **	Unknown	1.3
PCSCF FQDN List **	Unknown	1.3
IPv6 Address * ***	Unknown	1.9
IPv6 Gateway Address * ***	Unknown	1.9
Primary IPv6 DNS Address * ***	Unknown	1.7
Secondary IPv6 DNS Address * ***	Unknown	1.7
MTU * ***	Unknown	1.8
Domain Name List * ***	Unknown	1.8
IP Family * ***	Unknown	1.8
IM CN Flag *	Unknown	1.8
Technology Name * ***	Unknown	1.25
PCSCF IPv6 Address List * ***	Unknown	1.11
Operator Reserved Protocol Information * ***	1.37	1.37

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Profile Name **
Length	Var			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	string	profile_name	Var	<p>One or more bytes describing the profile. The description can be a user-defined name for the profile.</p> <p>QMI_ERR_ARG_TOO_LONG is returned when the profile_name is too long</p>
Type	0x11			1	PDP Type **
Length	1			2	
Value	→	enum8	pdp_type	1	<p>Specifies the type of data payload exchanged over the airlink when the packet data session is established with this profile. Values:</p> <ul style="list-style-type: none"> • WDS_PDP_TYPE_PDP_IPV4 (0x00) – PDP-IP (IPv4) • WDS_PDP_TYPE_PDP_PPP (0x01) – PDP-PPP • WDS_PDP_TYPE_PDP_IPV6 (0x02) – PDP-IPv6 • WDS_PDP_TYPE_PDP_IPV4V6 (0x03) – PDP-IPv4 and IPv6
Type	0x14			1	Context APN Name **
Length	Var			2	
Value	→	string	apn_name	Var	<p>String parameter that is a logical name used to select the GGSN and external packet data network. If the value is NULL or omitted, the subscription default value is requested.</p> <p>QMI_ERR_ARG_TOO_LONG is returned if the APN name is too long.</p>
Type	0x15			1	Primary DNS Address Preference * **
Length	4			2	
Value	→	uint32	primary_DNS_IPv4_address_preference	4	Used as a preference during negotiation with the network; if not specified, the wireless device attempts to obtain the DNS address automatically from the network. The negotiated value is provided to the host via DHCP.
Type	0x16			1	Secondary DNS Address Preference * **
Length	4			2	
Value	→	uint32	secondary_DNS_IPv4_address_preference	4	Used as a preference during negotiation with the network; if not specified, the wireless device attempts to obtain the DNS address automatically from the network. The negotiated value is provided to the host via DHCP.
Type	0x17			1	UMTS Requested QoS **
Length	33			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	enum8	traffic_class	1	<p>Traffic class. Values:</p> <ul style="list-style-type: none"> • WDS_TRAFFIC_CLASS_SUBSCRIBED (0x00) – Subscribed • WDS_TRAFFIC_CLASS_CONVERSATIONAL (0x01) – Conversational • WDS_TRAFFIC_CLASS_STREAMING (0x02) – Streaming • WDS_TRAFFIC_CLASS_INTERACTIVE (0x03) – Interactive • WDS_TRAFFIC_CLASS_BACKGROUND (0x04) – Background
		uint32	max_uplink_bitrate	4	Maximum uplink bitrate in bits per second.
		uint32	max_downlink_bitrate	4	Maximum downlink bitrate in bits per second.
		uint32	guaranteed_uplink_bitrate	4	Guaranteed uplink bitrate in bits per second.
		uint32	guaranteed_downlink_bitrate	4	Guaranteed downlink bitrate in bits per second.
		enum8	qos_delivery_order	1	<p>Values:</p> <ul style="list-style-type: none"> • WDS_QOS_DELIVERY_ORDER_SUBSCRIBE (0x00) – Subscribe • WDS_QOS_DELIVERY_ORDER_ON (0x01) – Delivery order on • WDS_QOS_DELIVERY_ORDER_OFF (0x02) – Delivery order off
		uint32	max_sdu_size	4	Maximum SDU size.
		enum8	sdu_error_ratio	1	<p>Target value for the fraction of SDUs lost or detected as erroneous. Values:</p> <ul style="list-style-type: none"> • 0 – Subscribe • 1 – 1×10^2 • 2 – 7×10^3 • 3 – 1×10^3 • 4 – 1×10^4 • 5 – 1×10^5 • 6 – 1×10^6 • 7 – 1×10^1

Field	Field value	Field type	Parameter	Size (byte)	Description
		enum8	residual_bit_error_ratio	1	<p>Target value for the undetected bit error ratio in the delivered SDUs. Values:</p> <ul style="list-style-type: none"> • 0 – Subscribe • 1 – 5×10^2 • 2 – 1×10^2 • 3 – 5×10^3 • 4 – 4×10^3 • 5 – 1×10^3 • 6 – 1×10^4 • 7 – 1×10^5 • 8 – 1×10^6 • 9 – 6×10^8
		enum8	delivery_erroneous_SDUs	1	<p>Delivery of erroneous SDUs. Indicates whether SDUs detected as erroneous are delivered or not. Values:</p> <ul style="list-style-type: none"> • WDS_DELIVERY_ERRONEOUS_SDUS_SUBSCRIBE (0x00) – Subscribe • WDS_DELIVERY_ERRONEOUS_SDUS_NO_DETECTION (0x01) – No detection • WDS_DELIVERY_ERRONEOUS_SDUS_YES (0x02) – Erroneous SDU is delivered • WDS_DELIVERY_ERRONEOUS_SDUS_NO (0x03) – Erroneous SDU is not delivered
		uint32	transfer_delay	4	Transfer delay. Indicates the targeted time between a request to transfer an SDU at one SAP to its delivery at the other SAP, in milliseconds; if the parameter is set to 0, the subscribed value is requested.
		uint32	traffic_handling_priority	4	Traffic handling priority. Specifies the relative importance for handling of SDUs that belong to the UMTS bearer, compared to the SDUs of other bearers. If the parameter is set to 0, the subscribed value is requested.
Type	0x19			1	GPRS Requested QoS **
Length	20			2	
Value	→	uint32	precedence_class	4	Precedence class
		uint32	delay_class	4	Delay class
		uint32	reliability_class	4	Reliability class
		uint32	peak_throughput_class	4	Peak throughput class
		uint32	mean_throughput_class	4	Mean throughput class
Type	0x1B			1	Username **
Length	Var			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	string	username	Var	<p>Username used during data network authentication.</p> <p>QMI_ERR_ARG_TOO_LONG is returned if the storage on the wireless device is insufficient in size to hold the value.</p>
Type	0x1D			1	Authentication Preference **
Length	1			2	
Value	→	mask8	authentication_preference	1	<p>Bitmap that indicates the authentication algorithm preference. Values:</p> <p>Bit 0 – PAP preference:</p> <ul style="list-style-type: none"> • 0 – PAP is never performed • 1 – PAP can be performed <p>Bit 1 – CHAP preference:</p> <ul style="list-style-type: none"> • 0 – CHAP is never performed • 1 – CHAP can be performed <p>All other bits are reserved and are ignored.</p> <p>If more than one bit is set, the device decides which authentication procedure is performed while setting up the data session. For example, the device can have a policy to select the most secure authentication mechanism.</p>
Type	0x1E			1	IPv4 Address Preference * **
Length	4			2	
Value	→	uint32	ipv4_address_preference	4	Preferred IPv4 address assigned to the TE. The actual assigned address is negotiated with the network and can differ from this value. If not specified, the IPv4 address is obtained automatically from the network. The assigned value is provided to the host via DHCP.
Type	0x1F			1	Profile Identifier **
Length	2			2	
Value	→	enum8	profile_type	1	<p>Values:</p> <ul style="list-style-type: none"> • WDS_PROFILE_TYPE_3GPP (0x00) – 3GPP • WDS_PROFILE_TYPE_3GPP2 (0x01) – 3GPP2 • WDS_PROFILE_TYPE_EPC (0x02) – EPC

Field	Field value	Field type	Parameter	Size (byte)	Description
		uint8	profile_index	1	Index of the profile whose settings are loaded before session parameter negotiation for the current call; if this TLV is not present, the data call parameters are based on the device default settings for each parameter.
Type	0x20			1	IPv4 Gateway Address * **
Length	4			2	
Value	→	uint32	ipv4_gateway_addr	4	Gateway address.
Type	0x21			1	IPv4 Subnet Mask * **
Length	4			2	
Value	→	uint32	ipv4_subnet_mask	4	Subnet mask.
Type	0x22			1	PCSCF Address Using PCO Flag **
Length	1			2	
Value	→	boolean	pcscf_addr_using_pco	1	Values: • 1 – TRUE – PCSCF address is requested using PCO • 0 – FALSE – PCSCF address is not requested
Type	0x23			1	PCSCF IPv4 Server Address List ** PCSCF IPv4 server address.
Length	Var			2	
Value	→	uint8	pcscf_ipv4_addr_list_len	1	Number of sets of the following elements: • pcscf_ipv4_address
		uint32	pcscf_ipv4_address	4	PCSCF IPv4 server address.
Type	0x24			1	PCSCF FQDN List **
Length	Var			2	
Value	→	uint8	fqdn_list_len	1	Number of sets of the following elements: • fqdn_len • fqdn
		uint16	fqdn_len	2	Number of sets of the following elements: • fqdn
		string	fqdn	Var	FQDN string.
Type	0x25			1	IPv6 Address * **
Length	17			2	
Value	→	uint8	ipv6_addr	16	IPv6 address (in network byte order). The address is a 16-element array of 8-bit numbers, each of which is in big-endian format.
		uint8	ipv6_prefix_length	1	IPv6 prefix length in number of bits. Range: 0 to 128.
Type	0x26			1	IPv6 Gateway Address * **
Length	17			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	uint8	ipv6_addr	16	IPv6 address (in network byte order). The address is a 16-element array of 8-bit numbers, each of which is in big-endian format.
		uint8	ipv6_prefix_length	1	IPv6 prefix length in number of bits. Range: 0 to 128.
Type	0x27			1	Primary IPv6 DNS Address * **
Length	16			2	
Value	→	uint8	primary_dns_IPv6_address	16	Primary IPv6 DNS address in network byte order; an 8-element array of 16-bit numbers, each of which is in big-endian format.
Type	0x28			1	Secondary IPv6 DNS Address * **
Length	16			2	
Value	→	uint8	secondary_dns_IPv6_address	16	Secondary IPv6 DNS address in network byte order; an 8-element array of 16-bit numbers, each of which is in big-endian format.
Type	0x29			1	MTU * **
Length	4			2	
Value	→	uint32	mtu	4	MTU.
Type	0x2A			1	Domain Name List * **
Length	Var			2	
Value	→	uint8	domain_name_list_len	1	Number of sets of the following elements: • domain_name_len • domain_name
		uint16	domain_name_len	2	Number of sets of the following elements: • domain_name
		string	domain_name	Var	Domain name.
Type	0x2B			1	IP Family * **
Length	1			2	
Value	→	enum8	ip_family	1	Values: • WDS_IP_FAMILY_IPV4 (0x04) – IPv4 • WDS_IP_FAMILY_IPV6 (0x06) – IPv6
Type	0x2C			1	IM CN Flag *
Length	1			2	
Value	→	boolean	im_cn_flag	1	Values: • 0 – FALSE • 1 – TRUE
Type	0x2D			1	Technology Name * **
Length	2			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	enum16	technology_name	2	<p>Technology on which current packet data session is in progress. Values:</p> <ul style="list-style-type: none"> • WDS TECHNOLOGY_NAME_CDMA (-32767) – 0x8001 – CDMA • WDS TECHNOLOGY_NAME_UMTS (-32764) – 0x8004 – UMTS • WDS TECHNOLOGY_NAME_WLAN_LOCAL_BRKOUT (-32736) – 0x8020 – WLAN_LOCAL_BRKOUT • WDS TECHNOLOGY_NAME_IWLAN_S2B (-32735) – 0x8021 – IWLAN_S2B • WDS TECHNOLOGY_NAME_EPC (-30592) – 0x8880 – EPC • WDS TECHNOLOGY_NAME_EMBMS (-30590) – 0x8882 – EMBMS • WDS TECHNOLOGY_NAME_MODEM_LINK_LOCAL (-30584) – 0x8888 – Modem link local <p>EPC is a logical interface to support LTE and eHRPD handoff; it is returned if the device supports IP session continuity. Modem Link Local is an interface for transferring data between entities on the AP and modem.</p>
Type	0x2E			1	PCSCF IPv6 Address List * ** PCSCF IPv6 server address (in network byte order); An 8-element array of 16-bit numbers, each of which is in big endian format.
Length	Var			2	
Value	→	uint8	pcscf_ipv6_addr_list_len	1	<p>Number of sets of the following elements:</p> <ul style="list-style-type: none"> • pcscf_ipv6_addr
		uint8	pcscf_ipv6_addr	16	PCSCF IPv6 server address (in network byte order); this is an 8-element array of 16-bit numbers, each of which is in big-endian format
Type	0x2F			1	<p>Operator Reserved Protocol Information ***</p> <p>Operator reserved PCO information that the device retrieved from the network. If there is no information available, a value of 0 is returned.</p>
Length	Var			2	
Value	→	uint16	mcc	2	A 16-bit integer representation of MCC. Range: 0 to 999.

Field	Field value	Field type	Parameter	Size (byte)	Description
		uint16	mnc	2	A 16-bit integer representation of MNC. Range: 0 to 999.
		boolean	mnc_includes_pcs_digit	1	Interprets the length of the corresponding MNC reported in the TLV. Values: <ul style="list-style-type: none">• TRUE – MNC is a three-digit value; for example, a reported value of 90 corresponds to an MNC value of 090• FALSE – MNC is a two-digit value; for example, a reported value of 90 corresponds to an MNC value of 90
		uint8	app_specific_info_len	1	Number of sets of the following elements: <ul style="list-style-type: none">• app_specific_info
		uint8	app_specific_info	Var	Points to the application-specific information from the network. The format for this field complies with 3GPP TS 24.008 . The field is populated in this format for both 3GPP and 3GPP2.
		uint16	container_id	2	Container ID of this PCO.

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 client or the message was corrupted during transmission
QMI_ERR_NO_MEMORY	Device could not allocate memory to formulate a response
QMI_ERR_OUT_OF_CALL	Request was issued when the packet data session was disconnected
QMI_ERR_INCOMPATIBLE_STATE	Request from a client whose subscription does not match the subscription of the current data session (incompatible subscription)

5.2.5.3. Description of QMI_WDS_GET_RUNTIME_SETTINGS REQ/RESP

This command retrieves the settings for the current data session. Note that these settings might not be identical to the referenced profile number, since the settings are negotiated with the network and the assigned values from the network can be different from the profile values. Also, some of the profile values can be overridden in the QMI_WDS_START_NETWORK_INTERFACE request, hence the preferred values are a combination of the profile values and those overrides.

The runtime settings are those in use for an active data session. If no data session has been started, there are no runtime settings. Password TLV is not returned.

6 Device Management Service (QMI_DMS)

The QMI_DMS provides applications running on a tethered device, such as Terminal Equipment (TE), with the following commands related to device management:

- Device identification (manufacturer, model, firmware revision, phone number, serial number)
- Device capabilities (data service type, SIM, data rate)
- Device power state (battery level, power source)

It is expected that user-level applications, for example, connection managers and/or device drivers on the TE, use QMI_DMS to access this functionality on the MSM™ device.

6.1. Theory of Operation

6.1.1. Generalized QMI Service Compliance

The QMI_DMS 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 80-VB816-1. Extensions to the generalized QMI service theory of operation are noted in subsequent sections of this chapter.

6.1.2. DMS Service Type

The DMS is assigned QMI service type 0x02.

6.1.3. Message Definition Template

6.1.3.1. Response Message Result TLV

This Type-Length-Value (TLV) is present in all Response messages defined in this document. It is not present in the Indication messages.

Name	Version introduced	Version last modified
Result Code	Corresponding command's <i>Version introduced</i>	N/A

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
			qmi_error	2	Error code – Possible error code values are described in the error codes section of each message definition

6.1.4. QMI_DMS Fundamental Concepts

The QMI_DMS service enables the control points to query device identification-related information.

Available information includes:

Manufacturer name, device model ID, software and hardware revision

Voice and network identification number of the device

Device capabilities, including the maximum channel rates, data service, SIM support, and radio technologies supported

Device serial numbers corresponding to the wireless technologies supported by the device

Power status information (such as power source and battery level)

UIM-related functions (verify, change, unblock pins, and set pin protection)

Device time

The QMI_DMS service also enables additional device management functionality. This includes:

Managing the operating mode of the device

User-controlled persistent lock state and code maintained by the device

The control point can generally obtain the above information via a polling mechanism (Request and Response messages). The power status change can also be reported via asynchronous indications. These are generated on a change in the value for all parameters, except the battery level. The notification of a battery level change is reported only when a threshold percentage (specified by a control point) is passed. These event-reporting settings registered by the control point are stored in the service state variables of the control point. The Reset command can be used to clear these settings, restoring them to their default values. The details for UIM and PIN-related terms used in Sections 6.2.9 through 6.2.12 of this document are located in 3GPP TS 31.102, 3GPP TS 51.011, and 80-V5329-1. The QMI_DMS UIM commands are only supported for 3GPP devices.

6.1.5. Service State Variables

6.1.5.1. Shared State Variables

No QMI_DMS state variables are shared across control points.

6.1.5.2. State Variables Per Control Point

Name	Description	Possible values	Default value
report_power_state	Indicates whether a power state change is reported to the control point	- TRUE - FALSE	FALSE
battery_lvl_lower_limit	Specifies the battery strength value (as %) below which a report to the control point is sent	0 to 100	0
battery_lvl_upper_limit	Battery strength value (as %) above which a report to the control point is sent	0 to 100	100
report_activation_state	Indicates whether a change in data-bearer technology is reported to the control point	- TRUE - FALSE	FALSE
report_operating_mode	Indicates whether an operating mode change is reported to the control point	- TRUE - FALSE	FALSE
report_uim_state	Indicates whether a UIM state change is reported to the control point	- TRUE - FALSE	FALSE
report_wireless_disable	Indicates whether a wireless disable state change is reported to the control point	- TRUE - FALSE	FALSE
report_prl_init	Indicates whether a PRL initialized notification is reported to the control point	- TRUE - FALSE	FALSE

6.2. QMI_DMS Messages

Command	ID	Description
QMI_DMS_RESET	0x0000	Resets the DMS state variables of the requesting control point.
QMI_DMS_SET_EVENT (QMI_DMS_SET_EVENT_REPORT)	0x0001	Sets the device management state reporting conditions for the requesting control point.
QMI_DMS_EVENT (QMI_DMS_EVENT_REPORT_IND)	0x0001	Indicates the device management state reporting conditions for the requesting control point.
QMI_DMS_GET_CAPS (QMI_DMS_GET_DEVICE_CAP)	0x0020	Requests the device capabilities.
QMI_DMS_GET_MANUFACTURER (QMI_DMS_GET_DEVICE_MFR)	0x0021	Requests the device manufacturer information.

QMI_DMS_GET_MODEL_ID	0x0022	Requests the device model identification.
QMI_DMS_GET_DEVICE_MODEL_ID		
QMI_DMS_GET_DEVICE_REV_ID	0x0023	Requests the device firmware revision identification.
QMI_DMS_GET_NUMBER	0x0024	Requests the assigned voice number.
QMI_DMS_GET_MSISDN		
QMI_DMS_GET_IDS (QMI_DMS_GET_DEVICE_SERIAL_NUMBERS)	0x0025	Requests the serial numbers of the device.
QMI_DMS_GET_POWER_STATE	0x0026	Requests the power status of the device.
QMI_DMS_SET_PIN_PROTECT	0x0027	Enables or disables protection of UIM contents by a specified PIN. (Deprecated)
QMI_DMS_UIM_SET_PIN_PROTECTION		
QMI_DMS_PIN_VERIFY	0x0028	Verifies the PIN before accessing the UIM contents. (Deprecated)
QMI_DMS_UIM_VERIFY_PIN		
QMI_DMS_PIN_UNBLOCK	0x0029	Unblocks a blocked PIN. (Deprecated)
QMI_DMS_UIM_UNBLOCK_PIN		
QMI_DMS_PIN_CHANGE	0x002A	Changes the PIN value. (Deprecated)
QMI_DMS_UIM_CHANGE_PIN		
QMI_DMS_GET_PIN_STATUS (QMI_DMS_UIM_GET_PIN_STATUS)	0x002B	Gets the status of a PIN. (Deprecated)
QMI_DMS_GET_MSM_ID	0x002C	Queries the hardware revision of the device.
QMI_DMS_GET_DEVICE_HARDWARE_REV		
QMI_DMS_GET_OPER_MODE (QMI_DMS_GET_OPERATING_MODE)	0x002D	Queries the current operating mode of the device.
QMI_DMS_SET_OPER_MODE (QMI_DMS_SET_OPERATING_MODE)	0x002E	Sets the operating mode of the device.
QMI_DMS_GET_TIME	0x002F	Queries the current time of the device.
QMI_DMS_GET_PRL_VERSION	0x0030	Queries the version of the active Preferred Roaming List (PRL) of the device.
QMI_DMS_GET_PRL_VER		
QMI_DMS_GET_ICCID (QMI_DMS_UIM_GET_ICCID)	0x003C	Queries the Integrated Circuit Card ID (ICCID) of the UIM for the device. (Deprecated)
QMI_DMS_GET_IMSI (QMI_DMS_UIM_GET_IMSI)	0x0043	Queries the International Mobile Station Identity (IMSI) of the UIM for the device. (Deprecated)
QMI_DMS_GET_UIM_STATE (QMI_DMS_UIM_GET_STATE)	0x0044	Queries the state of the UIM.(Deprecated)

QMI_DMS_GET_BAND_CAPS	0x0045	Queries the band capability of the device.
QMI_DMS_GET_FACTORY_ID	0x0046	Queries the factory provisioned Stock Keeping Unit (SKU).
QMI_DMS_GET_SW_VERSION	0x0051	Queries the software version from the device.

6.2.1. QMI_DMS_RESET

Resets the DMS state variables of the requesting control point.

- **DMS message ID**

0x0000

- **Version introduced**

Major - 1, Minor - 0

6.2.1.1. Request - QMI_DMS_RESET_REQ

- **Message type**

Request

- **Sender**

Control point

- **Mandatory TLVs**

None

- **Optional TLVs**

None

6.2.1.2. Response - QMI_DMS_RESET_RESP

- **Message type**

Response

- **Sender**

Service

- **Mandatory TLVs**

The Result Code TLV (defined in Section 6.1.3.1) 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

6.2.1.3. Description of QMI_DMS_RESET REQ/RESP

This command resets the issuing control point state (see Section 6.1.5.2) kept by the service. As a result, each shared state variable can change depending on its arbitration policy (see Section 6.1.5.1). Although it is performed as one operation, this is equivalent to closing the service and reopening it again, therefore the client ID of the requesting control point does not change. The control point state variables change to their default values before the response is issued.

6.2.2. QMI_DMS_SET_EVENT_REPORT

Sets the device management state reporting conditions for the requesting control point.

- **DMS message ID**

0x0001

- **Version introduced**

Major - 1, Minor - 0

6.2.2.1. Request - QMI_DMS_SET_EVENT_REPORT_REQ

- **Message type**

Request

- **Sender**

Control point

- **Mandatory TLVs**

None

- **Optional TLVs**

Name	Version introduced	Version last modified
Power State Reporting	1.0	1.0
Battery Level Report Limits	1.0	1.0
PIN State Reporting	Unknown	1.1
Activation State Reporting	Unknown	1.6
Operating Mode Reporting	Unknown	1.3
UIM State Reporting	Unknown	1.3
Wireless Disable State Reporting	Unknown	1.6
PRL Init Reporting	Unknown	1.7
CDMA Lock Mode Reporting	1.24	1.24

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Power State Reporting
Length	1			2	
Value	→	boolean	report_power_state	1	Values: • 0 – Do not report • 1 – Report on change in power state
Type	0x11			1	Battery Level Report Limits
Length	2			2	

6.2.2.2. Response - QMI_DMS_SET_EVENT_REPORT_RESP

- **Message type**

Response

- **Sender**

Service

- **Mandatory TLVs**

The Result Code TLV (defined in Section 6.1.3.1) is always present in the response.

- **Optional TLVs**

None

6.2.2.3. Indication - QMI_DMS_EVENT_REPORT_IND

- **Message type**

Indication

- **Sender**

Service

- **Indication scope**

Per control point (unicast)

- **Mandatory TLVs**

None

- **Optional TLVs**

Name	Version introduced	Version last modified
Power State	1.0	1.0
PIN 1 Status	Unknown	1.1
PIN 2 Status	Unknown	1.1
Activation State	Unknown	1.6
Operating Mode	1.3	1.12
UIM State	Unknown	1.3
Wireless Disable State	Unknown	1.6
PRL Init Notification	Unknown	1.7
CDMA Lock Mode State	1.24	1.24

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Power State
Length	2			2	
Value	→	mask8	power_status	1	<p>Power status flags. Values:</p> <ul style="list-style-type: none"> Bit 0 – Power source <ul style="list-style-type: none"> • 0 – Powered by battery • 1 – Powered by external source Bit 1 – Battery connected <ul style="list-style-type: none"> • 0 – Not connected • 1 – Connected Bit 2 – Battery charging <ul style="list-style-type: none"> • 0 – Not charging • 1 – Charging Bit 3 – Power fault <ul style="list-style-type: none"> • 0 – No power fault • 1 – Recognized power fault, calls inhibited
					<p>uint8</p> <p>battery_lvl</p> <p>1</p> <p>Level of the battery. Values:</p> <ul style="list-style-type: none"> • 0x00 – Battery is exhausted or the mobile device does not have a battery connected • 1 through 100 (0x64) – Percentage of battery capacity remaining
Type	0x11			1	PIN 1 Status
Length	3			2	
Value	→	enum8	status	1	<p>Current status of the PIN. Values:</p> <ul style="list-style-type: none"> • 0 – PIN is not initialized • 1 – PIN is enabled, not verified • 2 – PIN is enabled, verified • 3 – PIN is disabled • 4 – PIN is blocked • 5 – PIN is permanently blocked • 6 – PIN is unblocked • 7 – PIN is changed
					<p>uint8</p> <p>verify_retries_left</p> <p>1</p> <p>Number of retries left, after which the PIN is blocked.</p>
					<p>uint8</p> <p>unblock_retries_left</p> <p>1</p> <p>Number of unblock retries left, after which the PIN is permanently blocked, i.e., the UIM is unusable.</p>
Type	0x12			1	PIN 2 Status
Length	3			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	enum8	status	1	<p>Current status of the PIN. Values:</p> <ul style="list-style-type: none"> • 0 – PIN is not initialized • 1 – PIN is enabled, not verified • 2 – PIN is enabled, verified • 3 – PIN is disabled • 4 – PIN is blocked • 5 – PIN is permanently blocked • 6 – PIN is unblocked • 7 – PIN is changed
		uint8	verify_retries_left	1	Number of retries left, after which the PIN is blocked.
		uint8	unblock_retries_left	1	Number of unblock retries left, after which the PIN is permanently blocked, i.e., the UIM is unusable.
Type	0x13			1	Activation State
Length	2			2	
Value	→	enum16	activation_state	2	<p>Service activation state. Values:</p> <ul style="list-style-type: none"> • 0x00 – Service is not activated • 0x01 – Service is activated • 0x02 – Activation connecting – Network connection is in progress for automatic activation of service • 0x03 – Activation connected – Network connection is connected for automatic activation of service • 0x04 – OTASP security is authenticated • 0x05 – OTASP NAM is downloaded • 0x06 – OTASP MDN is downloaded • 0x07 – OTASP IMSI downloaded • 0x08 – OTASP PRL is downloaded • 0x09 – OTASP SPC is downloaded • 0x0A – OTASP settings are committed
Type	0x14			1	Operating Mode
Length	1			2	
Value	→	enum8	operating_mode	1	<p>Current operating mode. Values:</p> <ul style="list-style-type: none"> • 0 – Online • 1 – Low power • 2 – Factory Test mode • 3 – Offline • 4 – Resetting • 5 – Shutting down • 6 – Persistent low power • 7 – Mode-only low power • 8 – Conducting network test for GSM/WCDMA
Type	0x15			1	UIM State
Length	1			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	enum8	uim_state	1	UIM state. Values: • 0x00 – UIM initialization completed • 0x01 – UIM failed • 0x02 – UIM is not present • 0xFF – UIM state is currently unavailable
Type	0x16			1	Wireless Disable State
Length	1			2	
Value	→	enum8	wireless_disable_state	1	Wireless disable state. Values: • 0x00 – Wireless disable switch is turned off • 0x01 – Wireless disable switch is turned on
Type	0x17			1	PRL Init Notification
Length	1			2	
Value	→	enum8	prl_init	1	PRL initialized. Values: • 0x01 – PRL is completely loaded into the device (could be the default PRL).
Type	0x18			1	CDMA Lock Mode State
Length	4			2	
Value	→	enum	cdma_lock_mode_state	4	CDMA Lock mode state. Values: • DMS_CDMA_LOCK_MODE_OFF (0) – Phone is not CDMA locked • DMS_CDMA_LOCK_MODE_ON (1) – Phone is CDMA locked

Error codes

QMI_ERR_NONE	No error in the request
QMI_ERR_INTERNAL	Unexpected error occurred during processing
QMI_ERR_MALFORMED_MSG	The message was not formulated correctly by the control point or the message was corrupted during transmission
QMI_ERR_MISSING_ARG	One or more required TLVs were missing in the request
QMI_ERR_OP_DEVICE_UNSUPPORTED	Some of the TLVs, e.g., report_uim_state, are not supported because the device lacks underlying support. This error is returned even if the message contains a mix of supported and unsupported TLVs. The control point is expected to register separately for each event in such a situation.

6.2.2.4. Description of QMI_DMS_SET_EVENT_REPORT

The control point state variables controlling event reporting are modified according to the TLVs present in the request. The service maintains a separate set of state variables for each control point. See Section 6.1.5.2 for more details regarding control point state variables. Specific device management state changes are communicated to the registered DMS control point via the QMI_DMS_EVENT_REPORT_IND indicator message. The AT command equivalents to this command are **AT+CMER**, **AT+CIND**, and **AT+CIEV**.

This command is sent to specific control points when the device state corresponding to one of the previous TLVs has changed. The specific control points are those that previously registered for the corresponding state to be reported using the QMI_DMS_SET_EVENT_REPORT_REQ message.

The Power State TLV is included when any of the following occurs:

The control point sets the battery level limits, and the battery level triggers either the upper or lower limit, i.e., the Request message.

The control point enables power state reporting and the power state changes.

The PIN1 Status or PIN2 Status TLVs are included when the control point has enabled PIN status reporting and the PIN status has changed.

The AT command equivalents to this command are **AT+CMER**, **AT+CIND**, and **AT+CIEV**.

The Activation State TLV is included when the activation state of the device has changed.

The Operating Mode TLV is included when the control point has enabled Operating Mode reporting and the operating mode of the device has changed.

The UIM State TLV is included when the control point has enabled UIM state reporting and the UIM state of the device has changed.

The Wireless Disable TLV is included when the wireless disable signal state for the device has changed.

6.2.3. QMI_DMS_GET_DEVICE_CAP

Requests the device capabilities.

- **DMS message ID**

0x0020

- **Version introduced**

Major - 1, Minor - 0

6.2.3.1. Request - QMI_DMS_GET_DEVICE_CAP_REQ

- **Message type**

Request

- **Sender**

Control point

- **Mandatory TLVs**

None

- **Optional TLVs**

None

6.2.3.2. Response - QMI_DMS_GET_DEVICE_CAP_RESP

- **Message type**

Response

- **Sender**

Service

- **Mandatory TLVs**

The Result Code TLV (defined in Section 6.1.3.1) is always present in the response. This TLV is present if the result code is QMI_RESULT_SUCCESS.

Name	Version introduced	Version last modified
Device Capabilities	Unknown	1.10

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Device Capabilities
Length	Var			2	
Value	→	uint32	max_tx_channel_rate	4	Maximum Tx transmission rate in bits per second (bps) supported by the device. The value 0xFFFFFFFF implies a rate greater than or equal to 0xFFFFFFFF (4 Gbps). In multitechnology devices, this value is the greatest rate among all supported technologies.
		uint32	max_rx_channel_rate	4	Maximum Rx transmission rate in bits per second (bps) supported by the device. The value 0xFFFFFFFF implies rate greater than or equal to 0xFFFFFFFF (4 Gbps). In multitechnology devices, this value is the greatest rate among all supported technologies.
		enum8	data_service_capability	1	Values: <ul style="list-style-type: none">• 0 – No data services supported• 1 – Only circuit-switched (CS) services are supported• 2 – Only packet-switched (PS) services are supported• 3 – Simultaneous CS and PS• 4 – Nonsimultaneous CS and PS
		enum8	sim_capability	1	Values: <ul style="list-style-type: none">• 1 – SIM is not supported• 2 – SIM is supported
		uint8	radio_if_list_len	1	Number of sets of the following elements: <ul style="list-style-type: none">• radio_if_list
		enum8	radio_if_list	Var	List of N one-byte elements describing the radio interfaces supported by the device. Values: <ul style="list-style-type: none">• 1 – CDMA2000 1X• 2 – CDMA2000 HRPD (1xEV-DO)• 4 – GSM• 5 – UMTS• 8 – LTE• 9 – TDS

Optional TLVs

Name	Version introduced	Version last modified
Device Service Capability	1.11	1.11
Voice Support Capability	1.11	1.11
Simultaneous Voice and Data Capability	1.13	1.13
Device Multisim Capability	1.22	1.22

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Device Service Capability
Length	4			2	
Value	→	enum	device_service_capability	4	Values: • 1 – Only data services are supported • 2 – Only voice services are supported • 3 – Simultaneous voice and data • 4 – Nonsimultaneous voice and data
Type	0x11			1	Voice Support Capability
Length	8			2	
Value	→	mask	voice_support_capability	8	Bitmask of voice support available on device. Values: Bit 0 – GW CSFB • 0 – Not capable • 1 – Capable Bit 1 – 1x CSFB • 0 – Not capable • 1 – Capable Bit 2 – VoLTE • 0 – Not capable • 1 – Capable
Type	0x12			1	Simultaneous Voice and Data Capability
Length	8			2	
Value	→	mask	simul_voice_and_data_capability	8	Bitmask of simultaneous voice and data support available on device. Values: • Bit 0 – SVLTE capability • Bit 1 – SVDO capability Note: Zero bits set means that neither of the defined capabilities are supported.
Type	0x13			1	Device Multisim Capability Device capability for supporting multiple simultaneously active radio interfaces.
Length	Var			2	
Value	→	uint8	max_subscriptions	1	The maximum number of subscriptions that can be supported simultaneously.
		uint8	subscription_config_list_len	1	Number of sets of the following elements: • max_active • subscription_list
		uint8	max_active	1	The maximum number of subscriptions listed in this configuration that can be simultaneously active. If this number is less than max_subscriptions it implies that any combination of the subscriptions in this configuration can be active and the remaining can be in standby

Field	Field value	Field type	Parameter	Size (byte)	Description
		uint8	subscription_list_len	1	Number of sets of the following elements: • subscription_list
		mask	subscription_list	Var	An array of max_subscriptions entries where each entry is a mask of capabilities. The client ignores any bits in the mask that it does not recognize. Values: • DMS_SUBS_CAPABILITY_AMPS (0x00000001) – • DMS_SUBS_CAPABILITY_CDMA (0x00000002) – • DMS_SUBS_CAPABILITY_HDR (0x00000004) – • DMS_SUBS_CAPABILITY_GSM (0x00000008) – • DMS_SUBS_CAPABILITY_WCDMA (0x00000010) – • DMS_SUBS_CAPABILITY_LTE (0x00000020) – • DMS_SUBS_CAPABILITY_TDS (0x00000040) –

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

6.2.3.3. Description of QMI_DMS_GET_DEVICE_CAP REQ/RESP

This command obtains the high-level capabilities of the device. The AT command equivalent to this command is **AT+GCAP**

The Device Multisim Capability TLV includes a list of supported multisim configurations. Each entry in this list includes a max_active field and a max_subscriptions field. The order of the bitmask fields in this list does not correspond to any type of subscription index. Each bitmask field represents capabilities of a single subscription.

For example, consider a device where max_subscriptions is 3. One entry in the subscription configuration list has max_active = 2, with the following values :

```
subscription list[0] = DMS_SUBS_CAPABILITY_GSM | DMS_SUBS_CAPABILITY_WCDMA
subscription list[1] = DMS_SUBS_CAPABILITY_GSM
subscription list[2] = DMS_SUBS_CAPABILITY_GSM
```

This means the device supports a configuration with three GSM subscriptions, a configuration with two GSM subscriptions, and one WCDMA subscription. For any of these configurations the device supports any two being active simultaneously with any remaining subscriptions on standby

6.2.4. QMI_DMS_GET_DEVICE_MFR

Requests the device the manufacturer information.

- **DMS message ID**

0x0021

- **Version introduced**

Major - 1, Minor - 0

6.2.4.1. Request - QMI_DMS_GET_DEVICE_MFR_REQ

- **Message type**

Request

- **Sender**

Control point

- **Mandatory TLVs**

None

- **Optional TLVs**

None

6.2.4.2. Response - QMI_DMS_GET_DEVICE_MFR_RESP

- **Message type**

Response

- **Sender**

Service

- **Mandatory TLVs**

The Result Code TLV (defined in Section 6.1.3.1) is always present in the response. This TLV is present if the result code is QMI_RESULT_SUCCESS.

Name	Version introduced	Version last modified
Device Manufacturer	1.0	1.0

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Device Manufacturer
Length	Var			2	
Value	→	string	device_manufacturer	Var	String identifying the device manufacturer.

Optional TLVs

None

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 the response

6.2.4.3. Description of QMI_DMS_GET_DEVICE_MFR REQ/RESP

This command returns a string identifying the device manufacturer. The AT command equivalent to this command is **AT+GMI**

6.2.5. QMI_DMS_GET_DEVICE_MODEL_ID

Requests the device model identification.

- **DMS message ID**

0x0022

- **Version introduced**

Major - 1, Minor - 0

6.2.5.1. Request - QMI_DMS_GET_DEVICE_MODEL_ID_REQ

- **Message type**

Request

- Sender

Control point

- Mandatory TLVs

None

- Optional TLVs

None

6.2.5.2. Response - QMI_DMS_GET_DEVICE_MODEL_ID_RESP

- Message type

Response

- Sender

Service

- Mandatory TLVs

The Result Code TLV (defined in Section 6.1.3.1) is always present in the response. This TLV is present if the result code is QMI_RESULT_SUCCESS.

Name	Version introduced	Version last modified
Device Model	1.0	1.0

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Device Model
Length	Var			2	
Value	→	string	device_model_id	Var	String identifying the device model.

Optional TLVs

None

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

6.2.5.3. Description of QMI_DMS_GET_DEVICE_MODEL_ID REQ/RESP

This command returns a string identifying the model of the device. This usually corresponds to the manufacturer's model name under which the device is marketed. The AT command equivalent to this command is **AT+GMM**

6.2.6. QMI_DMS_GET_DEVICE_REV_ID

Requests the device firmware revision identification.

- **DMS message ID**

0x0023

- **Version introduced**

Major - 1, Minor - 0

6.2.6.1. Request - QMI_DMS_GET_DEVICE_REV_ID_REQ

- **Message type**

Request

- **Sender**

Control point

- **Mandatory TLVs**

None

- **Optional TLVs**

None

6.2.6.2. Response - QMI_DMS_GET_DEVICE_REV_ID_RESP

- **Message type**

Response

- **Sender**

Service

- **Mandatory TLVs**

The Result Code TLV (defined in Section 6.1.3.1) is always present in the response. This TLV is present if the result code is QMI_RESULT_SUCCESS.

Name	Version introduced	Version last modified
Revision ID	1.0	1.0

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Revision ID
Length	Var			2	
Value	→	string	device_rev_id	Var	String containing the device revision ID.

Optional TLVs

Name	Version introduced	Version last modified
Boot Code Revision	Unknown	1.6
PRI Revision	Unknown	1.6

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Boot Code Revision
Length	Var			2	
Value	→	string	boot_code_rev	Var	String containing the boot code revision.
Type	0x11			1	PRI Revision
Length	Var			2	
Value	→	string	pri_rev	Var	String containing the device PRI revision.

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

6.2.6.3. Description of QMI_DMS_GET_DEVICE_REV_ID REQ/RESP

This command returns a string identifying the firmware revision of the device. This usually corresponds to the manufacturer's software revision loaded on the device. The AT command equivalent to this command is **AT+GMR**

If supported by the device, one or more optional TLVs are also returned:

Boot Code revision – Revision of the boot software used to power up the device

PRI Revision – Revision of the factory configuration loaded to the device

6.2.7. QMI_DMS_GET_MSISDN

Requests the assigned voice number.

- **DMS message ID**

0x0024

- **Version introduced**

Major - 1, Minor - 0

6.2.7.1. Request - QMI_DMS_GET_MSISDN_REQ

- **Message type**

Request

- **Sender**

Control point

- **Mandatory TLVs**

None

- **Optional TLVs**

None

6.2.7.2. Response - QMI_DMS_GET_MSISDN_RESP

- **Message type**

Response

- **Sender**

Service

- **Mandatory TLVs**

The Result Code TLV (defined in Section 6.1.3.1) is always present in the response. This TLV is present if the result code is QMI_RESULT_SUCCESS.

Name	Version introduced	Version last modified
Voice Number	1.0	1.0

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Voice Number
Length	Var			2	
Value	→	string	voice_number	Var	String containing the voice number in use by the device.

Optional TLVs

Name	Version introduced	Version last modified
Mobile ID	Unknown	1.3
International Mobile Subscriber ID	Unknown	1.6

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Mobile ID
Length	Var			2	
Value	→	string	mobile_id_number	Var	String containing the mobile ID number of the device.
Type	0x11			1	International Mobile Subscriber ID
Length	Var			2	
Value	→	string	imsi	Var	String containing the international mobile subscriber ID of the device.

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_NOT_PROVISIONED	Device does not support voice service or the value is not provisioned in the device

6.2.7.3. Description of QMI_DMS_GET_MSISDN REQ/RESP

The voice number is the MDN or MSISDN assigned to the mobile. If it is available in the device provisioning, an optional mobile ID and IMSI is returned. The AT command equivalent to this command is **AT+CNUM**.

6.2.8. QMI_DMS_GET_DEVICE_SERIAL_NUMBERS

Requests the serial numbers of the device.

- DMS message ID

0x0025

- **Version introduced**

Major - 1, Minor - 0

6.2.8.1. Request - QMI_DMS_GET_DEVICE_SERIAL_NUMBERS_REQ

- **Message type**

Request

- **Sender**

Control point

- **Mandatory TLVs**

None

- **Optional TLVs**

None

6.2.8.2. Response - QMI_DMS_GET_DEVICE_SERIAL_NUMBERS_RESP

- **Message type**

Response

- **Sender**

Service

- **Mandatory TLVs**

The Result Code TLV (defined in Section 6.1.3.1) is always present in the response.

- **Optional TLVs**

Name	Version introduced	Version last modified
ESN	1.0	1.0
IMEI	1.0	1.0
MEID	1.0	1.0
IMEI SVN	Unknown	1.5

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	ESN
Length	Var			2	
Value	→	string	esn	Var	String containing the Electronic Serial Number (ESN) of the device.
Type	0x11			1	IMEI
Length	Var			2	
Value	→	string	imei	Var	String containing the International Mobile Equipment Identity (IMEI) of the device.
Type	0x12			1	MEID
Length	Var			2	
Value	→	string	meid	Var	String containing the Mobile Equipment Identifier (MEID) of the device.
Type	0x13			1	IMEI SVN
Length	Var			2	
Value	→	string	imeisv_svn	Var	IMEI software version number

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_NOT_PROVISIONED	Device does not support voice service or the value is not provisioned in the device

6.2.8.3. Description of QMI_DMS_GET_DEVICE_SERIAL_NUMBERS REQ/RESP

This command returns all serial numbers assigned to the device as follows:

ESN is included for 3GPP2 devices

IMEI is included for 3GPP devices

MEID is included for devices that support it, e.g., 3GPP or 3GPP2

IMEI software version number is included for 3GPP devices

The AT command equivalent to this command is **AT+GSN** .

6.2.9. QMI_DMS_GET_POWER_STATE

Requests the power status of the device.

- DMS message ID

0x0026

- Version introduced

Major - 1, Minor - 0

6.2.9.1. Request - QMI_DMS_GET_POWER_STATE_REQ

- **Message type**

Request

- **Sender**

Control point

- **Mandatory TLVs**

None

- **Optional TLVs**

None

6.2.9.2. Response - QMI_DMS_GET_POWER_STATE_RESP

- **Message type**

Response

- **Sender**

Service

- **Mandatory TLVs**

The Result Code TLV (defined in Section 6.1.3.1) is always present in the response. This TLV is present if the result code is QMI_RESULT_SUCCESS.

Name	Version introduced	Version last modified
Power State	1.0	1.0

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Power State
Length	2			2	
Value	→	mask8	power_status	1	Power status flags. Values: Bit 0 – Power source • 0 – Powered by battery • 1 – Powered by external source
					Bit 1 – Battery connected • 0 – Not connected • 1 – Connected
		uint8	battery_lvl	1	Bit 2 – Battery charging • 0 – Not charging • 1 – Charging
					Bit 3 – Power fault • 0 – No power fault • 1 – Recognized power fault, calls inhibited
				1	Level of the battery. Values: • 0x00 – Battery is exhausted or the mobile device does not have a battery connected • 1 through 100 (0x64) – Percentage of battery capacity remaining

Optional TLVs

None

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

6.2.9.3. Description of QMI_DMS_GET_POWER_STATE REQ/RESP

This command obtains information regarding the power status of the device. The information returned is described in the Mandatory TLVs section. The external power source can be one of the following:

Wall-mounted power source

USB charger

The AT command equivalent to this command is **AT+CBC**

6.2.10. QMI_DMS_UIM_SET_PIN_PROTECTION

Enables or disables protection of UIM contents by a specified PIN. (Deprecated)

- DMS message ID

0x0027

- Version introduced

Major - 1, Minor - 1

6.2.10.1. Request - QMI_DMS_UIM_SET_PIN_PROTECTION_REQ

- Message type

Request

- Sender

Control Point

- Mandatory TLVs

Name	Version introduced	Version last modified
PIN Protection Information	Unknown	1.1

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	PIN Protection Information
Length	Var			2	
Value	→	enum8	pin_id	1	Specifies the ID of the PIN to be enabled or disabled. Values: <ul style="list-style-type: none">• 1 – PIN1 (also called PIN)• 2 – PIN2
		boolean	protection_setting_enabled	1	Specifies whether the PIN is enabled. Values: <ul style="list-style-type: none">• 0 – Disable PIN• 1 – Enable PIN
		uint8	pin_value_len	1	Number of sets of the following elements: <ul style="list-style-type: none">• pin_value
		uint8	pin_value	Var	Specifies the PIN value of the PIN to be enabled/disabled. The protection setting is only changed if this value is successfully verified by the SIM.

Optional TLVs

None

6.2.10.2. Response - QMI_DMS_UIM_SET_PIN_PROTECTION_RESP

- **Message type**

Response

- **Sender**

Service

- **Mandatory TLVs**

The Result Code TLV (defined in Section 6.1.3.1) is always present in the response.

- **Optional TLVs**

Returned if the verify operation fails.

Name	Version introduced	Version last modified
Pin Retries Status	Unknown	1.1

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Pin Retries Status
Length	2			2	
Value	→	uint8	verify_retries_left	1	Number of retries left, after which the PIN is blocked.
		uint8	unblock_retries_left	1	Number of unblock retries left, after which the PIN is permanently blocked, i.e., the UIM is unusable.

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_NO_EFFECT	Operation had no effect
QMI_ERR_ARG_TOO_LONG	Device cannot handle the length of the PIN specified
QMI_ERR_INCORRECT_PIN	PIN specified in the request is incorrect
QMI_ERR_PIN_BLOCKED	PIN is blocked; an unblock operation needs to be issued
QMI_ERR_PIN_PERM_BLOCKED	PIN is permanently blocked; the UIM is unusable

QMI_ERR_UIM_NOT_INITIALIZED	PIN is not yet initialized because the SIM initialization has not finished; try the PIN operation later
QMI_ERR_OP_DEVICE_UNSUPPORTED	Operation is not supported by the device
QMI_ERR_MISSING_ARG	One or more required TLVs were missing in the request
QMI_ERR_INVALID_PINID	PIN specified in the request is invalid
QMI_ERR_ACCESS_DENIED	Operation cannot be performed because the UIM cannot be accessed

6.2.10.3. Description of QMI_DMS_UIM_SET_PIN_PROTECTION REQ/RESP

This command enables or disables the protection of the UIM contents by a specified PIN.

This command is deprecated. QMI_UIM_SET_PIN_PROTECTION is the equivalent command in the QMI_UIM service and should be used in its place

6.2.11. QMI_DMS_UIM_VERIFY_PIN

Verifies the PIN before accessing the UIM contents. (Deprecated)

- **DMS message ID**

0x0028

- **Version introduced**

Major - 1, Minor - 1

6.2.11.1. Request - QMI_DMS_UIM_VERIFY_PIN_REQ

- **Message type**

Request

- **Sender**

Control point

- **Mandatory TLVs**

Name	Version introduced	Version last modified
PIN Value	Unknown	1.1

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	PIN Value
Length	Var			2	
Value	→	enum8	pin_id	1	Specifies the ID of the PIN to be enabled or disabled. Values: <ul style="list-style-type: none">• 1 – PIN1 (also called PIN)• 2 – PIN2
		uint8	pin_value_len	1	Number of sets of the following elements: <ul style="list-style-type: none">• pin_value
		uint8	pin_value	Var	Specifies the PIN value of the PIN to be verified; the protection setting is only changed if this value is successfully verified by the SIM.

- Optional TLVs

None

6.2.11.2. Response - QMI_DMS_UIM_VERIFY_PIN_RESP

- Message type

Response

- Sender

Service

- Mandatory TLVs

The Result Code TLV (defined in Section 6.1.3.1) is always present in the response.

- Optional TLVs

Returned if the verify operation fails.

Name	Version introduced	Version last modified
PIN Retries Status	Unknown	1.1

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	PIN Retries Status
Length	2			2	
Value	→	uint8	verify_retries_left	1	Number of retries left, after which the PIN is blocked.
		uint8	unblock_retries_left	1	Number of unblock retries left, after which the PIN is permanently blocked, i.e., the UIM is unusable.

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 message was corrupted during transmission
QMI_ERR_NO_MEMORY	Device could not allocate memory to formulate a response
QMI_ERR_NO_EFFECT	Operation had no effect
QMI_ERR_ARG_TOO_LONG	Device cannot handle the length of the PIN specified in the request
QMI_ERR_INCORRECT_PIN	PIN specified in the request is incorrect
QMI_ERR_PIN_BLOCKED	PIN is blocked; an unblock operation needs to be issued
QMI_ERR_PIN_PERM_BLOCKED	PIN is permanently blocked; the SIM is unusable

QMI_ERR_UIM_NOT_INITIALIZED	PIN is not yet initialized because the SIM initialization has not finished; try the PIN operation later
QMI_ERR_OP_DEVICE_UNSUPPORTED	Operation is not supported by the device
QMI_ERR_MISSING_ARG	One or more required TLVs were missing in the request
QMI_ERR_INVALID_PINID	PIN specified in the request is invalid
QMI_ERR_ACCESS_DENIED	Operation cannot be performed because the UIM cannot be accessed

6.2.11.3. Description of QMI_DMS_UIM_VERIFY_PIN REQ/RESP

This command verifies the PIN before accessing the UIM contents.

This command is deprecated. QMI_UIM_VERIFY_PIN is the equivalent command in the QMI_UIM service and should be used in its place

6.2.12. QMI_DMS_UIM_UNBLOCK_PIN

Unblocks a blocked PIN. (Deprecated)

- DMS message ID

0x0029

- Version introduced

Major - 1, Minor - 1

6.2.12.1. Request - QMI_DMS_UIM_UNBLOCK_PIN_REQ

- Message type

Request

- Sender

Control point

- Mandatory TLVs

Name	Version introduced	Version last modified
PIN Unblock Information	Unknown	1.1

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	PIN Unblock Information
Length	Var			2	
Value	→	enum8	unblock_pin_id	1	Specifies the ID of the PIN to be unblocked. Values: <ul style="list-style-type: none">• 1 – PIN1 (also called PIN)• 2 – PIN2
		uint8	puk_value_len	1	Number of sets of the following elements: <ul style="list-style-type: none">• puk_value
		uint8	puk_value	Var	Specifies the PUK value (password) of the PIN to be unblocked.
		uint8	new_pin_value_len	1	Number of sets of the following elements: <ul style="list-style-type: none">• new_pin_value
		uint8	new_pin_value	Var	Specifies the new PIN value (password) for the PIN to be unblocked.

- Optional TLVs

None

6.2.12.2. Response - QMI_DMS_UIM_UNBLOCK_PIN_RESP

- Message type

Response

- **Sender**

Service

- **Mandatory TLVs**

The Result Code TLV (defined in Section 6.1.3.1) is always present in the response.

- **Optional TLVs**

Returned if the unblock operation failed

Name	Version introduced	Version last modified
PIN Retries Status	Unknown	1.1

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	PIN Retries Status
Length	2			2	
Value	→	uint8	verify_retries_left	1	Number of retries left, after which the PIN is blocked.
		uint8	unblock_retries_left	1	Number of unblock retries left, after which the PIN is permanently blocked, i.e., the UIM is unusable.

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 message was corrupted during transmission
QMI_ERR_NO_MEMORY	Device could not allocate memory to formulate a response
QMI_ERR_NO_EFFECT	Operation had no effect
QMI_ERR_ARG_TOO_LONG	Device cannot handle the length of the PIN specified in the request
QMI_ERR_INCORRECT_PIN	PIN or PUK specified in the request is incorrect
QMI_ERR_PIN_BLOCKED	PIN is blocked; an unblock operation needs to be issued
QMI_ERR_PIN_PERM_BLOCKED	PIN is permanently blocked; the SIM is unusable

QMI_ERR_UIM_NOT_INITIALIZED	PIN is not yet initialized because the SIM initialization has not finished; try the PIN operation later
QMI_ERR_OP_DEVICE_UNSUPPORTED	Operation is not supported by the device
QMI_ERR_MISSING_ARG	One or more required TLVs were missing in the request
QMI_ERR_INVALID_PINID	PIN specified in the request is invalid
QMI_ERR_ACCESS_DENIED	Operation cannot be performed because the UIM cannot be accessed

6.2.12.3. Description of QMI_DMS_UIM_UNBLOCK_PIN REQ/RESP

This command unblocks a blocked PIN using the PUK provided in the request TLV. The user must enter PUK1 for PIN1, or PUK2 for PIN2.

This command is deprecated. QMI_UIM_UNBLOCK_PIN is the equivalent command in the QMI_UIM service and should be used in its place

6.2.13. QMI_DMS_UIM_CHANGE_PIN

Changes the PIN value. (Deprecated)

- **DMS message ID**

0x002A

- **Version introduced**

Major - 1, Minor - 1

6.2.13.1. Request - QMI_DMS_UIM_CHANGE_PIN_REQ

- **Message type**

Request

- **Sender**

Control point

- **Mandatory TLVs**

Name	Version introduced	Version last modified
PIN Change Information	Unknown	1.1

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	PIN Change Information
Length	Var			2	
Value	→	enum8	pin_id	1	Specifies the ID of the PIN to be changed. Values: <ul style="list-style-type: none">• 1 – PIN1 (also called PIN)• 2 – PIN2
		uint8	old_pin_value_len	1	Number of sets of the following elements: <ul style="list-style-type: none">• old_pin_value
		uint8	old_pin_value	Var	Specifies the old PIN value (old password) of the PIN.
		uint8	new_pin_value_len	1	Number of sets of the following elements: <ul style="list-style-type: none">• new_pin_value
		uint8	new_pin_value	Var	Specifies the new PIN value (new password) of the PIN.

- Optional TLVs

None

6.2.13.2. Response - QMI_DMS_UIM_CHANGE_PIN_RESP

- Message type

Response

- Sender

Service

- Mandatory TLVs

The Result Code TLV (defined in Section 6.1.3.1) is always present in the response.

- Optional TLVs

Returned if the enable/disable operation failed.

Name	Version introduced	Version last modified
PIN Retries Status	Unknown	1.1

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	PIN Retries Status
Length	2			2	
Value	→	uint8	verify_retries_left	1	Number of retries left, after which the PIN is blocked.
			unblock_retries_left	1	Number of unblock retries left, after which the PIN is permanently blocked, i.e., the UIM is unusable.

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 message was corrupted during transmission
QMI_ERR_NO_MEMORY	Device could not allocate memory to formulate a response
QMI_ERR_NO_EFFECT	Operation had no effect
QMI_ERR_ARG_TOO_LONG	Device cannot handle the length of the PIN specified in the request
QMI_ERR_INCORRECT_PIN	PIN specified in the request is incorrect
QMI_ERR_PIN_BLOCKED	PIN is blocked. An unblock operation needs to be issued
QMI_ERR_PIN_PERM_BLOCKED	PIN is permanently blocked; the SIM is unusable

6.2.13.3. Description of QMI_DMS_UIM_CHANGE_PIN REQ/RESP

This command changes the old value of a specified PIN to the new value provided in the request TLV. This command is deprecated. QMI_UIM_CHANGE_PIN is the equivalent command in the QMI_UIM service and should be used in its place .

6.2.14. QMI_DMS_UIM_GET_PIN_STATUS

Gets the status of a PIN. (Deprecated)

- DMS message ID

0x002B

- Version introduced

Major - 1, Minor - 1

6.2.14.1. Request - QMI_DMS_UIM_GET_PIN_STATUS_REQ

- **Message type**

Request

- **Sender**

Control point

- **Mandatory TLVs**

None

- **Optional TLVs**

None

6.2.14.2. Response - QMI_DMS_UIM_GET_PIN_STATUS_RESP

- **Message type**

Response

- **Sender**

Service

- **Mandatory TLVs**

The Result Code TLV (defined in Section 6.1.3.1) is always present in the response.

- **Optional TLVs**

Name	Version introduced	Version last modified
PIN 1 Status	Unknown	1.1
PIN 2 Status	Unknown	1.1

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x11			1	PIN 1 Status
Length	3			2	
Value	→	enum8	status	1	Current status of the PIN. Values: • 0 – PIN is not initialized • 1 – PIN is enabled, not verified • 2 – PIN is enabled, verified • 3 – PIN is disabled • 4 – PIN is blocked • 5 – PIN is permanently blocked • 6 – PIN is unblocked • 7 – PIN is changed
		uint8	verify_retries_left	1	Number of retries left, after which the PIN is blocked.
		uint8	unblock_retries_left	1	Number of unblock retries left, after which the PIN is permanently blocked, i.e., the UIM is unusable.
Type	0x12			1	PIN 2 Status
Length	3			2	
Value	→	enum8	status	1	Current status of the PIN. Values: • 0 – PIN is not initialized • 1 – PIN is enabled, not verified • 2 – PIN is enabled, verified • 3 – PIN is disabled • 4 – PIN is blocked • 5 – PIN is permanently blocked • 6 – PIN is unblocked • 7 – PIN is changed
		uint8	verify_retries_left	1	Number of retries left, after which the PIN is blocked.
		uint8	unblock_retries_left	1	Number of unblock retries left, after which the PIN is permanently blocked, i.e., the UIM is unusable.

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_UIM_NOT_INITIALIZED	PIN is not yet initialized because SIM initialization has not finished; try the PIN operation later
QMI_ERR_OP_DEVICE_UNSUPPORTED	Operation is not supported by the device
QMI_ERR_ACCESS_DENIED	Operation cannot be performed because the UIM cannot be accessed

6.2.14.3. Description of QMI_DMS_UIM_GET_PIN_STATUS REQ/RESP

This command returns the status of PIN1 and PIN2.

This command is deprecated. QMI_UIM_GET_CARD_STATUS is the equivalent command in the QMI_UIM service and should be used in its place

6.2.15. QMI_DMS_GET_DEVICE_HARDWARE_REV

Queries the hardware revision of the device.

- **DMS message ID**

0x002C

- **Version introduced**

Major - 1, Minor - 2

6.2.15.1. Request - QMI_DMS_GET_DEVICE_HARDWARE_REV_REQ

- **Message type**

Request

- **Sender**

Control point

- **Mandatory TLVs**

None

- **Optional TLVs**

None

6.2.15.2. Response - QMI_DMS_GET_DEVICE_HARDWARE_REV_RESP

- **Message type**

Response

- **Sender**

Service

- **Mandatory TLVs**

The Result Code TLV (defined in Section 6.1.3.1) is always present in the response

Name	Version introduced	Version last modified
Hardware Revision	Unknown	1.2

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Hardware Revision
Length	Var			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	string	hardware_rev	Var	String containing the hardware revision of the device.

Optional TLVs

None

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

6.2.15.3. Description of QMI_DMS_GET_DEVICE_HARDWARE_REVREQ/RESP

This command queries the hardware revision of the device that returns an extension of the MSM version.

6.2.16. QMI_DMS_GET_OPERATING_MODE

Queries the current operating mode of the device.

- **DMS message ID**

0x002D

- **Version introduced**

Major - 1, Minor - 2

6.2.16.1. Request - QMI_DMS_GET_OPERATING_MODE_REQ

- **Message type**

Request

- **Sender**

Control point

- **Mandatory TLVs**

None

- **Optional TLVs**

None

6.2.16.2. Response - QMI_DMS_GET_OPERATING_MODE_RESP

- **Message type**

Response

- **Sender**

Service

- **Mandatory TLVs**

The Result Code TLV (defined in Section 6.1.3.1) is always present in the response.

Name	Version introduced	Version last modified
Operating Mode	1.2	1.12

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Operating Mode
Length	1			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	enum8	operating_mode	1	Selected operating mode. Values: • 0 – Online • 1 – Low power • 2 – Factory Test mode • 3 – Offline • 4 – Resetting • 5 – Shutting down • 6 – Persistent low power • 8 – Conducting network test for GSM/WCDMA

Optional TLVs

Name	Version introduced	Version last modified
Offline Reason	Unknown	1.6
Hardware-Restricted Mode	Unknown	1.6

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Offline Reason
Length	2			2	
Value	→	mask16	offline_reason	2	Offline reason bitmask. All unlisted bits are reserved for future use and are ignored. Values: • 0x0001 – Host image misconfiguration • 0x0002 – PRI image misconfiguration • 0x0004 – PRI version incompatible • 0x0008 – Device memory is full, cannot copy PRI information
Type	0x11			1	Hardware-Restricted Mode
Length	1			2	
Value	→	boolean	hardware_controlled_mode	1	Hardware-Restricted mode. Values: • 0x01 – TRUE

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

6.2.16.3. Description of QMI_DMS_GET_OPERATING_MODE REQ/RESP

This command queries the current operating mode of the device. The following operating modes are available:

Online – Indicates that the device can acquire a system and make calls
Low Power – Lowest power consumption state from which the device can return to Online mode; indicates that the device has temporarily disabled RF
Persistent Low Power – Same as Low Power mode, but persists even if the device is reset
Factory Test – Special mode for manufacturer use
Offline – Phone has deactivated RF and partially shutdown; the device must be power cycled before it can reacquire service from this mode
Resetting – Device is in the process of power cycling
Shutting Down – Device is in the process of shutting down
Device is conducting a network test for GSM/WCDMA. This mode cannot be set by clients

If the operating mode returned is Offline, an optional Offline Reason TLV is provided indicating the cause of the current state. If the offline reason is not known, the TLV is omitted.

If the current operating mode was set due to a hardware override, the optional Hardware-Restricted Mode

6.2.17. QMI_DMS_SET_OPERATING_MODE

Sets the operating mode of the device.

- **DMS message ID**

0x002E

- **Version introduced**

Major - 1, Minor - 2

6.2.17.1. Request - QMI_DMS_SET_OPERATING_MODE_REQ

- **Message type**

Request

- **Sender**

Control point

- **Mandatory TLVs**

Name	Version introduced	Version last modified
Operating Mode	Unknown	1.6

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Operating Mode
Length	1			2	
Value	→	enum8	operating_mode	1	Selected operating mode. Values: • 0 – Online • 1 – Low power • 2 – Factory Test mode • 3 – Offline • 4 – Resetting • 5 – Shutting down • 6 – Persistent low power • 7 – Mode-only low power

Optional TLVs

None

6.2.17.2. Response - QMI_DMS_SET_OPERATING_MODE_RESP

- **Message type**

Response

- **Sender**

Service

- **Mandatory TLVs**

The Result Code TLV (defined in Section 6.1.3.1) 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_DEVICE_IN_USE	Device is in use (i.e., in a call)
QMI_ERR_INVALID_ARG	Selected operating mode is invalid
QMI_ERR_INVALID_TRANSITION	Selected operating mode transition from the current operating mode is invalid
QMI_ERR_HARDWARE_RESTRICTED	Selected operating mode is invalid with the current wireless disable setting

6.2.17.3. Description of QMI_DMS_SET_OPERATING_MODE REQ/RESP

This command transitions operating modes based on the current mode of the device, and the mode selected.

Valid transitions include:

Online to low power, persistent low power, factory test, offline, or shut down

Low power to online, persistent low power, offline, or shut down

Persistent low power to online, low power, offline or shut down

Factory test to online

Offline to reset

Only Low Power mode can be used to change the device to Low Power mode, but does not modify the Persistent Low Power mode setting. If the device is not in Persistent Low Power mode, mode-only requests change the device to Low Power mode. If the device is already in Persistent Low Power mode, mode-only requests have no effect on the current mode.

NOTE

When in Persistent Low Power mode, only transitions to Online or regular Low Power mode cause the board to go online. Transitions to Offline (then Reset) and Shut Down power cycle the device, but upon startup, the device remains in Persistent Low Power mode.

Specifying an operating mode that is not in the valid range for the device elicits a QMI_ERR_INVALID_ARG error.

Specifying an operating mode that results in a transition not listed above elicits a QMI_ERR_INVALID_TRANSITION error.

For devices that allow hardware-controlled operating mode, it is possible that the current operating mode is enforced due to a hardware control. Changing the current operating mode to selected modes can be restricted by this hardware control, and any such requests elicit a QMI_ERR_HARDWARE_RESTRICTED error.

6.2.18. QMI_DMS_GET_TIME

Queries the current time of the device.

- **DMS message ID**

0x002F

- **Version introduced**

Major - 1, Minor - 3

6.2.18.1. Request - QMI_DMS_GET_TIME_REQ

- **Message type**

Request

- **Sender**

Control point

- **Mandatory TLVs**

None

- **Optional TLVs**

None

6.2.18.2. Response - QMI_DMS_GET_TIME_RESP

- **Message type**

Response

- **Sender**

Service

- **Mandatory TLVs**

The Result Code TLV (defined in Section 6.1.3.1) is always present in the response.

Name	Version introduced	Version last modified
Device Time	Unknown	1.3

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Device Time
Length	8			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	opaque	time_count	6	Count of 1.25 ms that have elapsed from the start of GPS Epoch time (January 6, 1980). A 6-byte integer in little-endian format.
		enum16	time_source	2	Source of the timestamp. Values: <ul style="list-style-type: none">• 0 – 32 kHz device clock• 1 – CDMA network• 2 – HDR network

Optional TLVs

Name	Version introduced	Version last modified
System Time in Milliseconds	Unknown	1.4
User Time in Milliseconds	Unknown	1.4

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	System Time in Milliseconds
Length	8			2	
Value	→	uint64	sys_time_in_ms	8	Count of system time in milliseconds that have elapsed from the start of GPS Epoch time (Jan 6, 1980).
Type	0x11			1	User Time in Milliseconds
Length	8			2	
Value	→	uint64	user_time_in_ms	8	Count of user time in milliseconds that have elapsed from the start of GPS Epoch time (Jan 6, 1980).

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

6.2.18.3. Description of QMI_DMS_GET_TIME REQ/RESP

This command queries the current time of the device and returns a count of 1.25 ms that have elapsed since Jan 6, 1980 (start of GPS Epoch time), or uptime if a valid timestamp is not available.

The source of the timestamp specifies how the timestamp was determined. If a network is active for the device, the first active time source is returned. Otherwise, if no networks are active, the 32 kHz slow-clock of the device is used.

If the slow-clock on the device has never been set, or if it was set but the device remained without power for an extended period of time, an accurate timestamp is not available. If valid time cannot be returned, the device returns a timestamp that is the uptime since the device has been powered. This semi-persistent uptime must not be used as a valid GPS timestamp.

The order of precedence for networks searched is:

CDMA

HDR

The System Time in Milliseconds TLV returns the count of milliseconds that have elapsed since Jan 6, 1980 (start of GPS Epoch time).

If the user time is available from the device, the User Time in Milliseconds TLV is also included in the response.

6.2.19. QMI_DMS_GET_PRL_VER

Queries the version of the active Preferred Roaming List (PRL) of the device.

- **DMS message ID**

0x0030

- **Version introduced**

Major - 1, Minor - 3

6.2.19.1. Request - QMI_DMS_GET_PRL_VER_REQ

- **Message type**

Request

- **Sender**

Control point

- **Mandatory TLVs**

None

- **Optional TLVs**

None

6.2.19.2. Response - QMI_DMS_GET_PRL_VER_RESP

- **Message type**

Response

- **Sender**

Service

- **Mandatory TLVs**

The Result Code TLV (defined in Section 6.1.3.1) is always present in the response.

Name	Version introduced	Version last modified
PRL Version	Unknown	1.3

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	PRL Version
Length	2			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	uint16	prl_version	2	PRL version.

Optional TLVs

Name	Version introduced	Version last modified
PRL-Only Preference	Unknown	1.5

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	PRL-Only Preference
Length	1			2	
Value	→	boolean	prl_only	1	Values: • 0 – Unset • 1 – Set

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_INVALID_ARG	Active PRL is invalid
QMI_ERR_INFO_UNAVAILABLE	Information is not available

6.2.19.3. Description of QMI_DMS_GET_PRL_VER REQ/RESP

This command queries the currently active PRL version of the device. This is valid only for CDMA devices (i.e., devices containing a PRL). An optional PRL-Only Preference TLV can be included in the response with information regarding whether the device is configured to register only on networks listed in the PRL (PRL-only preference).

Requesting the version of a PRL when the active PRL is invalid elicits a QMI_ERR_INVALID_ARG error.

Requests to read the PRL version on a UIM containing multiple sessions elicit a QMI_ERR_INFO_UNAVAILABLE error. Control points can either use the QMI_UIM service or QMI_DMS_GET_CURRENT_PRL_INFO (added in version 1.9) to retrieve the PRL version when this error is returned.

6.2.20. QMI_DMS_UIM_GET_ICCID

Queries the Integrated Circuit Card ID (ICCID) of the UIM for the device. (Deprecated)

- **DMS message ID**

0x003C

- **Version introduced**

Major - 1, Minor - 3

6.2.20.1. Request - QMI_DMS_UIM_GET_ICCID_REQ

- **Message type**

Request

- **Sender**

Control point

- **Mandatory TLVs**

None

- **Optional TLVs**

None

6.2.20.2. Response - QMI_DMS_UIM_GET_ICCID_RESP

- **Message type**

Response

- **Sender**

Service

- **Mandatory TLVs**

The Result Code TLV (defined in Section 6.1.3.1) is always present in the response.

Name	Version introduced	Version last modified
UIM ICCID	Unknown	1.3

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	UIM ICCID
Length	Var			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	string	uim_id	Var	String containing the UIM ICCID.

Optional TLVs

None

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_OP_DEVICE_UNSUPPORTED	Device does not support UIM
QMI_ERR_UIM_NOT_INITIALIZED	ID is not yet initialized because the UIM initialization has not finished; try UIM operation later

6.2.20.3. Description of QMI_DMS_UIM_GET_ICCID REQ/RESP

This command queries the UIM ICCID for the device if a UIM is present.

This command is deprecated. QMI_UIM_READ_TRANSPARENT (EF ID: 3F00 2FE2) is the equivalent command in the QMI_UIM service and should be used in its place.

6.2.21. QMI_DMS_UIM_GET_IMSI

Queries the International Mobile Station Identity (IMSI) of the UIM for the device. (Deprecated)

- **DMS message ID**

0x0043

- **Version introduced**

Major - 1, Minor - 3

6.2.21.1. Request - QMI_DMS_UIM_GET_IMSI_REQ

- **Message type**

Request

- **Sender**

Control point

- **Mandatory TLVs**

None

- **Optional TLVs**

None

6.2.21.2. Response - QMI_DMS_UIM_GET_IMSI_RESP

- **Message type**

Response

- **Sender**

Service

- **Mandatory TLVs**

The Result Code TLV (defined in Section 6.1.3.1) is always present in the response.

Name	Version introduced	Version last modified
International Mobile Subscriber ID	Unknown	1.3

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	International Mobile Subscriber ID
Length	Var			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	string	imsi	Var	String containing the international mobile subscriber ID.

Optional TLVs

None

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_OP_DEVICE_UNSUPPORTED	Device does not support UIM
QMI_ERR_UIM_NOT_INITIALIZED	ID is not yet initialized, because the UIM initialization has not finished; try the UIM operation later

6.2.21.3. Description of QMI_DMS_UIM_GET_IMSI REQ/RESP

This command queries the IMSI for the device, if it is available in the device provisioning.

This command is deprecated. QMI_UIM_READ_TRANSPARENT (EF ID: 3F00 7F20 6F07 for 2G card and 3F00 7FFF 6F07 for 3G card) is the equivalent command in the QMI_UIM service and should be used in its place.

6.2.22. QMI_DMS_UIM_GET_STATE

Queries the state of the UIM. (Deprecated)

- DMS message ID

0x0044

- Version introduced

Major - 1, Minor - 3

6.2.22.1. Request - QMI_DMS_UIM_GET_STATE_REQ

- **Message type**

Request

- **Sender**

Control point

- **Mandatory TLVs**

None

- **Optional TLVs**

None

6.2.22.2. Response - QMI_DMS_UIM_GET_STATE_RESP

- **Message type**

Response

- **Sender**

Service

- **Mandatory TLVs**

The Result Code TLV (defined in Section 6.1.3.1) is always present in the response.

Name	Version introduced	Version last modified
UIM State	Unknown	1.3

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	UIM State
Length	1			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	enum8	uim_state	1	UIM state. Values: • 0x00 – UIM initialization completed • 0x01 – UIM is locked or the UIM failed • 0x02 – UIM is not present • 0x03 – Reserved • 0xFF – UIM state is currently unavailable

Optional TLVs

None

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_OP_DEVICE_UNSUPPORTED	Operation is not supported by the device

6.2.22.3. Description of QMI_DMS_UIM_GET_STATE REQ/RESP

This command queries the state of the UIM for the device.

Requests to query the state when the message is not supported by the device elicit a QMI_ERR_OP_DEVICE_UNSUPPORTED error.

This command is deprecated. QMI_UIM_GET_CARD_STATUS is the equivalent command in the QMI_UIM service and should be used in its place.

6.2.23. QMI_DMS_GET_BAND_CAPABILITY

Queries the band capability of the device.

- DMS message ID

0x0045

- Version introduced

Major - 1, Minor - 3

6.2.23.1. Request - QMI_DMS_GET_BAND_CAPABILITY_REQ

- **Message type**

Request

- **Sender**

Control point

- **Mandatory TLVs**

None

- **Optional TLVs**

None

6.2.23.2. Response - QMI_DMS_GET_BAND_CAPABILITY_RESP

- **Message type**

Response

- **Sender**

Service

- **Mandatory TLVs**

The Result Code TLV (defined in Section 6.1.3.1) is always present in the response. This TLV is present if the result code is QMI_RESULT_SUCCESS.

Name	Version introduced	Version last modified
Band Capability	Unknown	1.3

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Band Capability
Length	8			2	
Value	→	mask	band_capability	8	Bitmask of bands supported by the device; see Appendix A for the definition of these values.

Optional TLVs

Name	Version introduced	Version last modified
LTE Band Capability	Unknown	1.21
TDS Band Capability	1.10	1.10

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	LTE Band Capability
Length	8			2	
Value	→	mask	lte_band_capability	8	This TLV is present on devices that support LTE bands. Bitmask of LTE bands supported by the device; see Appendix B for the definition of these values.
Type	0x11			1	TDS Band Capability
Length	8			2	
Value	→	mask	tds_band_capability	8	This TLV is present on devices that support TDS bands. Bitmask of TDS bands supported by the device. Values: <ul style="list-style-type: none">• Bit 0 – TDS Band A 1900 to 1920 MHz, 2010 to 2020 MHz• Bit 1 – TDS Band B 1850 to 1910 MHz, 1930 to 1990 MHz• Bit 2 – TDS Band C 1910 to 1930 MHz• Bit 3 – TDS Band D 2570 to 2620 MHz• Bit 4 – TDS Band E 2300 to 2400 MHz• Bit 5 – TDS Band F 1880 to 1920 MHz

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

6.2.23.3. Description of QMI_DMS_GET_BAND_CAPABILITY REQ/RESP

This command queries the band capability of the device.

6.2.24. QMI_DMS_GET_FACTORY_SKU

Queries the factory provisioned Stock Keeping Unit (SKU).

- **DMS message ID**

0x0046

- **Version introduced**

Major - 1, Minor - 6

6.2.24.1. Request - QMI_DMS_GET_FACTORY_SKU_REQ

- **Message type**

Request

- **Sender**

Control point

- **Mandatory TLVs**

None

- **Optional TLVs**

None

6.2.24.2. Response - QMI_DMS_GET_FACTORY_SKU_RESP

- **Message type**

Response

- **Sender**

Service

- **Mandatory TLVs**

The Result Code TLV (defined in Section 6.1.3.1) is always present in the response.

Name	Version introduced	Version last modified
Factory SKU	Unknown	1.6

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Factory SKU
Length	Var			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	string	factory_serial_number	Var	Factory serial number string in ASCII format (maximum 128 bytes).

Optional TLVs

None

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_OP_DEVICE_UNSUPPORTED	Operation is not supported by the device

6.2.24.3. Description of QMI_DMS_GET_FACTORY_SKU REQ/RESP

This command queries the SKU provisioned in the device by the factory.

Requests to query the serial number when no value has been provisioned elicit a QMI_ERR_NOT_PROVISIONED error.

6.2.25. QMI_DMS_GET_SW_VERSION

Queries the software version from the device.

- DMS message ID

0x0051

- Version introduced

Major - 1, Minor - 5

6.2.25.1. Request - QMI_DMS_GET_SW_VERSION_REQ

- **Message type**

Request

- **Sender**

Control point

- **Mandatory TLVs**

None

- **Optional TLVs**

None

6.2.25.2. Response - QMI_DMS_GET_SW_VERSION_RESP

- **Message type**

Response

- **Sender**

Service

- **Mandatory TLVs**

The Result Code TLV (defined in Section 6.1.3.1) is always present in the response.

Name	Version introduced	Version last modified
Software Version Information	Unknown	1.5

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Software Version Information
Length	Var			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	string	sw_version	Var	String representing the software version information.

Optional TLVs

None

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_NOT_PROVISIONED	Device could not retrieve the requested data because there is no provision in the device.

6.2.25.3. Description of QMI_DMS_GET_SW_VERSION REQ/RESP

This command queries the software version information of the device, which is represented by a string of characters.

7 Network Access Service (QMI_NAS)

QMI_NAS provides applications running on a host PC with commands related to network access:

- Signal strength
- Network registration and attach
- Serving system
- Network scan
- Home, preferred, and forbidden networks

It is expected that user-level applications, e.g., connection managers and/or device drivers on the Terminal Equipment (TE), use QMI_NAS to access this functionality on the MSM™ device.

7.1. Theory of Operation

7.1.1. Generalized QMI Service Compliance

The QMI_NAS 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 80-VB816-1. Extensions to the generalized QMI service theory of operation are noted in subsequent sections of this chapter.

7.1.2. NAS Service Type

NAS is assigned QMI service type 0x03.

7.1.3. Message Definition Template

7.1.3.1. Response Message Result TLV

This Type-Length-Value (TLV) is present in all Response messages defined in this document. It is not present in the Indication messages.

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
			qmi_error	2	Error code – Possible error code values are described in the error codes section of each message definition

7.1.4. QMI_NAS Fundamental Concepts

The QMI_NAS service provides NAS to its control points. These services include interfaces to control registration, attachment, and network selection performed by the device, as well as interfaces to obtain status information regarding the visible and serving networks.

To get service, the MSM device must register with a network and select the type of service it desires. The registration procedure is performed to notify the network of the mobile's presence and to validate that the user is allowed to use the network. The control point can select Automatic Registration mode, in which the device chooses the network with which to register. The control point can also select Manual Registration mode, in which it can specify a particular PLMN (MCC + MNC) with which to register. Note that the concept of user-driven manual registration is defined only in the 3GPP wireless standard. In 3GPP2 standards, the device always operates in Automatic Registration mode.

QMI_NAS also allows control points to perform a 3GPP network scan to discover the 3GPP networks that are currently visible to the device. The control point can then use this information to select a network for manual registration.

In the 3GPP wireless standard, the device must be attached to a service domain when it is registered on a network. This is a way to identify to the network which services may be used by the device during its registration. Service domains include Packet-Switched (PS) and Circuit-Switched (CS) data services. QMI control points can control this registration, network selection, and service domain attachment using QMI_NAS. The control points can also query the home network of the device. The home network of the device includes the MCC and MNC derived from the IMSI.

In the 3GPP wireless standard, there is a list of preferred and forbidden networks stored on a UIM, such as a SIM. The preferred networks list is a list of networks which the device prefers to register to in priority order. During automatic registration, the device gives preference to the listed networks over other visible

networks.

The forbidden networks list is a list of networks with which the device will not register.

QMI_NAS enables the control point to query and update these preferred and forbidden network lists.

QMI_NAS control points can also learn the network providing service and details of that service provided to the device. This includes the registration state, available service domains, registered network, and the radio technology in use.

A wireless device obtains a number of services, e.g., voice service and IP data service, via a radio that may act in accordance with different wireless standards. The radio technology indicates which wireless standard is currently in use by the device.

QMI control points may wish to monitor the signal strength measured by the device. Generally, the control point can obtain the above information via a polling mechanism (Request and Response messages). The signal strength change can also be reported via asynchronous indications. The control point can register signal strength thresholds. An asynchronous indication is sent when the current signal strength crosses one of the thresholds registered by the control point. These event-reporting settings registered by the control point are stored in the control point's service state variables. The Reset message can be used to clear these settings, restoring them to their default values.

7.1.5. Service State Variables

7.1.5.1. Shared State Variables

No QMI_NAS state variables are shared across control points.

7.1.5.2. State Variables Per Control Point

Name	Description	Possible values	Default value
report_signal_strength	Whether a change in signal strength is reported to a control point	• FALSE • TRUE	FALSE
report_signal_strength_threshold_list	Sequence of thresholds delimiting signal strength bands; threshold is a signed 1 byte value	-128 to +127	-128
report_rf_band_info	Whether a change in the radio interface is reported to a control point	• FALSE • TRUE	FALSE
report_reg_reject	Whether registration reject reasons are reported to a control point	• FALSE • TRUE	FALSE
report_rssi	Whether a change in RSSI is reported to a control point	• FALSE • TRUE	FALSE
rssi_delta	RSSI delta; an unsigned 1 byte value	0 to 255	N/A
report_ecio	Whether a change in ECIO is reported to a control point	• FALSE • TRUE	FALSE
ecio_delta	ECIO delta; an unsigned 1 byte value	0 to 255	N/A
report_io	Whether a change in IO is reported to a control point	• FALSE • TRUE	FALSE
io_delta	IO delta; an unsigned 1 byte value	0 to 255	N/A
report_sinr	Whether a change in SINR is reported to a control point	• FALSE • TRUE	FALSE
sinr_delta	SINR delta; an unsigned 1 byte value	0 to 255	N/A
report_rsrq	Whether a change in RSRQ is reported to a control point	• FALSE • TRUE	FALSE
rsrq_delta	RSRQ delta; an unsigned 1 byte value	5	N/A
report_lte_snr	Whether a change in LTE SNR is reported to a control point	• FALSE • TRUE	FALSE
lte_snr_delta	LTE SNR delta; an unsigned 2 byte value	0 to 255	N/A
report_lte_rsrp	Whether a change in LTE RSRP is reported to a control point	• FALSE • TRUE	FALSE
lte_rsrp_delta	LTE RSRP delta; an unsigned 1 byte value	0 to 255	N/A
req_serving_system	Whether serving system events are reported to a control point	• FALSE • TRUE	TRUE
reg_sys_sel_pref	Whether system selection preferences are reported to a control point	• FALSE • TRUE	FALSE
reg_ddtm_events	Whether DDTM events are reported to a control point	• FALSE • TRUE	FALSE

7.2. QMI_NAS Messages

Command	ID	Description
QMI_NAS_RESET	0x0000	Resets the NAS service state variables of the requesting control point.
QMI_NAS_ABORT	0x0001	Aborts a previously issued QMI_NAS command.
QMI_NAS_SET_EVENT	0x0002	Sets the NAS state reporting conditions for the requesting control point.
QMI_NAS_SET_EVENT_REPORT		
QMI_NAS_EVENT	0x0002	Indicates the NAS state change.(Deprecated)
QMI_NAS_EVENT_REPORT_IND	Indicates	
QMI_NAS_SET_REG_EVENT	0x0003	Sets the registration state for different QMI_NAS indications for the requesting control point.
QMI_NAS_INDICATION_REGISTER		
QMI_NAS_GET_RSSI	0x0020	Queries the current signal strength as measured by the device. (Deprecated)
QMI_NAS_GET_SIGNAL_STRENGTH		
QMI_NAS_SCAN_NETS	0x0021	Performs a scan for visible networks.
QMI_NAS_PERFORM_NETWORK_SCAN		
QMI_NAS_REGISTER_NET	0x0022	Initiates a network registration.
QMI_NAS_INITIATE_NETWORK_REGISTER		
QMI_NAS_ATTACH_DETACH	0x0023	Initiates a domain attach or detach action. (Deprecated)
QMI_NAS_INITIATE_ATTACH		
QMI_NAS_GET_SS_INFO	0x0024	Queries information regarding the system that currently provides service.
QMI_NAS_GET_SERVING_SYSTEM		
QMI_NAS_SS_INFO_IND	0x0024	Indicates a change in the current serving
QMI_NAS_SERVING_SYSTEM_IND	indication	system registration state and/or radio technology.
QMI_NAS_SET_SYSTEM_SELECTION_PREF	0x0033	Sets the different system selection preferences of the device.
QMI_NAS_SET_SYSTEM_SELECTION_PREFERENCE		
QMI_NAS_GET_SYSTEM_SELECTION_PREF	0x0034	Queries the different system selection preferences of the device.
QMI_NAS_GET_SYSTEM_SELECTION_PREFERENCE		

7.2.1.1. QMI_NAS_RESET

Resets the NAS service state variables of the requesting control point.

- **NAS message ID**

0x0000

- **Version introduced**

Major - 1, Minor - 0

7.2.1.2. Request - QMI_NAS_RESET_REQ_MSG

- **Message type**

Request

- **Sender**

Control point

- **Mandatory TLVs**

None

- **Optional TLVs**

None

7.2.1.3. Response - QMI_NAS_RESET_RESP_MSG

- **Message type**

Response

- **Sender**

Service

- **Mandatory TLVs**

The Result Code TLV (defined in Section 7.1.3.1) is always present in the response.

- **Optional TLVs**

None

Error codes

<code>QMI_ERR_NONE</code>	No error in the request
<code>QMI_ERR_INTERNAL</code>	Unexpected error occurred during processing
<code>QMI_ERR_MALFORMED_MSG</code>	Message was not formulated correctly by the control point or the message was corrupted during transmission

7.2.1.4. Description of QMI_NAS_RESET REQ/RESP

This command resets the issuing control point state (see Section 7.1.5.2) kept by the service. As a result, each shared state variable may change, depending on its arbitration policy (see Section 7.1.5.1).

This is equivalent to closing the service and reopening it again, although it is performed as one operation and, hence, the client ID of the requesting control point does not change. The control point state variables change to their default values before the response is issued.

7.2.2. QMI_NAS_ABORT

Aborts a previously issued QMI_NAS command.

- **NAS message ID**

0x0001

- **Version introduced**

Major - 1, Minor - 0

7.2.2.1. Request - QMI_NAS_ABORT_REQ_MSG

- **Message type**

Request

- **Sender**

Control point

- Mandatory TLVs

Name	Version introduced	Version last modified
TX_ID	Unknown	1.0

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	TX_ID
Length	2			2	
Value	→	uint16	tx_id	2	Transaction ID of the request to be aborted.

- Optional TLVs

None

7.2.2.2. Response - QMI_NAS_ABORT_RESP_MSG

- Message type

Response

- Sender

Service

- Mandatory TLVs

The Result Code TLV (defined in Section 7.1.3.1) 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_MISSING_ARG	One or more required TLVs were missing in the request
QMI_ERR_INVALID_TX_ID	Transaction ID supplied in the request does not match any pending transaction; i.e., either the transaction was not received or it is already executed by the device
QMI_ERR_UNABORTABLE_TRANSACTION	Specified transaction could not be aborted; none of the requests in the transaction were abortable

7.2.2.3. Description of QMI_NAS_ABORT REQ/RESP

This command aborts a previously issued QMI_NAS command. It is useful for requests that take a long time to execute, in the case where the user is no longer interested in the result.

The following QMI_NAS messages can be aborted:

QMI_NAS_PERFORM_NETWORK_SCAN_REQ

QMI_NAS_PERFORM_INCREMENTAL_NETWORK_SCAN_REQ

7.2.3. QMI_NAS_SET_EVENT_REPORT

Sets the NAS state reporting conditions for the requesting control point. (Deprecated)

- **NAS message ID**

0x0002

- **Version introduced**

Major - 1, Minor - 0

7.2.3.1. Request - QMI_NAS_SET_EVENT_REPORT_REQ_MSG

- **Message type**

Request

- **Sender**

Control point

- **Mandatory TLVs**

None

- **Optional TLVs**

At least one of the following optional TLVs must be included in this request.

Name	Version introduced	Version last modified
Signal Strength Indicator	Unknown	1.0
RF Band Information	Unknown	1.1
Registration Reject Reason**	Unknown	1.1
RSSI Indicator	Unknown	1.1
ECIO Indicator	Unknown	1.1
IO Indicator*	Unknown	1.1
SINR Indicator*	Unknown	1.1
Error Rate Indicator	Unknown	1.1
RSRQ Indicator*	Unknown	1.3
ECIO Threshold	Unknown	1.7
SINR Threshold	Unknown	1.7
LTE SNR Delta	1.15	1.40
RSRP Delta	1.15	1.15

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Signal Strength Indicator
Length	Var			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	boolean	report_signal_strength	1	Values: • 0 – Do not report • 1 – Report
		uint8	num_signal_strength_thresholds	1	Number of sets of the following elements: • report_signal_strength_threshold_list
		int8	report_signal_strength_threshold_list	Var	A sequence of thresholds delimiting signal strength Var bands. Each threshold specifies the signal strength (in dBm) at which an event report indication, including the current signal strength, will be sent to the requesting control point. Threshold is a signed 1 byte value. Valid values: -128 dBm to +127 dBm.
Type	0x11			1	RF Band Information
Length	1			2	
Value	→	boolean	report_rf_band_info	1	Values: • 0 – Do not report • 1 – Report
Type	0x12			1	Registration Reject Reason**
Length	1			2	
Value	→	boolean	report_reg_reject	1	Values: • 0 – Do not report • 1 – Report
Type	0x13			1	RSSI Indicator
Length	2			2	
Value	→	boolean	report_rssi	1	Values: • 0 – Do not report • 1 – Report
		uint8	rssi_delta	1	RSSI delta (in dBm) at which an event report indication, including the current RSSI, will be sent to the requesting control point. RSSI delta is an unsigned 1 byte value.
Type	0x14			1	ECIO Indicator
Length	2			2	
Value	→	boolean	report_ecio	1	Values: • 0 – Do not report • 1 – Report
		uint8	ecio_delta	1	ECIO delta at which an event report indication, ecio_delta including the current ECIO, will be sent to the requesting control point. ECIO delta is an unsigned 1 byte value that increments in negative 0.5 dB, e.g., ecio_delta of 2 means a change of -1 dB.
Type	0x15			1	IO Indicator*
Length	2			2	
Value	→	boolean	report_io	1	Values: • 0 – Do not report • 1 – Report

Field	Field value	Field type	Parameter	Size (byte)	Description
		uint8	io_delta	1	IO delta (in dBm) at which an event report indication, io_delta including the current IO, will be sent to the requesting control point. IO delta is an unsigned 1 byte value.
Type	0x16			1	SINR Indicator*
Length	2			2	
Value	→	boolean	report_sinr	1	Values: • 0 – Do not report • 1 – Report
		uint8	sinr_delta	1	SINR delta level at which an event report indication, sinr_delta including the current SINR, will be sent to the requesting control point. SINR delta level is an unsigned 1 byte value.
Type	0x17			1	Error Rate Indicator
Length	1			2	
Value	→	boolean	report_error_rate	1	Values: • 0 – Do not report • 1 – Report
Type	0x18			1	RSRQ Indicator*
Length	2			2	
Value	→	boolean	report_rsrq	1	Values: • 0 – Do not report • 1 – Report
		uint8	rsrq_delta	1	RSRQ delta level at which an event report indication, including the current RSRQ, will be sent to the requesting control point. RSRQ delta level is an unsigned 1 byte value.
Type	0x19			1	ECIO Threshold
Length	Var			2	
Value	→	boolean	report_ecio	1	Values: • 0 – Do not report • 1 – Report
		uint8	threshold_list_len	1	Number of sets of the following elements: • threshold_list
		int16	threshold_list	Var	A sequence of thresholds delimiting ECIO event reporting bands. Every time a new ECIO value crosses a threshold value, an event report indication message with the new ECIO value is sent to the requesting control point. For this field: • Each threshold value is a signed 2 byte value • Maximum number of threshold values is 10 • At least one value must be specified (if report_ecio is set)
Type	0x1A			1	SINR Threshold
Length	Var			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	boolean	report_sinr	1	Values: • 0 – Do not report • 1 – Report
		uint8	threshold_list_len	1	Number of sets of the following elements: • threshold_list
		uint8	threshold_list	Var	A sequence of thresholds delimiting SINR event reporting bands. Every time a new SINR value crosses a threshold value, an event report indication message with the new SINR value is sent to the requesting control point. For this field: • Each threshold value will be an unsigned 1 byte value • Maximum number of threshold values is 5 • At least one value must be specified (if report_sinr is set)
Type	0x1B			1	LTE SNR Delta
Length	3			2	
Value	→	boolean	report_lte_snr	1	Values: • 0 – Do not report • 1 – Report
		uint16	lte_snr_delta	2	LTE SNR delta level at which an event report indication, including the current SNR, will be sent to the requesting control point. LTE SNR delta level is an unsigned 2 byte value, representing the delta in units of 0.1 dB, e.g., lte_snr_delta of 3 means a change 0.3 dB.
Type	0x1C			1	RSRP Delta
Length	2			2	
Value	→	boolean	report_lte_rsrp	1	Values: • 0 – Do not report • 1 – Report
		uint8	lte_rsrp_delta	1	LTE RSRP delta level at which an event report indication, including the current RSRP, will be sent to the requesting control point. LTE RSRP delta level is an unsigned 1 byte value, representing the delta in dB.

7.2.3.2. Response - QMI_NAS_SET_EVENT_REPORT_RESP_MSG

- Message type

Response

- Sender

Service

- **Mandatory TLVs**

The Result Code TLV (defined in Section 7.1.3.1) is always present in the response.

- **Optional TLVs**

None

- **Error codes**

<code>QMI_ERR_NONE</code>	No error in the request
<code>QMI_ERR_INTERNAL</code>	Unexpected error occurred during processing
<code>QMI_ERR_MALFORMED_MSG</code>	Message was not formulated correctly by the control point or the message was corrupted during transmission
<code>QMI_ERR_MISSING_ARG</code>	One or more required TLVs were missing in the request
<code>QMI_ERR_ARG_TOO_LONG</code>	More than the maximum allowed thresholds were specified
<code>QMI_ERR_NO_THRESHOLDS</code>	No thresholds were specified in an enable signal strength request

7.2.3.3. Description of QMI_NAS_SET_EVENT_REPORT REQ/RESP

The control point state variables that control event reporting are modified to reflect the settings indicated in the TLVs that are present in the request message. The service maintains a separate set of state variables for each control point. See Section 7.1.5.2 for a list of state variables and their explanations.

The control point learns of changes in state via the QMI_NAS_EVENT_REPORT_IND indication.

The AT command equivalents to this command are AT+CMER, AT+CIND, and AT+CIEV (refer to 3GPP TS 27.007).

This command is deprecated. Use QMI_NAS_CONFIG_SIG_INFO2 to configure signal strength reporting thresholds. Use QMI_NAS_INDICATION_REGISTER to register for QMI_NAS_SIG_INFO_IND, QMI_NAS_ERR_RATE_IND, and/or QMI_NAS_RF_BAND_INFO_IND messages.

7.2.4. QMI_NAS_EVENT_REPORT_IND

Indicates the NAS state change. (Deprecated)

- **NAS message ID**

0x0002

- **Version introduced**

Major - 1, Minor - 0

7.2.4.1. Indication - QMI_NAS_EVENT_REPORT_IND_MSG

- **Message type**

Indication

- **Sender**

Service

- **Scope**

Per control point (unicast)

- **Mandatory TLVs**

None

- **Optional TLVs**

At least one of the following optional TLVs must be included in this indication.

Name	Version introduced	Version last modified
Signal Strength	Unknown	1.3
RF Band Information List	Unknown	1.142
Registration Reject Reason**	Unknown	1.2
RSSI	Unknown	1.3
ECIO	Unknown	1.1
IO*	Unknown	1.1
SINR*	Unknown	1.1
Error Rate	Unknown	1.1
RSRQ**	Unknown	1.3
LTE SNR	Unknown	1.15
LTE RSRP	Unknown	1.15

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Signal Strength
Length	2			2	
Value	→	int8	sig_strength	1	<p>Received signal strength in dBm:</p> <ul style="list-style-type: none"> • For CDMA and UMTS, this indicates forward link pilot Ec • For GSM, this indicates received signal strength • For LTE, this indicates the total received wideband power observed by the UE
		enum8	radio_if	1	<p>Radio interface technology of the signal being measured. Values:</p> <ul style="list-style-type: none"> • 0x00 – RADIO_IF_NO_SVC – None (no service) • 0x01 – RADIO_IF_CDMA_1X – cdma2000® 1X • 0x02 – RADIO_IF_CDMA_1xEVDO – cdma2000® HRPD (1xEV-DO) • 0x03 – RADIO_IF_AMPS – AMPS • 0x04 – RADIO_IF_GSM – GSM • 0x05 – RADIO_IF_UMTS – UMTS • 0x08 – RADIO_IF_LTE – LTE
Type	0x11			1	RF Band Information List
Length	Var			2	
Value	→	uint8	num_instances	1	<p>Number of sets of the following elements:</p> <ul style="list-style-type: none"> • radio_if • active_band • active_channel
		enum8	radio_if	1	<p>Radio interface currently in use. Values:</p> <ul style="list-style-type: none"> • 0x01 – cdma2000® 1X • 0x02 – cdma2000® HRPD (1xEV-DO) • 0x03 – AMPS • 0x04 – GSM • 0x05 – UMTS • 0x08 – LTE • 0x09 – TD-SCDMA
		enum16	active_band	2	<p>Active band class (see Table A-1 for details). Values:</p> <ul style="list-style-type: none"> • 00 to 39 – CDMA band classes • 40 to 79 – GSM band classes • 80 to 91 – WCDMA band classes • 120 to 161 – LTE band classes • 200 to 205 – TD-SCDMA band classes
		uint16	active_channel	2	Active channel. If the channel is not relevant to the technology, a value of 0 is returned.
Type	0x12			1	Registration Reject Reason**
Length	3			2	
Value	→	enum8	service_domain	1	<p>Network service domain that was rejected. Possible values:</p> <ul style="list-style-type: none"> • 1 – CIRCUIT_SWITCHED • 2 – PACKET_SWITCHED • 3 – CIRCUIT_AND_PACKET_SWITCHED

Field	Field value	Field type	Parameter	Size (byte)	Description
		uint16	reject_cause	2	Reject cause; refer to 3GPP TS 24.008 Sections 10.5.3.6 and 10.5.5.14, and 3GPP TS 24.301 Section 9.9.3.9.
Type	0x13			1	RSSI
Length	2			2	
Value	→	uint8	rssi	1	<p>RSSI represented as a positive value; control points need to convert this to negative to get actual value in dBm:</p> <ul style="list-style-type: none"> • For CDMA and UMTS, this indicates forward link pilot Ec • For GSM, this indicates received signal strength
		enum8	radio_if	1	<p>Radio interface technology of the signal being measured. Values:</p> <ul style="list-style-type: none"> • 0x00 – RADIO_IF_NO_SVC – None (no service) • 0x01 – RADIO_IF_CDMA_1X – cdma2000® 1X • 0x02 – RADIO_IF_CDMA_1XEVDO – cdma2000® HRPD (1xEV-DO) • 0x03 – RADIO_IF_AMPS – AMPS • 0x04 – RADIO_IF_GSM – GSM • 0x05 – RADIO_IF_UMTS – UMTS • 0x08 – RADIO_IF_LTE – LTE
Type	0x14			1	ECIO
Length	2			2	
Value	→	uint8	ecio	1	ECIO value representing negative 0.5 dB increments, i.e., 2 means -1 dB (14 means -7 dB, 63 means -31.5 dB).
		enum8	radio_if	1	<p>Radio interface technology of the signal being measured. Values:</p> <ul style="list-style-type: none"> • 0x00 – RADIO_IF_NO_SVC – None (no service) • 0x01 – RADIO_IF_CDMA_1X – cdma2000® 1X • 0x02 – RADIO_IF_CDMA_1XEVDO – cdma2000® HRPD (1xEV-DO) • 0x03 – RADIO_IF_AMPS – AMPS • 0x04 – RADIO_IF_GSM – GSM • 0x05 – RADIO_IF_UMTS – UMTS
Type	0x15			1	IO*
Length	4			2	
Value	→	int32	io	4	Received IO in dBm. IO is only applicable for 1xEV-DO.
Type	0x16			1	SINR*
Length	1			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	enum8	sinr	1	<p>SINR level. SINR is only applicable for 1xEV-DO. Valid levels are 0 to 8, where the maximum value for:</p> <ul style="list-style-type: none"> • 0x00 – SINR_LEVEL_0 is -9 dB • 0x01 – SINR_LEVEL_1 is -6 dB • 0x02 – SINR_LEVEL_2 is -4.5 dB • 0x03 – SINR_LEVEL_3 is -3 dB • 0x04 – SINR_LEVEL_4 is -2 dB • 0x05 – SINR_LEVEL_5 is +1 dB • 0x06 – SINR_LEVEL_6 is +3 dB • 0x07 – SINR_LEVEL_7 is +6 dB • 0x08 – SINR_LEVEL_8 is +9 dB
Type	0x17			1	Error Rate
Length	3			2	
Value	→	uint16	error_rate	2	<p>Error rate value corresponds to the RAT that is currently registered.</p> <p>For CDMA, the error rate reported is Frame Error Rate:</p> <ul style="list-style-type: none"> • Valid error rate values between 1 and 10000 are returned to indicate percentage, e.g., a value of 300 means the error rate is 3% • A value of 0xFFFF indicates that the error rate is unknown or unavailable <p>For HDR, the error rate reported is Packet Error Rate:</p> <ul style="list-style-type: none"> • Valid error rate values between 1 and 10000 are returned to indicate percentage, e.g., a value of 300 means the error rate is 3% • A value of 0xFFFF indicates that the error rate is unknown or unavailable <p>For GSM, the error rate reported is Bit Error Rate:</p> <ul style="list-style-type: none"> • Valid values are 0, 100, 200, 300, 400, 500, 600, and 700 • The reported value divided by 100 gives the error rate as an RxQual value as defined in 3GPP TS 45.008 Section 8.2.4, e.g., a value of 300 represents an RxQual value of 3 • A value of 25500 indicates No Data <p>For WCDMA, the error rate reported is Block Error Rate (BLER):</p> <ul style="list-style-type: none"> • Valid values are 1 to 10000 • The reported value divided by 100 provides the error rate in percentages, e.g., a value of 300 represents a BLER of 3% • A value of 0 indicates No Data

Field	Field value	Field type	Parameter	Size (byte)	Description
		enum8	radio_if	1	Radio interface technology of the signal being measured. Values: • 0x00 – RADIO_IF_NO_SVC – None (no service) • 0x01 – RADIO_IF_CDMA_1X – cdma2000® 1X • 0x02 – RADIO_IF_CDMA_1XEVDO – cdma2000® HRPD (1xEV-DO) • 0x03 – RADIO_IF_AMPS – AMPS • 0x04 – RADIO_IF_GSM – GSM • 0x05 – RADIO_IF_UMTS – UMTS
Type	0x18			1	RSRQ**
Length	2			2	
Value	→	int8	rsrq	1	RSRQ value in dB (signed integer value). Range: -3 to -20 (-3 means -3 dB, -20 means -20 dB).
		uint8	radio_if	1	Radio interface technology of the signal being measured. Values: • 0x08 – LTE
Type	0x19			1	LTE SNR
Length	2			2	
Value	→	int16	snr	2	LTE SNR level as a scaled integer in units of 0.1 dB; e.g., -16 dB has a value of -160 and 24.6 dB has a value of 246.
Type	0x1A			1	LTE RSRP
Length	2			2	
Value	→	int16	rsrp	2	Current LTE RSRP in dBm as measured by L1. Range: -44 to -140 (-44 means -44 dBm, -140 means -140 dBm).

7.2.4.2. Description of QMI_NAS_EVENT_REPORT_IND

This unsolicited indication is sent by the service to interested control points when the device state corresponding to any TLV listed previously changes. Interested control points are those that previously registered, using the QMI_NAS_SET_EVENT_REPORT_REQ message, for the corresponding state to be reported.

The Signal Strength TLV is included in the indication if the control point report_signal_strength state variable is set and the current signal strength moves past a threshold specified by the control point, relative to the last value indicated to the control point.

The AT command equivalents to this command are **AT+CMER**, **AT+CIND**, and **AT+CIEV** (refer to 3GPP TS 27.007).

The RF Band Information List TLV is included in the indication if the control point report_rf_band_info state variable is set and the current radio band or channel changes on a network to which the device is

registered.

The Registration Reject Reason TLV is included in the indication if the control point report_reg_reject_reason state variable is set and a registration request is rejected by the network. Presence of this indication means that the network rejected a registration request, but not that the serving system registration was affected. If the serving system registration changes, a new serving system indication is sent to the control point .

The RSSI TLV is included in the indication if the control point report_rssi state variable is set, and the difference between the current RSSI and the last value indicated to the control point crosses the delta specified by the control point.

The ECIO TLV is included in the indication if the control point report_ecio state variable is set, and the difference between the current ECIO and the last value indicated to the control point crosses the delta specified by the control point.

The IO TLV is included in the indication if the control point report_io state variable is set, and the difference between the current IO and the last value indicated to the control point crosses the delta specified by the control point.

The SINR TLV is included in the indication if the control point report_sinr state variable is set, and the difference between the current SINR and the last value indicated to the control point crosses the delta specified by the control point.

The Error Rate TLV is included in the indication if the control point report_error_rate state 10 variable is set along with one or more of report_signal_strength, report_ecio, report_io, report_sinr state variables, and if an error rate is available to report along with one or more of the Signal Strength, RSSI, ECIO, IO, or SINR TLVs.

The RSRQ TLV is included in the indication if the control point report_rsrq state variable is set, and the difference between the current RSRQ and the last value indicated to the control point crosses the delta specified by the control point.

This indication is deprecated. Use QMI_NAS_SIG_INFO_IND for signal strength-related information and QMI_NAS_ERR_RATE_IND for error rate-related information.

7.2.5. QMI_NAS_INDICATION_REGISTER

Sets the registration state for different QMI_NAS indications for the requesting control point.

- **NAS message ID**

0x0003

- Version introduced

Major - 1, Minor - 1

7.2.5.1. Request - QMI_NAS_INDICATION_REGISTER_REQ_MSG

- Message type

Request

- Sender

Control point

- Mandatory TLVs

None

- Optional TLVs

Name	Version introduced	Version last modified
System Selection Preference	Unknown	1.1
DDTM Events	Unknown	1.1
Serving System Events	Unknown	1.2
Dual Standby Preference	Unknown	1.7
Subscription Info	Unknown	1.7
Network Time	Unknown	1.8
Sys Info	Unknown	1.8
Signal Strength	Unknown	1.8
Error Rate	Unknown	1.8
HDR New UATI Assigned	Unknown	1.9
HDR Session Closed	Unknown	1.9
Managed Roaming	Unknown	1.11
Current PLMN Name	Unknown	1.14
eMBMS Status	Unknown	1.16
RF Band Information	Unknown	1.19
Network Reject Information	Unknown	1.22
Operator Name Data	1.24	1.24
CSP PLMN Mode Bit	1.24	1.24
RTRE Configuration	1.25	1.25
IMS Preference Status	1.51	1.51
E911 State Ready Status	1.66	1.66

Name	Version introduced	Version last modified
LTE SIB16 Network Time	1.73	1.73
LTE Physical Carrier Aggregation Information	1.81	1.81
Subscription Change	1.93	1.93
Service-Specific Access Class Barring	1.98	1.98
T3402 Timer Value	1.114	1.114
Access Class Barring	1.119	1.119
Data Subscription Priority	1.121	1.121
T3346 Timer Status Change	1.128	1.128
Call Mode Status	1.128	1.128
Service-Specific Access Class Barring Ext	1.136	1.136
Manual Network Scan Failure	1.139	1.139
Timer Expiry	1.147	1.147
Emergency Mode Status	1.148	1.148

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	System Selection Preference
Length	1			2	
Value	→	boolean	reg_sys_sel_pref	1	Values: • 0x00 – Disable • 0x01 – Enable
Type	0x12			1	DDTM Events
Length	1			2	
Value	→	boolean	reg_ddtm_events	1	Values: • 0x00 – Disable • 0x01 – Enable
Type	0x13			1	Serving System Events
Length	1			2	
Value	→	boolean	req_serving_system	1	Values: • 0x00 – Disable • 0x01 – Enable
Type	0x14			1	Dual Standby Preference
Length	1			2	
Value	→	boolean	dual_standby_pref	1	Values: • 0x00 – Disable • 0x01 – Enable
Type	0x15			1	Subscription Info
Length	1			2	
Value	→	boolean	subscription_info	1	Values: • 0x00 – Disable • 0x01 – Enable
Type	0x17			1	Network Time
Length	1			2	
Value	→	boolean	reg_network_time	1	Values: • 0x00 – Disable • 0x01 – Enable

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x18			1	Sys Info
Length	1			2	
Value	→	boolean	sys_info	1	Values: • 0x00 – Disable • 0x01 – Enable
Type	0x19			1	Signal Strength
Length	1			2	
Value	→	boolean	sig_info	1	Values: • 0x00 – Disable • 0x01 – Enable
Type	0x1A			1	Error Rate
Length	1			2	
Value	→	boolean	err_rate	1	Values: • 0x00 – Disable • 0x01 – Enable
Type	0x1B			1	HDR New UATI Assigned
Length	1			2	
Value	→	boolean	reg_hdr_uati	1	Controls the reporting of QMI_NAS_HDR_UATI_UPDATE_IND. Values: • 0x00 – Disable (default value) • 0x01 – Enable
Type	0x1C			1	HDR Session Closed
Length	1			2	
Value	→	boolean	reg_hdr_session_close	1	Controls the reporting of QMI_NAS_HDR_SESSION_CLOSE_IND. Values: • 0x00 – Disable (default value) • 0x01 – Enable
Type	0x1D			1	Managed Roaming
Length	1			2	
Value	→	boolean	reg_managed_roaming	1	Controls the reporting of QMI_NAS_MANAGED_ROAMING_IND. Values: • 0x00 – Disable (default value) • 0x01 – Enable
Type	0x1E			1	Current PLMN Name
Length	1			2	
Value	→	boolean	reg_current_plmn_name	1	Controls the reporting of QMI_NAS_CURRENT_PLMN_NAME_IND. Values: • 0x00 – Disable (default value) • 0x01 – Enable
Type	0x1F			1	eMBMS Status
Length	1			2	
Value	→	boolean	reg_embms_status	1	Controls the reporting of QMI_NAS_EMBMS_STATUS_IND. Values: • 0x00 – Disable (default value) • 0x01 – Enable
Type	0x20			1	RF Band Information

Field	Field value	Field type	Parameter	Size (byte)	Description
Length	1			2	
Value	→	boolean	reg_rf_band_info	1	Controls the reporting of QMI_NAS_RF_BAND_INFO_IND. Values: • 0x00 – Disable (default value) • 0x01 – Enable
Type	0x21			1	Network Reject Information
Length	2			2	
Value	→	boolean	reg_network_reject	1	Controls the reporting of QMI_NAS_NETWORK_REJECT_IND. Values: • 0x00 – Disable (default value) • 0x01 – Enable
		boolean	suppress_sys_info	1	Controls the reporting of QMI_NAS_SYS_INFO_IND when only the reject_cause field has changed. Values: • 0x00 – Do not suppress (default value) • 0x01 – Suppress
Type	0x22			1	Operator Name Data
Length	1			2	
Value	→	boolean	reg_operator_name_data	1	Controls the reporting of QMI_NAS_OPERATOR_NAME_DATA_IND. Values: • 0x00 – Disable • 0x01 – Enable (default value)
Type	0x23			1	CSP PLMN Mode Bit
Length	1			2	
Value	→	boolean	reg_csp_plmn_mode_bit	1	Controls the reporting of QMI_NAS_CSP_PLMN_MODE_BIT_IND. Values: • 0x00 – Disable • 0x01 – Enable (default value)
Type	0x24			1	RTRE Configuration
Length	1			2	
Value	→	boolean	reg_rtre_cfg	1	Controls the reporting of QMI_NAS_RTRE_CONFIG_IND. Values: • 0x00 – Disable (default value) • 0x01 – Enable
Type	0x25			1	IMS Preference Status
Length	1			2	
Value	→	boolean	reg_ims_pref_status	1	Controls the reporting of QMI_NAS_IMS_PREF_STATUS_IND. Values: • 0x00 – Disable (default value) • 0x01 – Enable
Type	0x26			1	E911 State Ready Status
Length	1			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	boolean	reg_e911_state_ready_status	1	Controls the reporting of QMI_NAS_E911_STATE_READY_IND. Values: <ul style="list-style-type: none">• 0x00 – Disable (default value)• 0x01 – Enable
Type	0x27			1	LTE SIB16 Network Time
Length	1			2	
Value	→	boolean	reg_lte_sib16_network_time	1	Controls the reporting of QMI_NAS_LTE_SIB16_NETWORK_TIME_IND. Values: <ul style="list-style-type: none">• 0x00 – Disable (default value)• 0x01 – Enable
Type	0x28			1	LTE Physical Carrier Aggregation Information
Length	1			2	
Value	→	boolean	reg_lte_cphy_ca	1	Controls the reporting of QMI_NAS_LTE_CPHY_CA_IND. Values: <ul style="list-style-type: none">• 0x00 – Disable (default value)• 0x01 – Enable
Type	0x29			1	Subscription Change
Length	1			2	
Value	→	boolean	reg_subscription_change	1	Controls the reporting of QMI_NAS_SUBSCRIPTION_CHANGE_IND. Values: <ul style="list-style-type: none">• 0x00 – Disable (default value)• 0x01 – Enable
Type	0x2A			1	Service-Specific Access Class Barring
Length	1			2	
Value	→	boolean	reg_ssac_info	1	Controls the reporting of QMI_NAS_SSAC_INFO_IND. Values: <ul style="list-style-type: none">• 0x00 – Disable (default value)• 0x01 – Enable
Type	0x2B			1	T3402 Timer Value
Length	1			2	
Value	→	boolean	reg_emm_t3402_change	1	Controls the reporting of QMI_NAS_EMM_T3402_CHANGED_IND. Values: <ul style="list-style-type: none">• 0x00 – Disable (default value)• 0x01 – Enable
Type	0x2C			1	Access Class Barring
Length	1			2	
Value	→	boolean	reg_acb_info_change	1	Controls the reporting of QMI_NAS_ACB_INFO_IND. Values: <ul style="list-style-type: none">• 0x00 – Disable (default value)• 0x01 – Enable
Type	0x2D			1	Data Subscription Priority
Length	1			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	boolean	reg_data_subs_priority_change	1	Controls the reporting of QMI_NAS_DATA_SUBS_PRIORITY_IND. Values: <ul style="list-style-type: none">• 0x00 – Disable (default value)• 0x01 – Enable
Type	0x2E			1	T3346 Timer Status Change
Length	1			2	
Value	→	boolean	reg_t3346_timer_status_change	1	Controls the reporting of QMI_NAS_T3346_TIMER_STATUS_CHANGE_IND. Values: <ul style="list-style-type: none">• 0x00 – Disable (default value)• 0x01 – Enable
Type	0x2F			1	Call Mode Status
Length	1			2	
Value	→	boolean	reg_call_mode_change	1	Controls the reporting of QMI_NAS_CALL_MODE_IND. Values: <ul style="list-style-type: none">• 0x00 – Disable (default value)• 0x01 – Enable
Type	0x30			1	Service-Specific Access Class Barring Ext
Length	1			2	
Value	→	boolean	reg_ssac_change_info	1	Controls the reporting of QMI_NAS_SSAC_CHANGE_INFO_IND. Values: <ul style="list-style-type: none">• 0x00 – Disable (default value)• 0x01 – Enable
Type	0x31			1	Manual Network Scan Failure
Length	1			2	
Value	→	boolean	reg_manual_scan_fail	1	Controls the reporting of QMI_NAS_MANUAL_SCAN_FAIL_IND. Values: <ul style="list-style-type: none">• 0x00 – Disable (default value)• 0x01 – Enable
Type	0x32			1	Timer Expiry
Length	1			2	
Value	→	boolean	reg_timer_expiry_ind	1	Controls the reporting of QMI_NAS_TIMER_EXPIRY_IND. Values: <ul style="list-style-type: none">• 0x00 – Disable (default value)• 0x01 – Enable
Type	0x33			1	Emergency Mode Status
Length	1			2	
Value	→	boolean	reg_emergency_mode_status_ind	1	Controls the reporting of QMI_NAS_EMERGENCY_MODE_STATUS_IND. Values: <ul style="list-style-type: none">• 0x00 – Disable (default value)• 0x01 – Enable

7.2.5.2. Response - QMI_NAS_INDICATION_REGISTER_RESP_MSG

- Message type

Response

- **Sender**

Service

- **Mandatory TLVs**

The Result Code TLV (defined in Section 7.1.3.1) is always present in the response.

- **Optional TLVs**

None

- **Error codes**

<code>QMI_ERR_NONE</code>	No error in the request
<code>QMI_ERR_INTERNAL</code>	Unexpected error occurred during processing
<code>QMI_ERR_MALFORMED_MSG</code>	Message was not formulated correctly by the control point or the message was corrupted during transmission
<code>QMI_ERR_NO_MEMORY</code>	Device could not allocate memory to formulate a response

7.2.5.3. Description of QMI_NAS_INDICATION_REGISTER REQ/RESP

This command is used by a control point to register/deregister for different QMI_NAS indications. The control point's registration state variables, controlling registration for indications, are modified to reflect the settings indicated in the TLVs that are present in the request message. At least one optional TLV must be present in the request.

The `reg_ddtm_events` field in the DDTM Events TLV must be set to Enable to register a control point for the DDTM events and Disable to deregister. When this registration is enabled, the control point learns of DDTM events via the `QMI_NAS_DDTM_IND` indication.

The `reg_sys_sel_pref` field in the System Selection Preference TLV must be set to Enable to register a control point for the system selection preference events and Disable to deregister. When this registration is enabled, the control point learns of system selection preference changes via the `QMI_NAS_SYSTEM_SELECTION_PREFERENCE_IND` indication.

By default, `QMI_NAS_SERVING_SYSTEM_IND` is a broadcast indication that is sent to all QMI_NAS control points when the serving system information changes. To deregister the control point from receiving this indication, the `req_serving_system` field in the Serving System Events TLV must be set to Disable.

The `sys_info` field must be set to Enable for a control point to receive the `QMI_NAS_SYS_INFO_IND`

indication. To deregister the control point from receiving this indication, the sys_info field must be set to Disable.

The Network Reject Information TLV must be sent to receive the QMI_NAS_NETWORK_REJECT_IND indication. The Network Reject Information TLV contains two fields to fill out, reg_network_reject and suppress_sys_info. The reg_network_reject field enables or disables sending the indication. If the reg_network_reject field is enabled, the suppress_sys_info field can be used to prevent sys_info indications from being sent if only the reject_cause field has changed. The suppress_sys_info field only takes effect if both the sys_info and reg_network_reject fields are enabled.

7.2.6. QMI_NAS_GET_SIGNAL_STRENGTH

Queries the current signal strength as measured by the device. (Deprecated)

- **NAS message ID**

0x0020

- **Version introduced**

Major - 1, Minor - 0

7.2.6.1. Request - QMI_NAS_GET_SIGNAL_STRENGTH_REQ_MSG

- **Message type**

Request

- **Sender**

Control point

- **Mandatory TLVs**

None

- **Optional TLVs**

Name	Version introduced	Version last modified
Request Mask	Unknown	1.1

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

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	mask16	request_mask	2	<p>Request additional signal information for:</p> <p>Bit 0 (0x01) – QMI_NAS_REQUEST_SIG_INFO_RSSI_MASK; values:</p> <ul style="list-style-type: none"> • 0 – Do not request additional information for RSSI • 1 – Request additional information for RSSI <p>Bit 1 (0x02) – QMI_NAS_REQUEST_SIG_INFO_ECIO_MASK; values:</p> <ul style="list-style-type: none"> • 0 – Do not request additional information for ECIO • 1 – Request additional information for ECIO <p>Bit 2 (0x04) – QMI_NAS_REQUEST_SIG_INFO_IO_MASK; values:</p> <ul style="list-style-type: none"> • 0 – Do not request additional information for IO • 1 – Request additional information for IO <p>Bit 3 (0x08) – QMI_NAS_REQUEST_SIG_INFO_SINR_MASK; values:</p> <ul style="list-style-type: none"> • 0 – Do not request additional information for SINR • 1 – Request additional information for SINR <p>Bit 4 (0x10) – QMI_NAS_REQUEST_SIG_INFO_ERROR_RATE_MASK; values:</p> <ul style="list-style-type: none"> • 0 – Do not request additional information for Error Rate • 1 – Request additional information for Error Rate <p>Bit 5 (0x20) – QMI_NAS_REQUEST_SIG_INFO_RSRQ_MASK; values:</p> <ul style="list-style-type: none"> • 0 – Do not request additional information for RSRQ • 1 – Request additional information for RSRQ <p>Bit 6 (0x40) – QMI_NAS_REQUEST_SIG_INFO_LTE_SNR_MASK; values:</p> <ul style="list-style-type: none"> • 0 – Do not request additional information for LTE SNR • 1 – Request additional information for LTE SNR <p>Bit 7 (0x80) – QMI_NAS_REQUEST_SIG_INFO_LTE_RSRP_MASK; values:</p> <ul style="list-style-type: none"> • 0 – Do not request additional information for LTE RSRP • 1 – Request additional information for LTE RSRP

7.2.6.2. Response - QMI_NAS_GET_SIGNAL_STRENGTH_RESP_MSG

- Message type

Response

- Sender

Service

- **Mandatory TLVs**

The Result Code TLV (defined in Section 7.1.3.1) is always present in the response. The following mandatory TLV is present if the result code is QMI_RESULT_SUCCESS.

Name	Version introduced	Version last modified
Signal Strength	Unknown	1.3

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Signal Strength
Length	2			2	
Value	→	int8	sig_strength	1	<p>Received signal strength in dBm:</p> <ul style="list-style-type: none"> • For CDMA and UMTS, this indicates forward link pilot Ec • For GSM, this indicates received signal strength • For LTE, this indicates the total received wideband power observed by the UE
		enum8	radio_if	1	<p>Radio interface technology of the signal being measured. Values:</p> <ul style="list-style-type: none"> • 0x00 – RADIO_IF_NO_SVC – None (no service) • 0x01 – RADIO_IF_CDMA_1X – cdma2000® 1X • 0x02 – RADIO_IF_CDMA_1XEVDO – cdma2000® HRPD (1xEV-DO) • 0x03 – RADIO_IF_AMPS – AMPS • 0x04 – RADIO_IF_GSM – GSM • 0x05 – RADIO_IF_UMTS – UMTS • 0x08 – RADIO_IF_LTE – LTE

Optional TLVs

Name	Version introduced	Version last modified
Signal Strength List	Unknown	1.0
RSSI List	Unknown	1.3
ECIO List	Unknown	1.1
IO	Unknown	1.1
SINR	Unknown	1.1

Name	Version introduced	Version last modified
Error Rate List	Unknown	1.1
RSRQ	Unknown	1.3
LTE SNR	Unknown	1.15
LTE RSRP	Unknown	1.15

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Signal Strength List
Length	Var			2	
Value	→	uint16	num_instances	2	Number of sets of the following elements: <ul style="list-style-type: none"> • sig_strength • radio_if
		int8	sig_strength	1	Received signal strength in dBm: <ul style="list-style-type: none"> • For CDMA and UMTS, this indicates forward link pilot Ec • For GSM, this indicates received signal strength
		enum8	radio_if	1	Radio interface technology of the signal being measured. Values: <ul style="list-style-type: none"> • 0x01 – RADIO_IF_CDMA_1X – cdma2000® 1X • 0x02 – RADIO_IF_CDMA_1XEVDO – cdma2000® HRPD (1xEV-DO)
Type	0x11			1	RSSI List
Length	Var			2	
Value	→	uint16	num_instances	2	Number of sets of the following elements: <ul style="list-style-type: none"> • rssi • radio_if
		uint8	rssi	1	RSSI represented as a positive value; control points need to convert this to negative to get actual value in dBm: <ul style="list-style-type: none"> • For CDMA and UMTS, this indicates forward link pilot Ec • For GSM, this indicates received signal strength
		enum8	radio_if	1	Radio interface technology of the signal being measured. Values: <ul style="list-style-type: none"> • 0x00 – RADIO_IF_NO_SVC – None (no service) • 0x01 – RADIO_IF_CDMA_1X – cdma2000® 1X • 0x02 – RADIO_IF_CDMA_1XEVDO – cdma2000® HRPD (1xEV-DO) • 0x03 – RADIO_IF_AMPS – AMPS • 0x04 – RADIO_IF_GSM – GSM • 0x05 – RADIO_IF_UMTS – UMTS • 0x08 – RADIO_IF_LTE – LTE
Type	0x12			1	ECIO List
Length	Var			2	
Value	→	uint16	num_instances	2	Number of sets of the following elements: <ul style="list-style-type: none"> • ecio • radio_if

Field	Field value	Field type	Parameter	Size (byte)	Description
		uint8	ecio	1	ECIO value representing negative 0.5 dB increments, i.e., 2 means -1 dB (14 means -7 dB, 63 means -31.5 dB).
		enum8	radio_if	1	Radio interface technology of the signal being measured. Values: • 0x00 – RADIO_IF_NO_SVC – None (no service) • 0x01 – RADIO_IF_CDMA_1X – cdma2000® 1X • 0x02 – RADIO_IF_CDMA_1XEVDO – cdma2000® HRPD (1xEV-DO) • 0x03 – RADIO_IF_AMPS – AMPS • 0x04 – RADIO_IF_GSM – GSM • 0x05 – RADIO_IF_UMTS – UMTS
Type	0x13			1	IO
Length	4			2	
Value	→	uint32	io	4	Received IO in dBm. IO is only applicable for 1xEV-DO.
Type	0x14			1	SINR
Length	1			2	
Value	→	enum8	sinr	1	SINR level. SINR is only applicable for 1xEV-DO. Valid levels are 0 to 8, where the maximum value for: • 0x00 – SINR_LEVEL_0 is -9 dB • 0x01 – SINR_LEVEL_1 is -6 dB • 0x02 – SINR_LEVEL_2 is -4.5 dB • 0x03 – SINR_LEVEL_3 is -3 dB • 0x04 – SINR_LEVEL_4 is -2 dB • 0x05 – SINR_LEVEL_5 is +1 dB • 0x06 – SINR_LEVEL_6 is +3 dB • 0x07 – SINR_LEVEL_7 is +6 dB • 0x08 – SINR_LEVEL_8 is +9 dB
Type	0x15			1	Error Rate List
Length	Var			2	
Value	→	uint16	num_instances	2	Number of sets of the following elements: • error_rate • radio_if

Field	Field value	Field type	Parameter	Size (byte)	Description
		uint16	error_rate	2	<p>Error rate value corresponds to the RAT that is currently registered.</p> <p>For CDMA, the error rate reported is Frame Error Rate:</p> <ul style="list-style-type: none"> • Valid error rate values between 1 and 10000 are returned to indicate percentage, e.g., a value of 300 means the error rate is 3% • A value of 0xFFFF indicates that the error rate is unknown or unavailable <p>For HDR, the error rate reported is Packet Error Rate:</p> <ul style="list-style-type: none"> • Valid error rate values between 1 and 10000 are returned to indicate percentage, e.g., a value of 300 means the error rate is 3% • A value of 0xFFFF indicates that the error rate is unknown or unavailable <p>For GSM, the error rate reported is Bit Error Rate:</p> <ul style="list-style-type: none"> • Valid values are 0, 100, 200, 300, 400, 500, 600, and 700 • The reported value divided by 100 gives the error rate as an RxQual value as defined in 3GPP TS 45.008 Section 8.2.4, e.g., a value of 300 represents an RxQual value of 3 • A value of 25500 indicates No Data <p>For WCDMA, the error rate reported is Block Error Rate (BLER):</p> <ul style="list-style-type: none"> • Valid values are 1 to 10000 • The reported value divided by 100 provides the error rate in percentages, e.g., a value of 300 represents a BLER of 3% • A value of 0 indicates No Data
		enum8	radio_if	1	<p>Radio interface technology of the signal being measured. Values:</p> <ul style="list-style-type: none"> • 0x00 – RADIO_IF_NO_SVC – None (no service) • 0x01 – RADIO_IF_CDMA_1X – cdma2000® 1X • 0x02 – RADIO_IF_CDMA_1XEVDO – cdma2000® HRPD (1xEV-DO) • 0x03 – RADIO_IF_AMPS – AMPS • 0x04 – RADIO_IF_GSM – GSM • 0x05 – RADIO_IF_UMTS – UMTS
Type	0x16			1	RSRQ
Length	2			2	
Value	→	int8	rsrq	1	RSRQ value in dB (signed integer value). Range: -3 to -20 (-3 means -3 dB, -20 means -20 dB).
		uint8	radio_if	1	Radio interface technology of the signal being measured. Values:
					<ul style="list-style-type: none"> • 0x08 – LTE

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x17			1	LTE SNR
Length	2			2	
Value	→	int16	snr	2	LTE SNR level as a scaled integer in units of 0.1 dB; e.g., -16 dB has a value of -160 and 24.6 dB has a value of 246. LTE SNR is included only when the current serving system is LTE.
Type	0x18			1	LTE RSRP
Length	2			2	
Value	→	int16	lte_rsrp	2	Current LTE RSRP in dBm as measured by L1. Range: -44 to -140 (-44 means -44 dBm, -140 means -140 dBm). LTE RSRP is included only if the current serving system is LTE.

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

7.2.6.3. Description of QMI_NAS_GET_SIGNAL_STRENGTH REQ/RESP

This command queries the current pilot signal strength (in dBm) and the associated radio technology as measured by the receiver.

The optional Signal Strength List TLV is present if the device has more than one signal strength to indicate, e.g., in 3GPP2 Hybrid mode, both the mandatory Signal Strength TLV and the optional Signal Strength List TLV is returned to indicate the signal strengths of CDMA and EV-DO technologies.

A sig_strength value of -125 dBm or lower is used to indicate No Signal.

The optional Request Mask TLV can be used in the request to query additional signal information, such as RSSI, ECIO, IO, SINR, and error rate, which are returned in the RSSI, ECIO List, IO, SINR, and Error Rate List TLVs respectively, if available. If the device has more than one signal, e.g., in 3GPP2 Hybrid mode, the signal information is returned as a list TLV, such as RSSI List, ECIO List, and Error Rate List.

The AT command equivalent to this command is **AT+CSQ**, as defined in 3GPP TS 27.007, 3GPP2 C.S0017-003-A, and TIA/EIA/IS-131.

This command is deprecated. Use QMI_NAS_GET_SIG_INFO.

7.2.7. QMI_NAS_SCAN_NETS (QMI_NAS_PERFORM_NETWORK_SCAN)

Performs a scan for visible networks.

- **NAS message ID**

0x0021

- **Version introduced**

Major - 1, Minor - 0

7.2.7.1. Request - QMI_NAS_PERFORM_NETWORK_SCAN_REQ_MSG

- **Message type**

Request

- **Sender**

Control point

- **Mandatory TLVs**

None

- **Optional TLVs**

Name	Version introduced	Version last modified
Network Type	1.9	1.60
Scan Type	1.41	1.155
Band Preference	1.83	1.83
LTE Band Preference	1.83	1.146
TDSCDMA Band Preference	1.83	1.83

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Network Type
Length	1			2	
Value	→	mask8	network_type	1	Bitmask representing the network type to scan. Values: • Bit 0 – GSM • Bit 1 – UMTS • Bit 2 – LTE • Bit 3 – TD-SCDMA Any combination of the bit positions can be used. If the mask is sent with no bits set, the scan is performed using the currently set preference.
Type	0x11			1	Scan Type
Length	4			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	enum	scan_type	4	Network scan type. Values: • NAS_SCAN_TYPE_PLMN (0x00) – PLMN (default) • NAS_SCAN_TYPE_CSG (0x01) – Closed subscriber group • NAS_SCAN_TYPE_MODE_PREF (0x02) – Mode preference • NAS_SCAN_TYPE_PCI (0x03) – Physical cell ID
Type	0x12			1	Band Preference
Length	8			2	
Value	→	mask	band_pref	8	Bitmask representing the band preference to be scanned. See Table A-2 for details.
Type	0x13			1	LTE Band Preference
Length	8			2	
Value	→	mask	lte_band_pref	8	Bitmask representing the LTE band preference to be scanned. See Table A-3 for details.
Type	0x14			1	TDSCDMA Band Preference
Length	8			2	
Value	→	mask	tdscdma_band_pref	8	Bitmask representing the TD-SCDMA band preference to be scanned. Values: • NAS_TDSCDMA_BAND_A (0x01) – TD-SCDMA Band A • NAS_TDSCDMA_BAND_B (0x02) – TD-SCDMA Band B • NAS_TDSCDMA_BAND_C (0x04) – TD-SCDMA Band C • NAS_TDSCDMA_BAND_D (0x08) – TD-SCDMA Band D • NAS_TDSCDMA_BAND_E (0x10) – TD-SCDMA Band E • NAS_TDSCDMA_BAND_F (0x20) – TD-SCDMA Band F All other bits are reserved and must be set to 0.

7.2.7.2. Response - QMI_NAS_PERFORM_NETWORK_SCAN_RESP_MSG

- Message type

Response

- Sender

Service

- **Mandatory TLVs**

The Result Code TLV (defined in Section 7.1.3.1) is always present in the response.

- **Optional TLVs**

Name	Version introduced	Version last modified
3GPP Network Information**	Unknown	1.0
Network Radio Access Technology**	Unknown	1.16
MNC PCS Digit Include Status	Unknown	1.10
Network Scan Result	1.30	1.30
CSG Information	1.41	1.41
CSG Signal Strength Information	1.91	1.91
Network Name Source	1.106	1.106
PCI Information	1.155	1.155

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	3GPP Network Information**
Length	Var			2	
Value	→	uint16	num_network_info_instances	2	Number of sets of the following elements: • mobile_country_code • mobile_network_code • network_status • network_description_length • network_description
		uint16	mobile_country_code	2	A 16-bit integer representation of MCC. Range: 0 to 999.
		uint16	mobile_network_code	2	A 16-bit integer representation of MNC. Range: 0 to 999.

Field	Field value	Field type	Parameter	Size (byte)	Description
		uint8	network_status	1	<p>Status of the network identified by MCC and MNC preceding it. The status is encoded in a bitmapped value as follows:</p> <p>Bits 0-1 – QMI_NAS_NETWORK_IN_USE_STATUS_BITS – In-use status</p> <ul style="list-style-type: none"> • 0 – QMI_NAS_NETWORK_IN_USE_STATUS_UNKNOWN – Unknown • 1 – QMI_NAS_NETWORK_IN_USE_STATUS_CURRENT_SERVING – Current serving • 2 – QMI_NAS_NETWORK_IN_USE_STATUS_AVAILABLE – Available <p>Bits 2-3 – QMI_NAS_NETWORK_ROAMING_STATUS_BITS – Roaming status</p> <ul style="list-style-type: none"> • 0 – QMI_NAS_NETWORK_ROAMING_STATUS_UNKNOWN – Unknown • 1 – QMI_NAS_NETWORK_ROAMING_STATUS_HOME – Home • 2 – QMI_NAS_NETWORK_ROAMING_STATUS_ROAM – Roam <p>Bits 4-5 – QMI_NAS_NETWORK_FORBIDDEN_STATUS_BITS – Forbidden status</p> <ul style="list-style-type: none"> • 0 – QMI_NAS_NETWORK_FORBIDDEN_STATUS_UNKNOWN – Unknown • 1 – QMI_NAS_NETWORK_FORBIDDEN_STATUS_FORBIDDEN – Forbidden • 2 – QMI_NAS_NETWORK_FORBIDDEN_STATUS_NOT_FORBIDDEN – Not forbidden <p>Bits 6-7 – QMI_NAS_NETWORK_PREFERRED_STATUS_BITS – Preferred status</p> <ul style="list-style-type: none"> • 0 – QMI_NAS_NETWORK_PREFERRED_STATUS_UNKNOWN – Unknown • 1 – QMI_NAS_NETWORK_PREFERRED_STATUS_PREFERRED – Preferred • 2 – QMI_NAS_NETWORK_PREFERRED_STATUS_NOT_PREFERRED – Not preferred
		uint8	network_description_length	1	<p>Number of sets of the following elements:</p> <ul style="list-style-type: none"> • network_description
		string	network_description	Var	An optional string containing the network name or description.
Type	0x11			1	Network Radio Access Technology**
Length	Var			2	
Value	→	uint16	num_inst	2	<p>Number of sets of the following elements:</p> <ul style="list-style-type: none"> • mcc • mnc • rat
		uint16	mcc	2	A 16-bit integer representation of MCC. Range: 0 to 999.

Field	Field value	Field type	Parameter	Size (byte)	Description
		uint16	mnc	2	A 16-bit integer representation of MNC. Range: 0 to 999.
		uint8	rat	1	Radio access technology. Values: <ul style="list-style-type: none">• 0x04 – GERAN• 0x05 – UMTS• 0x08 – LTE• 0x09 – TD-SCDMA
Type	0x12			1	MNC PCS Digit Include Status
Length	Var			2	
Value	→	uint16	mnc_includes_pcs_digit_len	2	Number of sets of the following elements: <ul style="list-style-type: none">• mcc• mnc• mnc_includes_pcs_digit
		uint16	mcc	2	A 16-bit integer representation of MCC. Range: 0 to 999.
		uint16	mnc	2	A 16-bit integer representation of MNC. Range: 0 to 999.
		boolean	mnc_includes_pcs_digit	1	This field is used to interpret the length of the corresponding MNC reported in the TLVs (in this table) with an mnc or mobile_network_code field. Values: <ul style="list-style-type: none">• TRUE – MNC is a three-digit value; e.g., a reported value of 90 corresponds to an MNC value of 090• FALSE – MNC is a two-digit value; e.g., a reported value of 90 corresponds to an MNC value of 90
Type	0x13			1	Network Scan Result
Length	4			2	
Value	→	enum	scan_result	4	Indicates the status of the network scan. Values: <ul style="list-style-type: none">• 0x00 – NAS_SCAN_SUCCESS – Network scan was successful• 0x01 – NAS_SCAN_AS_ABORT – Network scan was aborted• 0x02 – NAS_SCAN_REJ_IN_RLF – Network scan did not complete due to a radio link failure recovery in progress
Type	0x14			1	CSG Information
Length	Var			2	
Value	→	uint8	csg_info_len	1	Number of sets of the following elements: <ul style="list-style-type: none">• mcc• mnc• csg_list_cat• id• name_len• name
		uint16	mcc	2	A 16-bit integer representation of MCC. Range: 0 to 999.
		uint16	mnc	2	A 16-bit integer representation of MNC. Range: 0 to 999.

Field	Field value	Field type	Parameter	Size (byte)	Description
		enum	csg_list_cat	4	<p>Closed subscriber group category. Values:</p> <ul style="list-style-type: none"> • 0 – NAS_CSG_LIST_CAT_UNKNOWN – Unknown CSG list • 1 – NAS_CSG_LIST_CAT_ALLOWED – Allowed CSG list • 2 – NAS_CSG_LIST_CAT_OPERATOR – Operator CSG list
		uint32	id	4	Closed subscriber group identifier.
		uint8	name_len	1	Number of sets of the following elements:
		uint16	name	Var	Home Node B (HNB) or Home eNode B (HeNB) name in UTF-16. The network name is not guaranteed to be NULL terminated.
Type	0x15			1	CSG Signal Strength Information
Length	Var			2	
	→	uint8	csg_sig_info_len	1	<p>Number of sets of the following elements:</p> <ul style="list-style-type: none"> • mcc • mnc • csg_id • signal_strength
		uint16	mcc	2	A 16-bit integer representation of MCC. Range: 0 to 999.
		uint16	mnc	2	A 16-bit integer representation of MNC. Range: 0 to 999.
		uint32	csg_id	4	Closed subscriber group identifier.
		int32	signal_strength	4	Signal strength information in dBm.
Type	0x16			1	Network Name Source
Length	Var			2	
	→	uint8	nw_name_source_len	1	<p>Number of sets of the following elements:</p> <ul style="list-style-type: none"> • nw_name_source
		enum	nw_name_source	Var	<p>Network name source. Values:</p> <ul style="list-style-type: none"> • NAS_NW_NAME_SOURCE_UNKNOWN (0x00) – Unknown • NAS_NW_NAME_SOURCE_OPL_PNN (0x01) – Operator PLMN list and PLMN network name • NAS_NW_NAME_SOURCE_CPHS_ONS (0x02) – Common PCN handset specification and operator name string • NAS_NW_NAME_SOURCE_NITZ (0x03) – Network identity and time zone • NAS_NW_NAME_SOURCE_SE13 (0x04) – GSMA SE13 table • NAS_NW_NAME_SOURCE_MCC_MNC (0x05) – Mobile country code and mobile network code • NAS_NW_NAME_SOURCE_SPN (0x06) – Service provider name
Type	0x17			1	PCI Information

Field	Field value	Field type	Parameter	Size (byte)	Description
Length	Var			2	
Value	→	uint8	pci_cell_info_len	1	Number of sets of the following elements: • freq • cell_id • global_cell_id • mcc • mnc • mnc_includes_pcs_digit
		uint32	freq	4	Absolute cell's frequency. Range: 0 to 65535.
		uint16	cell_id	2	Cell ID
		uint32	global_cell_id	4	Global cell ID
		uint8	plmn_len	1	Number of sets of the following elements: • mcc • mnc • mnc_includes_pcs_digit
		uint16	mcc	2	A 16-bit integer representation of MCC. Range: 0 to 999.
		uint16	mnc	2	A 16-bit integer representation of MNC. Range: 0 to 999.
		boolean	mnc_includes_pcs_digit	1	This field is used to interpret the length of the corresponding MNC reported in this TLV. Values: • TRUE – MNC is a three-digit value; e.g., a reported value of 90 corresponds to an MNC value of 090 • FALSE – MNC is a two-digit value; e.g., a reported value of 90 corresponds to an MNC value of 90
		int16	rsrp	2	Combined RSRP
		int16	rsrp_rx0	2	Rx0 RSRP
		int16	rsrp_rx1	2	Rx1 RSRP
		int16	rsrq	2	Combined RSRQ
		int16	rsrq_rx0	2	Rx0 RSRQ
		int16	rsrq_rx1	2	Rx1 RSRQ

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_DEVICE_IN_USE	Operation cannot be performed; radio is currently in use, e.g., in a call
QMI_ERR_OP_DEVICE_UNSUPPORTED	Operation is not supported by the device
QMI_ERR_ABORTED	Operation was aborted by the control point using QMI_NAS_ABORT_REQ
QMI_ERR_INVALID_OPERATION	Value or configuration is not supported

7.2.7.3. Description of QMI_NAS_PERFORM_NETWORK_SCAN REQ/RESP

This command performs a network scan and returns a list of visible networks. If the Network Type TLV is not included in the request, the scan is performed on GSM, WCDMA, and LTE. If the Network Type TLV is included in the request, the scan is performed on the specified networks.

In the Network Type TLV, if the RAT bitmask is not valid, a QMI_ERR_INVALID_OPERATION error is returned.

The 3GPP Network Information TLV includes zero or more sets of parameters; each set describes a single visible network detected during the scan.

If nas_3gpp_network_info_len in the 3GPP Network Information TLV is 0, the Network Radio Access Technology TLV is not included.

The Scan Type TLV is used to select the type of network scanning: regular PLMN scanning vs closed subscriber group network scanning.

If the Band Preference, LTE Band Preference, or TDSCDMA Band Preference TLVs are provided, only the bands specified are scanned. If these TLVs are not included, all bands applicable to the RAT are scanned.

This operation is not supported on CDMA.

The AT command equivalent to this command is AT+COPS, as defined in 3GPP TS 27.007.

7.2.8. QMI_NAS_REGISTER_NET (QMI_NAS_INITIATE_NETWORK_REGISTER)

Initiates a network registration.

- **NAS message ID**

0x0022

- **Version introduced**

Major – 1, Minor – 0

7.2.8.1. Request – QMI_NAS_INITIATE_NETWORK_REGISTER_REQ_MSG

- **Message type**

Request

- **Sender**

Control point

- **Mandatory TLVs**

Name	Version introduced	Version last modified
Register Action	Unknown	1.0

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Register Action
Length	1			2	
Value	→	enum8	register_action	1	Specifies one of the following actions: • 0x01 – NAS_AUTO_REGISTER – Device registers according to its provisioning; optional TLVs supplied with the command are ignored • 0x02 – NAS_MANUAL_REGISTER – Device registers to a specified network; the optional Manual Network Register Information TLV must also be included for the command to process successfully; supported only for 3GPP

Optional TLVs

Name	Version introduced	Version last modified
Manual Network Register Information**	Unknown	1.17
Change Duration**	Unknown	1.5
MNC PCS Digit Include Status	Unknown	1.12

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Manual Network Register Information**
Length	5			2	
Value	→	uint16	mobile_country_code	2	A 16-bit integer representation of MCC. Range: 0 to 999.
			Uint16 mobile_network	2	A 16-bit integer representation of MNC. Range: 0 to 999.

			code		999.
	Enum8	radio_access_technology		1	Radio access technology for which to register. Values: • 0x04 – RADIO_IF_GSM – GSM • 0x05 – RADIO_IF_UMTS – UMTS • 0x08 – RADIO_IF_LTE – LTE • -1 – RADIO_IF_NO_CHANGE – No change in the mode preference
Type	0x11			1	Change Duration**
Length	1			2	
Value	→	enum8	change_duration	1	Duration of the change. Values: • 0x00 – Power cycle – Remains active until the next device power cycle • 0x01 – Permanent – Remains active through power cycles until changed by the client Note: The device will use “0x00 – Power cycle” as the default value if the TLV is omitted.
Type	0x12			1	MNC PCS Digit Include Status
Length	1			2	
Value	→	boolean	mnc_includes_pcs_digit	1	This TLV applies to the MNC field of the manual_network_register_info data structure. Values: • TRUE – MNC is a three-digit value • FALSE – MNC is a two-digit value If this TLV is not included in the case of a manual register option, the value of the MNC value specified in manual_network_register_info is interpreted as follows: • If the MNC value is less than 100, the MNC value provided is interpreted as a two-digit value. • If the MNC value is greater than or equal to 100, the MNC value provided is interpreted as a three-digit value.

7.2.8.2. Response – QMI_NAS_INITIATE_NETWORK_REGISTER_RESP_MSG

- Message type

Response

- Sender

Service

- Mandatory TLVs

The Result Code TLV (defined in Section 7.1.3.1) 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_MISSING_ARG	One or more required TLVs were missing in the request
QMI_ERR_DEVICE_IN_USE	Operation cannot be performed; radio is currently in use, e.g., in a call
QMI_ERR_OP_DEVICE_UNSUPPORTED	Operation is not supported by the device
QMI_ERR_INVALID_REGISTER_ACTION	Invalid register action value was specified in the request
QMI_ERR_NO_NETWORK_FOUND	Network specified in the manual registration request cannot be found
QMI_ERR_INVALID_ARG	Value field of one or more TLVs in the request message contains an invalid value

7.2.8.3. Description of QMI_NAS_INITIATE_NETWORK_REGISTER REQ/RESP

This command initiates an automatic or manual registration to the specified network.

If the Result Code TLV indicates success, the device has started the requested operation.

The control point must always process the QMI_NAS_SERVING_SYSTEM_IND indication to learn the current registration state of the device.

The AT command equivalent to this command is AT+COPS, as defined in 3GPP TS 27.007.

7.2.9. QMI_NAS_INITIATE_ATTACH

Initiates a domain attach or detach action. (Deprecated)

- **NAS message ID**

0x0023

- **Version introduced**

Major - 1, Minor - 0

7.2.9.1. Request - QMI_NAS_INITIATE_ATTACH_REQ_MSG

- **Message type**

Request

- **Sender**

Control point

- **Mandatory TLVs**

None

- **Optional TLVs**

The following optional TLV must be included in this request

Name	Version introduced	Version last modified
PS Attach Action**	Unknown	1.0

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	PS Attach Action**
Length	1			2	
Value	→	enum8	ps_attach_action	1	<p>Initiates a packet domain attach or detach action.</p> <p>Values:</p> <ul style="list-style-type: none"> • 0x01 – PS_ACTION_ATTACH – Initiates an immediate packet domain attach action • 0x02 – PS_ACTION_DETACH – Initiates an immediate packet domain detach action

7.2.9.2. Response - QMI_NAS_INITIATE_ATTACH_RESP_MSG

- **Message type**

Response

- **Sender**

Service

- **Mandatory TLVs**

The Result Code TLV (defined in Section 7.1.3.1) 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_MISSING_ARG	One or more required TLVs were missing in the request
QMI_ERR_DEVICE_IN_USE	Operation cannot be performed; radio is currently in use, e.g., in a call
QMI_ERR_OP_NETWORK_UNsupported	Operation is not supported by the network
QMI_ERR_INVALID_PS_ATTACH_ACTION	Invalid PS attach action value was specified in the request

7.2.9.3. Description of QMI_NAS_INITIATE_ATTACH REQ/RESP

This command initiates a domain attach or detach action.

If the Result Code TLV indicates success, this means the device has started the requested operation.

The control point must always process the QMI_NAS_SERVING_SYSTEM_IND indication to learn the current attachment state of the device.

CDMA networks do not have the concept of domain attachment. This command fails if the current mode is CDMA.

The AT command equivalent to this command is AT+CGATT, as defined in 3GPP TS 27.007.

This command is deprecated. Use QMI_NAS_SET_SYSTEM_SELECTION_PREFERENCE

7.2.10. QMI_NAS_GET_SS_INFO (QMI_NAS_GET_SERVING_SYSTEM)

Queries information regarding the system that currently provides service.

- **NAS message ID**

0x0024

- **Version introduced**

Major – 1, Minor – 0

7.2.10.1. Request – QMI_NAS_GET_SERVING_SYSTEM_REQ_MSG

- **Message type**

Request

- **Sender**

Control point

- **Mandatory TLVs**

None

- **Optional TLVs**

None

7.2.10.2. Response – QMI_NAS_GET_SERVING_SYSTEM_RESP_MSG

- **Message type**

Response

- **Sender**

Service

- **Mandatory TLVs**

The Result Code TLV (defined in Section 7.1.3.1) is always present in the response. The following mandatoryTLV is present if the result code is QMI_RESULT_SUCCESS.

Name	Version introduced	Version last modified			
Serving System	Unknown	1.3			
+					
Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Serving System
Length	Var			2	
Value	→	enum8	registration_state	1	Registration state of the mobile. Values: - 0x00 – NOT_REGISTERED – Not registered; mobile is not currently searching for a new network to provide service - 0x01 – REGISTERED – Registered with a network - 0x02 – NOT_REGISTERED_SEARCHING – Not

				registered, but mobile is currently searching for a new network to provide service • 0x03 – REGISTRATION_DENIED – Registration denied by the visible network • 0x04 – REGISTRATION_UNKNOWN – Registration state is unknown
	enum8	cs_attach_state	1	Circuit-switched domain attach state of the mobile. Values: • 0x00 – CS_UNKNOWN – Unknown or not applicable • 0x01 – CS_ATTACHED – Attached • 0x02 – CS_DETACHED – Detached
	enum8	ps_attach_state	1	Packet-switched domain attach state of the mobile. Values: • 0x00 – PS_UNKNOWN – Unknown or not applicable • 0x01 – PS_ATTACHED – Attached • 0x02 – PS_DETACHED – Detached
	enum8	selected_network	1	Type of selected radio access network. Values: • 0x00 – SELECTED_NETWORK_UNKNOWN – Unknown • 0x01 – SELECTED_NETWORK_3GPP2 – 3GPP2 network • 0x02 – SELECTED_NETWORK_3GPP – 3GPP network
	uint8	in_use_radio_if_li_st_num	1	Number of sets of the following elements: • radio_if
	enum8	radio_if	Var	Radio interface currently in use. Values: • 0x00 – RADIO_IF_NO_SVC – None (no service) • 0x01 – RADIO_IF_CDMA_1X – cdma2000® 1X • 0x02 – RADIO_IF_CDMA_1XEVDO – cdma2000® HRPD (1xEV-DO) • 0x03 – RADIO_IF_AMPS – AMPS • 0x04 – RADIO_IF_GSM – GSM • 0x05 – RADIO_IF_UMTS – UMTS • 0x08 – RADIO_IF_LTE – LTE

Optional TLVs

Name	Version introduced	Version last modified
Roaming Indicator Value	Unknown	1.0
Data Service Capability	Unknown	1.4
Current PLMN	Unknown	1.0
CDMA System ID	Unknown	1.1
CDMA Base Station Information	Unknown	1.1
Roaming Indicator List	Unknown	1.3
Default Roaming Indicator	Unknown	1.1
3GPP2 Time Zone	Unknown	1.1
CDMA P Rev in Use	Unknown	1.1
3GPP Time Zone	Unknown	1.4
3GPP Network Daylight Saving Adjustment	Unknown	1.4
3GPP Location Area Code	Unknown	1.5
At least one of the following optional TLVs must be included in this request.	Unknown	1.5

3GPP2 Concurrent Service Info	Unknown	1.5
3GPP2 PRL Indicator	Unknown	1.5
Dual Transfer Mode Indication	Unknown	1.5
Detailed Service Information	Unknown	1.5
CDMA System Info	Unknown	1.6
HDR Personality	Unknown	1.7
TAC Information for LTE	Unknown	1.7
Call Barring Status	Unknown	1.12
UMTS Primary Scrambling Code	Unknown	1.14
MNC PCS Digit Include Status	Unknown	1.17
HS Call Status	1.23	1.125
3GPP Network Name Source	1.113	1.113

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Roaming Indicator Value
Length	1			2	
Value	→	enum8	roaming_indicator	1	Roaming indicator. Values: • 0x00 – ROAMING_IND_ON – Roaming • 0x01 – ROAMING_IND_OFF – Home • 0x02 and above – Operator-defined values
Type	0x11			1	Data Service Capability
Length	Var			2	
Value	→	uint8	data_capability_list_len	1	Number of sets of the following elements: • data_capabilities
		enum8	data_capabilities	Var	List of data capabilities (each is 1 byte) of the current serving system. Possible values: • 0x01 – DATA_CAPABILITIES_GPRS – GPRS • 0x02 – DATA_CAPABILITIES_EDGE – EDGE • 0x03 – DATA_CAPABILITIES_HSDPA – HSDPA • 0x04 – DATA_CAPABILITIES_HSUPA – HSUPA • 0x05 – DATA_CAPABILITIES_WCDMA – WCDMA • 0x06 – DATA_CAPABILITIES_CDMA – CDMA • 0x07 – DATA_CAPABILITIES_EVDO_REV_0 – EV-DO REV 0 • 0x08 – DATA_CAPABILITIES_EVDO_REV_A – EV-DO REV A • 0x09 – DATA_CAPABILITIES_GSM – GSM • 0x0A – DATA_CAPABILITIES_EVDO_REV_B – EV-DO REV B • 0x0B – DATA_CAPABILITIES_LTE – LTE • 0x0C –

					DATA_CAPABILITIES_HSDPA_PLUS – HSDPA+ • 0x0D – DATA_CAPABILITIES_DC_HSDPA_PLUS – DC-HSDPA+
Type	0x12			1	Current PLMN
Length	Var			2	
Value	→	uint16	mobile_country_code	2	A 16-bit integer representation of MCC. Range: 0 to 999.
		Uint16	mobile_network_code	2	A 16-bit integer representation of MNC. Range: 0 to 999.
		Uint8	network_description_length	1	Number of sets of the following elements: • network_description
		string	network_description	Var	An optional string containing the network name or description.
Type	0x13			1	CDMA System ID
Length	4			2	
Value	→	uint16	sid	2	System ID.
		Uint16	nid	2	Network ID.
Type	0x14			1	CDMA Base Station Information
Length	10			2	
Value	→	uint16	base_id	2	Base station identification number.
		Int32	base_lat	4	Base station latitude in units of 0.25 sec, expressed as a two's complement signed number with positive numbers signifying North latitudes.
		Int32	base_long	4	Base station longitude in units of 0.25 sec, expressed as a two's complement signed number with positive numbers signifying East longitude.
Type	0x15			1	Roaming Indicator List
Length	Var			2	
Value	→	uint8	num_instances	1	Number of sets of the following elements: • radio_if • roaming_indicator
		enum8	radio_if	1	Radio interface currently in use. Values: • 0x01 – RADIO_IF_CDMA_1X – cdma2000® 1X • 0x02 – RADIO_IF_CDMA_1XEVDO – cdma2000® HRPD (1xEV-DO) • 0x03 – RADIO_IF_AMPS – AMPS • 0x04 – RADIO_IF_GSM – GSM • 0x05 – RADIO_IF_UTMS – UMTS • 0x08 – RADIO_IF_LTE – LTE
		enum8	roaming_indicator	1	Roaming indicator. Values: • 0x00 – ROAMING_IND_ON – Roaming • 0x01 – ROAMING_IND_OFF – Home Values from 2 onward are applicable only for 3GPP2. Refer to 3GPP2 C.R1001-F for the meanings of these values.

Type	0x16			1	Default Roaming Indicator
Length	1			2	
Value	→	enum8	def_roam_ind	1	<p>Roaming indicator. Values:</p> <ul style="list-style-type: none"> • 0x00 – ROAMING_IND_ON – Roaming • 0x01 – ROAMING_IND_OFF – Home <p>Values from 2 onward are applicable only for 3GPP2. Refer to 3GPP2 C.R1001-F for the meanings of these values.</p>
Type	0x17			1	3GPP2 Time Zone
Length	3			2	
Value	→	uint8	lp_sec	1	Number of leap seconds since the start of CDMA system time.
			ltm_offset	1	Offset of local time from system time in units of 30 min. The value in this field conveys the offset as an 8-bit two's complement number.
			daylt_savings	1	<p>Daylight saving indicator. Values:</p> <ul style="list-style-type: none"> • 0x00 – OFF (daylight saving not in effect) • 0x01 – ON (daylight saving in effect)
Type	0x18			1	CDMA P Rev in Use
Length	1			2	
Value	→	uint8	p_rev_in_use	1	P Rev that is currently in use.
Type	0x1A			1	3GPP Time Zone
Length	1			2	
Value	→	int8	time_zone	1	Offset from Universal time, i.e., difference between local time and Universal time, in increments of 15 min (signed value).
Type	0x1B			1	3GPP Network Daylight Saving Adjustment
Length	1			2	
Value	→	uint8	adj	1	<p>3GPP network daylight saving adjustment. Values:</p> <ul style="list-style-type: none"> • 0x00 – No adjustment for Daylight Saving Time • 0x01 – 1 hr adjustment for Daylight Saving Time • 0x02 – 2 hr adjustment for Daylight Saving Time
Type	0x1C			1	3GPP Location Area Code
Length	2			2	
Value	→	uint16	lac	2	Location area code.
Type	0x1D			1	3GPP Cell ID
Length	4			2	
Value	→	uint32	cell_id	4	3GPP cell ID.
Type	0x1E			1	3GPP2 Concurrent Service Info
Length	1			2	
Value	→	uint8	ccs	1	<p>3GPP2 concurrent service information. Values:</p> <ul style="list-style-type: none"> • 0x00 – Concurrent service not available • 0x01 – Concurrent service available
Type	0x1F			1	3GPP2 PRL Indicator
Length	1			2	
Value	→	uint8	prl_ind	1	3GPP2 PRL indicator. Values:
					<ul style="list-style-type: none"> • 0x00 – System not in PRL

					• 0x01 – System is in PRL
Type	0x20			1	Dual Transfer Mode Indication (GSM Only)
Length	1			2	
Value	→	uint8	dtm_ind	1	Dual Transfer mode indication. Values: • 0x00 – DTM not supported • 0x01 – DTM supported
Type	0x21			1	Detailed Service Information
Length	5			2	
Value	→	uint8	srv_status	1	Service status. Values: • 0x00 – No service • 0x01 – Limited service • 0x02 – Service available • 0x03 – Limited regional service • 0x04 – MS in power save or deep sleep
		uint8	srv_capability	1	System's service capability. Values: • 0x00 – No service • 0x01 – Circuit-switched only • 0x02 – Packet-switched only • 0x03 – Circuit-switched and-packet switched • 0x04 – MS found the right system but not yet registered/attached
		uint8	hdr_srv_status	1	HDR service status. Values: • 0x00 – No service • 0x01 – Limited service • 0x02 – Service available • 0x03 – Limited regional service • 0x04 – MS in power save or deep sleep
		uint8	hdr_hybrid	1	HDR hybrid information. Values: • 0x00 – System is not hybrid • 0x01 – System is hybrid
		uint8	is_sys_forbidden	1	Forbidden system information. Values: • 0x00 – System is not a forbidden system • 0x01 – System is a forbidden system
Type	0x22			1	CDMA System Info
Length	3			2	
Value	→	uint16	mcc	2	Mobile country code.
		Uint8	imsi_11_12	1	IMSI_11_12.
Type	0x23			1	HDR Personality
Length	1			2	
Value	→	enum8	hdr_personality	1	HDR personality information. Values: • 0x00 – Unknown • 0x01 – HRPD • 0x02 – eHRPD
Type	0x24			1	TAC Information for LTE
Length	2			2	
Value	→	uint16	tac	2	Tracking area code information for LTE.
Type	0x25			1	Call Barring Status
Length	8			2	
Value	→	enum	cs_bar_status	4	Call barring status for circuit-switched calls. Values: • 0x00 – – NAS_CELL_ACCESS_NORMAL_ONLY – Cell access is allowed for normal calls only

					<ul style="list-style-type: none"> • 0x01 – NAS_CELL_ACCESS_EMERGENCY_ONLY – Cell access is allowed for emergency calls only • 0x02 – NAS_CELL_ACCESS_NO_CALLS – Cell access is not allowed for any call type • 0x03 – NAS_CELL_ACCESS_ALL_CALLS – Cell access is allowed for all call types • -1 – NAS_CELL_ACCESS_UNKNOWN – Cell access type is unknown
	enum	ps_bar_status	4		<p>Call barring status for packet-switched calls.</p> <p>Values:</p> <ul style="list-style-type: none"> • 0x00 – NAS_CELL_ACCESS_NORMAL_ONLY – Cell access is allowed for normal calls only • 0x01 – NAS_CELL_ACCESS_EMERGENCY_ONLY – Cell access is allowed for emergency calls only • 0x02 – NAS_CELL_ACCESS_NO_CALLS – Cell access is not allowed for any call type • 0x03 – NAS_CELL_ACCESS_ALL_CALLS – Cell access is allowed for all call types • -1 – NAS_CELL_ACCESS_UNKNOWN – Cell access type is unknown
Type	0x26		1		UMTS Primary Scrambling Code
Length	2		2		
Value	→	uint16	umts_psc	2	Primary scrambling code.
Type	0x27		1		MNC PCS Digit Include Status
Length	5		2		
Value	→	uint16	mcc	2	<p>A 16-bit integer representation of MCC.</p> <p>Range: 0 to 999.</p>
	Uint16	mnc	2		<p>A 16-bit integer representation of MNC.</p> <p>Range: 0 to 999.</p>
	Boolean	mnc_includes_pcs_digit	1		<p>This field is used to interpret the length of the corresponding MNC reported in the TLVs (in this table) with an mnc or mobile_network_code field. Values:</p> <ul style="list-style-type: none"> • TRUE – MNC is a three-digit value; e.g., a reported value of 90 corresponds to an MNC value of 090 • FALSE – MNC is a two-digit value; e.g., a reported value of 90 corresponds to an MNC value of 90
Type	0x28		1		HS Call Status
Length	1		2		
Value	→	enum8	hs_call_status	1	<p>Call status on high speed (only applicable for WCDMA). Values:</p> <ul style="list-style-type: none"> • SYS HS IND HSDPA HSUPA UNSUPP

				CELL (0x00) – HSDPA and HSUPA are unsupported <ul style="list-style-type: none"> • SYS_HS_IND_HSDPA_SUPP_CELL (0x01) – HSDPA is supported • SYS_HS_IND_HSUPA_SUPP_CELL (0x02) – HSUPA is supported • SYS_HS_IND_HSDPA_HSUPA_SUPP_CELL (0x03) – HSDPA and HSUPA are supported • SYS_HS_IND_HSDPAPLUS_SUPP_CELL (0x04) – HSDPA+ is supported • SYS_HS_IND_HSDPAPLUS_HSUPA_SUPP_CELL (0x05) – HSDPA+ and HSUPA are supported • SYS_HS_IND_DC_HSDPAPLUS_SUPP_CELL (0x06) – Dual-cell HSDPA+ is supported • SYS_HS_IND_DC_HSDPAPLUS_HSUPA_SUPP_CELL (0x07) – Dual-cell HSDPA+ and HSUPA are supported • SYS_HS_IND_HSDPAPLUS_64QAM_HSUPA_SUPP_CELL (0x08) – Dual-cell HSDPA+, 64 QAM, and HSUPA are supported • SYS_HS_IND_HSDPAPLUS_64QAM_SUPP_CELL (0x09) – Dual-cell HSDPA+ and 64 QAM are supported • SYS_HS_IND_DC_HSDPAPLUS_DC_HSUPA_SUPP_CELL (0x0A) – Dual-cell HSDPA+ and dual-cell HSUPA are supported
Type	0x29		1	3GPP Network Name Source
Length	4		2	
Value	→	enum	nas_3gpp_nw_name_source	Network name source. Values: <ul style="list-style-type: none"> • NAS_NW_NAME_SOURCE_UNKNOWN (0x00) – Unknown • NAS_NW_NAME_SOURCE_OPL_PNN (0x01) – Operator PLMN list and PLMN network name • NAS_NW_NAME_SOURCE_CPHS_ONS (0x02) – Common PCN handset specification and operator name string • NAS_NW_NAME_SOURCE_NITZ (0x03) – Network identity and time zone • NAS_NW_NAME_SOURCE_SE13 (0x04) – GSMA SE13 table • NAS_NW_NAME_SOURCE_MCC_MNC (0x05) – Mobile country code and mobile network code

					• NAS_NW_NAME_SOURCE_SPN (0x06) – Service provider name
--	--	--	--	--	---

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_INFO_UNAVAILABLE	Information is not available at this time

7.2.10.3. Description of QMI_NAS_GET_SERVING_SYSTEM REQ/RESP

This command queries current serving system and registration information, including system identification, registration state, and radio technology information.

The roaming indicator and the current PLMN are not included in the response when the device is not registered.

If registered on the 3GPP network, and relevant information has been sent from the network, the 3GPP Time Zone and/or 3GPP Network Daylight Saving Adjustment TLVs are included.

The 3GPP Location Area Code and 3GPP Cell ID TLVs are included if the UE is registered on the 3GPP network. The 3GPP2 Concurrent Service Info and 3GPP2 PRL Indicator TLVs are included if the UE is registered on the 3GPP2 network. The Dual Transfer Mode Indication TLV is included if the UE is registered on the GSM network. The Detailed Service Information TLV is included so clients can retrieve detailed information about the Call Manager layer to fine-tune their internal states.

The Call Barring Status TLV is included only in GSM or WCDMA networks.

The AT command equivalent to this command is AT+CSS, as defined in 3GPP TS 27.007, 3GPP2 C.S0017-003-A, and TIA/EIA/IS-131.

The MNC PCS Digit Include Status TLV is used to indicate if pcs_digit is included in mnc. This TLV is present when the Current PLMN (TLV 0x12) is also present.

7.2.11. QMI_NAS_SS_INFO_IND (QMI_NAS_SERVING_SYSTEM_IND)

Indicates a change in the current serving system registration state and/or radio technology.

- **NAS message ID**

0x0024

- Version introduced

Major – 1, Minor – 0

7.2.11.1. 7.2.5.1. Indication – QMI_NAS_SERVING_SYSTEM_IND_MSG

- Message type

Indication

- Sender

Service

- Scope

Per control point (unicast)

- Mandatory TLVs

Name	Version introduced	Version last modified
Serving System	Unknown	1.3

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Serving System
Length	Var			2	
Value	→	enum8	registration_state	1	Registration state of the mobile. Values: • 0x00 – NOT_REGISTERED – Not registered; mobile is not currently searching for a new network to provide service • 0x01 – REGISTERED – Registered with a network • 0x02 – NOT_REGISTERED_SEARCHING – Not registered, but mobile is currently searching for a new network to provide service • 0x03 – REGISTRATION_DENIED – Registration denied by the visible network • 0x04 – REGISTRATION_UNKNOWN – Registration state is unknown
		enum8	cs_attach_state	1	Circuit-switched domain attach state of the mobile. Values: • 0x00 – CS_UNKNOWN – Unknown or not applicable • 0x01 – CS_ATTACHED – Attached • 0x02 – CS_DETACHED – Detached

	enum8	ps_attach_state	1	<p>Packet-switched domain attach state of the mobile.</p> <p>Values:</p> <ul style="list-style-type: none"> • 0x00 – PS_UNKNOWN – Unknown or not applicable • 0x01 – PS_ATTACHED – Attached • 0x02 – PS_DETACHED – Detached
	enum8	selected_network	1	<p>Type of selected radio access network.</p> <p>Values:</p> <ul style="list-style-type: none"> • 0x00 – SELECTED_NETWORK_UNKNOWN – Unknown • 0x01 – SELECTED_NETWORK_3GPP2 – 3GPP2 network • 0x02 – SELECTED_NETWORK_3GPP – 3GPP network
	uint8	in_use_radio_if_list_num	1	<p>Number of sets of the following elements:</p> <ul style="list-style-type: none"> • radio_if
	enum8	radio_if	Var	<p>Radio interface currently in use. Values:</p> <ul style="list-style-type: none"> • 0x00 – RADIO_IF_NO_SVC – None (no service) • 0x01 – RADIO_IF_CDMA_1X – cdma2000® 1X • 0x02 – RADIO_IF_CDMA_1XEVDO – cdma2000® HRPD (1xEV-DO) • 0x03 – RADIO_IF_AMPS – AMPS • 0x04 – RADIO_IF_GSM – GSM • 0x05 – RADIO_IF_UMTS – UMTS • 0x08 – RADIO_IF_LTE – LTE

Optional TLVs

Name	Version introduced	Version last modified
Roaming Indicator Value	Unknown	1.0
Data Service Capability	Unknown	1.4
Current PLMN	Unknown	1.0
CDMA System ID	Unknown	1.1
CDMA Base Station Information	Unknown	1.1
Roaming Indicator List	Unknown	1.3
At least one of the following optional TLVs must be included in this request.	Unknown	1.1
3GPP2 Time Zone	Unknown	1.1
CDMA P_Rev in Use	Unknown	1.1
3GPP PLMN Name Flag	Unknown	1.6
3GPP Time Zone	Unknown	1.4
3GPP Network Daylight Saving Adjustment	Unknown	1.4
3GPP Universal Time and Local Time Zone	Unknown	1.4
3GPP Location Area Code	Unknown	1.5
3GPP Cell ID	Unknown	1.5
3GPP2 Concurrent Service Info	Unknown	1.5
3GPP2 PRL Indicator	Unknown	1.5
Dual Transfer Mode Indication	Unknown	1.5
Detailed Service Information	Unknown	1.5

CDMA System Info Ext	Unknown	1.7
HDR Personality	Unknown	1.7
TAC Information for LTE	Unknown	1.7
Call Barring Status	Unknown	1.12
PLMN Change Status	Unknown	1.13
UMTS Primary Scrambling Code	Unknown	1.14
MNC PCS Digit Include Status	Unknown	1.17
At least one of the following optional TLVs must be included in this request.	1.23	1.125
3GPP Network Name Source	1.113	1.113

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Roaming Indicator Value
Length	1			2	
Value	→	enum8	roaming_indicator	1	Roaming indicator. Values: • 0x00 – ROAMING_IND_ON – Roaming • 0x01 – ROAMING_IND_OFF – Home • 0x02 – ROAMING_IND_FLASHING – Flashing • 0x03 and above – Operator-defined values
Type	0x11			1	Data Service Capability
Length	Var			2	
Value	→	uint8	data_capability_list_lent	1	Number of sets of the following elements: • data_capabilities
		enum8	data_capabilities	Var	List of data capabilities (each is 1 byte) of the current serving system. Possible values: • 0x01 – DATA_CAPABILITIES_GPRS – GPRS • 0x02 – DATA_CAPABILITIES_EDGE – EDGE • 0x03 – DATA_CAPABILITIES_HSDPA – HSDPA • 0x04 – DATA_CAPABILITIES_HSUPA – HSUPA • 0x05 – DATA_CAPABILITIES_WCDMA – WCDMA • 0x06 – DATA_CAPABILITIES_CDMA – CDMA • 0x07 – DATA_CAPABILITIES_EVDO_REV_0 – EV-DO REV 0 • 0x08 – DATA_CAPABILITIES_EVDO_REV_A – EV-DO REV A • 0x09 – DATA_CAPABILITIES_GSM – GSM • 0x0A – DATA_CAPABILITIES_EVDO_REV_B – EV-DO REV B • 0x0B – DATA_CAPABILITIES_LTE – LTE • 0x0C – DATA_CAPABILITIES_HSDPA_PLUS – HSDPA PLUS

					HSDPA+ • 0x0D – DATA_CAPABILITIES_DC_HSDPA_
Type	0x12			1	PLUS – DC-HSDPA+
Length	Var			2	Current PLMN
Value	→	uint16	mobile_country_code	2	A 16-bit integer representation of MCC. Range: 0 to 999.
		Uint16	mobile_network_code	2	A 16-bit integer representation of MNC. Range: 0 to 999.
		Uint8	network_description_length	1	Number of sets of the following elements: • network_description
		string	network_description	Var	An optional string containing the network name or description.
Type	0x13			1	CDMA System ID
Length	4			2	
Value	→	uint16	sid	2	System ID.
		Uint16	nid	2	Network ID.
Type	0x14			1	CDMA Base Station Information
Length	10			2	
Value	→	uint16	base_id	2	Base station identification number.
		Int32	base_lat	4	Base station latitude in units of 0.25 sec, expressed as a two's complement signed number with positive numbers signifying North latitudes.
		Int32	base_long	4	Base station longitude in units of 0.25 sec, expressed as a two's complement signed number with positive numbers signifying East longitude.
Type	0x15			1	Roaming Indicator List
Length	Var			2	
Value	→	uint8	num_instances	1	Number of sets of the following elements: • radio_if • roaming_indicator
		enum8	radio_if	1	Radio interface currently in use. Values: • 0x01 – RADIO_IF_CDMA_1X – cdma2000® 1X • 0x02 – RADIO_IF_CDMA_1XEVDO – cdma2000® HRPD (1xEV-DO) • 0x03 – RADIO_IF_AMPS – AMPS • 0x04 – RADIO_IF_GSM – GSM • 0x05 – RADIO_IF_UTMS – UMTS • 0x08 – RADIO_IF_LTE – LTE
		enum8	roaming_indicator	1	Roaming indicator. Values: • 0x00 – ROAMING_IND_ON – Roaming • 0x01 – ROAMING_IND_OFF – Home Values from 2 onward are applicable only for 3GPP2. Refer to 3GPP2 C.R1001-F for the meanings of these values.
Type	0x16			1	Default Roaming Indicator
Length	1			2	

Value	→	enum8	def_roam_ind	1	Roaming indicator. Values: • 0x00 – ROAMING_IND_ON – Roaming • 0x01 – ROAMING_IND_OFF – Home Values from 2 onward are applicable only for 3GPP2. Refer to 3GPP2 C.R1001-F for the meanings of these values.
Type	0x17			1	3GPP2 Time Zone
Length	3			2	
Value	→	uint8	lp_sec	1	Number of leap seconds since the start of CDMA system time.
		Int8	ltm_offset	1	Offset of local time from system time in units of 30 min. The value in this field conveys the offset as an 8-bit two's complement number.
		Boolean	daylt_savings	1	Daylight saving indicator. Values: • 0x00 – OFF (daylight saving not in effect) • 0x01 – ON (daylight saving in effect)
Type	0x18			1	CDMA P_Rev in Use
Length	1			2	
Value	→	uint8	p_rev_in_use	1	P_Rev that is currently in use.
Type	0x19			1	3GPP PLMN Name Flag
Length	1			2	
Value	→	boolean	plmn_description_changed	1	Flag indicating that the 3GPP EONS network description changed. Values: • 0x01 – PLMN name changed
Type	0x1A			1	3GPP Time Zone
Length	1			2	
Value	→	int8	time_zone	1	Offset from Universal time, i.e., difference between local time and Universal time, in increments of 15 min (signed value).
Type	0x1B			1	3GPP Network Daylight Saving Adjustment
Length	1			2	
Value	→	uint8	adj	1	3GPP network daylight saving adjustment. Values: • 0x00 – No adjustment for Daylight Saving Time • 0x01 – 1 hr adjustment for Daylight Saving Time • 0x02 – 2 hr adjustment for Daylight Saving Time
Type	0x1C			1	3GPP Universal Time and Local Time Zone
Length	8			2	
Value	→	uint16	year	2	Year.
		Uint8	month	1	Month.
		Uint8	day	1	Day.
		Uint8	hour	1	Hour.
		Uint8	minute	1	Minute.
		Uint8	second	1	Second.
		Int8	time_zone	1	Offset from Universal time, i.e., difference between local time and Universal time, in increments of 15 min (signed value).
Type	0x1D			1	3GPP Location Area Code
Length	2			2	

Value	→	uint16	lac	2	Location area code.
Type	0x1E			1	3GPP Cell ID
Length	4			2	
Value	→	uint32	cell_id	4	3GPP cell ID.
Type	0x1F			1	3GPP2 Concurrent Service Info
Length	1			2	
Value	→	uint8	ccs	1	3GPP2 concurrent service information. Values: • 0x00 – Concurrent service not available • 0x01 – Concurrent service available
Type	0x20			1	3GPP2 PRL Indicator
Length	1			2	
Value	→	uint8	prl_ind	1	3GPP2 PRL indicator. Values: • 0x00 – System not in PRL • 0x01 – System is in PRL
Type	0x21			1	Dual Transfer Mode Indication (GSM Only)
Length	1			2	
Value	→	uint8	dtm_ind	1	Dual Transfer mode indication. Values: • 0x00 – DTM not supported • 0x01 – DTM supported
Type	0x22			1	Detailed Service Information
Length	5			2	
Value	→	uint8	srv_status	1	Service status. Values: • 0x00 – No service • 0x01 – Limited service • 0x02 – Service available • 0x03 – Limited regional service • 0x04 – MS in power save or deep sleep
	uint8	srv_capability		1	System's service capability. Values: • 0x00 – No service • 0x01 – Circuit-switched only • 0x02 – Packet-switched only • 0x03 – Circuit-switched and-packet switched • 0x04 – MS found the right system but not yet registered/attached
	uint8	hdr_srv_status		1	HDR service status. Values: • 0x00 – No service • 0x01 – Limited service • 0x02 – Service available • 0x03 – Limited regional service • 0x04 – MS in power save or deep sleep
	uint8	hdr_hybrid		1	HDR hybrid information. Values: • 0x00 – System is not hybrid • 0x01 – System is hybrid
	uint8	is_sys_forbidden		1	Forbidden system information. Values: • 0x00 – System is not a forbidden system • 0x01 – System is a forbidden system
Type	0x23			1	CDMA System Info Ext
Length	3			2	
Value	→	uint16	mcc	2	Mobile country code.
	Uint8	imsi_11_12		1	IMSI_11_12.
Type	0x24			1	HDR Personality
Length	1			2	

Value	→	enum8	hdr_personality	1	HDR personality information. Values: • 0x00 – Unknown • 0x01 – HRPD • 0x02 – eHRPD
Type	0x25			1	TAC Information for LTE
Length	2			2	
Value	→	uint16	tac	2	Tracking area code information for LTE.
Type	0x26			1	Call Barring Status
Length	8			2	
Value	→	enum	cs_bar_status	4	Call barring status for circuit-switched calls. Values: • 0x00 – NAS_CELL_ACCESS_NORMAL_ONLY – Cell access is allowed for normal calls only • 0x01 – NAS_CELL_ACCESS_EMERGENCY_ONLY – Cell access is allowed for emergency calls only • 0x02 – NAS_CELL_ACCESS_NO_CALLS – Cell access is not allowed for any call type • 0x03 – NAS_CELL_ACCESS_ALL_CALLS – Cell access is allowed for all call types • -1 – NAS_CELL_ACCESS_UNKNOWN – Cell access type is unknown
		enum	ps_bar_status	4	Call barring status for packet-switched calls. Values: • 0x00 – NAS_CELL_ACCESS_NORMAL_ONLY – Cell access is allowed for normal calls only • 0x01 – NAS_CELL_ACCESS_EMERGENCY_ONLY – Cell access is allowed for emergency calls only • 0x02 – NAS_CELL_ACCESS_NO_CALLS – Cell access is not allowed for any call type • 0x03 – NAS_CELL_ACCESS_ALL_CALLS – Cell access is allowed for all call types • -1 – NAS_CELL_ACCESS_UNKNOWN – Cell access type is unknown
Type	0x27			1	PLMN Change Status
Length	1			2	
Value	→	boolean	srv_sys_no_change	1	Flag used to notify clients that a request to select a network ended with no change in the PLMN. Values: • 0x01 – No change in serving system information
Type	0x28			1	UMTS Primary Scrambling Code
Length	2			2	
Value	→	uint16	umts_psc	2	Primary scrambling code.
Type	0x29			1	MNC PCS Digit Include Status
Length	5			2	
Value	→	uint16	mcc	2	A 16-bit integer representation of MCC.

					Range: 0 to 999.
	Uint16	mnc	2	A 16-bit integer representation of MNC. Range: 0 to 999.	
	Boolean	mnc_includes_pcs_digit	1	This field is used to interpret the length of the corresponding MNC reported in the TLVs (in this table) with an mnc or mobile_network_code field. Values: <ul style="list-style-type: none"> • TRUE – MNC is a three-digit value; e.g., a reported value of 90 corresponds to an MNC value of 090 • FALSE – MNC is a two-digit value; e.g., a reported value of 90 corresponds to an MNC value of 90 	
Type	0x2A		1	HS Call Status	
Length	1		2		
Value	→	enum8	hs_call_status	1	<p>Call status on high speed (only applicable for WCDMA). Values:</p> <ul style="list-style-type: none"> • SYS_HS_IND_HSDPA_HSUPA_UNSUPP_CELL (0x00) – HSDPA and HSUPA are unsupported • SYS_HS_IND_HSDPA_SUPP_CELL (0x01) – HSDPA is supported • SYS_HS_IND_HSUPA_SUPP_CELL (0x02) – HSUPA is supported • SYS_HS_IND_HSDPA_HSUPA_SUPP_CELL (0x03) – HSDPA and HSUPA are supported • SYS_HS_IND_HSDPAPLUS_SUPP_CELL (0x04) – HSDPA+ is supported • SYS_HS_IND_HSDPAPLUS_HSUPA_SUPP_CELL (0x05) – HSDPA+ and HSUPA are supported • SYS_HS_IND_DC_HSDPAPLUS_SUPP_CELL (0x06) – Dual-cell HSDPA+ is supported • SYS_HS_IND_DC_HSDPAPLUS_HSUPA_SUPP_CELL (0x07) – Dual-cell HSDPA+ and HSUPA are supported • SYS_HS_IND_HSDPAPLUS_64QAM_HSUPA_SUPP_CELL (0x08) – Dual-cell HSDPA+, 64 QAM, and HSUPA are supported • SYS_HS_IND_HSDPAPLUS_64QAM_SUPP_CELL (0x09) – Dual-cell HSDPA+ and 64 QAM are supported • SYS_HS_IND_DC_HSDPAPLUS_DC_HSUPA_SUPP_CELL (0x0A) – Dual-cell HSDPA+ and dual-cell HSUPA are supported

Type	0x2B			1	3GPP Network Name Source
Length	4			2	
Value	→	enum	nas_3gpp_nw_name_source	4	<p>Network name source. Values:</p> <ul style="list-style-type: none"> • NAS_NW_NAME_SOURCE_UNKNOWN (0x00) – Unknown • NAS_NW_NAME_SOURCE_OPL_PNN (0x01) <ul style="list-style-type: none"> – Operator PLMN list and PLMN network name • NAS_NW_NAME_SOURCE_CPHS_ONS (0x02) – Common PCN handset specification and operator name string • NAS_NW_NAME_SOURCE_NITZ (0x03) <ul style="list-style-type: none"> – Network identity and time zone • NAS_NW_NAME_SOURCE_SE13 (0x04) <ul style="list-style-type: none"> – GSMA SE13 table • NAS_NW_NAME_SOURCE_MCC_MNC (0x05) – Mobile country code and mobile network code • NAS_NW_NAME_SOURCE_SPN (0x06) – Service provider name

7.2.11.2. Description of QMI_NAS_SERVING_SYSTEM_IND

This broadcast indication is sent (intended for all control points) when the current serving system registration state and/or radio technology changes.

The roaming indicator and the current PLMN are not included in the response when the device is not registered.

If registered on the 3GPP network and time zone, and relevant information has been sent from the network, the 3GPP Universal Time and Local Time Zone, 3GPP Time Zone, and/or 3GPP Network Daylight Saving Adjustment TLVs are included.

If registered, the presence of the optional 3GPP PLMN Name Flag TLV indicates that the operator name may have changed. The QMI_NAS_GET_PLMN_NAME command must be used to query the updated network name for the current PLMN.

The 3GPP Location Area Code and 3GPP Cell ID TLVs are included if the UE is registered on the 3GPP network. The 3GPP2 Concurrent Service Info and 3GPP2 PRL Indicator TLVs are included if the UE is registered on the 3GPP2 network. The Dual Transfer Mode Indication TLV is included if the UE is registered on the GSM network. The Detailed Service Information TLV is included so clients can retrieve detailed information about the Call Manager layer to fine-tune their internal states.

The Call Barring Status TLV is included only in GSM or WCDMA networks.

The AT command equivalent to this command is **AT+CSS**, defined in 3GPP2 C.S0017-003-A and TIA/EIA/IS-131, and **AT+CREG** is defined in 3GPP TS 27.007.

7.2.12. QMI_NAS_SET_SYSTEM_SELECTION_PREF

(QMI_NAS_SET_SYSTEM_SELECTION)

Sets the different system selection preferences of the device.

- **NAS message ID**

0x0033

- **Version introduced**

Major – 1, Minor – 1

7.2.12.1. Request – QMI_NAS_SET_SYSTEM_SELECTION_PREFERENCE_REQ_MSG

- **Message type**

Request

- **Sender**

Control point

- **Mandatory TLVs**

None

- Optional TLVs

Type	0x10			1	Emergency Mode
Length	1			2	
Value	→	boolean	emergency_mode	1	Values: • 0x00 – OFF (normal) • 0x01 – ON (emergency)
Type	0x11			1	Mode Preference
Length	2			2	
Value	→	mask16	mode_pref	2	Bitmask representing the radio technology mode preference to be set. Values: • Bit 0 (0x01) – QMI_NAS_RAT_MODE_PREF_CDMA2000_1X – cdma2000® 1X • Bit 1 (0x02) – QMI_NAS_RAT_MODE_PREF_CDMA2000_HRPD – cdma2000® HRPD (1xEV-DO) • Bit 2 (0x04) – QMI_NAS_RAT_MODE_PREF_GSM – GSM • Bit 3 (0x08) – QMI_NAS_RAT_MODE_PREF_UMTS – UMTS • Bit 4 (0x10) – QMI_NAS_RAT_MODE_PREF_LTE – LTE • Bit 5 (0x20) – QMI_NAS_RAT_MODE_PREF_TDSCDMA – TD-SCDMA All unlisted bits are reserved for future use and the service point ignores them if used.
Type	0x12			1	Band Preference
Length	8			2	
Value	→	mask	band_pref	8	Bitmask representing the band preference to be set. See TableA-2for details.
Type	0x13			1	CDMA PRL Preference
Length	2			2	
Value	→	enum16	prl_pref	2	PRL preference to be set for band class 0 (BC0) prl_pref. Values: • 0x0001 – PRL_PREF_A_SIDE_ONLY – Acquire available system only on the A side • 0x0002 – PRL_PREF_B_SIDE_ONLY – Acquire available system only on the B side • 0xFFFF – PRL_PREF_ANY – Acquire any available systems
Type	0x14			1	Roaming Preference
Length	2			2	
Value	→	enum16	roam_pref	2	Roaming preference to be set. Values: • 0x01 – ROAMING_PREF_OFF – Acquire only systems for which the roaming indicator is off • 0x02 – ROAMING_PREF_NOT_OFF – Acquire a system as long as its roaming indicator is not off • 0x03 – ROAMING_PREF_NOT_FLASING

					- Acquire only systems for which the roaming indicator is off or solid on, i.e., not flashing; CDMA only • 0xFF – ROAMING_PREF_ANY – Acquire systems, regardless of their roaming indicator
Type	0x15			1	LTE Band Preference (Deprecated; use LTE Band Preference Extended)
Length	8			2	
Value	→	mask	lte_band_pref	8	Bitmask representing the LTE band preference to be set. See TableA-3for details.
Type	0x16			1	Network Selection Preference
Length	5			2	
Value	→	enum8	net_sel_pref	1	Specifies one of the following actions: • 0x00 – NAS_NET_SEL_PREF_AUTOMATIC – Device registers according to its provisioning; mcc and mnc fields must also contain valid values if Radio Access Technology (TLV 0x22) is present. Otherwise, mcc and mnc are ignored. • 0x01 – NAS_NET_SEL_PREF_MANUAL – Device registers to specified network; mcc and mnc fields must also contain valid values. All other values are reserved.
	Uint16	mcc		2	A 16-bit integer representation of MCC. Range: 0 to 999.
	Uint16	mnc		2	A 16-bit integer representation of MNC. Range: 0 to 999.
Type	0x17			1	Change Duration
Length	1			2	
Value	→	enum8	change_duration	1	Duration of the change. Values: • 0x00 – Power cycle – Remains active until the next device power cycle • 0x01 – Permanent – Remains active through power cycles until changed by the client Note: The device will use “0x01 – Permanent” as the default value if the TLV is omitted.
Type	0x18			1	Service Domain
Length	4			2	
Value	→	enum	srv_domain_pre	f 4	Service domain preference. Values: • QMI_SRV_DOMAIN_PREF_CS_ONLY (0x00) – Circuit-switched only • QMI_SRV_DOMAIN_PREF_PS_ONLY (0x01) – Packet-switched only • QMI_SRV_DOMAIN_PREF_CS_PS (0x02) – Circuit-switched and packet-switched • QMI_SRV_DOMAIN_PREF_PS_ATTACH (0x03) – Packet-switched attach • QMI_SRV_DOMAIN_PREF_PS_DETACH (0x04) – Packet-switched detach • QMI_SRV_DOMAIN_PREF_PS_DETACH_NO_PREF_CHANGE (0x05) – Packet-switched

Name	Version introduced	Version last modified
Emergency Mode	Unknown	1.1
Mode Preference	Unknown	1.16
Band Preference	Unknown	1.16
CDMA PRL Preference	Unknown	1.1
Roaming Preference	Unknown	1.1
LTE Band Preference (Deprecated; use LTE Band Preference Extended)	1.16	1.138 (Deprecated)
Network Selection Preference	1.5	1.69
Change Duration	Unknown	1.5
Service Domain	1.34	1.121
GSM/WCDMA Acquisition Order	Unknown	1.11
MNC PCS Digit Include Status	Unknown	1.10
Service Domain Preference	1.34	1.121
GSM/WCDMA Acquisition Order Preference	Unknown	1.11
TDSCDMA Band Preference	Unknown	1.13
Acquisition Order Preference	Unknown	1.20
Network Selection Registration Restriction Preference	1.34	1.34
CSG ID	1.41	1.41
Usage Preference	1.67	1.67
Radio Access Technology	1.69	1.69
Voice Domain Preference	1.92	1.92
LTE Band Preference Extended	1.138	1.138
Force Preferences	1.140	1.140

Field	Field value	Field type	Parameter	Size (byte)	Description

					<ul style="list-style-type: none"> • QMI_SRV_DOMAIN_PREF_ON_DEMAND_PS_ATTACH (0x06) – Packet-switched detach with no change in the service domain preference • QMI_SRV_DOMAIN_PREF_FORCE_PS_DETACH (0x07) – Packet-switched detach where PS service loss is done forcibly by the modem.
Type	0x19			1	GSM/WCDMA Acquisition Order
Length	4			2	
Value	→	enum	gw_acq_order_pref	4	<p>GSM/WCDMA acquisition order preference. Values:</p> <ul style="list-style-type: none"> • 0x00 – NAS_GW_ACQ_ORDER_PREF_AUTOMATIC – Automatic • 0x01 – NAS_GW_ACQ_ORDER_PREF_GSM_WCDMA – GSM then WCDMA • 0x02 – NAS_GW_ACQ_ORDER_PREF_WCDMA_GSM – WCDMA then GSM
Type	0x1A			1	MNC PCS Digit Include Status
Length	1			2	
Value	→	boolean	mnc_includes_pcs_digit	1	<p>This field is used to interpret the length of the corresponding MNC reported in the Network Selection Preference TLV (0x16). Values:</p> <ul style="list-style-type: none"> • TRUE – MNC is a three-digit value; e.g., a reported value of 90 corresponds to an MNC value of 090 • FALSE – MNC is a two-digit value; e.g., a reported value of 90 corresponds to an MNC value of 90
Type	0x1B			1	Service Domain Preference (duplicate of 0x18)
Length	0			2	
Value	→	duplicate	srv_domain_pref	0	Duplicate of Service Domain Preference
Type	0x1C			1	GSM/WCDMA Acquisition Order Preference (duplicate of 0x19)
Length	0			2	
Value	→	duplicate	gw_acq_order_pref	0	<p>GSM/WCDMA acquisition order preference. Values:</p> <ul style="list-style-type: none"> • 0x00 – NAS_GW_ACQ_ORDER_PREF_AUTOMATIC – Automatic • 0x01 – NAS_GW_ACQ_ORDER_PREF_GSM_WCDMA – GSM then WCDMA • 0x02 – NAS_GW_ACQ_ORDER_PREF_WCDMA_GSM – WCDMA then GSM
Type	0x1D			1	TDSCDMA Band Preference
Length	8			2	

Value	→	mask	tdscdma_band_pref	8	Bitmask representing the TD-SCDMA band preference to be set. Values: • 0x01 – NAS_TDSCDMA_BAND_A – TD-SCDMA Band A • 0x02 – NAS_TDSCDMA_BAND_B – TD-SCDMA Band B • 0x04 – NAS_TDSCDMA_BAND_C – TD-SCDMA Band C • 0x08 – NAS_TDSCDMA_BAND_D – TD-SCDMA Band D • 0x10 – NAS_TDSCDMA_BAND_E – TD-SCDMA Band E • 0x20 – NAS_TDSCDMA_BAND_F – TD-SCDMA Band F All other bits are reserved.
Type	0x1E				1
Length	Var				2
Value	→	uint8	acq_order_len	1	Number of sets of the following elements: • acq_order
		enum8	acq_order	Var	Acquisition order preference to be set. Values: • 0x01 – NAS_RADIO_IF_CDMA_1X – cdma2000® 1X • 0x02 – NAS_RADIO_IF_CDMA_1XEVDO – cdma2000® HRPD (1xEV-DO) • 0x04 – NAS_RADIO_IF_GSM – GSM • 0x05 – NAS_RADIO_IF_UTMS – UMTS • 0x08 – NAS_RADIO_IF_LTE – LTE • 0x09 – NAS_RADIO_IF_TDSCDMA – TD-SCDMA
Type	0x1F				1
Length	4				2
Value	→	enum	srv_reg_restriction	4	Registration restriction preference. Specifies one of the following modifiers to net_sel_pref: • 0x00 – NAS_SRV_REG_RESTRICTION_UNRESTRICTED – Device follows the normal registration process • 0x01 – NAS_SRV_REG_RESTRICTION_CAMPED_ONLY – Device camps on the network according to its provisioning, but does not register • 0x02 – NAS_SRV_REG_RESTRICTION_LIMITED – Device selects the network for limited service All other values are reserved.
Type	0x20				1
Length	10				2
Value	→	uint16	mcc	2	A 16-bit integer representation of CSG MCC. Range: 0 to 999.
		Uint16	mnc	2	A 16-bit integer representation of CSG MNC. Range: 0 to 999.
		Boolean	mnc_includes_pcs_digit	1	This field is used to interpret the length of the corresponding MNC reported in the TLVs (in

					this table) with an mnc or mobile_network_code field. Values: • TRUE – MNC is a three-digit value; e.g., a reported value of 90 corresponds to an MNC value of 090 • FALSE – MNC is a two-digit value; e.g., a reported value of 90 corresponds to an MNC value of 90
	uint32	id	4	Closed subscriber group identifier.	
	Enum8	rat	1	Radio interface technology of the CSG network. Values: • 0x04 – RADIO_IF_GSM – GSM • 0x05 – RADIO_IF_UTMS – UMTS • 0x08 – RADIO_IF_LTE – LTE • 0x09 – RADIO_IF_TDSCDMA – TDS	
Type	0x21		1	Usage Preference	
Length	4		2		
Value	→	enum	usage_setting	4	Modem usage preference to be set. Values: • NAS_USAGE_VOICE_CENTRIC (1) – Voice centric • NAS_USAGE_DATA_CENTRIC (2) – Data centric
Type	0x22		1	Radio Access Technology	
Length	1		2		
Value	→	enum8	rat	1	Radio access technology for the corresponding PLMN ID in the Network Selection Preference TLV (0x16). If this TLV is present and the net_sel_pref field is set to automatic, the provided MCC, MNC, and RAT are searched for first. If they are not found, the selection falls back to automatic. This TLV can also be used with the net_sel_pref field set to manual to indicate the RAT of the specified MCC and MNC. Values: • 0x04 – NAS_RADIO_IF_GSM – GSM • 0x05 – NAS_RADIO_IF_UTMS – UMTS • 0x08 – NAS_RADIO_IF_LTE – LTE • 0x09 – NAS_RADIO_IF_TDSCDMA – TD-SCDMA
Type	0x23		1	Voice Domain Preference	
Length	4		2		
Value	→	enum	voice_domain_pref	4	Voice domain preference to be set. Values: • NAS_VOICE_DOMAIN_PREF_CS_ONLY (0x00) – Circuit-switched (CS) voice only • NAS_VOICE_DOMAIN_PREF_PS_ONLY (0x01) – Packet-switched (PS) voice only • NAS_VOICE_DOMAIN_PREF_CS_PREF (0x02) – CS is preferred; PS is secondary • NAS_VOICE_DOMAIN_PREF_PS_PREF (0x03) – PS is preferred; CS is secondary
Type	0x24		1	LTE Band Preference Extended	

Length	32			2	
Value	→	uint64	bits_1_64	8	Bits 1 to 64 of the 256-bit LTE E-UTRA Operating Band bitmask
		uint64	bits_65_128	8	Bits 65 to 128 of the 256-bit LTE E-UTRA Operating Band bitmask
		uint64	bits_129_192	8	Bits 129 to 192 of the 256-bit LTE E-UTRA Operating Band bitmask
		uint64	bits_193_256	8	Bits 193 to 256 of the 256-bit LTE E-UTRA Operating Band bitmask
Type	0x25			1	Force Preferences
Length	1			2	
Value	→	boolean	force	1	When TRUE, indicates that the UE cannot process the request due to an LPM transition, the lower layer is busy, etc. The request is buffered and processed as soon as possible instead of returning an error. The default value is FALSE.

7.2.12.2. Response – QMI_NAS_SET_SYSTEM_SELECTION_PREFERENCE_RESP_MSG

- **Message type**

Response

- **Sender**

Service

- **Mandatory TLVs**

The Result Code TLV (defined in Section 7.1.3.1) 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
At least one of the following optional TLVs must be included in this request.	Operation is not supported by the device
QMI_ERR_INVALID_ARG	Value field of one or more TLVs in the request message contains an invalid value
QMI_ERR_MISSING_ARG	One or more required TLVs were missing in the request
QMI_ERR_INVALID_OPERATION	Operation is not supported by the device

7.2.12.3. Description of QMI_NAS_SET_SYSTEM_SELECTION_PREFERENCE REQ/RESP

This command writes the specified system selection preference to the device. This setting is global to the device and is not unique to each control point. The preference is written to persistent storage to remain set after the device is power cycled.

A system selection preference is overwritten by a subsequent request to set the system selection preference.

Requests to set an invalid system selection preference for the current device configuration elicit a QMI_ERR_OP_DEVICE_UNSUPPORTED error.

Success of this command indicates that the specified change has been requested. The control point must always process the QMI_NAS_SYSTEM_SELECTION_PREFERENCE_IND indication to learn the current system selection of the device.

At least one optional TLV specifying a system selection preference must be present in the request. If not, a QMI_ERR_MISSING_ARG error is returned.

The control point must include the Emergency Mode TLV with a value set to ON if users want to enable Emergency mode. All other TLVs included in the command are ignored. To exit Emergency mode, the control point can either include the Emergency Mode TLV with a value set to OFF or include the Mode Preference TLV. When coming out of Emergency mode, the mode preference is set to whatever the Mode Preference TLV specifies (if the TLV is included) or to whatever mode preference that is set in persistent memory (if the Mode Preference TLV is not included).

When the Network Selection Preference TLV (0x16) is included, its information is used to control which networks the modem selects.

The Acquisition Order Preference TLV (0x1E) takes priority over the GSM/WCDMA Acquisition Order Preference TLV (0x1C); if both are sent, the Acquisition Order Preference TLV is used. If the Acquisition Order Preference TLV is not supported, a QMI_ERR_INVALID_OPERATION error is returned. Only the listed radio interfaces are supported. If a different radio interface is sent, or there are duplicates in the list, a QMI_ERR_INVALID_ARG error is returned.

The acquisition order preference list contains a list of RATs (1X, 1xEV-DO, .., LTE, TD-SCDMA, etc.). When the client attempts to change its order, the client must provide a list that contains the same RATs but in a different order. A RAT that was there previously cannot be removed and a new RAT cannot be added.

When the optional Service Domain Preference TLV (0x18) is sent as QMI_SRV_DOMAIN_PREF_PS_ATTACH, PS will be added to the current preference. If PS is already in the service domain preference, the request for the attach returns an error. If the TLV is sent as QMI_SRV_DOMAIN_PREF_PS_DETACH, PS is removed from the current preference. If the device was

already PS_ONLY, the UE moves to Power Save mode. A value of QMI_SRV_DOMAIN_PREF_PS_DETACH_NO_PREF_CHANGE performs the PS detach without modifying the service domain preference.

The optional Network Selection Registration Restriction Preference TLV (0x1F) is used to put the device into or pull it out of Limited mode or Camped Only mode. When using this TLV with a value other than NAS_SRV_REG_RESTRICTION_UNRESTRICTED, the Change Duration TLV (0x17) must be set to “Power cycle”.

Only one of the LTE Band Preference TLVs, either TLV 0x15 (deprecated) or TLV 0x24, can be included in the request; otherwise, a QMI_ERR_INVALID_ARG error is returned.

7.2.13. QMI_NAS_GET_SYSTEM_SELECTION_PREFERENCE

Queries the different system selection preferences of the device.

- **NAS message ID**

0x0034

- **Version introduced**

Major - 1, Minor - 1

7.2.13.1. Request-QMI_NAS_GET_SYSTEM_SELECTION_PREFERENCE_REQ_MSG

- **Message type**

Request

- **Sender**

Control point

- **Mandatory TLVs**

None

- **Optional TLVs**

None

7.2.13.2. Response - QMI_NAS_GET_SYSTEM_SELECTION_PREFERENCE_RESP_MSG

- **Message type**

Response

- **Sender**

Service

- **Mandatory TLVs**

The Result Code TLV (defined in Section 7.1.3.1) is always present in the response.

- **Optional TLVs**

At least one of the following optional TLVs are present if the result code is QMI_RESULT_SUCCESS.

Name	Version introduced	Version last modified
Emergency Mode	Unknown	1.1
Mode Preference	Unknown	1.16
Band Preference	Unknown	1.16
CDMA PRL Preference	Unknown	1.1
Roaming Preference	Unknown	1.1
LTE Band Preference (Deprecated; use LTE Band Preference Extended)	Unknown	1.138 (Deprecated)
Network Selection Preference	1.5	1.5
Service Domain Preference	Unknown	1.34
GSM/WCDMA Acquisition Order Preference	Unknown	1.11
TDSCDMA Band Preference	Unknown	1.13
Manual Network Selection PLMN	Unknown	1.19
Acquisition Order Preference	Unknown	1.20
Network Selection Registration Restriction Preference	1.34	1.34
CSG ID	1.41	1.41
Usage Preference	1.67	1.67
Voice Domain Preference	1.92	1.92
LTE Disable Cause	1.100	1.100
Disabled RAT Bitmask	1.132	1.132
LTE Band Preference Extended	1.138	1.138

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Emergency Mode
Length	1			2	
Value	→	boolean	emergency_mode	1	Values: • 0x00 – OFF (normal) • 0x01 – ON (emergency)
Type	0x11			1	Mode Preference
Length	2			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	mask16	mode_pref	2	<p>Bitmask representing the radio technology mode preference to be set. Values:</p> <ul style="list-style-type: none"> • Bit 0 (0x01) – QMI_NAS_RAT_MODE_PREF_CDMA2000_1X – cdma2000® 1X • Bit 1 (0x02) – QMI_NAS_RAT_MODE_PREF_CDMA2000_HRPD – cdma2000® HRPD (1xEV-DO) • Bit 2 (0x04) – QMI_NAS_RAT_MODE_PREF_GSM – GSM • Bit 3 (0x08) – QMI_NAS_RAT_MODE_PREF_UMTS – UMTS • Bit 4 (0x10) – QMI_NAS_RAT_MODE_PREF_LTE – LTE • Bit 5 (0x20) – QMI_NAS_RAT_MODE_PREF_TDSCDMA – TD-SCDMA <p>All unlisted bits are reserved for future use and the service point ignores them if used.</p>
Type	0x12			1	Band Preference
Length	8			2	
Value	→	mask	band_pref	8	Bitmask representing the band preference to be set. See Table A-2 for details.
Type	0x13			1	CDMA PRL Preference
Length	2			2	
Value	→	enum16	prl_pref	2	<p>PRL preference to be set for band class 0 (BC0) prl_pref. Values:</p> <ul style="list-style-type: none"> • 0x0001 – PRL_PREF_A_SIDE_ONLY – Acquire available system only on the A side • 0x0002 – PRL_PREF_B_SIDE_ONLY – Acquire available system only on the B side • 0x3FFF – PRL_PREF_ANY – Acquire any available systems
Type	0x14			1	Roaming Preference
Length	2			2	
Value	→	enum16	roam_pref	2	<p>Roaming preference to be set. Values:</p> <ul style="list-style-type: none"> • 0x01 – ROAMING_PREF_OFF – Acquire only systems for which the roaming indicator is off • 0x02 – ROAMING_PREF_NOT_OFF – Acquire a system as long as its roaming indicator is not off • 0x03 – ROAMING_PREF_NOT_FLASHING – Acquire only systems for which the roaming indicator is off or solid on, i.e., not flashing; CDMA only • 0xFF – ROAMING_PREF_ANY – Acquire systems, regardless of their roaming indicator
Type	0x15			1	LTE Band Preference (Deprecated; use LTE Band Preference Extended)
Length	8			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	uint64	band_pref_ext	8	<p>Bitmask representing the LTE band preference to be set. Values:</p> <ul style="list-style-type: none"> • Bit 0 – E-UTRA Operating Band 1 • Bit 1 – E-UTRA Operating Band 2 • Bit 2 – E-UTRA Operating Band 3 • Bit 3 – E-UTRA Operating Band 4 • Bit 4 – E-UTRA Operating Band 5 • Bit 5 – E-UTRA Operating Band 6 • Bit 6 – E-UTRA Operating Band 7 • Bit 7 – E-UTRA Operating Band 8 • Bit 8 – E-UTRA Operating Band 9 • Bit 9 – E-UTRA Operating Band 10 • Bit 10 – E-UTRA Operating Band 11 • Bit 11 – E-UTRA Operating Band 12 • Bit 12 – E-UTRA Operating Band 13 • Bit 13 – E-UTRA Operating Band 14 • Bit 16 – E-UTRA Operating Band 17 • Bit 17 – E-UTRA Operating Band 18 • Bit 18 – E-UTRA Operating Band 19 • Bit 19 – E-UTRA Operating Band 20 • Bit 20 – E-UTRA Operating Band 21 • Bit 23 – E-UTRA Operating Band 24 • Bit 24 – E-UTRA Operating Band 25 • Bit 32 – E-UTRA Operating Band 33 • Bit 33 – E-UTRA Operating Band 34 • Bit 34 – E-UTRA Operating Band 35 • Bit 35 – E-UTRA Operating Band 36 • Bit 36 – E-UTRA Operating Band 37 • Bit 37 – E-UTRA Operating Band 38 • Bit 38 – E-UTRA Operating Band 39 • Bit 39 – E-UTRA Operating Band 40 • Bit 40 – E-UTRA Operating Band 41 • Bit 41 – E-UTRA Operating Band 42 • Bit 42 – E-UTRA Operating Band 43 <p>All other bits are reserved.</p>
Type	0x16			1	Network Selection Preference
Length	1			2	
Value	→	enum8	net_sel_pref	1	<p>Network selection preference. Values:</p> <ul style="list-style-type: none"> • 0x00 – Automatic network selection • 0x01 – Manual network selection
Type	0x18			1	Service Domain Preference
Length	4			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	enum	srv_domain_pref	4	<p>Service domain preference. Values:</p> <ul style="list-style-type: none"> • 0x00 – QMI_SRV_DOMAIN_PREF_CS_ONLY – Circuit-switched only • 0x01 – QMI_SRV_DOMAIN_PREF_PS_ONLY – Packet-switched only • 0x02 – QMI_SRV_DOMAIN_PREF_CS_PS – Circuit-switched and packet-switched
Type	0x19			1	GSM/WCDMA Acquisition Order Preference
Length	4			2	
Value	→	enum	gw_acq_order_pref	4	<p>GSM/WCDMA acquisition order preference. Values:</p> <ul style="list-style-type: none"> • 0x00 – NAS_GW_ACQ_ORDER_PREF_AUTOMATIC – Automatic • 0x01 – NAS_GW_ACQ_ORDER_PREF_GSM_WCDMA – GSM then WCDMA • 0x02 – NAS_GW_ACQ_ORDER_PREF_WCDMA_GSM – WCDMA then GSM
Type	0x1A			1	TDSCDMA Band Preference
Length	8			2	
Value	→	mask	tdscdma_band_pref	8	<p>Bitmask representing the TD-SCDMA band preference to be set. Values:</p> <ul style="list-style-type: none"> • 0x01 – NAS_TDSCDMA_BAND_A – TD-SCDMA Band A • 0x02 – NAS_TDSCDMA_BAND_B – TD-SCDMA Band B • 0x04 – NAS_TDSCDMA_BAND_C – TD-SCDMA Band C • 0x08 – NAS_TDSCDMA_BAND_D – TD-SCDMA Band D • 0x10 – NAS_TDSCDMA_BAND_E – TD-SCDMA Band E • 0x20 – NAS_TDSCDMA_BAND_F – TD-SCDMA Band F <p>All other bits are reserved.</p>
Type	0x1B			1	Manual Network Selection PLMN
Length	5			2	
Value	→	uint16	mcc	2	A 16-bit integer representation of MCC. Range: 0 to 999.
			mnc	2	A 16-bit integer representation of MNC. Range: 0 to 999.

Field	Field value	Field type	Parameter	Size (byte)	Description
		boolean	mnc_includes_pcs_digit	1	<p>This field is used to interpret the length of the corresponding MNC reported in the TLVs (in this table) with an mnc or mobile_network_code field.</p> <p>Values:</p> <ul style="list-style-type: none"> • TRUE – MNC is a three-digit value; e.g., a reported value of 90 corresponds to an MNC value of 090 • FALSE – MNC is a two-digit value; e.g., a reported value of 90 corresponds to an MNC value of 90
Type	0x1C			1	Acquisition Order Preference
Length	Var			2	
Value	→	uint8	acq_order_len	1	<p>Number of sets of the following elements:</p> <ul style="list-style-type: none"> • acq_order
		enum8	acq_order	Var	<p>Acquisition order preference to be set. Values:</p> <ul style="list-style-type: none"> • 0x01 – NAS_RADIO_IF_CDMA_1X – cdma2000® 1X • 0x02 – NAS_RADIO_IF_CDMA_1XEVDO – cdma2000® HRPD (1xEV-DO) • 0x04 – NAS_RADIO_IF_GSM – GSM • 0x05 – NAS_RADIO_IF_UMTS – UMTS • 0x08 – NAS_RADIO_IF_LTE – LTE • 0x09 – NAS_RADIO_IF_TDSCDMA – TD-SCDMA
Type	0x1D			1	Network Selection Registration Restriction Preference
Length	4			2	
Value	→	enum	srv_reg_restriction	4	<p>Registration restriction preference. Specifies one of the following modifiers to net_sel_pref:</p> <ul style="list-style-type: none"> • 0x00 – NAS_SRV_REG_RESTRICTION_UNRESTRICTED – Device follows the normal registration process • 0x01 – NAS_SRV_REG_RESTRICTION_CAMPED_ONLY – Device camps on the network according to its provisioning, but does not register • 0x02 – NAS_SRV_REG_RESTRICTION_LIMITED – Device selects the network for limited service <p>All other values are reserved.</p>
Type	0x1E			1	CSG ID
Length	10			2	
Value	→	uint16	mcc	2	A 16-bit integer representation of CSG MCC. Range: 0 to 999.
		uint16	mnc	2	A 16-bit integer representation of CSG MNC. Range: 0 to 999.

Field	Field value	Field type	Parameter	Size (byte)	Description
		boolean	mnc_includes_pcs_digit	1	<p>This field is used to interpret the length of the corresponding MNC reported in the TLVs (in this table) with an mnc or mobile_network_code field.</p> <p>Values:</p> <ul style="list-style-type: none"> • TRUE – MNC is a three-digit value; e.g., a reported value of 90 corresponds to an MNC value of 090 • FALSE – MNC is a two-digit value; e.g., a reported value of 90 corresponds to an MNC value of 90
		uint32	id	4	Closed subscriber group identifier.
		enum8	rat	1	<p>Radio interface technology of the CSG network.</p> <p>Values:</p> <ul style="list-style-type: none"> • 0x04 – RADIO_IF_GSM – GSM • 0x05 – RADIO_IF_UMTS – UMTS • 0x08 – RADIO_IF_LTE – LTE • 0x09 – RADIO_IF_TDSCDMA – TDS
Type	0x1F			1	Usage Preference
Length	4			2	
Value	→	enum	usage_setting	4	<p>Modem usage preference to be set. Values:</p> <ul style="list-style-type: none"> • NAS_USAGE_UNKNOWN (0) – Unknown • NAS_USAGE_VOICE_CENTRIC (1) – Voice centric • NAS_USAGE_DATA_CENTRIC (2) – Data centric
Type	0x20			1	Voice Domain Preference
Length	4			2	
Value	→	enum	voice_domain_pref	4	<p>Voice domain preference. Values:</p> <ul style="list-style-type: none"> • NAS_VOICE_DOMAIN_PREF_CS_ONLY (0x00) – Circuit-switched (CS) voice only • NAS_VOICE_DOMAIN_PREF_PS_ONLY (0x01) – Packet-switched (PS) voice only • NAS_VOICE_DOMAIN_PREF_CS_PREF (0x02) – CS is preferred; PS is secondary • NAS_VOICE_DOMAIN_PREF_PS_PREF (0x03) – PS is preferred; CS is secondary
Type	0x21			1	LTE Disable Cause
Length	4			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	enum	lte_disable_cause	4	<p>LTE disable cause. Values:</p> <ul style="list-style-type: none"> • NAS_LTE_DISABLE_CAUSE_NONE (0x00) – LTE is not disabled • NAS_LTE_DISABLE_CAUSE_PERMANENT_DS (0x01) – LTE is disabled by DS permanently, e.g., T3316 expiry • NAS_LTE_DISABLE_CAUSE_TEMP_DS (0x02) – LTE is disabled by DS temporarily • NAS_LTE_DISABLE_CAUSE_DOM_SEL (0x03) – LTE disable procedure is called for domain selection purpose • NAS_LTE_DISABLE_CAUSE_DAM (0x04) – LTE disable procedure is called for device aggression management recovery • NAS_LTE_DISABLE_CAUSE_USER (0x05) – LTE disable procedure is called due to user action, e.g., mode_pref change or PS_DETACH triggered by ATCOP/QMI • NAS_LTE_DISABLE_CAUSE_NO_CHANGE (0x06) – No change in LTE disable cause
Type	0x22			1	Disabled RAT Bitmask
Length	2			2	
Value	→	mask16	rat_disabled_mask	2	<p>Bitmask representing the radio technologies that are disabled. Values:</p> <ul style="list-style-type: none"> • Bit 0 (0x01) – QMI_NAS_RAT_MODE_PREF_CDMA2000_1X – cdma2000® 1X • Bit 1 (0x02) – QMI_NAS_RAT_MODE_PREF_CDMA2000_HRPD – cdma2000® HRPD (1xEV-DO) • Bit 2 (0x04) – QMI_NAS_RAT_MODE_PREF_GSM – GSM • Bit 3 (0x08) – QMI_NAS_RAT_MODE_PREF_UMTS – UMTS • Bit 4 (0x10) – QMI_NAS_RAT_MODE_PREF_LTE – LTE • Bit 5 (0x20) – QMI_NAS_RAT_MODE_PREF_TDSCDMA – TD-SCDMA <p>All unlisted bits are reserved for future use and the service point ignores them if used.</p>
Type	0x23			1	LTE Band Preference Extended
Length	32			2	
Value	→	uint64	bits_1_64	8	Bits 1 to 64 of the 256-bit LTE E-UTRA Operating Band bitmask
			bits_65_128	8	Bits 65 to 128 of the 256-bit LTE E-UTRA Operating Band bitmask
			bits_129_192	8	Bits 129 to 192 of the 256-bit LTE E-UTRA Operating Band bitmask

Field	Field value	Field type	Parameter	Size (byte)	Description
		uint64	bits_193_256	8	Bits 193 to 256 of the 256-bit LTE E-UTRA Operating Band bitmask

7.2.13.3. Indication - QMI_NAS_SYSTEM_SELECTION_PREFERENCE_IND_MSG

- Message type

Indication

- Sender

Service

- Scope

Per control point (unicast)

- Mandatory TLVs

None

- Optional TLVs

At least one of the following optional TLVs are present if the result code is QMI_RESULT_SUCCESS.

Name	Version introduced	Version last modified
Emergency Mode	Unknown	1.1
Mode Preference	Unknown	1.16
Band Preference	Unknown	1.16
CDMA PRL Preference	Unknown	1.1
Roaming Preference	Unknown	1.1
LTE Band Preference (Deprecated; use LTE Band Preference Extended)	1.16	1.138 (Deprecated)
Network Selection Preference	1.5	1.5
Service Domain Preference	Unknown	1.34
GSM/WCDMA Acquisition Order Preference	Unknown	1.11
TDSCDMA Band Preference	Unknown	1.13
Manual Network Selection PLMN	Unknown	1.19
Acquisition Order Preference	Unknown	1.20
Network Selection Registration Restriction Preference	1.34	1.34
CSG ID	1.41	1.41
Usage Preference	1.67	1.67
Voice Domain Preference	1.92	1.92
LTE Disable Cause	1.100	1.100

Name	Version introduced	Version last modified
Disabled RAT Bitmask	1.132	1.132
LTE Band Preference Extended	1.138	1.138

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Emergency Mode
Length	1			2	
Value	→	boolean	emergency_mode	1	Values: • 0x00 – OFF (normal) • 0x01 – ON (emergency)
Type	0x11			1	Mode Preference
Length	2			2	
Value	→	mask16	mode_pref	2	Bitmask representing the radio technology mode preference to be set. Values: • Bit 0 (0x01) – QMI_NAS_RAT_MODE_PREF_CDMA2000_1X – cdma2000® 1X • Bit 1 (0x02) – QMI_NAS_RAT_MODE_PREF_CDMA2000_HRPD – cdma2000® HRPD (1xEV-DO) • Bit 2 (0x04) – QMI_NAS_RAT_MODE_PREF_GSM – GSM • Bit 3 (0x08) – QMI_NAS_RAT_MODE_PREF_UMTS – UMTS • Bit 4 (0x10) – QMI_NAS_RAT_MODE_PREF_LTE – LTE • Bit 5 (0x20) – QMI_NAS_RAT_MODE_PREF_TDSCDMA – TD-SCDMA All unlisted bits are reserved for future use.
Type	0x12			1	Band Preference
Length	8			2	
Value	→	mask	band_pref	8	Bitmask representing the band preference to be set. See Table A-2 for details.
Type	0x13			1	CDMA PRL Preference
Length	2			2	
Value	→	enum16	prl_pref	2	PRL preference to be set for band class 0 (BC0) prl_pref. Values: • 0x0001 – PRL_PREF_A_SIDE_ONLY – Acquire available system only on the A side • 0x0002 – PRL_PREF_B_SIDE_ONLY – Acquire available system only on the B side • 0x3FFF – PRL_PREF_ANY – Acquire any available systems
Type	0x14			1	Roaming Preference
Length	2			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	enum16	roam_pref	2	Roaming preference to be set. Values: • 0x01 – ROAMING_PREF_OFF – Acquire only systems for which the roaming indicator is off • 0x02 – ROAMING_PREF_NOT_OFF – Acquire a system as long as its roaming indicator is not off • 0x03 – ROAMING_PREF_NOT_FLASING – Acquire only systems for which the roaming indicator is off or solid on, i.e., not flashing; CDMA only • 0xFF – ROAMING_PREF_ANY – Acquire systems, regardless of their roaming indicator
Type	0x15			1	LTE Band Preference (Deprecated; use LTE Band Preference Extended)
Length	8			2	
Value	→	mask	lte_band_pref	8	Bitmask representing the LTE band preference to be set. See Table A-3 for details.
Type	0x16			1	Network Selection Preference
Length	1			2	
Value	→	enum8	net_sel_pref	1	Network selection preference. Values: • 0x00 – Automatic network selection • 0x01 – Manual network selection
Type	0x18			1	Service Domain Preference
Length	4			2	
Value	→	enum	srv_domain_pref	4	Service domain preference. Values: • 0x00 – QMI_SRV_DOMAIN_PREF_CS_ONLY – Circuit-switched only • 0x01 – QMI_SRV_DOMAIN_PREF_PS_ONLY – Packet-switched only • 0x02 – QMI_SRV_DOMAIN_PREF_CS_PS – Circuit-switched and packet-switched
Type	0x19			1	GSM/WCDMA Acquisition Order Preference
Length	4			2	
Value	→	enum	gw_acq_order_pref	4	GSM/WCDMA acquisition order preference. Values: • 0x00 – NAS_GW_ACQ_ORDER_PREF_AUTOMATIC – Automatic • 0x01 – NAS_GW_ACQ_ORDER_PREF_GSM_WCDMA – GSM then WCDMA • 0x02 – NAS_GW_ACQ_ORDER_PREF_WCDMA_GSM – WCDMA then GSM
Type	0x1A			1	TDSCDMA Band Preference
Length	8			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	mask	tdscdma_band_pref	8	<p>Bitmask representing the TD-SCDMA band preference to be set. Values:</p> <ul style="list-style-type: none"> • 0x01 – NAS_TDSCDMA_BAND_A – TD-SCDMA Band A • 0x02 – NAS_TDSCDMA_BAND_B – TD-SCDMA Band B • 0x04 – NAS_TDSCDMA_BAND_C – TD-SCDMA Band C • 0x08 – NAS_TDSCDMA_BAND_D – TD-SCDMA Band D • 0x10 – NAS_TDSCDMA_BAND_E – TD-SCDMA Band E • 0x20 – NAS_TDSCDMA_BAND_F – TD-SCDMA Band F <p>All other bits are reserved.</p>
Type	0x1B			1	Manual Network Selection PLMN
Length	5			2	
Value	→	uint16	mcc	2	A 16-bit integer representation of MCC. Range: 0 to 999.
		uint16	mnc	2	A 16-bit integer representation of MNC. Range: 0 to 999.
		boolean	mnc_includes_pcs_digit	1	<p>This field is used to interpret the length of the corresponding MNC reported in the TLVs (in this table) with an mnc or mobile_network_code field.</p> <p>Values:</p> <ul style="list-style-type: none"> • TRUE – MNC is a three-digit value; e.g., a reported value of 90 corresponds to an MNC value of 090 • FALSE – MNC is a two-digit value; e.g., a reported value of 90 corresponds to an MNC value of 90
Type	0x1C			1	Acquisition Order Preference
Length	Var			2	
Value	→	uint8	acq_order_len	1	<p>Number of sets of the following elements:</p> <ul style="list-style-type: none"> • acq_order
		enum8	acq_order	Var	<p>Acquisition order preference to be set. Values:</p> <ul style="list-style-type: none"> • 0x01 – NAS_RADIO_IF_CDMA_1X – cdma2000® 1X • 0x02 – NAS_RADIO_IF_CDMA_1XEVDO – cdma2000® HRPD (1xEV-DO) • 0x04 – NAS_RADIO_IF_GSM – GSM • 0x05 – NAS_RADIO_IF_UTMS – UMTS • 0x08 – NAS_RADIO_IF_LTE – LTE • 0x09 – NAS_RADIO_IF_TDSCDMA – TD-SCDMA
Type	0x1D			1	Network Selection Registration Restriction Preference
Length	4			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	enum	srv_reg_restriction	4	<p>Registration restriction preference. Specifies one of the following modifiers to net_sel_pref:</p> <ul style="list-style-type: none"> • 0x00 – NAS_SRV_REG_RESTRICTION_UNRESTRICTED – Device follows the normal registration process • 0x01 – NAS_SRV_REG_RESTRICTION_CAMPED_ONLY – Device camps on the network according to its provisioning, but does not register • 0x02 – NAS_SRV_REG_RESTRICTION_LIMITED – Device selects the network for limited service <p>All other values are reserved.</p>
Type	0x1E			1	CSG ID
Length	10			2	
Value	→	uint16	mcc	2	A 16-bit integer representation of CSG MCC. Range: 0 to 999.
		uint16	mnc	2	A 16-bit integer representation of CSG MNC. Range: 0 to 999.
		boolean	mnc_includes_pcs_digit	1	<p>This field is used to interpret the length of the corresponding MNC reported in the TLVs (in this table) with an mnc or mobile_network_code field.</p> <p>Values:</p> <ul style="list-style-type: none"> • TRUE – MNC is a three-digit value; e.g., a reported value of 90 corresponds to an MNC value of 090 • FALSE – MNC is a two-digit value; e.g., a reported value of 90 corresponds to an MNC value of 90
		uint32	id	4	Closed subscriber group identifier.
		enum8	rat	1	<p>Radio interface technology of the CSG network.</p> <p>Values:</p> <ul style="list-style-type: none"> • 0x04 – RADIO_IF_GSM – GSM • 0x05 – RADIO_IF_UMTS – UMTS • 0x08 – RADIO_IF_LTE – LTE • 0x09 – RADIO_IF_TDSCDMA – TDS
Type	0x1F			1	Usage Preference
Length	4			2	
Value	→	enum	usage_setting	4	<p>Usage preference to be set. Values:</p> <ul style="list-style-type: none"> • NAS_USAGE_UNKNOWN (0) – Unknown • NAS_USAGE_VOICE_CENTRIC (1) – Voice centric • NAS_USAGE_DATA_CENTRIC (2) – Data centric
Type	0x20			1	Voice Domain Preference
Length	4			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	enum	voice_domain_pref	4	<p>Voice domain preference. Values:</p> <ul style="list-style-type: none"> • NAS_VOICE_DOMAIN_PREF_CS_ONLY (0x00) – Circuit-switched (CS) voice only • NAS_VOICE_DOMAIN_PREF_PS_ONLY (0x01) – Packet-switched (PS) voice only • NAS_VOICE_DOMAIN_PREF_CS_PREF (0x02) – CS is preferred; PS is secondary • NAS_VOICE_DOMAIN_PREF_PS_PREF (0x03) – PS is preferred; CS is secondary
Type	0x21			1	LTE Disable Cause
Length	4			2	
Value	→	enum	lte_disable_cause	4	<p>LTE disable cause. Values:</p> <ul style="list-style-type: none"> • NAS_LTE_DISABLE_CAUSE_NONE (0x00) – LTE is not disabled • NAS_LTE_DISABLE_CAUSE_PERMANENT_DS (0x01) – LTE is disabled by DS permanently, e.g., T3316 expiry • NAS_LTE_DISABLE_CAUSE_TEMP_DS (0x02) – LTE is disabled by DS temporarily • NAS_LTE_DISABLE_CAUSE_DOM_SEL (0x03) – LTE disable procedure is called for domain selection purpose • NAS_LTE_DISABLE_CAUSE_DAM (0x04) – LTE disable procedure is called for device aggression management recovery • NAS_LTE_DISABLE_CAUSE_USER (0x05) – LTE disable procedure is called due to user action, e.g., mode_pref change or PS_DETACH triggered by ATCOP/QMI • NAS_LTE_DISABLE_CAUSE_NO_CHANGE (0x06) – No change in LTE disable cause
Type	0x22			1	Disabled RAT Bitmask
Length	2			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	mask16	rat_disabled_mask	2	<p>Bitmask representing the radio technologies that are disabled. Values:</p> <ul style="list-style-type: none"> • Bit 0 (0x01) – QMI_NAS_RAT_MODE_PREF_CDMA2000_1X – cdma2000® 1X • Bit 1 (0x02) – QMI_NAS_RAT_MODE_PREF_CDMA2000_HRPD – cdma2000® HRPD (1xEV-DO) • Bit 2 (0x04) – QMI_NAS_RAT_MODE_PREF_GSM – GSM • Bit 3 (0x08) – QMI_NAS_RAT_MODE_PREF_UMTS – UMTS • Bit 4 (0x10) – QMI_NAS_RAT_MODE_PREF_LTE – LTE • Bit 5 (0x20) – QMI_NAS_RAT_MODE_PREF_TDSCDMA – TD-SCDMA <p>All unlisted bits are reserved for future use and the service point ignores them if used.</p>
Type	0x23			1	LTE Band Preference Extended
Length	32			2	
Value	→	uint64	bits_1_64	8	Bits 1 to 64 of the 256-bit LTE E-UTRA Operating Band bitmask
		uint64	bits_65_128	8	Bits 65 to 128 of the 256-bit LTE E-UTRA Operating Band bitmask
		uint64	bits_129_192	8	Bits 129 to 192 of the 256-bit LTE E-UTRA Operating Band bitmask
		uint64	bits_193_256	8	Bits 193 to 256 of the 256-bit LTE E-UTRA Operating Band bitmask

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_OP_DEVICE_UNSUPPORTED	Operation is not supported by the device
QMI_ERR_INVALID_ARG	Value field of one or more TLVs in the request message contains an invalid value
QMI_ERR_MISSING_ARG	One or more required TLVs were missing in the request

7.2.13.4. Description of QMI_NAS_GET_SYSTEM_SELECTION_PREFERENCE

This command queries the preferred system selection settings for the device.

For more information regarding the preference settings and the description of

QMI_NAS_SET_SYSTEM_SELECTION_PREFERENCE, see Section 7.2.12.1.

The Manual Network Selection PLMN TLV is included only when the Network Selection Preference TLV is set to “Manual network selection”.

If the Emergency Mode TLV is set to “ON”, the Mode Preference TLV will be populated with the last received non-emergency value.

The CSG ID TLV is included only when the PLMN listed is a CSG network.

Description of QMI_NAS_SYSTEM_SELECTION_PREFERENCE_IND

This indication communicates the current preferred system selection settings for the device.

The Manual Network Selection PLMN TLV is included only when the current network selection preference is set to manual.

If the Emergency Mode TLV is set to “ON”, the Mode Preference TLV will be populated with the last received non-emergency value.

The CSG ID TLV is included only when the PLMN listed is a CSG network.

8 Wireless Messaging Service (QMI_WMS)

QMI_WMS provides commands related to wireless messaging to applications running on a host PC, including:

Sending raw data

Writing, reading, and deleting data to/from device memory

Modifying tags

Setting and reading routes

Reading and setting Short Message Service Center (SMSC) addresses

It is expected that user-level applications, e.g., connection managers and/or device drivers residing on the Terminal Equipment (TE), will use QMI_WMS to access such functionality on the MSM™ device.

QMI_WMS is a QMI native service, conforming to the generalized behavior for QMI services, as defined in 80-VB816-1.

8.1. Theory of Operation

8.1.1. Generalized QMI Service Compliance

The QMI_WMS 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 80-VB816-1. Extensions to the generalized QMI service theory of operation are noted in subsequent sections of this chapter.

8.1.2. WMS Service Type

WMS is assigned QMI service type 0x05.

8.1.3. Message Definition Template

8.1.3.1. Response Message Result TLV

This Type-Length-Value (TLV) is present in all Response messages defined in this document. It is not

present in the Indication messages.

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

8.1.4. QMI_WMS Fundamental Concepts

8.1.4.1. Wireless Message Network Architecture

A network supports wireless messaging with three main components:

A wireless MSM device supporting WMS is designated as an Endpoint (EP) (refer to 3GPP2 C.S0015-A) within a larger network. WMS Eps are capable of both originating and terminating WMS messages.

A wireless network may include one or more SMSCs (refer to 3GPP2 C.S0015-A). These are responsible for routing WMS messages between the origination and destination Eps.

Relay points are included in the wireless network and are responsible for safely transferring messages between Eps and SMSCs within the network.

These components are the main building blocks that make up a short messaging network and can be found in both CDMA and WCDMA networks, although the names may be slightly different.

When the MSM device sends a WMS message, it is submitted to the wireless network using a Base Station (BS). The BS relays the WMS message to the SMSC, which acknowledges the message, then the BS, in turn, relays the acknowledgment back to the MSM device. The SMSC is then responsible for routing and delivery of the WMS to the destination EP.

The WMS architecture for a CDMA network can be found in 3GPP2 C.S0015-A Figure 1.5.1. The WMS architecture for a WCDMA network can be found in 3GPP TS 23.040 Figure 4 and Figure 5.

8.1.4.2. Wireless Message Types

QMI_WMS supports the message types defined in the standardized protocols for CDMA in 3GPP2 C.S0015-A and WCDMA in 3GPP TS 23.040. Both CDMA and WCDMA support Point-to-Point (PP) and Broadcast (BC) (refer to 3GPP2 C.S0015-A) message functionality. Messages are further classified into Mobile-Originated (MO) and Mobile-Terminated (MT) messages (refer to 3GPP2 C.S0015-A), relative to the control point.

The WMS protocol dictates that a PP WMS message solicits a response or Acknowledgment (ACK) (refer to 3GPP2 C.S0015-A) to the network upon receipt by the addressee. The ACK is relayed to the network SMSC verifying delivery, but not to the originator unless requested in the original message.

QMI_WMS supports point-to-point messaging and associated WMS types, and broadcast messaging. It also supports sending ACKs to the network.

8.1.4.3. WMS Client/Service Architecture

The WMS service provides its clients the means to send messages over the wireless network, read and write messages to persistent storage on the device, and to configure various WMS service configuration options.

The WMS service running on the MSM device supports multiple clients. In addition, other WMS service clients may operate within the MSM device.

Note that, even if no QMI_WMS or other WMS clients are active, the WMS service is still running on the MSM device. This allows the MSM device to accept, store (if configured to allow), and acknowledge delivery of incoming WMS messages.

8.1.4.4. Incoming Message Indication

Each QMI_WMS control point may independently enable indications of new MT messages. When the WMS service accepts a new MT message from the wireless network, a QMI_WMS indication message is sent to each QMI_WMS control point that has enabled notification.

Resetting the QMI_WMS control point returns an MT message indication back to the default disabled state.

After each reset, the control point must again register for these indications using the QMI_WMS_SET_EVENT_REPORT message.

8.1.4.5. WMS Message Layers

The WMS message layers are:

WMS teleservice layer – This layer is also known as the Transfer Protocol data unit (TPDU) layer in GSM/WCDMA. In this layer, the message is sent, received, and presented to users. The message structure in this layer includes a message body encoded with a specified encoding, a message identifier that enables the MSM device to transfer messages to/from the wireless network, the date of reception, etc. Refer to 3GPP2 C.S0015-A Section 4 and 3GPP TS 23.040 Section 9.2.3 for details of the parameters defined for this layer.

WMS transport layer – In addition to carrying the WMS teleservice layer message, the message in this layer is considered as a sequence of octets containing information, such as a teleservice ID, message originator or recipient address, bearer reply option in CDMA, or service center address in GSM/WCDMA. Refer to 3GPP2 C.S0015-A Section 3.4 and 3GPP TS 23.040 Section 9.2.3.24 for details of the parameters defined for this layer.

8.1.4.6. Raw Message Parameters

The raw QMI_WMS messages defined later in this document take or return transport layer encoded messages as parameters.

8.1.4.7. Routes

A message category is defined as a unique tuple of:

WMS message type (PP or BC)

WMS message class

For each message type, PP or BC, there are one or more message classes, depending on the message protocol in use. CDMA defines one message class, while WCDMA defines five unique classes.

A message action is defined as a unique tuple of:

WMS action, when receiving a message of this type and class

WMS storage type (for store actions)

When a new message arrives, its type and class determine how the message is processed. When the message is delivered from the network, there are four possibilities: discard, store and notify, transfer only, or transfer and ACK. Discard accepts the message and then deletes it without storing the message. Store and notify writes the message to the designated memory storage on the MSM device and then sends notification to all QMI_WMS control points that have enabled incoming message notification. Transfer only transfers the message to the client and lets the client send the ACK to the network. Transfer and ACK transfers the message to the client and sends the ACK to the network.

There are other routing actions that are not supported by QMI_WMS are returned as unknown by the QMI_WMS_GET_ROUTES response message. If one of these actions is set by an external MSM WMS client, unexpected behavior results.

A message route refers to the action associated with a message category. Consequently, a message route is described by its message category and the action performed when a message matching that category is received by the device.

8.1.4.8. Device Memory Storage

The types of memory that are available on the MSM device to store messages are:

User Identity Module (UIM) – Removable media used by the phone

Nonvolatile (NV) – Persistent memory located within the phone

Each WMS protocol supporting these storage types is allocated its own storage. These storage types are unique to each protocol and cannot be accessed by the other protocols.

8.1.5. Service State Variables

8.1.5.1. Shared State Variables

The following is a shared state variable for all control points using the QMI_WMS service:

Name	Description	Possible values
message_mode	System mode used for a WMS message	<ul style="list-style-type: none"> • CDMA • WCDMA

Note: If the device is capable of supporting more than one message protocol, this shared state variable will not be maintained.

8.1.5.2. State Variables Per Control Point

The following are nonshared state variables for each QMI_WMS control point:

Name	Description	Possible values	Default value
report_mt_message	Whether new MT messages are reported to a control point	<ul style="list-style-type: none"> • FALSE • TRUE 	FALSE
report_call_control_info	Whether MO SMS call control information is reported to a control point	<ul style="list-style-type: none"> • FALSE • TRUE 	FALSE
report_mwi_message	Whether new MWI messages are reported to a control point	<ul style="list-style-type: none"> • FALSE • TRUE 	FALSE

8.2. QMI_WMS Messages

Table 7: QMI_WMS messages

Command	ID	Description
QMI_WDS_START_NET	0x0020	Activates a packet data session (if not already started) on behalf of the requesting control point.
QMI_WMS_RESET	0x0000	Resets the WMS service state variables of the requesting control point.
QMI_WMS_SET_EVENT	0x0001	Sets the WMS event reporting conditions for the control point.
QMI_WMS_SET_EVENT_REPORT		
QMI_WMS_EVENT	0x0001	Indicates a QMI_WMS event.
QMI_WMS_EVENT_REPORT_IND		
QMI_WMS_RAW_SEND	0x0020	Sends a new message in its raw format.
QMI_WMS_RAW_READ	0x0022	Reads a message from the device memory storage and returns the message in its raw format.
QMI_WMS MODIFY_TAG	0x0023	Modifies the metadata tag of a message in the MSM device storage.
QMI_WMS_DELETE	0x0024	Deletes the message in a specified memory location.
QMI_WMS_GET_MSG_PROTOCOL	0x0030	Queries the message protocol currently in use for the WMS client.
QMI_WMS_GET_MESSAGE_PROTOCOL		
QMI_WMS_GET_MSG_LIST	0x0031	Requests a list of WMS message indices and meta information within the specified memory storage, matching a specified message tag.
QMI_WMS_LIST_MESSAGES		
QMI_WMS_SET_ROUTES	0x0032	Sets the action performed upon WMS message receipt for the specified message routes. It also sets the action performed upon WMS receipt of status reports.
QMI_WMS_GET_ROUTES	0x0033	Queries the currently configured action performed upon WMS message receipt for the specified message routes. It also queries the action performed upon WMS receipt of status reports.
QMI_WMS_GET_SMSC_ADD	0x0034	Queries the currently configured SMSC

QMI_WMS_GET_SMSC_ADDRESS**		address.
QMI_WMS_SET_SMSC_ADDR	0x0035	Sets the SMSC address used when storing or saving SMS messages.
QMI_WMS_SET_SMSC_ADDRESS**		
QMI_WMS_GET_MSG_LIST_MAX	0x0036	Queries the maximum number of messages that can be stored per memory storage, as well as the number of slots currently available.
QMI_WMS_GET_STORE_MAX_SIZE		
QMI_WMS_GET_DOMAIN_PREF	0x0040	Queries the GW domain preference.(Deprecated)
QMI_WMS_GET_DOMAIN_PREF**		
QMI_WMS_GET_DOMAIN_PREF	0x0041	Sets the GW domain preference.(Deprecated)
QMI_WMS_SET_DOMAIN_PREF**		

8.2.1. QMI_WMS_RESET

Resets the WMS service state variables of the requesting control point.

- **WMS message ID**

0x0000

- **Version introduced**

Major - 1, Minor - 1

8.2.1.1. Request - QMI_WMS_RESET_REQ

- **Message type**

Request

- **Sender**

Control point

- **Mandatory TLVs**

None

- **Optional TLVs**

None

8.2.1.2. Response - QMI_WMS_RESET_RESP

- **Message type**

Response

- **Sender**

Service

- **Mandatory TLVs**

The Result Code TLV (defined in Section 8.1.3.1) 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

8.2.1.3. Description of QMI_WMS_RESET REQ/RESP

This command resets the issuing control point's state kept by the service.

As a result, each shared state variable may change according to its arbitration policy (see Section 8.1.5.2).

Although it is performed as one operation, this is equivalent to closing the service and reopening it; therefore, the client ID of the requesting control point does not change.

The control point's state variables change to their default values before the response is issued.

8.2.2. QMI_WMS_SET_EVENT (QMI_WMS_SET_EVENT_REPORT)

Sets the WMS event reporting conditions for the control point.

- **WMS message ID**

0x0001

- Version introduced

Major - 1, Minor - 1

8.2.2.1. Request - QMI_WMS_SET_EVENT_REPORT_REQ

- Message type

Request

- Sender

Control point

- Mandatory TLVs

None

- Optional TLVs

At least one of the following optional TLVs must be included in this request.

Name	Version introduced	Version last modified
New MT Message Indicator	Unknown	1.1
MO SMS Call Control Information	1.16	1.16
MWI Message Indicator	1.17	1.17

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	New MT Message Indicator
Length	1			2	
Value	→	boolean	report_mt_message	1	Report new MT messages. Values: • 0x00 – Disable • 0x01 – Enable
Type	0x11			1	MO SMS Call Control Information
Length	1			2	
Value	→	boolean	report_call_control_info	1	Report MO SMS call control information. Values: • 0x00 – Disable • 0x01 – Enable

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x12			1	MWI Message Indicator
Length	1			2	
Value	→	boolean	report_mwi_message	1	Report new MWI messages. Values: • 0x00 – Disable • 0x01 – Enable

8.2.2.2. Response - QMI_WMS_SET_EVENT_REPORT_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_MISSING_ARG	A required TLV was not provided

8.2.2.3. Description of QMI_WMS_SET_EVENT_REPORT REQ/RESP

The control point's event reporting state variables are modified according to the settings specified in the TLVs included in the request message. The service maintains a set of state variables for each control point.

See Section 8.1.5.2 for more details regarding control point state variables.

Specified events are communicated to the registered WMS control point via QMI_WMS_EVENT_REPORT_IND.

The MWI Indicator TLV must be set to 1 if the control point needs the MWI PDU (sent via QMI_WMS_EVENT_REPORT_IND) for parsing the information. The default setting is to send the decoded information via QMI_WMS_MESSAGE_WAITING_IND.

8.2.3. Indication - QMI_WMS_SET_EVENT (QMI_WMS_EVENT_REPORT_IND)

- **Message type**

Indication

- **Sender**

Service

- **Indication scope**

Unicast (per control point)

- **Mandatory TLVs**

None

- **Optional TLVs**

At least one of the following optional TLVs shall be included in this indication.

8.2.3.1. Description of QMI_WMS_EVENT_REPORT_IND

This unsolicited indication is sent to specified control points when the device state that corresponds to any TLV listed above changes. Specified control points are those that previously registered for the corresponding state to be reported using the QMI_WMS_SET_EVENT_REPORT_REQ message.

This indication with the MT message received TLV or transfer route MT message TLV is generated when a new MT message is received by the device. The MT message TLV is sent in the indication when the route for the MT message is store and notify. The transfer route MT message TLV is sent in the indication when the route for the MT message is transfer only or transfer and ACK.

The Call Control Result TLV is sent when MO SMS initiated by other WMS clients has the call control result as disallowed, allowed, or allowed with modifications.

8.2.4. QMI_WMS_RAW_SEND

Sends a new message in its raw format.

- **WMS message ID**

0x0020

- **Version introduced**

Major - 1, Minor - 1

8.2.4.1. Request - QMI_WMS_RAW_SEND_REQ

- **Message type**

Request

- **Sender**

Control point

- **Mandatory TLVs**

Name	Version introduced	Version last modified
Raw Message Data	Unknown	1.1

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Raw Message Data
Length	Var			2	
Value	→	enum8	format	1	Message format. Values: • 0x00 – MESSAGE_FORMAT_CDMA – CDMA • 0x02 to 0x05 – Reserved • 0x06 – MESSAGE_FORMAT_GW_PP – GW_PP
		uint16	len	2	Number of sets of the following elements: • raw_message
		uint8	raw_message	Var	Raw message data.

Optional TLVs

Name	Version introduced	Version last modified
Force on DC*	Unknown	1.1
Follow on DC*	Unknown	1.1
Link Control**	Unknown	1.2
SMS on IMS	1.4	1.9
Retry Message	Unknown	1.5
Retry Message ID	Unknown	1.5
Link Control Enabling Information**	1.15	1.15

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Force on DC*
Length	2			2	
Value	→	boolean	force_on_dc	1	Force the message to be sent on the CDMA dedicated channel. Values: <ul style="list-style-type: none">• 0x00 – Do not care about the channel on which the message is sent• 0x01 – Request to send the message over the dedicated channel
			enum8 so	1	Service option. Values: <ul style="list-style-type: none">• 0x00 – SO_AUTO – AUTO (choose the best service option while setting up the DC)• 0x06 – SO_6 – Service option 6• 0x0E – SO_14 – Service option 14
Type	0x11			1	Follow on DC*
Length	1			2	
Value	→	enum8	follow_on_dc	1	Flag to request to not disconnect the CDMA dedicated channel after the send operation is completed; this TLV can be included if more messages are expected to follow. Values: <ul style="list-style-type: none">• 0x01 – FOLLOW_ON_DC_ON – On (do not disconnect the DC after the send operation) Any value other than 0x01 in this field is treated as an absence of this TLV.
Type	0x12			1	Link Control**
Length	1			2	
Value	→	uint8	link_timer	1	Keeps the GW SMS link open for the specified number of seconds; can be enabled if more messages are expected to follow
Type	0x13			1	SMS on IMS
Length	1			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	boolean	sms_on_ims	1	<p>Indicates whether the message is to be sent on IMS. Values:</p> <ul style="list-style-type: none"> • 0x00 – Message is not to be sent on IMS • 0x01 – Message is to be sent on IMS • 0x02 to 0xFF – Reserved <p>Note: In minor version 9, the implementation was changed in such a way that inclusion of this TLV may affect the SMS routing differently.</p>
Type	0x14			1	Retry Message
Length	1			2	
Value	→	enum8	retry_message	1	<p>Indicates this message is a retry message. Values:</p> <ul style="list-style-type: none"> • 0x01 – WMS_MESSAGE_IS_A_RETRY – Message is a retry message <p>Note: Any value other than 0x01 in this field is treated as an absence of this TLV.</p>
Type	0x15			1	Retry Message ID
Length	4			2	
Value	→	uint32	retry_message_id	4	<p>Message ID to be used in the retry message. The message ID specified here is used instead of the message ID encoded in the raw message.</p> <p>Note: This TLV is valid only if the Retry Message TLV is specified and set to 0x01.</p>
Type	0x16			1	Link Control Enabling Information**
Length	1			2	
Value	→	boolean	link_enable_mode	1	<p>Indicates whether to keep the link control enabled, until the option is modified by the client. Values:</p> <ul style="list-style-type: none"> • 0x00 – Enable link control once so that the lower layer keeps the link up for a specified time until the next MO SMS is requested or the timer expires • 0x01 – Always enable link control <p>Note: This TLV is valid only if the Link Control TLV is specified and is set to a valid timer value.</p>

8.2.4.2. Response - QMI_WMS_RAW_SEND_RESP

- **Message type**

Response

- **Sender**

Service

- **Mandatory TLVs**

The Result Code TLV (defined in Section 8.1.3.1) is always present in the response. The following mandatory TLVs are always present in the response.

Name	Version introduced	Version last modified
Result Code	1.1	1.1
Message ID	1.1	1.19

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Message ID
Length	2			2	
Value	→	uint16	message_id	2	WMS message ID.

- **Optional TLVs**

If the Result Code TLV indicates failure and the qmi_error fifield is set to QMI_ERR_CAUSE_CODE, the following parameters are returned.

Name	Version introduced	Version last modified
Cause Code*	1.1	1.1
Error Class*	Unknown	1.2
GW Cause Info**	Unknown	1.2
Message Delivery Failure Type	Unknown	1.4
Message Delivery Failure Cause	Unknown	1.5
Call Control Modified Information**	Unknown	1.5

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Cause Code*
Length	2			2	
Value	→	enum16	cause_code	2	WMS cause code per 3GPP2 N.S0005-0 Section 6.5.2.125; see Table A-1 for more information

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x11			1	Error Class*
Length	1			2	
Value	→	enum8	error_class	1	Error class. Values: • 0x00 – ERROR_CLASS_TEMPORARY • 0x01 – ERROR_CLASS_PERMANENT
Type	0x12			1	GW Cause Info**
Length	3			2	
Value	→	enum16	rp_cause	2	GW RP cause per 3GPP TS 24.011 Section 8.2.5.4; see Table A-2 for more information.
		enum8	tp_cause	1	GW TP cause per 3GPP TS 23.040 Section 9.2.3.22; see Table A-3 for more information.
Type	0x13			1	Message Delivery Failure Type
Length	1			2	
Value	→	enum8	message_delivery_failure_type	1	Message delivery failure type. Values: • 0x00 – WMS_MESSAGE_DELIVERY_FAILURE_TEMPORARY • 0x01 – WMS_MESSAGE_DELIVERY_FAILURE_PERMANENT
Type	0x14			1	Message Delivery Failure Cause
Length	1			2	
Value	→	enum8	message_delivery_failure_cause	1	Message delivery failure cause. Values: • 0x00 – WMS_MESSAGE_BLOCKED_DUE_TO_CALL_CONTROL
Type	0x15			1	Call Control Modified Information**
Length	Var			2	
Value	→	uint8	alpha_id_len	1	Number of sets of the following elements: • alpha_id
		uint8	alpha_id	Var	Alpha ID.

- Error codes

QMI_ERR_NONE	No error in the request
QMI_ERR_INTERNAL	Unexpected error occurred during processing
QMI_ERR_MALFORMED_MSG	Message was not formulated correctly by the control point or the message was corrupted during transmission
QMI_ERR_NO_MEMORY	Device could not allocate memory to formulate a response
QMI_ERR_ARG_TOO_LONG	Argument passed in a TLV was larger than the available storage in the device
QMI_ERR_MISSING_ARG	A required TLV was not provided
QMI_ERR_INVALID_ARG	One of the parameters specified contains an invalid value

QMI_ERR_CAUSE_CODE	SMS cause code: For CDMA, refer to 3GPP2 N.S0005-0 Section 6.5.2.125; for GW, refer to 3GPP TS 27.005 Section 3.2.5
QMI_ERR_ENCODING	Message is not encoded properly
QMI_ERR_INVALID_MESSAGE_ID	Message ID specified for the message is invalid
QMI_ERR_MESSAGE_NOT_SENT	Message could not be sent
QMI_ERR_MESSAGE_DELIVERY_FAILURE	Message could not be delivered
QMI_ERR_DEVICE_NOT_READY	Device is not ready to send the message
QMI_ERR_NETWORK_NOT_READY	Network is not ready to send the message
QMI_ERR_OP_DEVICE_UNSUPPORTED	Selected operation is not supported by the device
QMI_ERR_OP_NETWORK_UNSUPPORTED	Selected operation is not supported by the network
QMI_ERR_SMSC_ADDR	SMSC address specified is invalid
QMI_ERR_CALL_FAILED	Cannot bring up the CDMA dedicated channel
QMI_ERR_MSG_BLOCKED	Message is blocked because the recipient is not on the FDN
QMI_ERR_INVALID_OPERATION	SMS on IMS TLV is set to TRUE; however, IMS is not registered

8.2.4.3. Description of QMI_WMS_RAW_SEND REQ/RESP

This command requests that a WMS message be sent by the MSM device.

Raw send can be used only with transport layer-encoded messages:

For 3GPP2 devices, transport layer messages are in Layer 3 format (refer to 3GPP2 C.S0015-A). The control point must ensure that the raw message has these fields encoded (3GPP2 C.S0015-A Section 3.4.2) for a detailed description of these fields):

- Teleservice ID
- Destination Address
- Bearer Reply Option – Used to configure the setting to get the transport layer acknowledgment (only if the control point is interested in receiving the transport layer acknowledgment)

For 3GPP devices, transport layer messages are in PDU format (refer to 3GPP TS 27.005). The raw message in PDU format must include the SMSC address length identifier as the first byte of the message. If this byte is set to zero, the SMSC provisioned for the device is used (as specified using QMI_WMS_SET_SMSC_ADDRESS). Otherwise, the first byte indicates the length, in bytes, of the SMSC address that is included after the first byte, but before the start of the actual PDU message. The equivalent AT command for this request is **AT+CMGS** (refer to 3GPP TS 27.005).

If a raw message is not in transport layer format or includes transport layer parameters that cannot be processed for any reason, the command fails and returns a QMI_ERR_ENCODING error. A successful result value in the response implies that the given message send request is complete. The message is not stored in memory; it is only sent by the MSM device. To store the message in memory, the QMI_WMS_RAW_WRITE command must be used.

The behaviors of the Force on DC and Follow on DC TLVs are as follows:

For 3GPP2 devices, the Force on DC TLV can be included in the request, with value TRUE, to send the message over the CDMA dedicated channel. If the service fails to bring up the dedicated channel, a QMI_ERR_CALL_FAILED error is returned in the response.

If more messages are expected, the Follow on DC TLV can be included in the request.

If the Follow on DC TLV is absent and the Force on DC TLV is present (with value TRUE or FALSE), the service attempts to tear down the CDMA dedicated channel after the send operation. However, this disconnection is not guaranteed immediately, e.g., if there are pending messages. The service does not wait for the disconnection to send the QMI_WMS_RAW_SEND_RESP.

The Follow on DC TLV is ignored if it is sent in the absence of the Force on DC TLV in the request.

For GW, if more messages are expected, the Link Control TLV can be included. The link is kept open for the specified number of seconds. The link can be kept open for a maximum of 5 sec; setting the link timer to a value greater than 5 elicits a QMI_ERR_INVALID_ARG error. The suggested value for the link timer is 5 sec. If multiple messages are expected, the link control can be kept enabled by setting the optional Link Control Enabling Information TLV to 1. If this optional TLV is not present, the default behavior is to keep the link open for the number of seconds specified in the Link Control TLV. The Link Control TLV is required to enable link control; setting the Link Control Enabling Information TLV without the Link Control TLV elicits a QMI_ERR_MISSING_ARG error.

If the Result Code TLV indicates failure and the qmi_error field is set to QMI_ERR_CAUSE_CODE, 3GPP2 devices return the Cause Code and the Error Class TLVs. 3GPP devices return the GW Cause Information TLV.

If the Result Code TLV indicates failure and the qmi_error field is set to QMI_ERR_MESSAGE_DELIVERY_FAILURE, the mobile may return the Message Delivery Failure Type TLV.

If the message was successfully sent but modified due to call control, the mobile may return the Call Control Modified Information TLV.

The Retry Message TLV may be included to indicate this is a retry message. Sending a message as a

retry changes the behavior of the message; a message should be specified as a retry only after the message has been sent once and failed. There are two options for setting the message ID for a retry message:

Retry Message ID TLV not included – The message ID encoded in the raw message is left unchanged.

Retry Message ID TLV included – The message ID encoded in the raw message is updated with this specified value.

Messages should be sent one at a time. The client should wait for the response from the previous message before sending the next message.

If the SMS on IMS TLV is not included, WMS uses IMS whenever possible, i.e., IMS is the preferred transport. If the TLV is included with value 0x00 (FALSE), WMS does not use IMS as the transport. If the TLV is included with value 0x01 (TRUE) and IMS cannot be used, a QMI_ERR_INVALID_OPERATION error is returned.

8.2.5. QMI_WMS_RAW_READ

Reads a message from the device memory storage and returns the message in its raw format.

- **WMS message ID**

0x0022

- **Version introduced**

Major - 1, Minor - 1

8.2.5.1. Request - QMI_WMS_RAW_READ_REQ

- **Message type**

Request

- **Sender**

Control point

- Mandatory TLVs

Name	Version introduced	Version last modified
Raw Message Data	Unknown	1.1

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Raw Message Data
Length	Var			2	
Value	→	enum8	format	1	Message format. Values: • 0x00 – MESSAGE_FORMAT_CDMA – CDMA • 0x02 to 0x05 – Reserved • 0x06 – MESSAGE_FORMAT_GW_PP – GW_PP
		uint16	len	2	Number of sets of the following elements: • raw_message
		uint8	raw_message	Var	Raw message data.

Optional TLVs

Name	Version introduced	Version last modified
Force on DC*	Unknown	1.1
Follow on DC*	Unknown	1.1
Link Control**	Unknown	1.2
SMS on IMS	1.4	1.9
Retry Message	Unknown	1.5
Retry Message ID	Unknown	1.5
Link Control Enabling Information**	1.15	1.15

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Force on DC*
Length	2			2	
Value	→	boolean	force_on_dc	1	Force the message to be sent on the CDMA dedicated channel. Values: • 0x00 – Do not care about the channel on which the message is sent • 0x01 – Request to send the message over the dedicated channel
		enum8	so	1	Service option. Values: • 0x00 – SO_AUTO – AUTO (choose the best service option while setting up the DC) • 0x06 – SO_6 – Service option 6 • 0x0E – SO_14 – Service option 14
Type	0x11			1	Follow on DC*
Length	1			2	
Value	→	enum8	follow_on_dc	1	Flag to request to not disconnect the CDMA dedicated channel after the send operation is completed; this TLV can be included if more messages are expected to follow. Values: • 0x01 – FOLLOW_ON_DC_ON – On (do not disconnect the DC after the send operation) Any value other than 0x01 in this field is treated as an absence of this TLV.
Type	0x12			1	Link Control**
Length	1			2	
Value	→	uint8	link_timer	1	Keeps the GW SMS link open for the specified number of seconds; can be enabled if more messages are expected to follow
Type	0x13			1	SMS on IMS
Length	1			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	boolean	sms_on_ims	1	<p>Indicates whether the message is to be sent on IMS. Values:</p> <ul style="list-style-type: none"> • 0x00 – Message is not to be sent on IMS • 0x01 – Message is to be sent on IMS • 0x02 to 0xFF – Reserved <p>Note: In minor version 9, the implementation was changed in such a way that inclusion of this TLV may affect the SMS routing differently.</p>
Type	0x14			1	Retry Message
Length	1			2	
Value	→	enum8	retry_message	1	<p>Indicates this message is a retry message. Values:</p> <ul style="list-style-type: none"> • 0x01 – WMS_MESSAGE_IS_A_RETRY – Message is a retry message <p>Note: Any value other than 0x01 in this field is treated as an absence of this TLV.</p>
Type	0x15			1	Retry Message ID
Length	4			2	
Value	→	uint32	retry_message_id	4	<p>Message ID to be used in the retry message. The message ID specified here is used instead of the message ID encoded in the raw message.</p> <p>Note: This TLV is valid only if the Retry Message TLV is specified and set to 0x01.</p>
Type	0x16			1	Link Control Enabling Information**
Length	1			2	
Value	→	boolean	link_enable_mode	1	<p>Indicates whether to keep the link control enabled, until the option is modified by the client. Values:</p> <ul style="list-style-type: none"> • 0x00 – Enable link control once so that the lower layer keeps the link up for a specified time until the next MO SMS is requested or the timer expires • 0x01 – Always enable link control <p>Note: This TLV is valid only if the Link Control TLV is specified and is set to a valid timer value.</p>

8.2.5.2. Response - QMI_WMS_RAW_SEND_RESP

- Message type

Response

- Sender

Service

- Mandatory TLVs

The Result Code TLV (defined in Section 8.1.3.1) is always present in the response. The following mandatory TLVs are always present in the response.

Name	Version introduced	Version last modified
Result Code	1.1	1.1
Message ID	1.1	1.19

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Message ID
Length	2			2	
Value	→	uint16	message_id	2	WMS message ID.

- Optional TLVs

If the Result Code TLV indicates failure and the qmi_error field is set to QMI_ERR_CAUSE_CODE, the following parameters are returned.

Name	Version introduced	Version last modified
Cause Code*	1.1	1.1
Error Class*	Unknown	1.2
GW Cause Info**	Unknown	1.2
Message Delivery Failure Type	Unknown	1.4
Message Delivery Failure Cause	Unknown	1.5
Call Control Modified Information**	Unknown	1.5

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Cause Code*
Length	2			2	
Value	→	enum16	cause_code	2	WMS cause code per 3GPP2 N.S0005-0 Section 6.5.2.125; see Table A-1 for more information

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x11			1	Error Class*
Length	1			2	
Value	→	enum8	error_class	1	Error class. Values: • 0x00 – ERROR_CLASS_TEMPORARY • 0x01 – ERROR_CLASS_PERMANENT
Type	0x12			1	GW Cause Info**
Length	3			2	
Value	→	enum16	rp_cause	2	GW RP cause per 3GPP TS 24.011 Section 8.2.5.4; see Table A-2 for more information.
		enum8	tp_cause	1	GW TP cause per 3GPP TS 23.040 Section 9.2.3.22; see Table A-3 for more information.
Type	0x13			1	Message Delivery Failure Type
Length	1			2	
Value	→	enum8	message_delivery_failure_type	1	Message delivery failure type. Values: • 0x00 – WMS_MESSAGE_DELIVERY_FAILURE_TEMPORARY • 0x01 – WMS_MESSAGE_DELIVERY_FAILURE_PERMANENT
Type	0x14			1	Message Delivery Failure Cause
Length	1			2	
Value	→	enum8	message_delivery_failure_cause	1	Message delivery failure cause. Values: • 0x00 – WMS_MESSAGE_BLOCKED_DUE_TO_CALL_CONTROL
Type	0x15			1	Call Control Modified Information**
Length	Var			2	
Value	→	uint8	alpha_id_len	1	Number of sets of the following elements: • alpha_id
		uint8	alpha_id	Var	Alpha ID.

Error codes

QMI_ERR_NONE	No error in the request
QMI_ERR_INTERNAL	Unexpected error occurred during processing
QMI_ERR_MALFORMED_MSG	Message was not formulated correctly by the control point or the message was corrupted during transmission
QMI_ERR_NO_MEMORY	Device could not allocate memory to formulate a response
QMI_ERR_ARG_TOO_LONG	Argument passed in a TLV was larger than the available storage in the device
QMI_ERR_MISSING_ARG	A required TLV was not provided
QMI_ERR_INVALID_ARG	One of the parameters specified contains an invalid value

QMI_ERR_CAUSE_CODE	SMS cause code: For CDMA, refer to 3GPP2 N.S0005-0 Section 6.5.2.125; for GW, refer to 3GPP TS 27.005 Section 3.2.5
QMI_ERR_ENCODING	Message is not encoded properly
QMI_ERR_INVALID_MESSAGE_ID	Message ID specified for the message is invalid
QMI_ERR_MESSAGE_NOT_SENT	Message could not be sent
QMI_ERR_MESSAGE_DELIVERY_FAILURE	Message could not be delivered
QMI_ERR_DEVICE_NOT_READY	Device is not ready to send the message
QMI_ERR_NETWORK_NOT_READY	Network is not ready to send the message
QMI_ERR_OP_DEVICE_UNSUPPORTED	Selected operation is not supported by the device
QMI_ERR_OP_NETWORK_UNSUPPORTED	Selected operation is not supported by the network
QMI_ERR_SMSC_ADDR	SMSC address specified is invalid
QMI_ERR_CALL_FAILED	Cannot bring up the CDMA dedicated channel
QMI_ERR_MSG_BLOCKED	Message is blocked because the recipient is not on the FDN
QMI_ERR_INVALID_OPERATION	SMS on IMS TLV is set to TRUE; however, IMS is not registered

8.2.5.3. Description of QMI_WMS_RAW_READ REQ/RESP

This command reads a WMS message from memory storage on the MSM device.

The message is returned in the response in its raw, teleservice layer encoding without being decoded.

For 3GPP2 devices, transport layer messages are in Layer 3 format (refer to 3GPP2 C.S0015-A).

For 3GPP devices, transport layer messages are in PDU format (refer to 3GPP TS 27.005). The raw message returned in PDU format includes the SMSC address length identifier as the first byte of the message. This byte indicates the length, in bytes, of the SMSC address that is included after the first byte, but before the start of the actual PDU message. The equivalent AT command for this request is **AT+CMGR** (refer to 3GPP TS 27.005).

The response also includes metadata for the message, including the tag and format

For 3GPP devices, requests to read messages of an invalid TPDU type (refer to 3GPP TS 27.005) elicit a QMI_ERR_TPDU_TYPE error.

The Message Mode TLV must be included if the device is capable of supporting more than one protocol. If the TLV is not included, a QMI_ERR_MISSING_ARG error is returned.

8.2.6. QMI_WMS MODIFY_TAG

Modifies the metadata tag of a message in the MSM device storage.

- WMS message ID

0x0023

- Version introduced

Major - 1, Minor - 1

8.2.6.1. Request - QMI_WMS MODIFY_TAG_REQ

- Message type

Request

- Sender

Control point

- Mandatory TLVs

Name		Version introduced	Version last modified
WMS Message Tag		Unknown	1.1

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	WMS Message Tag
Length	6			2	
Value	→	enum8	storage_type	1	Memory storage. Values: • 0x00 – STORAGE_TYPE_UIM • 0x01 – STORAGE_TYPE_NV
		uint32	storage_index	4	Memory index.
		enum8	tag_type	1	Message tag. Values: • 0x00 – TAG_TYPE_MT_READ • 0x01 – TAG_TYPE_MT_NOT_READ • 0x02 – TAG_TYPE_MO_SENT • 0x03 – TAG_TYPE_MO_NOT_SENT

Optional TLVs

Name		Version introduced	Version last modified
Message Mode		Unknown	1.2

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Message Mode
Length	1			2	
Value	→	enum8	message_mode	1	Message mode. Values: • 0x00 – MESSAGE_MODE_CDMA – CDMA • 0x01 – MESSAGE_MODE_GW – GW

8.2.6.2. Response - QMI_WMS MODIFY_TAG RESP

- **Message type**

Response

- **Sender**

Service

- **Mandatory TLVs**

The Result Code TLV (defined in Section 8.1.3.1) is always present in the response.

- **Optional TLVs**

None

- **Error codes**

<code>QMI_ERR_NONE</code>	No error in the request
<code>QMI_ERR_INTERNAL</code>	Unexpected error occurred during processing
<code>QMI_ERR_MALFORMED_MSG</code>	Message was not formulated correctly by the control point or the message was corrupted during transmission
<code>QMI_ERR_NO_MEMORY</code>	Device could not allocate memory to formulate a response
<code>QMI_ERR_INVALID_ARG</code>	One of the parameters specified contains an invalid value
<code>QMI_ERR_INVALID_INDEX</code>	Memory storage index specified in the request is invalid
<code>QMI_ERR_NO_ENTRY</code>	No message exists at the specified memory storage designation
<code>QMI_ERR_MISSING_ARG</code>	A required TLV was not provided
<code>QMI_ERR_OP_DEVICE_UNSUPPORTED</code>	Selected operation is not supported by the device

8.2.6.3. Description of QMI_WMS MODIFY_TAG REQ/RESP

This command modifies the metadata tag of the message at the specified index in the specified memory storage.

The response is sent after all necessary operations are complete.

If the request attempts to modify the tag of an empty storage index, a `QMI_ERR_NO_ENTRY` error results.

The Message Mode TLV must be included if the device is capable of supporting more than one protocol. If

the TLV is not included, a QMI_ERR_MISSING_ARG error is returned.

8.2.7. QMI_WMS_DELETE

Deletes the message in a specified memory location.

- **WMS message ID**

0x0024

- **Version introduced**

Major - 1, Minor - 1

8.2.7.1. Request - QMI_WMS_DELETE_REQ

- **Message type**

Request

- **Sender**

Control point

- Mandatory TLVs

Name	Version introduced	Version last modified
Memory Storage	Unknown	1.1

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Memory Storage
Length	1			2	
Value	→	enum8	storage_type	1	Memory storage. Values: • 0x00 – STORAGE_TYPE_UIM • 0x01 – STORAGE_TYPE_NV

Optional TLVs

Name	Version introduced	Version last modified
Memory Index	Unknown	1.1
Message Tag	Unknown	1.1
Message Mode	Unknown	1.2

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Memory Index
Length	4			2	
Value	→	uint32	index	4	Indicates the storage index of the relevant message.

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x11			1	Message Tag
Length	1			2	
Value	→	enum8	tag_type	1	Message tag. Values: • 0x00 – TAG_TYPE_MT_READ • 0x01 – TAG_TYPE_MT_NOT_READ • 0x02 – TAG_TYPE_MO_SENT • 0x03 – TAG_TYPE_MO_NOT_SENT
Type	0x12			1	Message Mode
Length	1			2	
Value	→	enum8	message_mode	1	Message mode. Values: • 0x00 – MESSAGE_MODE_CDMA – CDMA • 0x01 – MESSAGE_MODE_GW – GW

8.2.7.2. Response - QMI_WMS_DELETE_RESP

- **Message type**

Response

- **Sender**

Service

- **Mandatory TLVs**

The Result Code TLV (defined in Section 8.1.3.1) 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_ARG	One of the parameters specified contains an invalid value
QMI_ERR_INVALID_INDEX	Memory storage index specified in the request is invalid
QMI_ERR_NO_ENTRY	No message exists at the specified memory storage designation
QMI_ERR_MISSING_ARG	A required TLV was not provided
QMI_ERR_OP_DEVICE_UNSUPPORTED	Selected operation is not supported by the device

8.2.7.3. Description of QMI_WMS_DELETE REQ/RESP

This command deletes one or more WMS messages from a given memory storage on the MSM device.

If no optional TLVs are specified, all messages are deleted from the storage location specified in the mandatory message store parameter.

The optional storage index and message tag parameters narrow the range of messages being deleted. If a message index is specified, the single message at that index from the specified memory store is deleted. If a message tag is specified, all messages in the specified memory store with a tag that matches the specified tag are deleted.

There are three ways to use this message:

Specify the memory storage only – Deletes all messages from the memory storage

Specify the memory storage and a message tag – Deletes all messages from the memory storage that

match the specific message tag

Specify the memory storage and a message index – Deletes only the message at the specific index from the memory storage

The message index and message tag TLVs may not be specified in the same request message. Doing so results in the QMI_ERR_INVALID_ARG error.

The Message Mode TLV must be included if the device is capable of supporting more than one protocol. If the TLV is not included, a QMI_ERR_MISSING_ARG error is returned.

All deletions are complete when the response is sent.

8.2.8. QMI_WMS_GET_MESSAGE_PROTOCOL

Queries the message protocol currently in use for the WMS client.

- **WMS message ID**

0x0030

- **Version introduced**

Major - 1, Minor - 1

8.2.8.1. Request - QMI_WMS_GET_MESSAGE_PROTOCOL_REQ

- **Message type**

Request

- **Sender**

Control point

- **Mandatory TLVs**

None

- **Optional TLVs**

None

8.2.8.2. Response - QMI_WMS_GET_MESSAGE_PROTOCOL_RESP

- **Message type**

Response

- **Sender**

Service

- **Mandatory TLVs**

The Result Code TLV (defined in Section 8.1.3.1) is always present in the response. The following mandatory TLV is present if the result code is QMI_RESULT_SUCCESS.

Name	Version introduced	Version last modified
Message Protocol	Unknown	1.1

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Message Protocol
Length	1			2	
Value	→	enum8	message_protocol	1	WMS message protocol. Values: • 0x00 – MESSAGE_PROTOCOL_CDMA • 0x01 – MESSAGE_PROTOCOL_WCDMA

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_MISSING_ARG	A required TLV was not provided
QMI_ERR_NO_MEMORY	Device could not allocate memory to formulate a response
QMI_ERR_OP_DEVICE_UNSUPPORTED	Selected operation is not supported by the device

8.2.8.3. Description of QMI_WMS_GET_MESSAGE_PROTOCOL REQ/RESP

This command queries the current messaging mode of the device.

If the device is capable of supporting more than one message protocol, a

QMI_ERR_OP_DEVICE_UNSUPPORTED error is returned.

8.2.9. QMI_WMS_LIST_MESSAGES

Requests a list of WMS message indices and meta information within the specified memory storage, matching a specified message tag.

- **WMS message ID**

0x0031

- **Version introduced**

Major - 1, Minor - 1

8.2.9.1. 3.11.1 Request - QMI_WMS_LIST_MESSAGES_REQ

- **Message type**

Request

- **Sender**

Control point

- **Mandatory TLVs**

Name	Version introduced	Version last modified
Requested Memory Storage	Unknown	1.1

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Requested Memory Storage
Length	1			2	
Value	→	enum8	storage_type	1	Memory storage. Values: • 0x00 – STORAGE_TYPE_UIM • 0x01 – STORAGE_TYPE_NV

Optional TLVs

Name	Version introduced	Version last modified
Requested Tag	Unknown	1.1
Message Mode	Unknown	1.2

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

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	enum8	tag_type	1	Message tag. Values: • 0x00 – TAG_TYPE_MT_READ • 0x01 – TAG_TYPE_MT_NOT_READ • 0x02 – TAG_TYPE_MO_SENT • 0x03 – TAG_TYPE_MO_NOT_SENT
Type	0x11			1	Message Mode
Length	1			2	
Value	→	enum8	message_mode	1	Message mode. Values: • 0x00 – MESSAGE_MODE_CDMA – CDMA • 0x01 – MESSAGE_MODE_GW – GW

8.2.9.2. Response - QMI_WMS_LIST_MESSAGES_RESP

- Message type

Response

- Sender

Service

- Mandatory TLVs

The Result Code TLV (defined in Section 8.1.3.1) is always present in the response. The following mandatory TLVs are present if the result code is QMI_RESULT_SUCCESS.

Name	Version introduced	Version last modified
Message List	Unknown	1.1

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Message List
Length	Var			2	
Value	→	uint32	N_messages	4	Number of sets of the following elements: <ul style="list-style-type: none">• message_index• tag_type
		uint32	message_index	4	Message index of each matched message.
		enum8	tag_type	1	Message tag. Values: <ul style="list-style-type: none">• 0x00 – TAG_TYPE_MT_READ• 0x01 – TAG_TYPE_MT_NOT_READ• 0x02 – TAG_TYPE_MO_SENT• 0x03 – TAG_TYPE_MO_NOT_SENT

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_MISSING_ARG	A required TLV was not provided
QMI_ERR_INVALID_ARG	One of the parameters specified contains an invalid value
QMI_ERR_OP_DEVICE_UNSUPPORTED	Selected operation is not supported by the device

8.2.9.3. Description of QMI_WMS_LIST_MESSAGES REQ/RESP

This command generates and returns the number of WMS messages within the specified MSM memory storage.

An optional tag can be used to narrow the search criteria. When this optional tag is specified, only messages within the specified memory storage that match the specified tag are returned.

A successful response includes a count of messages matching the search criteria, along with a list of

indices and tags for each matching message.

The Message Mode TLV must be included if the device is capable of supporting more than one protocol. If the TLV is not included, a QMI_ERR_MISSING_ARG error is returned.

8.2.10. QMI_WMS_SET_ROUTES

Sets the action performed upon WMS message receipt for the specified message routes. It also sets the action performed upon WMS receipt of status reports.

- **WMS message ID**

0x0032

- **Version introduced**

Major - 1, Minor - 1

8.2.11. Request - QMI_WMS_SET_ROUTES_REQ

- **Message type**

Request

- **Sender**

Control point

- **Mandatory TLVs**

Name	Version introduced	Version last modified
Route List	Unknown	1.1

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Route List
Length	Var			2	
Value	→	uint16	n_routes	2	Number of sets of the following elements: • message_type • message_class • route_storage • receipt_action
		enum8	message_type	1	Message type matching this route. Values: • 0x00 - MESSAGE_TYPE_POINT_TO_POINT - Point-to-Point

Field	Field value	Field type	Parameter	Size (byte)	Description
		enum8	message_class	1	<p>Message class. Values:</p> <ul style="list-style-type: none"> • 0x00 – MESSAGE_CLASS_0 – Class 0 • 0x01 – MESSAGE_CLASS_1 – Class 1 • 0x02 – MESSAGE_CLASS_2 – Class 2 • 0x03 – MESSAGE_CLASS_3 – Class 3 • 0x04 – MESSAGE_CLASS_NONE – Class None • 0x05 – MESSAGE_CLASS_CDMA – Class CDMA
		enum8	route_storage	1	<p>If the action is store, where to store the incoming message. Values:</p> <ul style="list-style-type: none"> • 0x00 – STORAGE_TYPE_UIM • 0x01 – STORAGE_TYPE_NV • -1 – STORAGE_TYPE_NONE
		enum8	receipt_action	1	<p>Action to be taken on receipt of a message matching the specified type and class for this route. Values:</p> <ul style="list-style-type: none"> • 0x00 – DISCARD – Incoming messages for this route are discarded by the WMS service without notifying QMI_WMS clients • 0x01 – STORE_AND_NOTIFY – Incoming messages for this route are stored to the specified device memory, and new message notifications are sent to registered clients • 0x02 – TRANSFER_ONLY – Incoming messages for this route are transferred to the client, and the client is expected to send ACK to the network • 0x03 – TRANSFER_AND_ACK – Incoming messages for this route are transferred to the client, and ACK is sent to the network

Optional TLVs

Name	Version introduced	Version last modified
Transfer Status Report**	Unknown	1.2

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Transfer Status Report**
Length	1			2	
Value	→	enum8	transfer_ind	1	Values: • 0x01 – TRANSFER_IND_CLIENT – Status reports are transferred to the client

8.2.11.1. Response - QMI_WMS_SET_ROUTES_RESP

- Message type

Response

- Sender

Service

- Mandatory TLVs

The Result Code TLV (defined in Section 8.1.3.1) 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_MISSING_ARG	A required TLV was not provided
QMI_ERR_ARG_TOO_LONG	Argument passed in a TLV was larger than the available storage in the device
QMI_ERR_INVALID_ARG	One of the parameters specified contains an invalid value

8.2.11.2. Description of QMI_WMS_SET_ROUTES REQ/RESP

This command sets the routing action taken upon receipt of incoming WMS messages per message class. A storage location on the MSM device, and whether indications must be sent to interested WMS clients, may be specified separately for each message category.

All routes need not be set at the same time. Routes not specified in the request message are left unchanged.

Specifying route_instances as zero results in a QMI_ERR_INVALID_ARG error. Similarly, if the number of route tuples given does not match route_instances, a QMI_ERR_INVALID_ARG error is returned.

When multiple routes are specified, error checking is performed on all specified routes before any routes values are changed. If any of the specified routes contains an unsupported or invalid value, the entire requested action is cancelled and no route modifications are made.

For transfer-only and transfer and ACK routes, the route_storage field is ignored in the request.

When the optional Transfer Status Report TLV is present, status reports are transferred to the client. If this TLV is not present, status reports are stored on the SIM if a matching MO message is found on the SIM; otherwise, they are transferred to the client.

A successful response indicates that the specified message routes have been changed.

Under some circumstances, the route setting by the client is not honored.

In the following section:

- + indicates that the route change is implementation-specific
- ++ indicates that the route change is based on an interpretation of the standards

The route is modified by the AMSS WMS module in the following instances:

- For WAP messages, the route is set to transfer and ACK+
- For broadcast messages, the route is set to transfer and ACK+

For MT CDMA messages:

- In the following cases, the route is set to store and notify:
 - Voicemails, message waiting indications+
 - Card Application Toolkit Protocol Teleservice (CATPT) and PP download messages, if the services are not available++
- In the following case, the route is set to transfer and ACK:
 - Flash messages+

For MT GW PP messages:

- For voicemails, the route is set to store and notify if the message needs to be stored, or to transfer and ACK if the message needs to be discarded+
- For messages with PID = 0x40 (short message type 0), the route is set to transfer and ACK+
- If the QMI_WMS_SET_PRIMARY_CLIENT request has been used to set the client as the primary

client:

- In the following case, the route is set to transfer only.

8.2.12. QMI_WMS_GET_ROUTES

Queries the currently configured action performed upon WMS message receipt for the specified message routes. It also queries the action performed upon WMS receipt of status reports.

- **WMS message ID**

0x0033

- **Version introduced**

Major - 1, Minor - 1

8.2.13. Request - QMI_WMS_GET_ROUTES_REQ

- **Message type**

Request

- **Sender**

Control point

- **Mandatory TLVs**

None

- **Optional TLVs**

None

8.2.13.1. Response - QMI_WMS_GET_ROUTES_RESP

- **Message type**

Response

- **Sender**

Service

- **Mandatory TLVs**

The Result Code TLV (defined in Section 8.1.3.1) is always present in the response. The following

mandatory TLVs are present if the result code is QMI_RESULT_SUCCESS.

Name	Version introduced	Version last modified
Route List	Unknown	1.5

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Route List
Length	Var			2	
Value	→	uint16	route_instances	2	Number of sets of the following elements: <ul style="list-style-type: none"> • route_type • route_class • route_memory • route_value
		enum8	route_type	1	Message type matching this route. Values: <ul style="list-style-type: none"> • 0x00 – MESSAGE_TYPE_POINT_TO_POINT – Point-to-Point
		enum8	route_class	1	Message class. Values: <ul style="list-style-type: none"> • 0x00 – MESSAGE_CLASS_0 – Class 0 • 0x01 – MESSAGE_CLASS_1 – Class 1 • 0x02 – MESSAGE_CLASS_2 – Class 2 • 0x03 – MESSAGE_CLASS_3 – Class 3 • 0x04 – MESSAGE_CLASS_NONE – Class None • 0x05 – MESSAGE_CLASS_CDMA – Class CDMA
		enum8	route_memory	1	Memory storage. Values: <ul style="list-style-type: none"> • 0x00 – STORAGE_TYPE_UIM • 0x01 – STORAGE_TYPE_NV • -1 – STORAGE_TYPE_NONE

Field	Field value	Field type	Parameter	Size (byte)	Description
		enum8	route_value	1	<p>Route value. Values:</p> <ul style="list-style-type: none"> • 0x00 – DISCARD – Incoming messages for this route are discarded by the WMS service, and no notification is sent to clients • 0x01 – STORE_AND_NOTIFY – Incoming messages for this route are stored to the specified device memory, and new message notifications are sent to registered clients • 0x02 – TRANSFER_ONLY – Incoming messages for this route are transferred to the client, and the client is expected to send ACK to the network • 0x03 – TRANSFER_AND_ACK – Incoming messages for this route are transferred to the client, and ACK is sent to the network • -1 – UNKNOWN – Incoming messages for this route are handled in a way that is unknown or unsupported by QMI_WMS

Optional TLVs

Name	Version introduced	Version last modified
Transfer Status Report**	Unknown	1.2

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Transfer Status Report**
Length	1			2	
Value	→	enum8	transfer_ind	1	<p>Values:</p> <ul style="list-style-type: none"> • 0x00 – TRANSFER_IND_SIM – Status reports are stored on the SIM if a matching MO record is found on the SIM; otherwise, status reports are transferred to the client • 0x01 – TRANSFER_IND_CLIENT – Status reports are transferred to the client

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

8.2.13.2. Description of QMI_WMS_GET_ROUTES REQ/RESP

This command queries the behavior used to route new MT messages to MSM memory storage and WMS clients.

The response indicates the current actions for all messaging routes on the MSM device taken upon receipt of an incoming WMS message matching that route.

If the optional TLV Transfer Status Report is included, it indicates the current action taken upon receipt of an incoming Status Report.

8.2.14. QMI_WMS_GET_SMSC_ADDRESS

Queries the currently configured SMSC address.

- **WMS message ID**

0x0034

- **Version introduced**

Major - 1, Minor - 1

8.2.14.1. Request - QMI_WMS_GET_SMSC_ADDRESS_REQ

- **Message type**

Request

- **Sender**

Control point

- **Mandatory TLVs**

None

- Optional TLVs

Name	Version introduced	Version last modified
SMSC Address Index	1.21	1.21

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	SMSC Address Index
Length	1			2	
Value	→	uint8	index	1	Memory index to read a SMSC address from a specific index in EF-SMSP.

8.2.14.2. Response - QMI_WMS_GET_SMSC_ADDRESS_RESP

- Message type

Response

- Sender

Service

- Mandatory TLVs

The Result Code TLV (defined in Section 8.1.3.1) is always present in the response. The following mandatory TLVs are present if the result code is QMI_RESULT_SUCCESS.

Name	Version introduced	Version last modified
SMSC Address	Unknown	1.1

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	SMSC Address
Length	Var			2	
Value	→	char	smsc_address_type	3	Type of SMSC address given in ASCII digits (must be three digits long, with leading zeros used as placeholders)
		uint8	smsc_address_length	1	Number of sets of the following elements: <ul style="list-style-type: none">• smsc_address_digits
		char	smsc_address_digits	Var	Address of the SMSC given in ASCII digits; can be prefixed with + (maximum 20 digits, not including the +)

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_OP_DEVICE_UNSUPPORTED	Selected operation is not supported by the device
QMI_ERR_DEVICE_NOT_READY	Device has not yet read this value
QMI_ERR_NOT_PROVISIONED	Device does not have this value provisioned
QMI_ERR_INVALID_INDEX	Storage index specified in the request is invalid

8.2.14.3. Description of QMI_WMS_GET_SMSC_ADDRESS REQ/RESP

This command queries the SMSC address that is currently configured for the device. The AT command equivalent to this command is AT+CSCA (refer to 3GPP TS 27.005).

The control point may provide the optional SMSC Address Index TLV to read the SMSC address from a specific index in EF-SMSP. If the optional TLV is missing, the SMSC address is read from EF-SMSP at index 0 (if no valid record is found) or the index of the most recent valid record.

The SMSC address is applicable to 3GPP devices only. Attempts to read the SMSC address setting from a non-3GPP device elicit a QMI_ERR_OP_DEVICE_UNSUPPORTED error.

8.2.15. QMI_WMS_SET_SMSC_ADDRESS

Sets the SMSC address used when storing or saving SMS messages.

- **WMS message ID**

0x0035

- **Version introduced**

Major - 1, Minor - 1

8.2.15.1. Request - QMI_WMS_SET_SMSC_ADDRESS_REQ

- **Message type**

Request

- **Sender**

Control point

- Mandatory TLVs

Name	Version introduced	Version last modified
SMSC Address	Unknown	1.1

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	SMSC Address
Length	Var			2	
Value	→	string	smsc_address_digits	Var	NULL-terminated string containing the address of the SMSC, given in ASCII digits; can be prefixed with + (maximum 20 digits, not including the +)

Optional TLVs

Name	Version introduced	Version last modified
SMSC Address Type	Unknown	1.1
SMSC Address Index	1.20	1.20

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	SMSC Address Type
Length	Var			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	string	smsc_address_type	Var	NULL-terminated string containing the type of SMSC address, given in ASCII digits (maximum three digits)
Type	0x11			1	SMSC Address Index
Length	1			2	
Value	→	uint8	index	1	Indicates the record index where the SMSC address needs to be written.

8.2.15.2. Response - QMI_WMS_SET_SMSC_ADDRESS_RESP

- Message type

Response

- Sender

Service

- **Mandatory TLVs**

The Result Code TLV (defined in Section 8.1.3.1) is always present in the response.

- **Optional TLVs**

None

- **Error codes**

<code>QMI_ERR_NONE</code>	No error in the request
<code>QMI_ERR_INTERNAL</code>	Unexpected error occurred during processing
<code>QMI_ERR_MALFORMED_MSG</code>	Message was not formulated correctly by the control point or the message was corrupted during transmission
<code>QMI_ERR_MISSING_ARG</code>	A required TLV was not provided
<code>QMI_ERR_ARG_TOO_LONG</code>	Argument passed in a TLV was larger than the available storage in the device
<code>QMI_ERR_INVALID_LOG</code>	One of the parameters specified contains an invalid value
<code>QMI_ERR_OP_DEVICE_UNSUPPORTED</code>	Device does not support this message
<code>QMI_ERR_INVALID_INDEX</code>	Storage index specified in the request is invalid
<code>QMI_ERR_DEVICE_STORAGE_FULL</code>	SIM storage is full

8.2.15.3. Description of QMI_WMS_SET_SMSC_ADDRESS REQ/RESP

This command sets the SMSC address that is used by the device when sending or storing SMS messages. If the SMSC address provided is prefixed with a plus sign (+), the SMSC address type defaults to 145, regardless of whether the type is specified. If the optional SMSC address type is not provided and the address is not prefixed with a +, the address type defaults to 129. The AT command equivalent to this command is **AT+CSCA** (refer to 3GPP TS 27.005).

The control point may provide the optional SMSC Address Index TLV to store the SMSC address in a specific index in EF-SMSP. If the optional TLV is missing, the SMSC address is written to EF-SMSP at index 0 (if no valid record is found) or the index of the most recent valid record.

The SMSC address is applicable to 3GPP devices only. Attempts to set the SMSC address from a non-3GPP device elicit a `QMI_ERR_OP_DEVICE_UNSUPPORTED` error.

8.2.16. QMI_WMS_GET_STORE_MAX_SIZE

Queries the maximum number of messages that can be stored per memory storage, as well as the number of slots currently available.

- **WMS message ID**

0x0036

- Version introduced

Major - 1, Minor - 1

8.2.16.1. Request - QMI_WMS_GET_STORE_MAX_SIZE_REQ

- Message type

Request

- Sender

Control point

- Mandatory TLVs

Name	Version introduced	Version last modified
Memory Store	Unknown	1.1

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Memory Store
Length	1			2	
Value	→	enum8	storage_type	1	Memory storage. Values: • 0x00 – STORAGE_TYPE_UIM • 0x01 – STORAGE_TYPE_NV

Optional TLVs

Name	Version introduced	Version last modified
Message Mode	Unknown	1.2

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Message Mode
Length	1			2	
Value	→	enum8	message_mode	1	Message mode. Values: • 0x00 – MESSAGE_MODE_CDMA • 0x01 – MESSAGE_MODE_GW

8.2.16.2. Response - QMI_WMS_GET_STORE_MAX_SIZE_RESP

- **Message type**

Response

- **Sender**

Service

- **Mandatory TLVs**

The Result Code TLV (defined in Section 8.1.3.1) is always present in the response. The following mandatory TLV is present if the result code is QMI_RESULT_SUCCESS.

Name	Version introduced	Version last modified
Memory Store Size	Unknown	1.1

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Memory Store Size
Length	4			2	
Value	→	uint32	mem_store_max_size	4	Maximum number of messages for this memory storage.

Optional TLVs

Name	Version introduced	Version last modified
Memory Available	Unknown	1.2

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Memory Available
Length	4			2	
Value	→	uint32	free_slots	4	Number of slots currently available for this memory storage.

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_MISSING_ARG	A required TLV was not provided
QMI_ERR_NO_MEMORY	Device could not allocate memory to formulate a response

8.2.16.3. Description of QMI_WMS_GET_STORE_MAX_SIZE REQ/RESP

This command queries for the maximum size of a specified memory storage.

If the optional Memory Available TLV is included, it indicates the number of available slots in the specified memory storage.

The Message Mode TLV must be included if the device is capable of supporting more than one protocol. If the TLV is not included, a QMI_ERR_MISSING_ARG error is returned.

8.2.17. QMI_WMS_GET_DOMAIN_PREF

Queries the GW domain preference. (Deprecated)

- **WMS message ID**

0x0040

- **Version introduced**

Major - 1, Minor - 2

8.2.17.1. Request - QMI_WMS_GET_DOMAIN_PREF_REQ

- **Message type**

Request

- **Sender**

Control point

- **Mandatory TLVs**

None

- **Optional TLVs**

None

8.2.17.2. Response - QMI_WMS_GET_DOMAIN_PREF_RESP

- **Message type**

Response

- Sender

Service

- **Mandatory TLVs**

The Result Code TLV (defined in Section 8.1.3.1) is always present in the response. The following mandatory TLVs are present if the result code is QMI_RESULT_SUCCESS.

Name	Version introduced	Version last modified
Domain Pref	Unknown	1.2

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Domain Pref
Length	1			2	
Value	→	enum8	domain_pref	1	GW domain preference. Values: • 0x00 – DOMAIN_PREF_CS – CS preferred • 0x01 – DOMAIN_PREF_PS – PS preferred • 0x02 – DOMAIN_PREF_CS_ONLY – CS only • 0x03 – DOMAIN_PREF_PS_ONLY – PS only

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_OP_DEVICE_UNSUPPORTED	Selected operation is not supported by the device
QMI_ERR_MISSING_ARG	A required TLV was not provided
QMI_ERR_INVALID_ARG	One of the parameters specified contains an invalid value

8.2.17.3. Description of QMI_WMS_GET_DOMAIN_PREF_REQ/RESP

This command queries the GW domain preference.

The GW domain preference is applicable to 3GPP devices only. Attempts to retrieve the GW domain

preference from a non-3GPP device elicit a QMI_ERR_OP_DEVICE_UNSUPPORTED error.

This command is deprecated. Use QMI_WMS_GET_DOMAIN_PREF_CONFIG to get the GW domain preference.

8.2.18. QMI_WMS_SET_DOMAIN_PREF

Sets the GW domain preference. (Deprecated)

- **WMS message ID**

0x0041

- **Version introduced**

Major - 1, Minor - 2

8.2.18.1. Request - QMI_WMS_SET_DOMAIN_PREF_REQ

- **Message type**

Request

- **Sender**

Control point

- **Mandatory TLVs**

Name	Version introduced	Version last modified
Domain Pref	Unknown	1.2

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Domain Pref
Length	1			2	
Value	→	enum8	domain_pref	1	GW domain preference. Values: • 0x00 – DOMAIN_PREF_CS – CS preferred • 0x01 – DOMAIN_PREF_PS – PS preferred • 0x02 – DOMAIN_PREF_CS_ONLY – CS only • 0x03 – DOMAIN_PREF_PS_ONLY – PS only

8.2.18.2. Response - QMI_WMS_SET_DOMAIN_PREF_RESP

- **Message type**

Response

- **Sender**

Service

- **Mandatory TLVs**

The Result Code TLV (defined in Section 8.1.3.1) is always present in the response.

- **Optional TLVs**

None

- **Error codes**

<code>QMI_ERR_NONE</code>	No error in the request
<code>QMI_ERR_INTERNAL</code>	Unexpected error occurred during processing
<code>QMI_ERR_MALFORMED_MSG</code>	Message was not formulated correctly by the control point or the message was corrupted during transmission
<code>QMI_ERR_MISSING_ARG</code>	A required TLV was not provided
<code>QMI_ERR_INVALID_ARG</code>	One of the parameters specified contains an invalid value
<code>QMI_ERR_OP_DEVICE_UNSUPPORTED</code>	Selected operation is not supported by the device

8.2.18.3. Description of QMI_WMS_SET_DOMAIN_PREF REQ/RESP

This command sets the GW domain preference.

The GW domain preference is applicable to 3GPP devices only. Attempts to set the GW domain preference from a non-3GPP device elicit a `QMI_ERR_OP_DEVICE_UNSUPPORTED` error.

This command is deprecated. Use `QMI_WMS_SET_DOMAIN_PREF_CONFIG` to set the GW domain preference.

9 Position Determination Service (QMI_PDS)

This specification documents Major Version 1 of the Qualcomm Messaging Interface (QMI) for Position Determination Service (QMI_PDS).

QMI_PDS provides applications running on a tethered device, i.e., Terminal Equipment (TE), with commands related to Position Determination (PD) to:

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

It is expected that user-level applications (e.g., connection managers and/or device drivers on the TE) use QMI_PDS to access this functionality on the MSM device

9.1. Theory of Operation

9.1.1. Generalized QMI Service Compliance

The QMI_PDS service complies with the generalized QMI service specification, including the rules for messages, indications and responses, byte ordering, arbitration, constants, result.

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

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

9.1.2. PDS Service Type

PDS is assigned QMI service type 0x06.

9.1.3. Message Definition Template

9.1.3.1. Response Message Result TLV

This Type-Length-Value (TLV) is present in all Response messages defined in this document. It is not present in the Indication messages.

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

9.1.4. QMI_PDS Fundamental Concepts

9.1.4.1. GPS

GPS uses a network of orbiting satellites to provide the MSM device with accurate location measurements. Refer to for a detailed explanation.

A wireless MSM device supporting QMI_PDS provides control and accessibility to the GPS service functionality of the device.

9.1.4.2. Position Determination Methods

The location engine supports three methods for determining the position: MS-assisted, MS-based, and Standalone. 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. A standalone fix is one in which the entire position fix process is done within the location engine without interaction with the PDE/PDM. 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 and MS-based call flows. Since standalone does not interact with the network (PDE, PDM, etc.), it has no call flow.

9.1.4.3. 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. 2.4.2.1.1 MS-assisted Call Flow Below feature illustrates the steps required to perform an MS-assisted PD, where the MS position is calculated by the PDE.

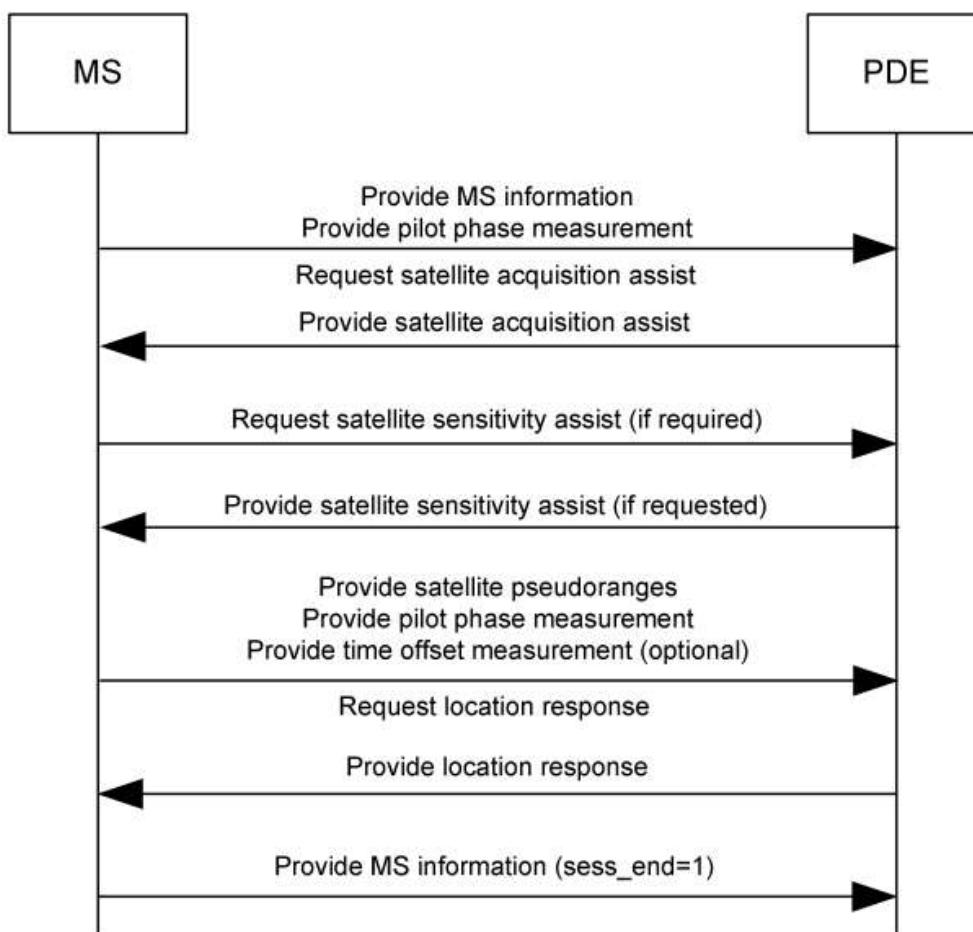


Figure 10: MS-assisted call Flow Example

9.1.4.4. 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_PDS_SET_DEFAULT_TRACKING_SESSION (session_operation=0x02).
2. The PD process begins when the client application sends one of the following messages:
 - QMI_PDS_START_TRACKING_SESSION
 - QMI_PDS_DETERMINE_POSITION
 - QMI_PDS_SET_AUTO_TRACKING_STATE (auto_tracking_state = 0x01)

9.1.4.5. 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

9.1.4.6. MS-based Call Flow

The below figure illustrates an example of an MS-based call flow.

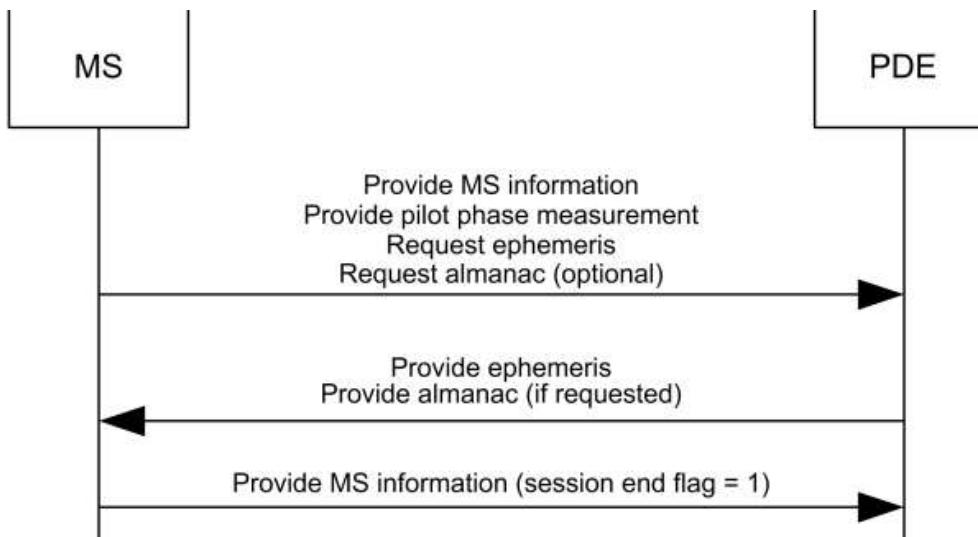


Figure 11: MS-Based Call Flow Example

9.1.4.6.1. 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_PDS_SET_DEFAULT_TRACKING_SESSION (session_operation=0x01).

2. The PD process begins when the client application sends one of the following messages:

- QMI_PDS_START_TRACKING_SESSION
- QMI_PDS_DETERMINE_POSITION
- QMI_PDS_SET_AUTO_TRACKING_STATE (auto_tracking_state = 0x01)

9.1.4.6.2. 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.

9.1.4.6.3. 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_PDS_SET_DEFAULT_TRACKING_SESSION (session_operation=0x00).

2. The PD process begins when the client application sends one of the following messages:

- QMI_PDS_START_TRACKING_SESSION
- QMI_PDS_DETERMINE_POSITION

- QMI_PDS_SET_AUTO_TRACKING_STATE (auto_tracking_state = 0x01)

9.1.4.7. Tracking Sessions

Tracking sessions are used to request and obtain a continuous stream of fixes from the MSM GPS service. There are three types of tracking sessions: QMI autotracking, NMEA port autotracking, and QMI manual tracking.

QMI autotracking allows the client to initiate and continue a tracking session without having to periodically strobe the GPS engine. The QMI_PDS_SET_AUTO_TRACKING_STATE is used to start and stop the tracking session. Once the tracking session is started, the MSM GPS service continues to compute position fixes at the desired interval.

When using NMEA port autotracking, NMEA sentences can be generated automatically by the MSM GPS service until there is a request to stop. This functionality can be initiated by:

- An application opening the serial COM port (if this configuration is enabled); does not require a QMI_PDS control point
- A QMI_PDS control point issuing an autotracking-enabled request

Automatic tracking starts when at least one QMI_PDS control point requests to enable it or the serial COM port is opened. Multiple QMI_PDS control points can simultaneously request for autotracking to be enabled as well. The autotracking session will remain active and fixes are continuously generated until all of the entities that requested autotracking to be enabled no longer require it. Ending a tracking session can be performed in the following ways:

- A QMI PDS control point explicitly disables it.
- A QMI PDS control point resets.
- A QMI PDS control point is closed.
- A QMI PDS control point disables the NMEA COM port.

NMEA data output is configurable;

After a tracking session has ended, any control point may then start a new tracking session if additional fixes are required.

Any control point may disable the MSM GPS service at any time, even if there is a tracking session active that the control point had not initiated.

For NMEA and QMI autotracking sessions, a default tracking session configuration is stored and used for automatic tracking sessions. This configuration can be updated by using the QMI_PDS_SET_DEFAULT_TRACKING_SESSION request. This sets the default configuration without starting a new tracking session.

9.1.4.8. Manual Tracking Sessions

NOTE

Manual tracking is documented here for legacy reasons; it is recommended that future implementation use autotracking.

Manual tracking sessions are used to instruct the MSM GPS service to determine the position for a finite number of fix requests.

Starting a tracking session activates the MSM GPS service and generates the first position fix. If more than one fix is requested the control point must request additional fixes for the duration of the tracking session. These requests should be performed at the time interval that was specified when the tracking session was started. Once a position determination request has been completed the fix data will generate an event report indication sent to all control points that have previously registered for a specific data format. NMEA data output is also configurable.

When all of the requested fixes have been performed the MSM GPS service automatically ends the tracking session. Sessions may also be ended early if no further fixes are needed.

Only one tracking session may be run at a time by the MSM GPS service. If the tracking session was started by a QMI_PDS control point, only that control point may request position determination fixes (if the tracking session control method is manual) or end the tracking session early. However, any control point may disable the MSM GPS service at any time, even if there is a tracking session active that the control point did not initiate.

If auto-NMEA sentence generation is supported by the device, a default tracking session configuration is stored and used for automatic tracking sessions. This configuration can be updated two ways:

- Start tracking session request – Sets the default configuration and then starts a tracking session
- Set default tracking session request – Sets just the default configuration without starting a new tracking session

9.1.4.9. Single-Shot PD

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

9.1.4.10. NMEA Sentence Data

QMI_PDS offers two separate mechanisms for providing GPS data in the NMEA sentence [S1] format to existing TE-based applications.

The first method is data transferred over a serial COM port controlled by the GPS service on the MSM device. In this way, the NMEA sentences are sent to the TE-based application in a raw form on the serial communications port, without any QMUX or QMI service message encoding, i.e., the NMEA I/O channel is orthogonal to the QMI I/O channel used to control the GPS service on the MSM device.

The second method is data transferred to the QMI_PDS control point directly. Control points that have optionally registered for this event report will receive the same NMEA sentence sent as a QMI indication. The NMEA sentence is delivered in the same format as is output to the serial communications port but encapsulated in a QMI indication message.

QMI_PDS provides messages to configure various NMEA settings:

- The serial COM port, if any, on which to send NMEA sentences
- Which NMEA sentence types are to be generated
- How often NMEA sentences are generated (while the fix is being performed)

While many serial COM ports may exist, NMEA sentences are only delivered to an individual port. Additionally, the NMEA configuration is global and any change made by a QMI_PDS control point or any other MSM-based GPS client affects the generation of NMEA sentence data for all clients.

9.1.4.11. External Information Injection

9.1.4.11.1. External Time Injection

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

9.1.4.11.2. Coarse Position Injection

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

9.1.4.11.3. Wi-Fi Position Injection

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

9.1.4.12. gpsOneXTRA Satellite Database Information

QMI_PDS can support an extension of the standalone position determination method known as eXTended Receiver Assistance (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.

The MSM GPS service contains an embedded client that can accept XTRA download requests from the GPS engine. The data is obtained from an XTRA server on the network over the WWAN interface. Alternatively, QMI_PDS can be configured to send download requests to an external XTRA client sitting above QMI. This allows for the client to use other network interfaces to download the XTRA data and then inject it to the MSM GPS service using the QMI_PDS_INJECT_XTRA_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_PDS 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.

To inject an XTRA file from an external XTRAClient:

1. Disable the embedded data client state using QMI_PDS_SET_XTRA_PARAMETERS (this step must only be done once, since the setting is stored persistently in NV memory).
2. Register for the external XTRA database request in QMI_PDS_SET_EVENT_REPORT (this step only needs to be done once at power-up).
3. 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_PDS_FORCE_XTRA_DOWNLOAD or if the MSM GPS service needs a fresh download at the beginning of a positioning session).
4. Download the XTRA database file from the server using at least one of the URLs in the indication.
5. Inject the downloaded XTRA database file using QMI_PDS_INJECT_XTRA_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 2.

9.1.4.13. 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 Global Navigation Satellite Services (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 MSM GPS service attempts to make use of the SBAS.

9.1.4.14. External Sensor Data Input

The MSM GPS service has the ability to utilize various types of sensor data injected by a control point. A control point must register for sensor data requests from the MSM GPS service using the QMI_PDS_SET_EVENT_REPORT message. The MSM GPS service will indicate when it is ready/not ready to receive sensor data inputs from the control point by sending the appropriate TLVs in the QMI_PDS_EVENT_REPORT_IND message. A separate TLV will be used for each sensor type that is supported (e.g., 3-axis accelerometer, 3-axis gyro). The control point must inject sensor data using the QMI_PDS_INJECT_SENSOR_DATA message. In the case of data sourced from the device sensors, the control point must inject sensor data using the QMI_PDS_INJECT_SENSOR_DATA message. Separately, in the case of data sourced from sensors mounted on a vehicle, the control point must inject sensor data using the QMI_PDS_INJECT_VEHICLE_SENSOR_DATA message. The MSM GPS service supports concurrent injection of data sourced from device and vehicle sensors.

Time synchronization between the GPS processor and external sensor processor is crucial for the MSM 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 TLV using the QMI_PDS_SET_EVENT_REPORT message. The MSM GPS service will periodically send a time sync request in the QMI_PDS_EVENT_REPORT_IND message with a reference counter. The control point is then expected to return this counter along with the sensor processor's time to the MSM GPS service using the QMI_PDS_INJECT_TIME_SYNC_DATA message. This handshake allows the MSM GPS service to maintain a time correlation between the processors and use the sensor data effectively.

Sensor-specific configuration parameters can be queried and set by a control point using the QMI_PDS_GET_SENSOR_CONFIG and QMI_PDS_SET_SENSOR_CONFIG messages. QMI PDS defines sensor time or sensor processor time as a monotonically increasing counter with jitter ≤ 1 msec. This counter must never be stopped until the processor is rebooted. This time source must be used in the QMI_PDS_INJECT_SENSOR_DATA and QMI_PDS_INJECT_TIME_SYNC_DATA messages.

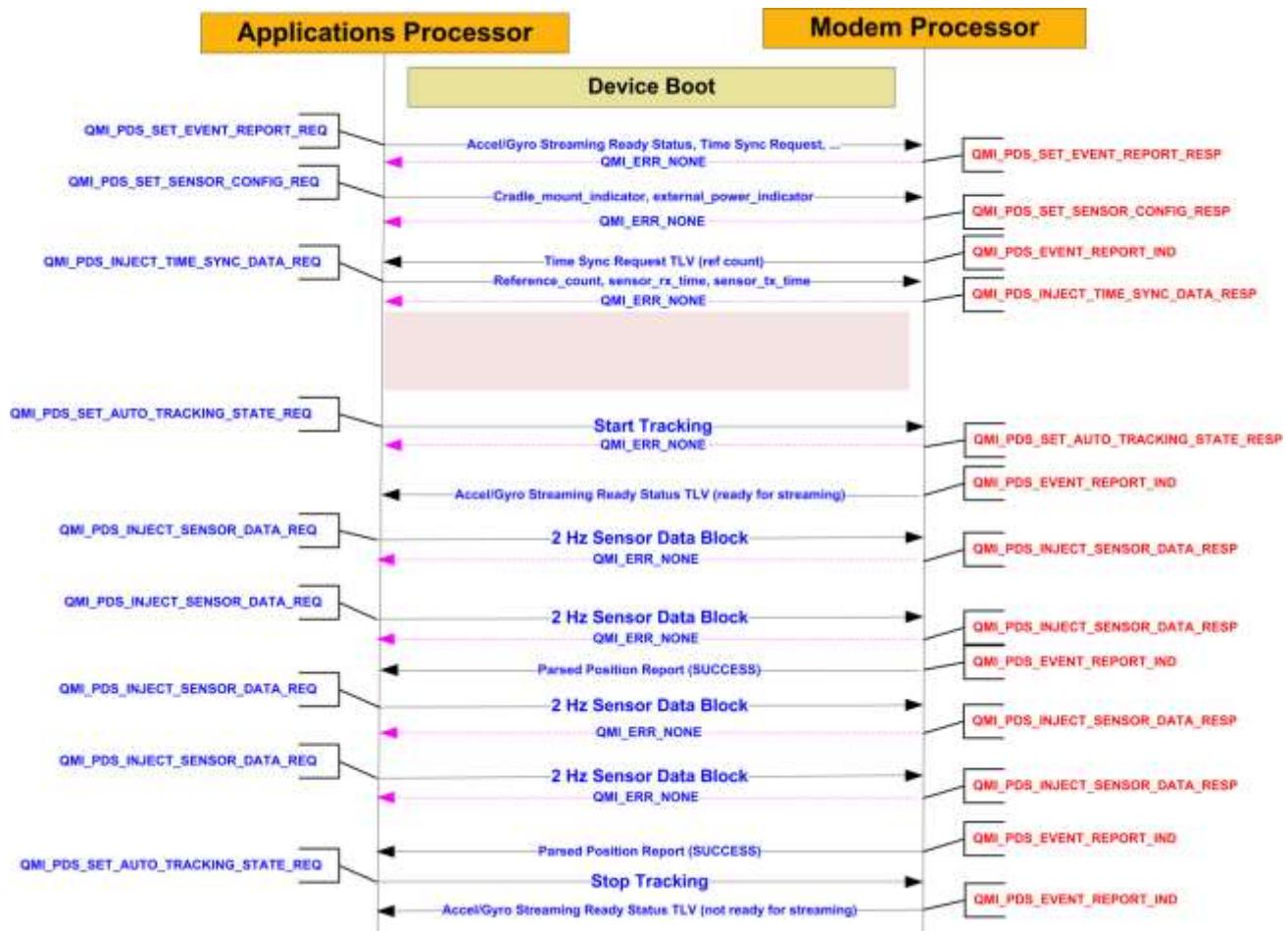


Figure 12: Call Flow for Configuration, Request, and Injection of Sensor and Time sync Data

9.1.4.15. Device Accelerometer Measurements

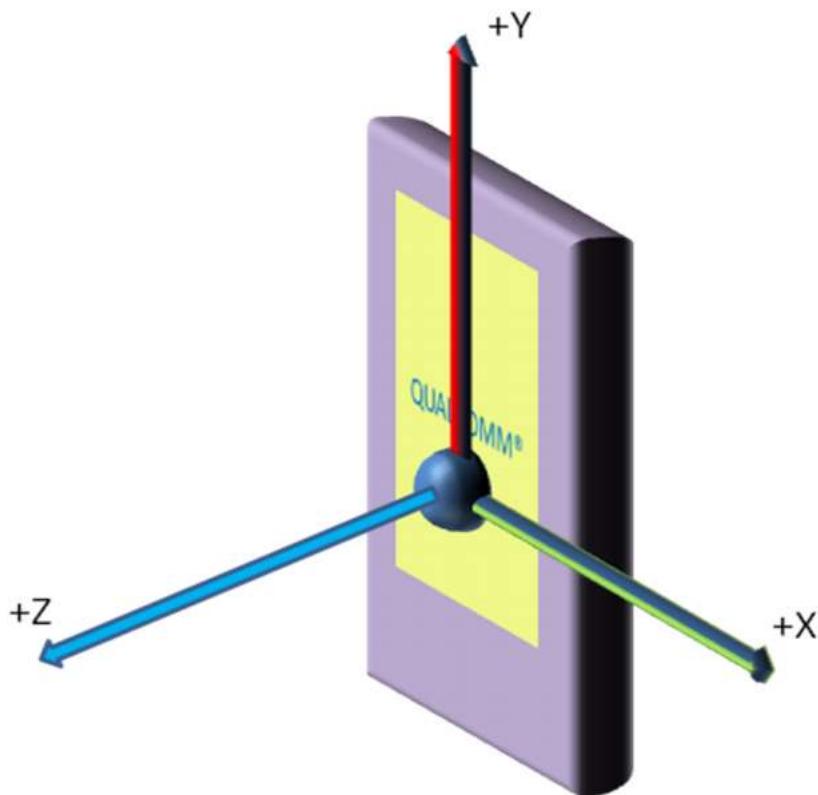


Figure 13: Orientation of coordinate axes for acceleration measurements

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/s²) 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 above figure is approximately +9.81 m/s², 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 QMI_PDS_INJECT_SENSOR_DATA_REQ message.

9.1.4.16. Device Gyroscope Measurements

The gyroscope reading of a given 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. Below figure illustrates the positive rotation right-hand rule.

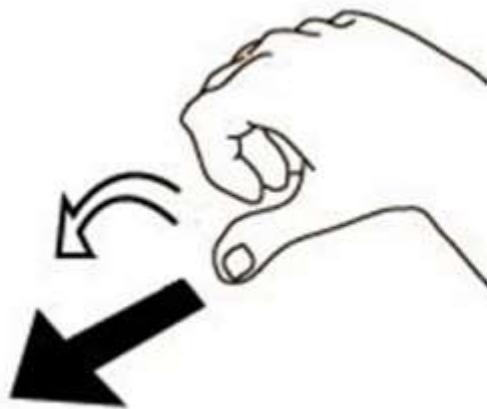


Figure 14: Right-hand rule

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

9.1.4.17. Vehicle Sensor Measurements

The information in this section pertains to the QMI_PDS_INJECT_VEHICLE_SENSOR_DATA message. This message may be used to inject data obtained from sensors mounted on a vehicle. Supported types of vehicle sensor measurements include a 3-axis accelerometer, 3-axis gyroscope, and odometer.

9.1.4.18. Accelerometer and Gyroscope Measurements

As specified by the Society of Automotive Engineers (SAE) standards, acceleration and angular rate measurements are defined relative to the axes illustrated in Figure 2-6.

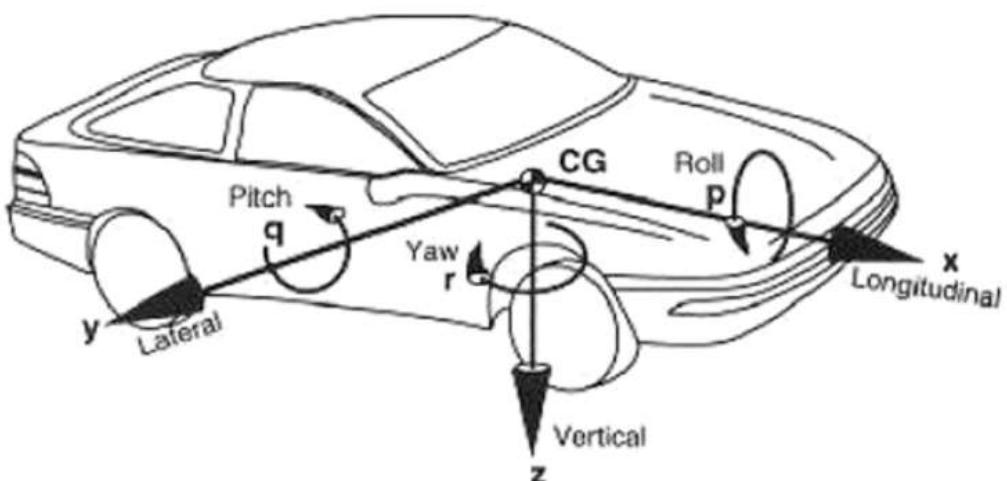


Figure 15: Vehicle Axes

Where:

x=Vehicle forward and roll

y=Vehicle right and pitch

z=Vehicle down and yaw/heading

Examples and further details regarding the axes sign convention are:

- Sign of acceleration – For a vehicle accelerating forward, a positive value for x acceleration is expected.
- Gravity effects – The values reported here are not expected to try to remove gravity effects on accelerometers. Acceleration, as typically measured by accelerometers, is with respect to “free-fall” space, e.g., where the ground pushes up on the car at 1g times the mass of the car (in order to fully offset the feel-fall the gravity would otherwise provide).
- Stationary example – For a stationary vehicle, the approximate value reported in the (downward pointing) z-axis must be -9.8 m/s².
- Sign of angular rate – Direction of angular rotation follows the right-hand-rule around these axes.
- Angular rate example – As an example for a forward moving vehicle, a positive angular rate around the z-axis implies that the vehicle is turning to the right.

9.1.4.19. Odometry Measurement

Distance traveled (odometry) is to be reported in a continuously accumulating way from device power up. It may be an incremental distance starting at zero, another arbitrary point, or the absolute distance traveled by the vehicle. For a vehicle traveling in the reverse direction, the accumulated distance traveled may decrease, increase, or not represent the motion of the vehicle. The former is the preferred representation.

A slowly varying scale factor is expected as the major source of error in the odometry measurement. If calibration of the scale factor is performed externally, care must be taken to prevent an abrupt change in the scale factor during normal operation.

The nature of the odometry data, including the detection of certain “error events,” must be specified using the flags field bitmasks as follows:

- PDS_MASK_VEHICLE_ODOMETRY_REVERSE_MOVEMENT – Odometry data includes at least some data where the vehicle may have been moving in the reverse direction. If odometry data may be in reverse, this bit must be set. If odometry data is all in the forward direction, this bit must not be set.
- PDS_MASK_VEHICLE_ODOMETRY_AFFECTED_BY_ERRORS – Odometry data includes at least some data affected by a major error source affecting distance-traveled 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. Error due a slowly varying scale factor is expected by design and should not be indicated by this flag.
- PDS_MASK_VEHICLE_ODOMETRY_ABSOLUTE_MEASUREMENT – Odometry data is an absolute amount since the vehicle began service, and is the same vehicle that is regularly used with the device (so that the offset of this value, since the last time this measurement was used by the location engine can be safely used as an likely correct estimate of distance traveled since last use).

9.1.4.20. External Time Synchronization

The change_in_time_scales field is to be used in conjunction with an external timesync mechanism that aligns the vehicle sensor time scale with the device sensor time scale. In the event that the time offset between these two timescales changes, this field is applied for correct interpretation of the sensor time of the samples. If an external timesync mechanism does not exist, e.g., if only the vehicle time is provided, this field may be left at zero.

The dynamic range of the field is approximately -2100 sec to +2100 sec, where full-scale (minimum and maximum value) must be input if the value is greater than the range (i.e., saturate instead of rollover), and is interpreted as greater than or equal to this value of an offset change.

9.1.4.21. Emergency Positioning Session

The section outlines the call flow that occurs during an emergency positioning session. The service sends a QMI_PDS_EVENT_REPORT_IND message with an Emergency Position Session Event TLV that denotes the type of emergency position session. The client can use this message and TLV to distinguish an emergency positioning session from normal network-initiated positioning sessions. The QMI_PDS_EVENT_REPORT_IND message containing the Emergency Position Session Event TLV is sent to inform the client of the following:

- During an emergency call, the GNSS engine starts a position session even before the network-initiated location request corresponding to the emergency position session is received. This

is done in order to reduce the time spent in obtaining a position during an emergency. In this case, the service sets the value in the Emergency Position Session Event TLV to ePDS_EMERGENCY_POSITION_SESSION_EARLY_START.

- If the GNSS engine receives a network-initiated location request over the GSM control plane, the service sets the value in the Emergency Position Session Event TLV to ePDS_EMERGENCY_POSITION_SESSION_GSM_CP_NI_LR.
- If the GNSS engine receives a network-initiated location request over the WCDMA control plane, the service sets the value in the Emergency Position Session Event TLV to ePDS_EMERGENCY_POSITION_SESSION_WCDMA_CP_NI_LR.
- If the GNSS engine receives a network-initiated location request over the WCDMA Secure User Plane Location (SUPL), the service sets the value in the Emergency Position Session Event TLV to ePDS_EMERGENCY_POSITION_SESSION_WCDMA_SUPL_NI_LR.
- If the GNSS engine receives a network-initiated location request over the LTE control plane, the service sets the value in the Emergency Position Session Event TLV to ePDS_EMERGENCY_POSITION_SESSION_LTE_CP_NI_LR.
- If the GNSS engine receives a network-initiated location request over the LTE user plane, the service sets the value in the Emergency Position Session Event TLV to ePDS_EMERGENCY_POSITION_SESSION_LTE_UP_NI_LR.

Below figures illustrate the different call flows that may occur during an emergency position session on GSM and WCDMA networks.



Figure 16: Call Flow for GSM Control Plane Network-Initiated Location Request



Figure 17: Call Flow for WCDMA Control Plane Network-Initiated Location Request



Figure 18: Call Flow for WCDMA SUPL Network-Initiated Location Request

9.1.5. Service State Variables

9.1.5.1. Shared State Variables

No QMI_PDS state variables are shared across control points.

9.1.5.2. State Variables Per Control Point

Name	Description	Possible values	Default value
report_position_data_nmea	Whether new NMEA sentences are reported	• FALSE • TRUE	FALSE
report_position_data	Whether new parsed position data is reported	• FALSE • TRUE	FALSE
report_xtra_data_request	Whether XTRA data requests are reported	• FALSE • TRUE	FALSE
report_time_injection_request	Whether time injection requests are reported	• FALSE • TRUE	FALSE
report_satellite_info	Whether new satellite information is reported	• FALSE • TRUE	FALSE
report_network_initiated_prompts	Whether network-initiated prompts are reported	• FALSE • TRUE	FALSE
report_pds_comm_events	Whether PDS Communication events are reported	• FALSE • TRUE	FALSE
report_wifi_position_request	Whether Wi-Fi position requests are reported	• FALSE • TRUE	FALSE
report_accel_data_streaming_ready_status	Whether accelerometer data requests are reported	• FALSE • TRUE	FALSE
report_gyro_data_streaming_ready_status	Whether gyro data requests are reported	• FALSE • TRUE	FALSE
report_time_sync_request	Whether time synchronization requests are reported	• FALSE • TRUE	FALSE
report_position_reliability_indicator	Whether position reliability indications are reported	• FALSE • TRUE	FALSE
report_time_source_information	Whether time source information is reported	• FALSE • TRUE	FALSE
report_heading_unc_information	Whether heading uncertainty information is reported	• FALSE • TRUE	FALSE
report_nmea_debug_strings	Whether NMEA debug strings are reported	• FALSE • TRUE	FALSE
report_extended_xtra_data_request	Whether external XTRA data requests from the MSM GPS service are reported	• FALSE • TRUE	FALSE
report_service_reset_status	Whether the status of the engine during reset is reported	• FALSE • TRUE	FALSE
report_vehicle_accel_streaming_ready_status	Whether the vehicle accelerometer streaming ready status is reported	• FALSE • TRUE	FALSE

Name	Description	Possible values	Default value
report_vehicle_angular_rate_streaming_ready_status	Whether the vehicle angular streaming ready status is reported	• FALSE • TRUE	FALSE
report_vehicle_odometry_streaming_ready_status	Whether the vehicle odometry streaming ready status is reported	• FALSE • TRUE	FALSE
report_on_device_bias_estimated	Whether on-device data is reported	• FALSE • TRUE	FALSE
report_is801_cp_ni_reject_reason	Whether the IS-801 CP NI session reject reason is reported	• FALSE • TRUE	FALSE
report_supl_session_end_status	Whether the SUPL session end status is reported	• FALSE • TRUE	FALSE
report_emergency_position_session_event	Whether the emergency position session is reported	• FALSE • TRUE	FALSE

9.2. QMI_PDS Messages

Command	ID	Description
QMI_PDS_RESET	0x0000	Resets the PDS service state variables of the requesting control point
QMI_PDS_SET_EVENT QMI_PDS_SET_EVENT_REPORT	0x0001	Sets the PDS event reporting conditions for the requesting control point. A control point must enable TLVs it would like to receive from the MSM GPS service.
QMI_PDS_EVENT	0x0001	Indication the PDS event reporting indication
QMI_PDS_GET_STATE QMI_PDS_GET_GPS_SERVICE_STATE	0x0020	Queries the current state of the MSM GPS service.
QMI_PDS_STATE_IND	0x0020	Indication the current state of the MSM GPS service.
QMI_PDS_GET_AUTOTRACK QMI_PDS_GET_AUTO_TRACKING_STATE	0x0030	Queries the state of the QMI_PDS autotracking behavior for the control point.
QMI_PDS_SET_AUTOTRACK QMI_PDS_SET_AUTO_TRACKING_STATE	0x0031	Sets the state of the QMI_PDS autotracking behavior for the control point.

9.2.1. QMI_PDS_RESET

Resets the PDS service state variables of the requesting control point.

- **PDS message ID**

0x0000

- **Version introduced**

Major - 1, Minor - 0

9.2.1.1. Request - QMI_PDS_RESET_REQ

- **Message type**

Request

- **Sender**

Control point

- **Mandatory TLVs**

None

- **Optional TLVs**

None

9.2.1.2. Response - QMI_PDS_RESET_RESP

- **Message type**

Response

- **Sender**

Service

- **Mandatory TLVs**

The Result Code TLV (defined in Section 9.1.3.1) is always present in the response.

- **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_OPERATION	Operation is not allowed due to the current state of the location engine

9.2.1.3. Description of QMI_PDS_RESET REQ/RESP

This command resets the issuing control point's state kept by the service. Each shared state variable may change as a result according to its arbitration policy (see Section 9.1.4.3).

The resetting is equivalent to closing the service and reopening it again, although it is done as one operation, and hence, the client ID of the requesting control point does not change.

The control point's state variables will have changed to their default values before the response is issued.

9.2.2. QMI_PDS_SET_EVENT_REPORT

Sets the PDS event reporting conditions for the requesting control point. A control point must enable TLVs it would like to receive from the MSM GPS service.

- **PDS message ID**

0x0001

- **Version introduced**

Major - 1, Minor - 0

9.2.3. Request - QMI_PDS_SET_EVENT_REPORT_REQ

- **Message type**

Request

- **Sender**

Control point

- **Mandatory TLVs**

None

- **Optional TLVs**

In the TLV descriptions shown in this section, the term “report” means that the GPS service will send the corresponding TLVs (within the QMI_PDS_EVENT_REPORT_IND) to the control points that register for them by setting them to Enable.

At least one of the following optional TLVs must be included in this request.

Name	Version introduced	Version last modified
Position Data NMEA* **	Unknown	1.0
Extended Position NMEA* **	Unknown	1.10
Parsed Position Data* **	Unknown	1.0
External XTRA Data Request* **	Unknown	1.0
External Time Injection Request* **	Unknown	1.0
External WiFi Position Request* **	Unknown	1.0
Satellite Information* **	Unknown	1.0
Network-Initiated Prompt - VX*	Unknown	1.0
Network-Initiated Prompt - SUPL**	Unknown	1.0
Network-Initiated Prompt - UMTS CP**	Unknown	1.0
PDS Comm Events* **	Unknown	1.0
Accelerometer Data Streaming Ready Status	Unknown	1.7
Gyro Data Streaming Ready Status	Unknown	1.7
Time Sync Request	Unknown	1.7

Name	Version introduced	Version last modified
Position Reliability Indicator	Unknown	1.7
Sensor Data Usage Indicator	Unknown	1.7
Time Source Information	Unknown	1.8
Heading Uncertainty Information	Unknown	1.10
NMEA Debug Strings	Unknown	1.10
Extended External XTRA Data Request	Unknown	1.10
Location Engine Reset Status Indication	1.20	1.20
Vehicle Accelerometer Streaming Ready Status	1.28	1.28
Vehicle Angular Rate Streaming Ready Status	1.28	1.28
Vehicle Odometry Streaming Ready Status	1.28	1.28
On-Device Sensor Bias Estimate	1.29	1.29
IS801 CP NI Session Reject Reason	1.32	1.32
SUPL Session End Status	1.33	1.33
Emergency Position Session Event	1.33	1.33

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Position Data NMEA* **
Length	1			2	
Value	→	boolean	report_position_data_nmea	1	Report new position data in NMEA format. Values: <ul style="list-style-type: none">• 0x00 - Disable• 0x01 - Enable
Type	0x11			1	Extended Position NMEA* **
Length	1			2	
Value	→	boolean	report_extended_position_data_nmea	1	Report new extended position data in NMEA format. Values: <ul style="list-style-type: none">• 0x00 – Disable• 0x01 – Enable
Type	0x12			1	Parsed Position Data* **
Length	1			2	
Value	→	boolean	report_position_data	1	Report new position data in raw format. Values: <ul style="list-style-type: none">• 0x00 – Disable• 0x01 – Enable
Type	0x13			1	External XTRA Data Request* **
Length	1			2	
Value	→	boolean	report_xtra_data_request	1	Report external XTRA data requests from the MSM GPS service. Values: <ul style="list-style-type: none">• 0x00 – Disable• 0x01 – Enable
Type	0x14			1	External Time Injection Request* **
Length	1			2	
Value	→	boolean	report_time_injection_request	1	Report external time injection requests from the MSM GPS service. Values: <ul style="list-style-type: none">• 0x00 – Disable• 0x01 – Enable

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x15			1	External WiFi Position Request* **
Length	1			2	
Value	→	boolean	report_wifi_position_request	1	Report external Wi-Fi position requests from the MSM GPS service. Values: • 0x00 – Disable • 0x01 – Enable
Type	0x16			1	Satellite Information* **
Length	1			2	
Value	→	boolean	report_satellite_info	1	Report satellite information. Values: • 0x00 – Disable • 0x01 – Enable
Type	0x17			1	Network-Initiated Prompt - VX*
Length	1			2	
Value	→	boolean	report_network_initiated_prompts_vx	1	Report VX network-initiated request prompts. Values: • 0x00 – Disable • 0x01 – Enable
Type	0x18			1	Network-Initiated Prompt - SUPL**
Length	1			2	
Value	→	boolean	report_network_initiated_prompts_supl	1	Report SUPL network-initiated request prompts. Values: • 0x00 – Disable • 0x01 – Enable
Type	0x19			1	Network-Initiated Prompt - UMTS CP**
Length	1			2	
Value	→	boolean	report_network_initiated_prompts_umts_cp	1	Report UMTS CP network-initiated request prompts. Values: • 0x00 – Disable • 0x01 – Enable
Type	0x1A			1	PDS Comm Events* **
Length	1			2	
Value	→	boolean	report_pds_comm_events	1	Report PDS Comm events. Values: • 0x00 – Disable • 0x01 – Enable
Type	0x1B			1	Accelerometer Data Streaming Ready Status
Length	1			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	boolean	report_accel_data_streaming_ready_status	1	<p>Report accelerometer data streaming ready status. Values:</p> <ul style="list-style-type: none"> • 0x00 – Disable • 0x01 – Enable <p>When enabled, the GPS service may send the Accelerometer Data Streaming Ready Status TLV to a control point in QMI_PDS_EVENT_REPORT_IND. This TLV is sent to a control point when the GPS service is ready/not ready to receive accelerometer data from a control point.</p>
Type	0x1C			1	Gyro Data Streaming Ready Status
Length	1			2	
Value	→	boolean	report_gyro_data_streaming_ready_status	1	<p>Report gyro data streaming ready status. Values:</p> <ul style="list-style-type: none"> • 0x00 – Disable • 0x01 – Enable <p>When enabled, the GPS service may send the Gyro Data Streaming Ready Status TLV to a control point in QMI_PDS_EVENT_REPORT_IND. This TLV is sent to a control point when the GPS service is ready/not ready to receive gyro data from a control point.</p>
Type	0x1D			1	Time Sync Request
Length	1			2	
Value	→	boolean	report_time_sync_request	1	<p>Report time sync request. Values:</p> <ul style="list-style-type: none"> • 0x00 – Disable • 0x01 – Enable <p>When enabled, the GPS service may send the Time Sync Request TLV to a control point in QMI_PDS_EVENT_REPORT_IND. The GPS service sends this TLV to the control point when it needs to synchronize its clock with the sensor processor's clock.</p>
Type	0x1E			1	Position Reliability Indicator

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	boolean	report_position_reliability_indicator	1	<p>Report position reliability indicator.</p> <p>Values:</p> <ul style="list-style-type: none"> • 0x00 – Disable • 0x01 – Enable <p>When enabled, the GPS service sends the Position Reliability Indicator TLV with the Parsed Position Data and Position Session Status TLVs when the session status is set to In Progress or Success.</p>
Type	0x1F			1	Sensor Data Usage Indicator
Length	1			2	
Value	→	boolean	report_sensor_data_usage_indicator	1	<p>Report sensor data usage indicator.</p> <p>Values:</p> <ul style="list-style-type: none"> • 0x00 – Disable • 0x01 – Enable <p>When enabled, the GPS service sends the Sensor Data Usage Indicator TLV with the Parsed Position Data and Position Session Status TLVs when the session status is set to In Progress or Success.</p>
Type	0x20			1	Time Source Information
Length	1			2	
Value	→	boolean	report_time_source_information	1	<p>Report time source information. Values:</p> <ul style="list-style-type: none"> • 0x00 – Disable • 0x01 – Enable <p>When enabled, the GPS service sends the periodic Time Source Information TLV with the Satellite Information TLV when the session status is set to In Progress or Success.</p>
Type	0x21			1	Heading Uncertainty Information
Length	1			2	
Value	→	boolean	report_heading_unc_information	1	<p>Report heading uncertainty information.</p> <p>Values:</p> <ul style="list-style-type: none"> • 0x00 – Disable • 0x01 – Enable <p>When enabled, the GPS service sends the Heading Uncertainty Information TLV.</p>
Type	0x22			1	NMEA Debug Strings
Length	1			2	
Value	→	boolean	report_nmea_debug_strings	1	<p>Report NMEA debug strings. Values:</p> <ul style="list-style-type: none"> • 0x00 – Disable • 0x01 – Enable <p>When enabled, the GPS service sends the proprietary NMEA Debug Strings TLV.</p>
Type	0x23			1	Extended External XTRA Data Request
Length	1			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	boolean	report_extended_xtra_data_request	1	<p>Report external XTRA data requests from the MSM GPS service. Values:</p> <ul style="list-style-type: none"> • 0x00 – Disable • 0x01 – Enable <p>This TLV is to be used in lieu of TLV 0x13 of QMI_PDS_SET_EVENT_REPORT_REQ message when the XTRA file size is more than 64 K. If a client registers for this, TLV 0x26 in QMI_PDS_EVENT_REPORT_IND is generated.</p>
Type	0x24			1	Location Engine Reset Status Indication
Length	1			2	
Value	→	boolean	report_service_reset_status	1	<p>Report the status of the engine during reset. Values:</p> <ul style="list-style-type: none"> • 0x00 – Disable • 0x01 – Enable <p>If a client registers for this, TLV 0x27 in QMI_PDS_EVENT_REPORT_IND is generated.</p>
Type	0x25			1	Vehicle Accelerometer Streaming Ready Status
Length	1			2	
Value	→	boolean	report_vehicle_accel_streaming_ready_status	1	<p>Report the vehicle accelerometer streaming ready status. Values:</p> <ul style="list-style-type: none"> • 0x00 – Disable • 0x01 – Enable <p>When enabled, the location service may send the Vehicle Accelerometer Streaming Ready Status TLV to a control point in QMI_PDS_EVENT_REPORT_IND. This TLV is sent to a control point when the location service is ready/not ready to receive vehicle accelerometer data from a control point.</p>
Type	0x26			1	Vehicle Angular Rate Streaming Ready Status
Length	1			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	boolean	report_vehicle_angular_rate_streaming_ready_status	1	<p>Report the vehicle angular streaming ready status. Values:</p> <ul style="list-style-type: none"> • 0x00 – Disable • 0x01 – Enable <p>When enabled, the location service may send the Vehicle Angular Rate Streaming Ready Status TLV to a control point in QMI_PDS_EVENT_REPORT_IND. This TLV is sent to a control point when the location service is ready/not ready to receive vehicle angular rate data from a control point.</p>
Type	0x27			1	Vehicle Odometry Streaming Ready Status
Length	1			2	
Value	→	boolean	report_vehicle_odometry_streaming_ready_status	1	<p>Report the vehicle odometry streaming ready status. Values:</p> <ul style="list-style-type: none"> • 0x00 – Disable • 0x01 – Enable <p>When enabled, the location service may send the Vehicle Odometry Streaming Ready Status TLV to a control point in QMI_PDS_EVENT_REPORT_IND. This TLV is sent to a control point when the location service is ready/not ready to receive vehicle odometry data from a control point.</p>
Type	0x28			1	On-Device Sensor Bias Estimate
Length	1			2	
Value	→	boolean	report_on_device_bias_estimated	1	<p>Report on device data. Values:</p> <ul style="list-style-type: none"> • 0x00 – Disable • 0x01 – Enable <p>When enabled, the location service may send the On-Device Sensor Bias Estimate TLV to a control point in QMI_PDS_EVENT_REPORT_IND.</p>
Type	0x29			1	IS801 CP NI Session Reject Reason
Length	1			2	
Value	→	boolean	report_is801_cp_ni_reject_reason	1	<p>Report the IS-801 CP NI session reject reason. Values:</p> <ul style="list-style-type: none"> • 0x00 – Disable • 0x01 – Enable <p>When enabled, the location service reports when an IS-801 CP NI session is rejected and gives the reason for rejecting the NI session.</p>
Type	0x2A			1	SUPL Session End Status

Field	Field value	Field type	Parameter	Size (byte)	Description
Length	1			2	
Value	→	boolean	report_supl_session_end_status	1	Report SUPL session end status. Values: • 0x00 – Disable • 0x01 – Enable When enabled, the location service reports when a SUPL session ends and indicates whether the SUPL session was successful.
Type	0x2B			1	Emergency Position Session Event
Length	1			2	
Value	→	boolean	report_emergency_position_session_event	1	Report emergency position session events. Values: • 0x00 – Disable • 0x01 – Enable When enabled, the location service reports an emergency position session event that contains the type of event.

9.2.3.1. Response - QMI_PDS_SET_EVENT_REPORT_RESP

- **Message type**

Response

- **Sender**

Service

- **Mandatory TLVs**

The Result Code TLV (defined in Section 9.1.3.1) is always present in the response.

- **Optional TLVs**

None

9.2.3.2. Indication - QMI_PDS_EVENT_REPORT_IND

- **Message type**

Indication

- **Sender**

Service

- **Indication scope**

Unicast (per control point)

- **Mandatory TLVs**

None

- **Optional TLVs**

At least one of the following optional TLVs must be included in this request.

Name	Version introduced	Version last modified
Position Data NMEA***	Unknown	1.13
Extended Position Data NMEA***	Unknown	1.10
Position Session Status***	Unknown	1.0
Parsed Position Data***	Unknown	1.13
External XTRA Database Request***	Unknown	1.0
External Time Injection Request***	Unknown	1.0
External WiFi Position Request***	Unknown	1.0
Satellite Information***	Unknown	1.0
Network-Initiated Prompt - VX*	Unknown	1.0
Network-Initiated Prompt - SUPL**	Unknown	1.13
Network-Initiated Prompt - UMTS CP**	Unknown	1.13
PDS Comm Events***	1.13	1.35
Position Source***	Unknown	1.2
Accelerometer Streaming Ready Status	Unknown	1.7
Gyroscope Streaming Ready Status	Unknown	1.7
Time Sync Request	Unknown	1.7
Position Reliability Indicator	Unknown	1.7
Sensor Data Usage	Unknown	1.7
Time Source Information	Unknown	1.8
Encrypted Position Information	Unknown	1.10
Heading Uncertainty Information	Unknown	1.10
Proprietary NMEA Debug Sentences	Unknown	1.10
Extended External XTRA Database Request***	Unknown	1.10
Location Service Reset Status***	1.20	1.20
Extended Position Failure	1.25	1.25
SUPL Trigger Type	1.25	1.25
SUPL Emergency Notification	1.25	1.25
SUPL INIT Source	1.25	1.25
Vehicle Accelerometer Streaming Ready Status	1.28	1.28
Vehicle Angular Rate Streaming Ready Status	1.28	1.28
Vehicle Odometry Streaming Ready Status	1.28	1.28
On-Device Sensor Bias Estimate	1.29	1.29

Name	Version introduced	Version last modified
IS801 CP NI Session Reject Reason	1.32	1.32
SUPL Session End Status	1.33	1.33
Emergency Position Session Event	1.33	1.34

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Position Data NMEA* **
Length	Var			2	
Value	→	string	position_data_nmea	Var	String containing the position data in NMEA sentence format (maximum 200 bytes).
Type	0x11			1	Extended Position Data NMEA* **
Length	Var			2	
Value	→	enum8	position_data_mode	1	Determines the operation mode. Values: • -1 – Unknown (final fix determination is pending or cannot be determined)
		uint8	position_data_nmea_len	1	Number of sets of the following elements: • position_data_nmea
		string	position_data_nmea	Var	String containing the position data in NMEA sentence format (maximum 200 bytes).
Type	0x12			1	Position Session Status* **
Length	1			2	
Value	→	enum8	position_session_status	1	Status of current session. Values: • 0x00 – Success • 0x01 – In progress • 0x02 – General failure • 0x03 – Timeout • 0x04 – User ended the session • 0x05 – Bad parameter • 0x06 – Phone is offline • 0x07 – Engine is locked • 0x08 – E911 session in progress This TLV is always sent while a fix is being generated or if it halts due to an error. The Parsed Position Data TLV (0x13) is included if the session status is Success (0x00) or In Progress (0x01); otherwise this TLV is passed alone.
Type	0x13			1	Parsed Position Data* **
Length	105			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	mask32	valid_mask	4	<p>Bitmask of valid values in this TLV:</p> <ul style="list-style-type: none"> • 0x00000001 – Timestamp calendar (based on GPS time) • 0x00000002 – Timestamp UTC • 0x00000004 – Leap seconds • 0x00000008 – Time uncertainty • 0x00000010 – Latitude • 0x00000020 – Longitude • 0x00000040 – Ellipsoid altitude • 0x00000080 – Mean sea level altitude • 0x00000100 – Horizontal speed • 0x00000200 – Vertical speed • 0x00000400 – Heading • 0x00000800 – Horizontal uncertainty circular • 0x00001000 – Horizontal uncertainty ellipse semi-major • 0x00002000 – Horizontal uncertainty ellipse semi-minor • 0x00004000 – Horizontal uncertainty ellipse orient azimuth • 0x00008000 – Vertical uncertainty • 0x00010000 – Horizontal velocity uncertainty • 0x00020000 – Vertical velocity uncertainty • 0x00040000 – Horizontal confidence • 0x00080000 – Position Dilution of Precision (DOP) • 0x00100000 – Horizontal DOP • 0x00200000 – Vertical DOP • 0x00400000 – Operating mode used
	uint16		calendar_year	2	Current year.
	enum8		calendar_month	1	<p>Current month. Values:</p> <ul style="list-style-type: none"> • 0x01 – January • 0x02 – February • 0x03 – March • 0x04 – April • 0x05 – May • 0x06 – June • 0x07 – July • 0x08 – August • 0x09 – September • 0x0A – October • 0x0B – November • 0x0C – December

Field	Field value	Field type	Parameter	Size (byte)	Description
		float	altitude_wrt_sea_level	4	Height of the MS above the mean sea level (in meters). Note: Value is in single float format (refer to [S2]).
		float	horizontal_speed	4	Horizontal speed of the MS (in meters/sec). Note: Value is in single float format (refer to [S2]).
		float	vertical_speed	4	Vertical speed of the MS (in meters/sec). Note: Value is in single float format (refer to [S2]).
		float	heading	4	Direction of the MS, counting degrees from 0 in the north and increasing clockwise. <ul style="list-style-type: none">• Units: Decimal degrees• Range: 0 to 359.99 Note: Value is in single float format (refer to [S2]).
		float	horizontal_unc.circular	4	Circular horizontal uncertainty (in meters). Error circular confidence is specified by the horizontal_confidence field. Note: Value is in single float format (refer to [S2]).
		float	horizontal_unc.ellipse_semi_major	4	Size of the axis along the major angle specified for location uncertainty (in meters) for the error ellipse. Error ellipse confidence is specified by the horizontal_confidence field. Note: Value is in single float format (refer to [S2]).
		float	horizontal_unc.ellipse_semi_minor	4	Size of the axis along the minor angle specified for location uncertainty (in meters) for the error ellipse. Error ellipse confidence is specified by the horizontal_confidence field. Note: Value is in single float format (refer to [S2]).
		float	horizontal_unc.ellipse_orient_azimuth	4	Azimuth of the major for the location uncertainty ellipse. <ul style="list-style-type: none">• Units: Decimal degrees• Range: 1 to 180 Note: Value is in single float format (refer to [S2]).
		float	vertical_unc	4	Vertical uncertainty (in meters) provided at a 1-sigma (68 percent) confidence. Note: Value is in single float format (refer to [S2]).

Field	Field value	Field type	Parameter	Size (byte)	Description
		enum8	calendar_day_of_week	1	<p>Current day of the week. Values:</p> <ul style="list-style-type: none"> • 0x00 – Sunday • 0x01 – Monday • 0x02 – Tuesday • 0x03 – Wednesday • 0x04 – Thursday • 0x05 – Friday • 0x06 – Saturday
		uint8	calendar_day_of_month	1	Current day of the month (in days).
		uint8	calendar_hour	1	Current hour (in hours).
		uint8	calendar_minute	1	Current minute (in minutes) within an hour.
		uint8	calendar_second	1	Current second (in seconds) within a minute.
		uint16	calendar_millisecond	2	Current millisecond (in milliseconds) within a second.
		uint8	leap_seconds	1	Number of leap seconds (in seconds).
		uint64	timestamp_utc	8	Timestamp in ms; time is UTC time, starting Jan 1, 1970.
		uint32	time_unc	4	<p>One-sided time uncertainty for the calendar and UTC timestamp (in milliseconds). This uncertainty is provided at 99 percent confidence.</p>
		double	latitude	8	<p>Latitude position referenced to the WGS-84 reference ellipsoid, counting positive angles north of the equator and negative angles south of the equator.</p> <ul style="list-style-type: none"> • Units: Decimal degrees • Range: -90 to +90 <p>Note: Value is in double float format (refer to [S2]).</p>
		double	longitude	8	<p>Longitude position referenced to the WGS-84 reference ellipsoid, counting positive angles east of the Greenwich Meridian and negative angles west of the Greenwich Meridian.</p> <ul style="list-style-type: none"> • Units: Decimal degrees • Range: -180 to +180 <p>Note: Value is in double float format (refer to [S2]).</p>
		float	altitude_wrt_ellipsoid	4	<p>Height above the WGS-84 reference ellipsoid. Value conveys height (in meters) plus 500 m. Range: -500 to 15883.</p> <p>Note: Value is in single float format (refer to [S2]).</p>

Field	Field value	Field type	Parameter	Size (byte)	Description
		float	horizontal_vel_unc	4	<p>Horizontal velocity uncertainty (in meters/sec). Up to QMI_PDS version 1.5, this represents a 2-D circular uncertainty provided at 99 percent confidence. In QMI_PDS version 1.6 and above, this represents a 2-D circular uncertainty provided at 63 percent confidence.</p> <p>Note: Value is in single float format (refer to [S2]).</p>
		float	vertical_vel_unc	4	<p>Vertical velocity uncertainty (in meters/sec). Up until QMI_PDS version 1.5, this represents a 1-D uncertainty provided at 99 percent confidence. In QMI_PDS version 1.6 and above, this represents a 1-D uncertainty provided at 68 percent confidence.</p> <p>Note: Value is in single float format (refer to [S2]).</p>
		uint8	horizontal_confidence	1	Confidence value of the location horizontal uncertainty specified as 0 percent to 99 percent.
		float	position_dop	4	<p>Position dilution of position. Range: 1.0 (highest accuracy) to 50.0 (lowest accuracy).</p> $\text{PDOP} = \sqrt{\text{HDOP}^2 + \text{VDOP}^2}$ <p>Note: Value is in single float format (refer to [S2]).</p>
		float	horizontal_dop	4	<p>Horizontal dilution of position. Range: 1.0 (highest accuracy) to 50.0 (lowest accuracy).</p> <p>Note: Value is in single float format (refer to [S2]).</p>
		float	vertical_dop	4	<p>Vertical dilution of position. Range: 1.0 (highest accuracy) to 50.0 (lowest accuracy).</p> <p>Note: Value is in single float format (refer to [S2]).</p>
		enum8	position_op_mode	1	<p>Operating mode used to calculate a fix.</p> <p>Values:</p> <ul style="list-style-type: none"> • 0x00 – Standalone • 0x01 – MS-based • 0x02 – MS-assisted • -1 – Unknown (final fix determination is pending or cannot be determined)
Type	0x14			1	External XTRA Database Request* **.
Length	Var			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	uint16	max_file_size	2	Maximum size of the database file that can be injected.
		uint8	urls_len	1	Number of sets of the following elements: <ul style="list-style-type: none">• url_data_len• url_data
		uint8	url_data_len	1	Number of sets of the following elements: <ul style="list-style-type: none">• url_data
		string	url_data	Var	String containing the URL; maximum length is 128.
Type	0x15			1	External Time Injection Request* **
Length	Var			2	
Value	→	uint32	delay_threshold	4	Recommended to skip this service if one-way data exceeds this threshold (in milliseconds).
		uint8	urls_len	1	Number of sets of the following elements: <ul style="list-style-type: none">• url_data_len• url_data
		uint8	url_data_len	1	Number of sets of the following elements: <ul style="list-style-type: none">• url_data
		string	url_data	Var	String containing the URL; maximum length is 128.
Type	0x16			1	External WiFi Position Request* **
Length	3			2	
Value	→	enum8	wifi_request_type	1	Type of request for Wi-Fi positioning. Values: <ul style="list-style-type: none">• 0x00 – Start periodic fixes; high frequency• 0x01 – Start periodic fixes; keep warm• 0x02 – Stop periodic fixes• 0x04 – Suspend
		uint16	wifi_request_tbf	2	Time between fixes for requested fixes (in milliseconds). Note: Only valid when the request type is 0x00 or 0x01; otherwise this is set to 0 and is to be ignored.
Type	0x17			1	Satellite Information* **
Length	Var			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	mask32	valid_mask	4	<p>Mask of valid satellite information elements. Values:</p> <ul style="list-style-type: none"> • 0x00000001 – Iono validity • 0x00000002 – Satellite count • 0x00000004 – Satellite list <p>Each bit set causes the corresponding value to be specified in this optional TLV. If the bit is not set, the value is set to 0 and is to be ignored.</p> <p>All unlisted bits are reserved for future use and are set to 0.</p>
	uint8	iono_valid		1	Ionospheric corrections are valid.
	uint8	svns_len		1	<p>Number of sets of the following elements:</p> <ul style="list-style-type: none"> • svn_valid_mask • svn_system • svn_prn • svn_health_status • svn_process_status • svn_ephemeris_state • svn_almanac_state • svn_elevation • svn_azimuth • svn_cno
	mask32	svn_valid_mask		4	<p>Mask of valid satellite information elements. Values:</p> <ul style="list-style-type: none"> • 0x00000001 – System • 0x00000002 – Pseudorandom Noise (PRN) • 0x00000004 – Health status • 0x00000008 – Process status • 0x00000010 – Ephemeris state • 0x00000020 – Almanac state • 0x00000040 – Elevation • 0x00000080 – Azimuth • 0x00000100 – CN0 <p>Each bit set causes the corresponding value to be specified in this optional TLV. If the bit is not set, the value is set to 0 and is to be ignored.</p> <p>All unlisted bits are reserved for future use and are set to 0.</p>

Field	Field value	Field type	Parameter	Size (byte)	Description
		enum8	svn_system	1	System to which this Satellite Vehicle (SV) belongs. Values: • 0x01 – GPS • 0x02 – Galileo • 0x03 – Satellite-Based Augmentation System (SBAS) • 0x04 – Compass • 0x05 – Global Navigation Satellite System (GLONASS)
		uint8	svn_prn	1	Pseudorandom noise of the GPS SVs. SV ID of the GLONASS SVs.
		enum8	svn_health_status	1	SV health level. Values: • 0x00 – Unhealthy • 0x01 – Healthy
		enum8	svn_process_status	1	SV processing status. Values: • 0x01 – Idle • 0x02 – Search • 0x03 – Search verify • 0x04 – Bit edge • 0x05 – Track
		enum8	svn_ephemeris_state	1	SV ephemeris state. Values: • 0x00 – Unavailable • 0x01 – Available
		enum8	svn_almanac_state	1	SV almanac state. Values: • 0x00 – Unavailable • 0x01 – Available
		float	svn_elevation	4	SV elevation. • Units: Degrees • Range: -90 to 90 Note: Value is in single float format (refer to [S2]).
		uint16	svn_azimuth	2	SV azimuth. 360/2 ¹⁶ degrees per Least Significant Bit (LSB). • Units: Degrees • Range: 0 to 360
		uint16	svn_cno	2	SV signal-to-(noise spectrum density) ratio (in dB Hz) at the matched filter output; 0.1 dB per LSB.
Type	0x18			1	Network-Initiated Prompt - VX*
Length	Var			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	mask32	valid_mask	4	<p>Mask of valid Version x (VX) data.</p> <p>Values:</p> <ul style="list-style-type: none"> • 0x00000001 – VX privacy • 0x00000002 – VX Quality of Service (QoS) • 0x00000004 – VX count • 0x00000008 – VX interval • 0x00000010 – VX mode • 0x00000020 – VX requestor ID
		enum8	vx_privacy	1	<p>Privacy of the notification prompt.</p> <p>Values:</p> <ul style="list-style-type: none"> • 0x00 – No notify/no verify • 0x01 – Notify only • 0x02 – Notify/verify; allow no response • 0x03 – Notify/verify; require response • 0x04 – Privacy override
		uint8	vx_qos	1	Requested VX position quality of service.
		uint32	vx_count	4	Requested VX position count.
		uint32	vx_interval	4	Requested VX interval between fixes (in seconds).
		enum8	vx_mode	1	<p>Requested VX position operating mode.</p> <p>Values:</p> <ul style="list-style-type: none"> • 0x00 – MS-assisted only • 0x01 – MS-based only • 0x02 – MS-assisted preferred, MS-based allowed • 0x03 – MS-based preferred, MS-assisted allowed
		enum8	vx_requestor_id_dcs	1	<p>Data coding scheme of the requestor ID.</p> <p>Values:</p> <ul style="list-style-type: none"> • 0x00 – Octet • 0x01 – EXN protocol message • 0x02 – ASCII • 0x03 – IA5 • 0x04 – Unicode • 0x05 – Shift JIS • 0x06 – Korean • 0x07 – Latin Hebrew • 0x08 – Latin • 0x09 – GSM
		uint8	vx_requestor_id_len	1	Number of sets of the following elements:
		string	vx_requestor_id	Var	String containing the requestor ID in the specified data coding scheme.
Type	0x19			1	Network-Initiated Prompt - SUPL**

Field	Field value	Field type	Parameter	Size (byte)	Description
Length	Var			2	
Value	→	mask32	valid_mask	4	<p>Mask of valid Secure User Plane Location (SUPL) data. Values:</p> <ul style="list-style-type: none"> • 0x00000001 – Privacy • 0x00000002 – SUPL_INIT hash • 0x00000004 – Mode • 0x00000008 – SUPL Location Platform (SLP) session ID • 0x00000010 – SLP server address – IPv4 • 0x00000020 – SLP server address – IPv6 • 0x00000040 – SLP server address – URL • 0x00000080 – Data coding scheme • 0x00000100 – Requestor ID • 0x00000200 – Client name • 0x00000400 – Quality of Position (QoP) horizontal accuracy • 0x00000800 – QoP vertical accuracy • 0x00001000 – QoP maximum location age • 0x00002000 – QoP delay
		enum8	supl_privacy	1	<p>Privacy of the notification prompt. Values:</p> <ul style="list-style-type: none"> • 0x00 – No notify/no verify • 0x01 – Notify only • 0x02 – Notify/verify; allow no response • 0x03 – Notify/verify; require response • 0x04 – Privacy override
		uint64	supl_hash	8	Hash for SUPL_INIT.
		enum8	supl_mode	1	<p>Requested SUPL position operating mode. Values:</p> <ul style="list-style-type: none"> • 0x00 – MS-assisted • 0x01 – MS-based • 0x02 – MS-assisted preferred • 0x03 – MS-based preferred • 0x04 – Standalone • 0x05 – AFLT • 0x06 – ECID • 0x07 – EOTD • 0x08 – OTDOA • 0x09 – No position
		uint32	supl_slp_session_id	4	SUPL location platform session ID.
		uint32	supl_slp_server_ipv4_port	4	Port of the server (IPv4 format).
		uint32	supl_slp_server_ipv4_addr	4	IP address of the server (IPv4 format).
		uint32	supl_slp_server_ipv6_port	4	Port of the server (IPv6 format).
		opaque	supl_slp_server_ipv6_addr	16	IP address of the server (IPv6 format).

Field	Field value	Field type	Parameter	Size (byte)	Description
		uint8	supl_slp_server_url_len	1	Number of sets of the following elements: <ul style="list-style-type: none">• supl_slp_server_url
		string	supl_slp_server_url	Var	String containing the URL of the SLP server; maximum length is 255.
		enum8	supl_dcs	1	Data coding scheme for the SUPL request. Values: <ul style="list-style-type: none">• 0x00 – UTF8• 0x01 – UCS2• 0x02 – GSM default• -1 – Unknown
		enum8	supl_requestor_id_dcs	1	Data coding scheme for the SUPL requestor ID. Values: <ul style="list-style-type: none">• 0x00 – Logical name• 0x01 – Email address• 0x02 – MSISDN• 0x03 – URL• 0x04 – SIP URL• 0x05 – MIN• 0x06 – MDN• -1 – Unknown
		uint8	supl_requestor_id_len	1	Number of sets of the following elements: <ul style="list-style-type: none">• supl_requestor_id
		string	supl_requestor_id	Var	Requestor ID string.
		enum8	supl_client_name_dcs	1	Data coding scheme for the SUPL client name. Values: <ul style="list-style-type: none">• 0x00 – Logical name• 0x01 – Email address• 0x02 – MSISDN• 0x03 – URL• 0x04 – SIP URL• 0x05 – MIN• 0x06 – MDN• -1 – Unknown
		uint8	supl_client_name_len	1	Number of sets of the following elements: <ul style="list-style-type: none">• supl_client_name
		string	supl_client_name	Var	Client name string.
		uint8	supl_qop_horiz_acc	1	Quality of position horizontal accuracy.
		uint8	supl_qop_vert_acc	1	Quality of position vertical accuracy.
		uint8	supl_qop_max_loc_age	1	Quality of position maximum location age.
		uint8	supl_qop_delay	1	Quality of position delay.
Type	0x1A			1	Network-Initiated Prompt - UMTS CP**
Length	Var			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	mask32	valid_mask	4	<p>Mask of valid UMTS Control Point (CP) data. Values:</p> <ul style="list-style-type: none"> • 0x00000001 – Privacy • 0x00000002 – Invoke ID • 0x00000004 – Notification text • 0x00000008 – Client address • 0x00000010 – Location type • 0x00000020 – Requestor ID • 0x00000040 – Codeword string • 0x00000080 – Service type ID
		enum8	umts_cp_privacy	1	<p>Privacy of the notification prompt. Values:</p> <ul style="list-style-type: none"> • 0x00 – No notify/no verify • 0x01 – Notify only • 0x02 – Notify/verify; allow no response • 0x03 – Notify/verify; require response • 0x04 – Privacy override
		uint8	umts_cp_invoke_id	1	UMTS control point invoke ID.
		enum8	umts_cp_notification_text_dcs	1	Data coding scheme for the UMTS CP notification text. Values:
					<ul style="list-style-type: none"> • 0x00 – German • 0x01 – English • 0x02 – Italian • 0x03 – French • 0x04 – Spanish • 0x05 – Dutch • 0x06 – Swedish • 0x07 – Danish • 0x08 – Portuguese • 0x09 – Finnish • 0x0A – Norwegian • 0x0B – Greek • 0x0C – Turkish • 0x0D – Hungarian • 0x0E – Polish • -1 – Unknown
		uint8	umts_cp_notification_text_len	1	Number of sets of the following elements:
		string	umts_cp_notification_text	Var	Notification text.
		uint8	umts_cp_client_address_len	1	Number of sets of the following elements:
		string	umts_cp_client_address	Var	External client address; maximum length is 255.

Field	Field value	Field type	Parameter	Size (byte)	Description
		enum8	umts_cp_location_type	1	Type of location requested. Values: <ul style="list-style-type: none"> • 0x00 – Current • 0x01 – Current or last known • 0x02 – Initial
		enum8	umts_cp_requestor_id_dcs	1	Data coding scheme for the UMTS CP requestor ID. Values: <ul style="list-style-type: none"> • 0x00 – German • 0x01 – English • 0x02 – Italian • 0x03 – French • 0x04 – Spanish • 0x05 – Dutch • 0x06 – Swedish • 0x07 – Danish • 0x08 – Portuguese • 0x09 – Finnish • 0x0A – Norwegian • 0x0B – Greek • 0x0C – Turkish • 0x0D – Hungarian • 0x0E – Polish • -1 – Unknown
		uint8	umts_cp_requestor_id_len	1	Number of sets of the following elements: <ul style="list-style-type: none"> • umts_cp_requestor_id
		string	umts_cp_requestor_id	Var	Requestor ID string.
		enum8	umts_cp_codeword_dcs	1	Data coding scheme for the UMTS CP codeword. Values: <ul style="list-style-type: none"> • 0x00 – German • 0x01 – English • 0x02 – Italian • 0x03 – French • 0x04 – Spanish • 0x05 – Dutch • 0x06 – Swedish • 0x07 – Danish • 0x08 – Portuguese • 0x09 – Finnish • 0x0A – Norwegian • 0x0B – Greek • 0x0C – Turkish • 0x0D – Hungarian • 0x0E – Polish • -1 – Unknown
		uint8	umts_cp_codeword_len	1	Number of sets of the following elements: <ul style="list-style-type: none"> • umts_cp_codeword
		string	umts_cp_codeword	Var	Codeword string.

Field	Field value	Field type	Parameter	Size (byte)	Description
		uint8	umts_cp_service_id	1	UMTS CP service type ID.
Type	0x1B			1	PDS Comm Events* **
Length	2			2	
Value	→	enum8	pds_comm_event_type	1	<p>Type of event. Values:</p> <ul style="list-style-type: none"> • 0x00 – Begin – Connection to the network server was initiated • 0x01 – Connected – Connection to the network server was established • 0x02 – Failure – Connection to the network server failed • 0x03 – Done – Connection to the network server terminated because the communication was completed • 0x04 – Other failure – Connection to the network server was successful but another problem occurred

Field	Field value	Field type	Parameter	Size (byte)	Description
		enum8	pds_comm_event_protocol	1	<p>Protocol/data type communicating with the network server. Values:</p> <ul style="list-style-type: none"> • ePDS_COMM_EVENT_PROTOCOL_UMTS_USER_PLANE_SUPL (0) – UMTS user plane SUPL • ePDS_COMM_EVENT_PROTOCOL_1X_USER_PLANE (1) – 1X user plane • ePDS_COMM_EVENT_PROTOCOL_UMTS_CONTROL_PLANE_WCDMA (2) – UMTS control plane WCDMA • ePDS_COMM_EVENT_PROTOCOL_UMTS_CONTROL_PLANE_GSM (3) – UMTS control plane GSM • ePDS_COMM_EVENT_PROTOCOL_V1_OR_V2 (4) – V1/V2 • ePDS_COMM_EVENT_PROTOCOL_KDDI (5) – JCDMA • ePDS_COMM_EVENT_PROTOCOL_XTRA_DATA_DOWNLOAD (6) – XTRA data download • ePDS_COMM_EVENT_PROTOCOL_SNTP_TIME_DOWNLOAD (7) – SNTP time download • ePDS_COMM_EVENT_PROTOCOL_1X_CONTROL_PLANE (8) – 1X control plane • ePDS_COMM_EVENT_PROTOCOL_LTE_UP (9) – LTE user plane • ePDS_COMM_EVENT_PROTOCOL_LTE_CP (10) – LTE control plane • ePDS_COMM_EVENT_PROTOCOL_UNKNOWN (-1) – Unknown
Type	0x1C			1	Position Source* **
Length	4			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	mask32	pds_position_source	4	<p>This TLV is always sent with the Parsed Position Data TLV (0x13) when the latitude/longitude is marked as valid.</p> <p>Source of the position (bitmask). Values:</p> <ul style="list-style-type: none"> • 0x00000001 – GPS • 0x00000002 – Cell ID • 0x00000004 – GLONASS • 0x00000008 – Network • 0x00000010 – External position injection <p>Multiple sources can potentially be indicated in the bitmask.</p> <p>Future versions of the GPS service may use reserved bits to represent the new position sources. A control point must treat any bits that it considers to be reserved as “unknown” or “other” source.</p>
Type	0x1D			1	Accelerometer Streaming Ready Status
Length	1			2	
Value	→	boolean	accel_streaming_ready_status	1	<p>The GPS service uses this TLV to let a control point know when it is ready or not ready to receive accelerometer data input. Values:</p> <ul style="list-style-type: none"> • 0x00 – Not ready for streaming • 0x01 – Ready for streaming
Type	0x1E			1	Gyroscope Streaming Ready Status
Length	1			2	
Value	→	boolean	gyro_streaming_ready_status	1	<p>The GPS service uses this TLV to let a control point know when it is ready or not ready to receive gyroscope data input. Values:</p> <ul style="list-style-type: none"> • 0x00 – Not ready for streaming • 0x01 – Ready for streaming
Type	0x1F			1	Time Sync Request
Length	4			2	
Value	→	uint32	ref_counter	4	This TLV is sent to registered control points when the GPS service needs to synchronize GPS and control point/sensor processor times.
Type	0x20			1	Position Reliability Indicator
Length	2			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	enum8	reliability_indicator_horizontal	1	<p>Reliability of horizontal position.</p> <p>Values:</p> <ul style="list-style-type: none"> • 0x00 – Not set • 0x01 – Very low (e.g., use at your own risk) • 0x02 – Low (e.g., little or no cross-check is possible) • 0x03 – Medium (e.g., limited cross-check passed) • 0x04 – High (e.g., strong cross-check passed) <p>This optional TLV may be provided if a specific reliability of the user wants to inform the GNSS engine of the expected level of reliability of the horizontal portion of the injected position.</p>
			reliability_indicator_vertical	1	<p>Reliability of vertical position. Values:</p> <ul style="list-style-type: none"> • 0x00 – Not set • 0x01 – Very low (e.g., use at your own risk) • 0x02 – Low (e.g., little or no cross-check is possible) • 0x03 – Medium (e.g., limited cross-check passed) • 0x04 – High (e.g., strong cross-check passed) <p>This optional TLV may be provided if a specific reliability of the user wants to inform the GNSS engine of the expected level of reliability of the vertical portion of the injected position.</p>
Type	0x21			1	Sensor Data Usage
Length	4			2	
Value	→	mask16	sensor_data_usage_mask	2	<p>This TLV is always sent with the Parsed Position Data TLV (0x13) when the latitude/longitude is marked as valid.</p> <p>Sensor data usage mask. Values:</p> <ul style="list-style-type: none"> • 0x0001 – Accelerometer • 0x0002 – Gyro <p>A bit set to 1 bit indicates that data from the corresponding sensor was used to compute the content (position, velocity, heading, etc.) in this packet.</p> <p>Future versions of the GPS service may use reserved bits to represent new sensor types. A control point must ignore any bits that it considers to be reserved.</p>

Field	Field value	Field type	Parameter	Size (byte)	Description
		mask16	sensor_aiding_indicator_mask	2	<p>This TLV is always sent with the Parsed Position Data TLV (0x13) when the latitude/longitude is marked as valid.</p> <p>Sensor aiding indicator mask. Values:</p> <ul style="list-style-type: none"> • 0x0001 – Heading aided with sensor data • 0x0002 – Speed aided with sensor data • 0x0004 – Position aided with sensor data • 0x0008 – Velocity aided with sensor data <p>A bit set to 1 indicates that certain fields (position, velocity, heading, etc.) in the Parsed Position Data TLV were aided with sensor data.</p>
Type	0x22			1	Time Source Information
Length	32			2	
Value	→	boolean	valid_time	1	<p>Time estimate validity. Values:</p> <ul style="list-style-type: none"> • 0x00 – FALSE – Time estimate is invalid • 0x01 – TRUE – Time estimate is valid
		enum8	time_source	1	<p>Estimate of time source. Values:</p> <ul style="list-style-type: none"> • 0x00 – Invalid • 0x01 – Network time transfer • 0x02 – Network time tagging • 0x03 – External input • 0x04 – Time of week decode • 0x05 – Time of week confirmed • 0x06 – Time of week and week confirmed • 0x07 – Time alignment • 0x08 – Navigation solution • 0x09 – Solve for time <p>Other values are reserved for future use.</p>
		float	time_uncertainty_ms	4	<p>Time uncertainty in millisecond units. This uncertainty is provided at 99 percent confidence.</p> <p>Note: Value is in single float format (refer to [S2]).</p>
		uint16	gps_week	2	GPS week (full GPS week).
		uint32	gps_time_of_week_ms	4	GPS time of week in milliseconds.
		uint32	reserved_1	4	Reserved field.
		uint32	reserved_2	4	Reserved field.
		uint32	reserved_3	4	Reserved field.
		uint32	reserved_4	4	Reserved field.
		uint32	reserved_5	4	Reserved field.
Type	0x23			1	Encrypted Position Information

Field	Field value	Field type	Parameter	Size (byte)	Description
Length	Var			2	
Value	→	enum8	encryption_algo	1	Encryption algorithm used. Values: • -1 – None • 0 – SHA-1 • 1 – Maximum
		uint8	encryption_data_len	1	Number of sets of the following elements: • encryption_data
		uint8	encryption_data	Var	Encryption data.
Type	0x24			1	Heading Uncertainty Information
Length	16			2	
Value	→	float	heading_unc	4	Heading uncertainty in radians. Note: Value is in single format float (refer to [S2]). If inertial sensors are used (as indicated by the Sensor Data Usage TLV), this uncertainty corresponds to the sensors filtered heading. When sensors are not used, this is identical to the GNSS heading uncertainty. This uncertainty is provided at 68 percent confidence.
		float	gnss_heading_unc	4	GNSS-only heading uncertainty in radians. Note: Value is in single format float (refer to [S2]). This field denotes the heading uncertainty as determined from GNSS measurements for the current epoch. This uncertainty is provided at 68 percent confidence.
		uint32	reserved_1	4	Reserved field.
		uint32	reserved_2	4	Reserved field.
Type	0x25			1	Proprietary NMEA Debug Sentences
Length	Var			2	
Value	→	string	nmea_debug	Var	String containing proprietary NMEA debug sentences (maximum 200 bytes).
Type	0x26			1	Extended External XTRA Database Request* ** (used in lieu of the External XTRA Database Request TLV when the file size is greater than a 16-bit value)
Length	Var			2	
Value	→	uint32	max_file_size	4	Maximum size of the database file that can be injected. Range: $0 \leq \text{max_file_size} \leq ((2^{32}) - 1)$

Field	Field value	Field type	Parameter	Size (byte)	Description
		uint8	urls_len	1	Number of sets of the following elements: <ul style="list-style-type: none">• url_data_len• url_data
		uint8	url_data_len	1	Number of sets of the following elements: <ul style="list-style-type: none">• url_data
		string	url_data	Var	String containing the URL; maximum length is 128.
Type	0x27			1	Location Service Reset Status* ***
Length	4			2	
Value	→	enum	location_service_reset_state	4	Reset status indicator. Values: <ul style="list-style-type: none">• 0x00 – Reset of the location engine is in progress; during this state no requests are accepted and no indications are generated• 0x01 – Reset of the location engine has completed; after this indication, the clients can resume normal operation• 0x02 – Reset of the location engine could not be initiated• 0x03 – Reset of the location engine could not be initiated because an E911 call is in progress
Type	0x28			1	Extended Position Failure
Length	4			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	enum	position_session_failure_ext	4	<p>If position_session_status is General Failure, this TLV, if present, gives the reason for the failure. Values:</p> <ul style="list-style-type: none"> • ePDS_POSITION_SESSION_FAILURE_EXT_NO_SERVICE (0) – No service • ePDS_POSITION_SESSION_FAILURE_EXT_NO_CONNECTION (1) – No connection with the PDE • ePDS_POSITION_SESSION_FAILURE_EXT_CONNECTION_FAIL (2) – Connection failure with the PDE • ePDS_POSITION_SESSION_FAILURE_EXT_PDE_REJECT (3) – Reject from the PDE • ePDS_POSITION_SESSION_FAILURE_EXT_STALE_BS_INFO (4) – Base station information is stale • ePDS_POSITION_SESSION_FAILURE_EXT_NORMAL_ENDING (5) – Network indicated a normal ending of the session • ePDS_POSITION_SESSION_FAILURE_EXT_RESET_LOCATION_SERVICE (6) – Request for reset location service was received • ePDS_POSITION_SESSION_FAILURE_EXT_PPM_POST_FIX_TIMEOUT (7) – PPM calculation started at the request of the IS-801 PDE timed out • ePDS_POSITION_SESSION_FAILURE_EXT_PPM_PRE_FIX_TIMEOUT (8) – PPM calculation started at the beginning of the IS-801 session timed out • ePDS_POSITION_SESSION_FAILURE_EXT_IS801_SERVER_TIMEOUT (9) – PDE did not send a response to the UE within the IS-801 session timer • ePDS_POSITION_SESSION_FAILURE_EXT_IS801_PAYLOAD_ERROR (10) – Position Determination Data message received from the PDE has errors and the decoding failed

Field	Field value	Field type	Parameter	Size (byte)	Description
			position_session_failure_ext (cont.)		<ul style="list-style-type: none"> • ePDS_POSITION_SESSION_FAILURE_EXT_VX_AUTH_FAIL (11) – VX LCS agent authentication failure • ePDS_POSITION_SESSION_FAILURE_EXT_FIX_METHOD_FAILURE (12) – Preferred fix method failed
Type	0x29			1	SUPL Trigger Type
Length	4			2	
Value	→	enum	supl_trigger_type	4	<p>Specifies the type of session trigger requested in the SUPL POS INIT message (refer to [S3]). Values:</p> <ul style="list-style-type: none"> • 0xFFFFFFFF – TRIGGER_TYPE_SINGLE_SHOT – Single shot • 0x00000000 – TRIGGER_TYPE_PERIODIC – Periodic • 0x00000001 – TRIGGER_TYPE_AREA_EVENT – Area event <p>For more details, refer to OMA-TS-ULP-V2_0-20120417-A [S3].</p>
Type	0x2A			1	<p>SUPL Emergency Notification (This specifies that the corresponding NI notification is an emergency notification. Emergency notification can be given even without an ESLP address.)</p>
Length	Var			2	
Value	→	uint8	eslp_url_len	1	<p>Number of sets of the following elements:</p> <ul style="list-style-type: none"> • eslp_url
		string	eslp_url	Var	Emergency SUPL location platform URL. Maximum length: 255 bytes.
Type	0x2B			1	SUPL INIT Source
Length	4			2	
Value	→	enum	supl_init_source	4	<p>SUPL INIT source. Values:</p> <ul style="list-style-type: none"> • ePDS_SUPL_INIT_SOURCE_TYPE_SMS_1X (0) – SUPL INIT came over 1X SMS • ePDS_SUPL_INIT_SOURCE_TYPE_SMS_GSM (1) – SUPL INIT came over GSM SMS • ePDS_SUPL_INIT_SOURCE_TYPE_WAP_PUSH (2) – SUPL INIT came over WAP Push • ePDS_SUPL_INIT_SOURCE_TYPE_UDP (3) – SUPL INIT came over UDP

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x2C			1	Vehicle Accelerometer Streaming Ready Status
Length	1			2	
Value	→	boolean	vehicle_accel_streaming_ready_status	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 for streaming• 0x01 – Ready for streaming
Type	0x2D			1	Vehicle Angular Rate Streaming Ready Status
Length	1			2	
Value	→	boolean	vehicle-angular_rate_streaming_ready_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: <ul style="list-style-type: none">• 0x00 – Not ready for streaming• 0x01 – Ready for streaming
Type	0x2E			1	Vehicle Odometry Streaming Ready Status
Length	1			2	
Value	→	boolean	vehicle_odometry_streaming_ready_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: <ul style="list-style-type: none">• 0x00 – Not ready for streaming• 0x01 – Ready for streaming
Type	0x2F			1	On-Device Sensor Bias Estimate (contains on-device sensor data for both gyroscope and accelerometer)
Length	42			2	
Value	→	uint16	sequence_number	2	Monotonically increasing sequence number identifying the Sensor-Bias report to the client.
		mask	sensor_valid_mask (for gyroscope)	8	Indicates which of the axes have valid data. Valid bitmasks: <ul style="list-style-type: none">• PDS_MASK_X_AXIS (0x0000000000000001) – The x-axis is valid• PDS_MASK_Y_AXIS (0x0000000000000002) – The y-axis is valid• PDS_MASK_Z_AXIS (0x0000000000000004) – The z-axis is valid.

Field	Field value	Field type	Parameter	Size (byte)	Description
		float	x_axis_value (for gyroscope)	4	x-axis value. <ul style="list-style-type: none"> Type: Floating point Units gyroscope: (radians)/(seconds)
		float	y_axis_value (for gyroscope)	4	y-axis value. <ul style="list-style-type: none"> Type: Floating point Units gyroscope: (radians)/(seconds)
		float	z_axis_value (for gyroscope)	4	z-axis value. <ul style="list-style-type: none"> Type: Floating point Units gyroscope: (radians)/(seconds)
		mask	sensor_valid_mask (for accelerometer)	8	Indicates which of the axes have valid data. Valid bitmasks: <ul style="list-style-type: none"> PDS_MASK_X_AXIS (0x0000000000000001) – The x-axis is valid PDS_MASK_Y_AXIS (0x0000000000000002) – The y-axis is valid PDS_MASK_Z_AXIS (0x0000000000000004) – The z-axis is valid.
		float	x_axis_value (for accelerometer)	4	x-axis value. <ul style="list-style-type: none"> Type: Floating point Units accelerometer: (meters)/(seconds^2)
		float	y_axis_value (for accelerometer)	4	y-axis value. <ul style="list-style-type: none"> Type: Floating point Units accelerometer: (meters)/(seconds^2)
		float	z_axis_value (for accelerometer)	4	z-axis value. <ul style="list-style-type: none"> Type: Floating point Units accelerometer: (meters)/(seconds^2)
Type	0x30			1	IS801 CP NI Session Reject Reason
Length	4			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	enum	is801_cp_ni_reject_reason	4	The location service sends this TLV when an IS-801 CP NI session is rejected. The TLV contains the reason for rejecting the NI session. Values: <ul style="list-style-type: none">• ePDS_IS801_CP_NI_REJECT_REASON_DECODE_ERROR (0) – Decode error• ePDS_IS801_CP_NI_REJECT_REASON_NO_RESOURCES (1) – No resources could be allocated to service the session• ePDS_IS801_CP_NI_REJECT_REASON_ENGINE_LOCKED (2) – GNSS engine is locked• ePDS_IS801_CP_NI_REJECT_REASON_ENGINE_RESET_IN_PROGRESS (3) – GNSS engine reset is in progress• ePDS_IS801_CP_NI_REJECT_REASON_PRIORITIZED_MO_SESSION_IN_PROGRESS (4) – GNSS engine is busy servicing an MO session and the device is configured to prioritize MO sessions over NI sessions• ePDS_IS801_CP_NI_REJECT_REASON_INTERNAL_ERROR (5) – Internal error in the GNSS engine
Type	0x31			1	SUPL Session End Status
Length	4			2	
Value	→	enum	supl_session_end_status	4	Identifies the status of the SUPL session end. Values: <ul style="list-style-type: none">• ePDS_SUPL_SESSION_END_STATUS_SUCCESS (0) – Session ended with success• ePDS_SUPL_SESSION_END_STATUS_FAILURE (1) – Session ended with a failure
Type	0x32			1	Emergency Position Session Event
Length	4			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	enum	emergency_position_session_event	4	<p>The location service sends this TLV during an emergency positioning session. The TLV identifies the type of emergency session event. Values:</p> <ul style="list-style-type: none"> • ePDS_EMERGENCY_POSITION_SESSION_EARLY_START (0) – An emergency call has been placed and the GNSS engine has started a positioning session “early”, i.e., before a network-initiated location request corresponding to the emergency call has been received • ePDS_EMERGENCY_POSITION_SESSION_GSM_CP_NI_LR (1) – GNSS engine received a network-initiated control plane location request on a GSM network during an emergency call • ePDS_EMERGENCY_POSITION_SESSION_WCDMA_CP_NI_LR (2) – GNSS engine received a network-initiated control plane location request on a WCDMA network during an emergency call • ePDS_EMERGENCY_POSITION_SESSION_WCDMA_SUPL_NI_LR (3) – GNSS engine received a network-initiated SUPL location request on a WCDMA network during an emergency call • ePDS_EMERGENCY_POSITION_SESSION_LTE_CP_NI_LR (4) – GNSS engine received a network-initiated control plane location request on an LTE network during an emergency call • ePDS_EMERGENCY_POSITION_SESSION_LTE_UP_NI_LR (5) – GNSS engine received a network-initiated user plane location request on an LTE network during an emergency call

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_MISSING_ARG	One or more required TLVs were missing in the request
QMI_ERR_INVALID_OPERATION	Operation is not allowed due to the current state of the location engine

9.2.3.3. Description of QMI_PDS_SET_EVENT_REPORT

This command sets the PDS event reporting conditions for the requesting control point. A control point must enable the TLVs it would like to receive from the GPS service.

The control point's event reporting state variables will be modified according to the settings specified in the TLVs included in the request message. The service maintains a set of state variables for each control point.

See Section 9.1.5.2 for more details regarding control point state variables.

Events of interest are communicated to the registered PDS control point via the QMI_PDS_EVENT_REPORT_IND indication.

Description of QMI_PDS_EVENT_REPORT_IND

This unsolicited indication is sent to interested control points when the device state corresponding to any TLV listed in the indication section changes. Interested control points are those that previously registered for the corresponding state to be reported using the QMI_PDS_SET_EVENT_REPORT_REQ message.

This indication is generated when one or more of the following events occur:

- NMEA Sentence – A new NMEA sentence was generated by the MSM GPS service.
- Extended NMEA Sentence – A new NMEA sentence was generated by the MSM GPS service. This includes the operating mode that was used to generate the fix (which may be different from the mode that was requested)
- Parsed Position Session Status – Status of fix generation from the MSM GPS service; this TLV is always sent if Parsed Position Data reporting is enabled.
- Parsed Position Data – A fix was generated by the MSM GPS service and delivered as discrete data elements. This TLV is only present if Parsed Position Data reporting is enabled and the parsed position_session_status is Success or In Progress.
- External XTRA Database Request – Request from the MSM GPS service to download XTRA data from a specified server; the XTRA data must be injected to the MSM GPS service once obtained from the server.
- External Time Injection Request – Request from the MSM GPS service to download time data from a

specified server; the time data must be injected to the MSM GPS service once obtained from the server.

- External WiFi Position Request – Request from the MSM GPS service to control fixes is determined externally via Wi-Fi; when requested, the Wi-Fi position data must be injected to the MSM GPS service.
- Satellite Info – Information regarding the current satellites is used to determine fixes by the MSM GPS service. This is only sent to control points at a 1 Hz frequency when the MSM GPS service is active .
- Network-Initiated Prompt VX – VX network request to perform a fix; the prompt is sent to the QMI control point registered for this event. The control point is to decide how to respond to this event (present a prompt to the user and send back the user response, send a response without presenting it to the user, etc.).
- Network-Initiated Prompt SUPL – SUPL network request to perform a fix; the prompt is sent to the QMI control point registered for this event. The control point is to decide how to respond to this event (present a prompt to the user and send back the user response, send a response without presenting it to the user, etc.).
- Network-Initiated Prompt UMTS CP – UMTS CP network request to perform a fix; the prompt is sent to the QMI control point registered for this event. The control point is to decide how to respond to this event (present a prompt to the user and send back the user response, send a response without presenting it to the user, etc.).
- PDS Comm Events – Events indicating the status of the connection between the device and network server.
- Accelerometer Data Streaming Ready Status – This TLV is sent to a registered control point when the GPS service is ready/not ready to accept accelerometer data injection. Following enablement, the registered control point commences streaming data using the QMI_PDS_INJECT_SENSOR_DATA command. This information is used by the GPS service to enhance the operation of the navigation filter. If the control point sends data when the GPS service is in a Not Ready status, the GPS service ignores the data and returns a QMI_ERR_INVALID_OPERATION error (assuming the message is received and decoded successfully, which may result in other errors such as QMI_ERR_INVALID_ARG).
- Gyroscope Data Streaming Ready Status – This TLV is sent to a registered control point when the GPS service is ready/not ready to accept gyroscope data injection. Following enablement, the registered control point commences streaming data using the QMI_PDS_INJECT_SENSOR_DATA command. This information is used by the GPS service to enhance the operation of the navigation filter. If the control point sends data when the GPS service is in a Not Ready status, the GPS service ignores the data and returns a QMI_ERR_INVALID_OPERATION error (assuming the message is received and decoded successfully, which may result in other errors such as QMI_ERR_INVALID_ARG).
- Time Sync Request – This TLV may be sent to a registered control point on a periodic basis. Upon receipt, the control point must immediately read its sensor processor time (not phone time or absolute GPS time) and send it back along with the reference_count that it received in the Time Sync Request TLV, along with a sensor processor time that represents the time when the QMI_PDS_INJECT_TIME_SYNC_DATA command was sent to the GPS processor. If there is less than 1 ms latency between the time that the control point decodes the Time Sync Request TLV and

when it outputs a QMI_PDS_INJECT_TIME_SYNC_DATA command, the two sensor processor time values (Rx and Tx) may be identical. This is acceptable, but the GPS processor expects both values to be present in the QMI_PDS_INJECT_TIME_SYNC_DATA command. The GPS processor uses the data inside the QMI_PDS_INJECT_TIME_SYNC_DATA command to correlate the separate processor times. The control point must not alter the reference_count before sending it back to the

- Position Reliability Indicator – The reliability indicator is used to show the probability of a position outlier, which is defined as the truth position being grossly inconsistent with the Gaussian-model uncertainty parameters, e.g., not within 10 standard deviations. A higher reliability metric means a lower probability of having a position outlier. Basically, this indicator tells to what extent the position can be trusted (four levels of the reliability metric are defined in the TLV). Note that this is an optional input/output that the control point may choose to use for certain advanced functionality. Pre-existing position inputs and outputs will continue to work as they do currently. Use or nonuse of this TLV does not affect the underlying GNSS system performance.

The benefit of this indicator is that, together with other information in the parsed position data (such as position uncertainty), it provides additional information for a user to make better use of a specified position. For example, a very low reliability output may be used simply for preloading likely needed-soon map information tiles, but not for an end-user display. Additionally, if external positioning sources are available, the reliability of the GNSS solution (cross-check with an injected position, if available) may be used for other system-level comparisons.

The definition of cross-check is using redundant information to verify a position fix. This information includes GNSS measurements, externally provided positions, and any other input that can help the positioning process.

- Sensor Data Usage Indicator – This indicator is used to show which sensors, if any, were used to help determine various components of a position. Time Source Information – Information regarding the time estimate used by the MSM GPS service; this is only sent to control points periodically (1 Hz rate approximately) when the MSM GPS service is active .
- Secure Position Reports – When a secure client receives an encrypted position report, the encrypted report contains this TLV that has some of the encryption information.
- Heading uncertainty information – This TLV contains uncertainties associated with a computed heading. When sensors are not used in computing a heading, the heading_unc and gnss_heading_unc fields contain the heading uncertainty as computed from GNSS measurements.
- Location Service Reset Status – This TLV notifies the control point when a hard reset of the engine has commenced or completed. No new messages are accepted while the engine is performing a hard reset.
- Extended Position Failure – Extended status information of a fix from the MSM GPS service; this TLV is sent if an extended reason for failure is present and is also sent when position_session_status is General Failure.
- SUPL Trigger Type – Notifies the trigger type used in SUPL.
- SUPL Emergency Notification – Specifies that the SUPL INIT has the emergency flag set; the emergency SLP address is optional in the notification.
- SUPL INIT Source – Identifies the source of SUPL INIT.
- Vehicle Accelerometer Streaming Ready Status – This TLV is sent to a registered control point when the location service is ready/not ready to accept on-vehicle accelerometer data injection. Following enablement, the registered control point commences streaming data using the

QMI_PDS_INJECT_VEHICLE_SENSOR_DATA command. If the control point sends data when the location service is in a Not Ready status, the location service ignores the data and returns a QMI_ERR_INVALID_OPERATION error (assuming the message is received and decoded).

- Vehicle Angular Rate Streaming Ready Status – This TLV is sent to a registered control point when the location service is ready/not ready to accept on-vehicle angular rate data injection. Following enablement, the registered control point commences streaming data using the QMI_PDS_INJECT_VEHICLE_SENSOR_DATA command. If the control point sends data when the location service is in a Not Ready status, the location service ignores the data and returns a QMI_ERR_INVALID_OPERATION error (assuming the message is received and decoded successfully, which may result in other errors such as QMI_ERR_INVALID_ARG).
- Vehicle Odometry Streaming Ready Status – This TLV is sent to a registered control point when the location service is ready/not ready to accept on-vehicle odometry data injection. Following enablement, the registered control point commences streaming data using the QMI_PDS_INJECT_VEHICLE_SENSOR_DATA command. If the control point sends data when the location service is in a Not Ready status, the location service ignores the data and returns a QMI_ERR_INVALID_OPERATION error (assuming the message is received and decoded successfully, which may result in other errors such as QMI_ERR_INVALID_ARG).
- On-Device Sensor Bias Estimate – This TLV is sent to a registered control point and contains gyroscope bias and accelerometer bias for on-device sensors.
- IS801 CP NI Session Reject Reason – This TLV is sent to a registered control point when an IS-801 CP NI session is rejected. The TLV contains the reason for rejecting the NI session.
- SUPL Session End Status – This TLV is sent to a registered control point to inform when a SUPL session ends. The TLV denotes if the SUPL session ended with success or ended due to a failure.
- Emergency Position Session Event – This TLV is sent to a registered control point during an emergency positioning session. The TLV contains the type of emergency position session event. A control point can use this TLV to distinguish the network-initiated requests that occur during an emergency call from those that occur during normal operation. The call flow during an emergency positioning session is described in Section 9.1.4.14.

9.2.4. QMI_PDS_GET_AUTO_TRACKING_STATE

Queries the state of the QMI_PDS autotracking behavior for the control point.

- **PDS message ID**

0x0030

- **Version introduced**

Major - 1, Minor - 0

9.2.4.1. Request - QMI_PDS_GET_AUTO_TRACKING_STATE_REQ

- **Message type**

Request

- **Sender**

Control point

- **Mandatory TLVs**

None

- **Optional TLVs**

None

9.2.4.2. Response - QMI_PDS_GET_AUTO_TRACKING_STATE_RESP

- **Message type**

Response

- **Sender**

Service

- **Mandatory TLVs**

The Result Code TLV (defined in Section 9.1.3.1) is always present in the response. The following mandatory TLV is present if the result code is QMI_RESULT_SUCCESS.

Name	Version introduced	Version last modified
Autotracking State	Unknown	1.0

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Autotracking State
Length	1			2	
Value	→	boolean	auto_tracking_state	1	Current autotracking state for this control point. Values: <ul style="list-style-type: none">• 0x00 – Disable• 0x01 – Enable

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_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

9.2.4.3. Description of QMI_PDS_GET_AUTO_TRACKING_STATE REQ/RESP

This command retrieves the state of the QMI_PDS autotracking for this control point.

9.2.5. QMI_PDS_SET_AUTO_TRACKING_STATE

Sets the state of the QMI_PDS autotracking behavior for the control point.

- PDS message ID

0x0031

- Version introduced

Major - 1, Minor - 0

9.2.5.1. Request - QMI_PDS_SET_AUTO_TRACKING_STATE_REQ

- Message type

Request

- Sender

Control point

- Mandatory TLVs

Name	Version introduced	Version last modified
Autotracking State	Unknown	1.0

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Autotracking State
Length	1			2	
Value	→	boolean	auto_tracking_state	1	Current autotracking state for this control point. Values: • 0x00 – Disable • 0x01 – Enable

9.2.5.2. Response - QMI_PDS_SET_AUTO_TRACKING_STATE_RESP

- Message type

Response

- Sender

Service

- Mandatory TLVs

The Result Code TLV (defined in Section 9.1.3.1) is always present in the response.

- 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_MISSING_ARG	One or more required TLVs were missing in the request
QMI_ERR_SESSION_INACTIVE	MSM GPS session is inactive
QMI_ERR_OP_DEVICE_UNSUPPORTED	Operation is not supported by the MSM GPS service
QMI_ERR_NO_EFFECT	Request had no effect on the current state
QMI_ERR_INVALID_OPERATION	Operation is not allowed due to the current state of the location engine

9.2.5.3. Description of QMI_PDS_SET_AUTO_TRACKING_STATE REQ/RESP

This command sets the state of autotracking fix generation for this QMI_PDS control point. Enabling autotracking results in a tracking session being started using the default session configuration as defined in Section 9.1.4.3. Once started, autotracking continues to generate fixes indefinitely until requested to be disabled.

Requests to disable autotracking can only be made if the QMI_PDS control point has first successfully enabled it; a QMI_ERR_SESSION_INACTIVE error is elicited if the current state is disabled. Disable requests can return success without the autotracking session being stopped if all conditions for it to be disabled are not met; these conditions are defined in Section 9.1.4.3.

10 Voice Service (QMI_VOICE)

This specification documents Major Version 2 of the Qualcomm Messaging Interface (QMI) for Voice Service (QMI_VOICE).

QMI_VOICE provides applications running on a host PC with commands related to voice service:

- Call origination
- Call end
- Call answer
- Flash
- Dual-Tone Multifrequency (DTMF)
- Supplementary services

It is expected that user-level applications, for example, connection managers or device drivers on the TE.

10.1. Theory of Operation

10.1.1. Generalized QMI Service Compliance

The QMI_VOICE 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 80-VB816-1. Extensions to the generalized QMI service theory of operation are noted in subsequent sections of this chapter.

10.1.2. VOICE Service Type

VOICE is assigned QMI service type 0x09.

10.1.3. Message Definition Template

10.1.3.1. Response Message Result TLV

This Type-Length-Value (TLV) is present in all Response messages defined in this document. It is not present in the Indication messages.

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

10.1.4. QMI_VOICE Fundamental Concepts

QMI_VOICE provides VOICE service to its control points. These services include interfaces to control voice call origination, tear down, answer, send Flash, DTMF, and Supplementary Service requests to the network, and to receive indications to report the call state, DTMF events, and other asynchronous indications from the network to convey caller ID, display, signal information and supplementary service notifications, and so on.

A dial string must always be provided to originate a voice call. A unique call ID is assigned to the call by the service. This call ID must be used as a key to identify the call to perform operations such as Answer, End, and so on. Any asynchronous indications associated with a call are sent with its corresponding call ID parameter.

Certain QMI_VOICE indications might be of interest to some QMI control points only. A mechanism that lets the control point register/deregister for certain indications is provided in which these registration settings for a control point are stored in the service state variables of the control point.

10.1.4.1. Dual SIM

The Dual SIM feature requires explicit support of the High Level Operating System (HLOS). One possible implementation is for the HLOS to create two instances of the modem interface, one for each subscription. In this design there could be two instances of the QMI_VOICE client, with one instance bound to the primary subscription and the other instance bound to the secondary subscription.

10.1.5. Service State Variables

10.1.5.1. Shared State Variables

No QMI_VOICE state variables are shared across control points.

10.1.5.2. State Variables Per Control Point

Name	Description	Possible values	Default value
reg_dtmf_events	Whether DTMF events are reported to a control point	• FALSE • TRUE	FALSE
reg_voice_privacy_events	Whether Voice Privacy events are reported to a control point	• FALSE • TRUE	FALSE
supps_notification_events	Whether Supplementary Service Notification events are reported to a control point	• FALSE • TRUE	FALSE
call_events	Whether Call Notification events are reported to a control point	• FALSE • TRUE	TRUE
handover_events	Whether Handover events are reported to a control point	• FALSE • TRUE	FALSE
speech_events	Whether Speech Codec events are reported to a control point	• FALSE • TRUE	FALSE

Name	Description	Possible values	Default value
ussd_notification_events	Whether USSD Notification events are reported to a control point	• FALSE • TRUE	TRUE
modification_events	Whether Modification events are reported to a control point	• FALSE • TRUE	TRUE
uus_events	Whether UUS events are reported to a control point	• FALSE • TRUE	TRUE
aoc_events	Whether AOC events are reported to a control point	• FALSE • TRUE	FALSE
conference_events	Whether Conference events are reported to a control point	• FALSE • TRUE	FALSE
ext_burst_intl_events	Whether Extended Burst Type International Information events are reported to a control point	• FALSE • TRUE	FALSE
page_miss_events	Whether MT Page Miss Information events are reported to a control point	• FALSE • TRUE	FALSE
cc_result_events	Whether Call Control Result Information events are reported to a control point	• FALSE • TRUE	FALSE
conf_participants_events	Whether Conference Participants events are reported to a control point	• FALSE • TRUE	FALSE
tty_info_events	Whether TTY Info events are reported to a control point	• FALSE • TRUE	FALSE
orig_fail_events	Whether E911 Call Origination Failure events are reported to a control point	• FALSE • TRUE	FALSE
vs_status_events	Whether Videoshare Status events are reported to a control point	• FALSE • TRUE	FALSE
audio_rat_change_events	Whether Audio Rat Change events are reported to a control point	• FALSE • TRUE	FALSE
additional_call_info_events	Whether Additional Call Information events are reported to a control point	• FALSE • TRUE	FALSE

10.2. 3 QMI_VOICE Messages

Command	ID	Description
QMI_VOICE_CANCEL_USSD	0x003C	Aborts an ongoing USSD operation (applicable only for 3GPP).
QMI_VOICE_USSD_RELEASE_IND	0x003D	Notifies clients that the USSD session is terminated by the network (applicable only for 3GPP).
QMI_VOICE_USSD_IND	0x003E	Notifies clients about any USSD requests or notifications from the network (applicable only for 3GPP).

QMI_VOICE_SUPS_IND	0x0043	Notifies clients about the USSDresponses received from the QMI_VOICE_ORIG_USSD_NO_WAIT_REQrequest (applicable only for 3GPP).
QMI_VOICE_ORIG_USSD_NO_WAIT_I ND	indicatio n	QMI_VOICE_ORIG_USSD_NO_WAIT_REQrequest (applicable only for 3GPP).

10.2.1. QMI_VOICE_CANCEL_USSD

Aborts an ongoing USSD operation (applicable only for 3GPP).

- **VOICE message ID**

0x003C

- **Version introduced**

Major - 2, Minor - 0

10.2.1.1. Request - QMI_VOICE_CANCEL_USSD_REQ

- **Message type**

Request

- **Sender**

Control point

- **Mandatory TLVs**

None

- **Optional TLVs**

None

10.2.1.2. Response - QMI_VOICE_CANCEL_USSD_RESP

- **Message type**

Response

- **Sender**

Service

- **Mandatory TLVs**

The Result Code TLV (defined in Section 10.1.3.1) is always present in the response.

- **Optional TLVs**

None

Error codes

<code>QMI_ERR_NONE</code>	No error in the request
<code>QMI_ERR_INTERNAL</code>	Unexpected error occurred during processing
<code>QMI_ERR_MALFORMED_MSG</code>	Message was not formulated correctly by the control point or the message was corrupted during transmission
<code>QMI_ERR_NO_MEMORY</code>	Device could not allocate memory to formulate a response
<code>QMI_ERR_NO_RADIO</code>	Radio is not available
<code>QMI_ERR_NOT_SUPPORTED</code>	Request is currently not supported
<code>QMI_ERR_INCOMPATIBLE_STATE</code>	Operation is not supported in the current state

10.2.1.3. Description of QMI_VOICE_CANCEL_USSD REQ/RESP

Only one USSD operation is possible at a time and that will be aborted.

Refer to 3GPP TS 22.090 for more details regarding USSD.

This command is applicable only in 3GPP devices.

10.3. QMI_VOICE_USSD_RELEASE_IND

Notifies clients that the USSD session is terminated by the network (applicable only for 3GPP).

- **VOICE message ID**

0x003D

- **Version introduced**

Major - 2, Minor - 0

10.3.1. Indication - QMI_VOICE_USSD_RELEASE_IND

- **Message type**

Indication

- **Sender**

Service

- **Scope**

Broadcast

- **Mandatory TLVs**

None

- **Optional TLVs**

None

10.3.1.1. Description of QMI_VOICE_USSD_RELEASE_IND

This indication is sent for user-initiated, and might or might not be sent for network-initiated, USSD requests upon termination (normal/abort) of the USSD requests by the network. Because there can be only one USSD operation at a time, this indication notifies that the existing USSD operation has been terminated. For more details, refer to 3GPP TS 23.090. This indication is applicable only in 3GPP devices.

10.4. QMI_VOICE_USSD_IND

Notifies clients about any USSD requests or notifications from the network (applicable only for 3GPP).

- **VOICE message ID**

0x003E

- **Version introduced**

Major - 2, Minor - 0

10.4.1.1. Description of QMI_VOICE_USSD_IND

If the notification_type is 0x02, it means the network expects the user to respond. The user response can be sent via the QMI_VOICE_ANSWER_USSD command.

USS data (if any) sent by the network is relayed to the control point through the optional USS Data from Network TLV.

For more details, refer to 3GPP TS 23.090.

The optional USS Data from Network in UTF-16 Encoding TLV is sent whenever the USS Data from Network TLV is sent.

This indication is applicable only in 3GPP devices

10.4.1.2. Indication - QMI_VOICE_USSD_IND

- Message type

Indication

- Sender

Service

- Scope

Unicast (per control point)

- Mandatory TLVs

Name	Version introduced	Version last modified
Notification Type	Unknown	2.0

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Notification Type
Length	1			2	
Value	→	enum8	notification_type	1	Notification type. Values: • 0x01 – FURTHER_USER_ACTION_NOT_REQUIRED – No further action is required • 0x02 – FURTHER_USER_ACTION_REQUIRED – Further action is required

Optional TLVs

Name	Version introduced	Version last modified
USS Data from Network	Unknown	2.0
USS Data from Network in UTF-16 Encoding	2.13	2.13

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	USS Data from Network
Length	Var			2	
Value	→	enum8	uss_dcs	1	Unstructured supplementary service data coding scheme. Values: • 0x01 – USS_DCS_ASCII – ASCII coding scheme • 0x02 – USS_DCS_8BIT – 8-bit coding scheme per 3GPP TS 23.038 • 0x03 – USS_DCS_UCS2 – UCS2
			uss_len	1	Number of sets of the following elements: • uss_data
			uss_data	Var	USS data per the coding scheme.
Type	0x11			1	USS Data from Network in UTF-16 Encoding
Length	Var			2	
Value	→	uint8	uss_info_utf16_len	1	Number of sets of the following elements: • uss_info_utf16
		uint16	uss_info_utf16	Var	Unstructured supplementary service information in UTF-16 encoding.

10.4.2. QMI_VOICE_ORIG_USSD_NO_WAIT_IND

Notifies clients about the USSD responses received from the QMI_VOICE_ORIG_USSD_NO_WAIT_REQ request (applicable only for 3GPP).

- **VOICE message ID**

0x0043

- **Version introduced**

Major - 2, Minor - 3

10.4.2.1. Indication - QMI_VOICE_ORIG_USSD_NO_WAIT_IND

- **Message type**

Indication

- **Sender**

Service

- **Scope**

Unicast (per control point)

- **Mandatory TLVs**

None

- **Optional TLVs**

Name	Version introduced	Version last modified
Error	Unknown	2.3
Failure Cause	2.3	2.27
USS Data from Network	Unknown	2.3
Alpha Identifier	Unknown	2.3
USS Data from Network in UTF-16 Encoding	2.13	2.13

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Error
Length	2			2	
Value	→	enum16	error	2	Type of error (if any).
Type	0x11			1	Failure Cause
Length	2			2	
Value	→	enum16	failure_cause	2	Supplementary services failure cause; see Table A-3 for more information.

10.4.2.2. Description of QMI_VOICE_ORIG_USSD_NO_WAIT_IND

This indication is received as a response for the QMI_VOICE_ORIG_USSD_NO_WAIT_REQ request. The failure_cause is present if a QMI_ERR_SUPS_FAILURE_CAUSE error is returned.

The optional Alpha Identifier TLV is used to pass the alpha (if any) given by the SIM/R-UIM after call control. For more details, refer to 3GPP TS 11.14 Section 9.1.3.

Refer to 3GPP TS 22.090 and 3GPP TS 23.090 for more details on USSD.

The optional USS Data from Network in UTF-16 Encoding TLV is sent whenever the optional USS Data from Network TLV is sent.

This indication is applicable only in 3GPP devices.

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x12			1	USS Data from Network
Length	Var			2	
Value	→	enum8	uss_dcs	1	Unstructured supplementary service data coding scheme. Values: <ul style="list-style-type: none">• 0x01 – USS_DCS_ASCII – ASCII coding scheme• 0x02 – USS_DCS_8BIT – 8-bit coding scheme per 3GPP TS 23.038• 0x03 – USS_DCS_UCS2 – UCS2
					Number of sets of the following elements: <ul style="list-style-type: none">• uss_data
					uss data per the coding scheme.
Type	0x13			1	Alpha Identifier
Length	Var			2	
Value	→	enum8	alpha_dcs	1	Alpha coding scheme. Values: <ul style="list-style-type: none">• 0x01 – ALPHA_DCS_GSM – SMS default 7-bit coded alphabet as defined in 3GPP TS 23.038 with bit 8 set to 0• 0x02 – ALPHA_DCS_UCS2 – UCS2
					Number of sets of the following elements: <ul style="list-style-type: none">• alpha_text
					Data encoded per alpha_dcs.
Type	0x14			1	USS Data from Network in UTF-16 Encoding
Length	Var			2	
Value	→	uint8	uss_info_utf16_len	1	Number of sets of the following elements: <ul style="list-style-type: none">• uss_info_utf16
					Unstructured supplementary service information in UTF-16 encoding.

Error codes

QMI_ERR_INTERNAL	Unexpected error occurred during processing
QMI_ERR_SUPS_FAILURE_CAUSE	Indicates supplementary services failure information; see Table A-3 for failure cause
QMI_ERR_NETWORK_ABORTED	Operation was released abruptly by the network

11 User Identity Module (QMI_UIM)

This specification documents Major Version 1 of the Qualcomm Messaging Interface (QMI) for the User Identity Module (QMI_UIM).

11.1. Theory of Operation

11.1.1. Generalized QMI Service Compliance

The QMI_UIM 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 80-VB816-1. Extensions to the generalized QMI service theory of operation are noted in subsequent sections of this chapter.

11.1.2. UIM Service Type

The UIM is assigned QMI service type 0x0B.

11.1.3. Message Definition Template

11.1.3.1. Response Message Result TLV

This Type-Length-Value (TLV) is present in all Response messages defined in this document. It is not present in the Indication messages.

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

11.1.4. QMI_UIM Fundamental Concepts

11.1.4.1. UIM Overview

The UIM module is used to access the card available on the device. The implementation supports SIM and USIM cards used for GSM/WCDMA devices, as well as RUIM and CSIM cards used for CDMA devices.

11.1.4.2. Commands Supported via QMI_UIM

11.1.4.3. Multiple Applications on the Same Card

Multimode UICC cards allow true multimode operation with a single smart card. Previously, only one application was accessed. Up to two active provisioning applications are now possible, and application-specific requests are required to specify the application to which the request pertains.

11.1.4.4. Sessions

To access each application available on the card, sessions are used, as described in 80-VT475-1. The service automatically opens six sessions to the provisioning applications for both 1X and GSM/WCDMA and to the card in each slot for the files under MF. Depending on the capabilities of the target, some of these sessions might be invalid. The same notion of sessions is also used in the case of ICC cards, even

though they do not support logical channels or applications.

In addition to providing access to files in the provisioning applications, the service also makes it possible to access other files, specifying the Application Identifier (AID) or the slot of the card; in these cases, an additional session is opened on-the-fly and maintained by the QMI service.

11.1.4.5. Commands Supported via QMI_UIM

The QMI_UIM module enables access to several features of the card.

- Access to files on the card
 - Read transparent and records
 - Write transparent and records
 - Get file attributes
- PIN operation
 - Enable and disable
 - Verify
 - Unblock
 - Change
- Other tasks
 - Support for file refresh operations
 - Support for personalization
 - Power up/power down the card
 - Authentication
 - Select provisioning applications from the card
 - Get modem configuration
 - Send raw APDUs to the card
 - SIM Access Profile

A control point can register for notification of card events. The QMI layer maintains a cached status of the card and updates the status with notifications from the modem.

11.1.4.6. Refresh Procedure via QMI_UIM

The card can automatically start a refresh procedure. Refresh is one of the proactive commands; however, it must be handled by the modem in a special way, because it might impact many other modules (e.g., call manager, phonebook). During the refresh procedure, the control point receives multiple events at different stages. Details about the events and the call flows for the various refresh scenarios are described in 80-VM566-1.

11.1.5. Service State Variables

11.1.5.1. Shared State Variables

No QMI_UIM state variables are shared across control points.

11.2. QMI_UIM Messages

Command	ID	Description
QMI_UIM_READ_TRANSPARENT	0x0020	Provides read access to any transparent file in the card and provides access by the path.
QMI_UIM_READ_RECORD	0x0021	Provides read access to a specific record in a linear fixed or cyclic file in the card and provides access by the path.
QMI_UIM_WRITE_TRANSPARENT	0x0022	Provides write access to any transparent file in the card and provides access by the path.
QMI_UIM_WRITE_RECORD	0x0023	Provides write access to a specific record in a linear fixed or cyclic file in the card and provides access by the path.
QMI_UIM_GET_FILE_ATTRIBUTES	0x0024	Retrieves the file attributes for any EF or DF in the card and provides access by the path.
QMI_UIM_VERIFY_PIN	0x0026	Verifies the PIN before the card content is accessed.
QMI_UIM_EVENT_REGISTRATION	0x002E	Registers for event notifications from the card.
QMI_UIM_EVENT_REG		
QMI_UIM_GET_CARD_STATUS	0x002F	Retrieves the current status of the card. QMI_UIM_POWER_DOWN 0x0030 Powers down the card.QMI_UIM_POWER_UP 0x0031 Powers up the card.
QMI_UIM_GET_CARD_STATUS_EVENT	0x0032	Indicates that the status of the card has changed.
QMI_UIM_STATUS_CHANGE_IND		

11.2.1. QMI_UIM_READ_TRANSPARENT

Provides read access to any transparent file in the card and provides access by the path.

- **UIM message ID**

0x0020

- Version introduced

Major - 1, Minor - 0

11.2.1.1. Request - QMI_UIM_READ_TRANSPARENT_REQ

- Message type

Request

- Sender

Control point

- Mandatory TLVs

Name	Version introduced	Version last modified
Session Information	1.0	1.30
File ID	1.0	1.0
Read Transparent	1.0	1.0

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Session Information
Length	Var			2	
Value	→	enum8	session_type	1	Indicates the session type. Valid values: 0 through 24. See Table A-1 for the list of available session types with their values and descriptions.
		uint8	aid_len	1	Number of sets of the following elements: <ul style="list-style-type: none">• aid
		uint8	aid	Var	Application identifier value or channel ID. This value is required for nonprovisioning and for logical channel session types. It is ignored in all other cases.
Type	0x02			1	File ID
Length	Var			2	
Value	→	uint16	file_id	2	File ID.

Field	Field value	Field type	Parameter	Size (byte)	Description
		uint8	path_len	1	Number of sets of the following elements: • path
		uint8	path	Var	File path. This value must be the complete path of the file, which is a sequence block of 2 bytes (e.g., 0x3F00 0x7FFF).
Type	0x03			1	Read Transparent
Length	4			2	
Value	→	uint16	offset	2	Offset for the Read operation.
		uint16	length	2	Length of the content to be read. The value 0 is used to read the complete file.

Optional TLVs

Name	Version introduced	Version last modified
Response in Indication	1.12	1.12
Encrypt Data	1.18	1.18
Encrypt Request Data	1.45	1.45

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Response in Indication
Length	4			2	
Value	→	uint32	indication_token	4	When this TLV is present, it indicates that the result must be provided in a subsequent indication.
Type	0x11			1	Encrypt Data
Length	1			2	
Value	→	boolean	encryption	1	Indicates whether the data read from the card is to be encrypted.
Type	0x12			1	Encrypt Request Data
Length	1			2	
Value	→	boolean	encryption_ext	1	Indicates whether the session information, file ID, and data read from the card are to be encrypted.

11.2.1.2. Response - QMI_UIM_READ_TRANSPARENT_RESP

- Message type

Response

- Sender

Service

- **Mandatory TLVs**

The Result Code TLV (defined in Section 11.1.3.1) is always present in the response.

Name	Version introduced	Version last modified
Result Code	1.0	1.29

Optional TLVs

Name	Version introduced	Version last modified
Card Result	1.0	1.0
Read Result	1.0	1.0
Response in Indication	1.12	1.12
Encrypted Data	1.18	1.18
Requested Length	1.20	1.20
Long Data Token	1.35	1.35
Read Result With Request Information	1.45	1.45

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Card Result
Length	2			2	
Value	→	uint8	sw1	1	SW1 status code received from the card.
		uint8	sw2	1	SW2 status code received from the card.
Type	0x11			1	Read Result
Length	Var			2	
Value	→	uint16	content_len	2	Number of sets of the following elements: • content
		uint8	content	Var	Read content; sequence of bytes as read from the card.
Type	0x12			1	Response in Indication
Length	4			2	
Value	→	uint32	indication_token	4	When this TLV is present, it indicates that the result is provided in a subsequent indication.
Type	0x13			1	Encrypted Data
Length	1			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	boolean	encryption	1	Indicates whether the data from the card passed in read_result is encrypted.
Type	0x14			1	Requested Length
Length	2			2	
Value	→	uint16	file_length	2	When this TLV is present, the requested length exceeds the maximum size supported by the QMI UIM. QMI_ERR_INSUFFICIENT_RESOURCES is returned to the client and this value indicates the total length.
Type	0x15			1	Long Data Token
Length	4			2	
Value	→	uint32	long_data_token	4	When this TLV is present, the requested length exceeds the maximum size supported by QMI UIM. QMI_ERR_INSUFFICIENT_RESOURCES is returned to the client, and the result is provided in subsequent indications. Long data token is used to reconstruct the entire read response, possibly spanning multiple indications.
Type	0x16			1	Read Result With Request Information
Length	Var			2	
Value	→	uint16	read_result_ext_len	2	Number of sets of the following elements: <ul style="list-style-type: none">• read_result_ext
		opaque	read_result_ext	Var	Read request information and content. Includes session information, file ID as provided in the read request, and the sequence of bytes as read from the card. See Section D.5 for the result format.

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 the response.
QMI_ERR_ACCESS_DENIED	Access to the requested file is denied. This can occur when there is an attempt to access a PIN-protected file.
QMI_ERR_SIM_FILE_NOT_FOUND	File is not present on the card.
QMI_ERR_ARG_TOO_LONG	Path in the request was too long.
QMI_ERR_INVALID_ARG	Parameters passed as input were invalid (e.g., an odd length of the path).
QMI_ERR_DEVICE_NOT_READY	Device is not ready.

QMI_ERR_INSUFFICIENT_RESOURCES	Response is longer than the maximum supported size.
QMI_ERR_NOT_PROVISIONED	File is not provisioned on the card.

11.2.1.3. Description of QMI_UIM_READ_TRANSPARENT REQ/RESP

This function provides read access to any transparent file in the card and access by the path. The client using this function is only able to access files that have a flat data structure. An error is reported for all other cases.

The response contains the status code received from the card (SW1 and SW2) when the card responded to the read request.

The client can pass a token in the request to receive the result in a subsequent QMI_UIM_READ_TRANSPARENT_IND indication. In this case, the immediate response indicates only whether the command was received correctly for processing. The service passes the token back in the immediate response to indicate that it supports the indication. The service then passes the same token in the indication, so that the client can match it with the original request.

A client must be able to accommodate the case where the optional indication is not supported by the service and the return of the complete result in the response, even when the request indicates that the indication is to be used.

A client can request to encrypt the data read from the card. This procedure guarantees protection against a man-in-the-middle attack between the service and the client. The security algorithm and keys used for the encryption are not part of this document. The service might not support the encryption and provide a non-encrypted response.

A client can request to encrypt the session and file information in addition to the data read from the card.

A client can request either to encrypt the data read from the card or to encrypt session and file information along with the data read from the card.

The service returns QMI_ERR_INSUFFICIENT_RESOURCES when the file being read is too large, in which case the file_length field contains the total length of the file. The client can use this information to request the data with multiple operations that read smaller blocks. Additionally, the Long Data Token TLV is sent. The service then sends one or more QMI_UIM_READ_TRANSPARENT_LONG_IND indications to the client. The client can use the long_data_token and file_length fields to reconstruct the final data for the read operation.

Other error codes not listed in the preceding table are handled as QMI_ERR_INTERNAL.

11.2.2. QMI_UIM_READ_RECORD

Provides read access to a specific record in a linear fixed or cyclic file in the card and provides access by the path.

- **UIM message ID**

0x0021

- **Version introduced**

Major - 1, Minor - 0

11.2.2.1. Request - QMI_UIM_READ_RECORD_REQ

- **Message type**

Request

- **Sender**

Control point

- Mandatory TLVs

Name	Version introduced	Version last modified
Session Information	1.0	1.30
File ID	1.0	1.0
Read Record	1.0	1.0

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Session Information
Length	Var			2	
Value	→	enum8	session_type	1	Indicates the session type. Valid values: 0 through 24. See Table A-1 for the list of available session types with their values and descriptions.
		uint8	aid_len	1	Number of sets of the following elements: <ul style="list-style-type: none">• aid
		uint8	aid	Var	Application identifier value or channel ID. This value is required for nonprovisioning and for logical channel session types. It is ignored in all other cases.
Type	0x02			1	File ID
Length	Var			2	
Value	→	uint16	file_id	2	File ID.

Field	Field value	Field type	Parameter	Size (byte)	Description
		uint8	path_len	1	Number of sets of the following elements: <ul style="list-style-type: none">• path
		uint8	path	Var	File path. This value must be the complete path of the file, which is a sequence block of 2 bytes (e.g., 0x3F00 0x7FFF).
Type	0x03			1	Read Record
Length	4			2	
Value	→	uint16	record	2	Record number (starting from 1).
		uint16	length	2	Length of the content to be read. The value 0 is used to read the complete record.

Optional TLVs

Name	Version introduced	Version last modified
Last Record	1.8	1.8
Response in Indication	1.12	1.12

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Last Record
Length	2			2	
Value	→	uint16	last_record	2	Last record. This value is used to read multiple records at the same time.
Type	0x11			1	Response in Indication
Length	4			2	
Value	→	uint32	indication_token	4	When this TLV is present, it indicates that the result must be provided in a subsequent indication.

11.2.2.2. Response - QMI_UIM_READ_RECORD_RESP

- Message type

Response

- Sender

Service

- Mandatory TLVs

The Result Code TLV (defined in Section 11.1.3.1) is always present in the response.

Name	Version introduced	Version last modified
Result Code	1.0	1.29

Optional TLVs

Name	Version introduced	Version last modified
Card Result	1.0	1.0
Read Result	1.0	1.0
Additional Read Result	1.8	1.8
Response in Indication	1.12	1.12

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Card Result
Length	2			2	
Value	→	uint8	sw1	1	SW1 status code received from the card.
		uint8	sw2	1	SW2 status code received from the card.
Type	0x11			1	Read Result
Length	Var			2	
Value	→	uint16	content_len	2	Number of sets of the following elements: <ul style="list-style-type: none">• content
		uint8	content	Var	Read content. The content is the sequence of bytes as read from the card.
Type	0x12			1	Additional Read Result
Length	Var			2	
Value	→	uint16	additional_record_len	2	Number of sets of the following elements: <ul style="list-style-type: none">• additional_record
		uint8	additional_record	Var	Read content of all the additional records. Each record has the same size as the first record.
Type	0x13			1	Response in Indication
Length	4			2	
Value	→	uint32	indication_token	4	When this TLV is present, it indicates that the result is provided in a subsequent indication.

Error codes

<code>QMI_ERR_NONE</code>	No error in the request.
<code>QMI_ERR_INTERNAL</code>	Unexpected error occurred during processing.
<code>QMI_ERR_MALFORMED_MSG</code>	Message was not formulated correctly by the control point or the message was corrupted during transmission.
<code>QMI_ERR_NO_MEMORY</code>	Device could not allocate memory to formulate the response.
<code>QMI_ERR_ACCESS_DENIED</code>	Access to the requested file is denied. This can occur when an attempt is made to access a PIN-protected file.
<code>QMI_ERR_SIM_FILE_NOT_FOUND</code>	File is not present on the card.
<code>QMI_ERR_ARG_TOO_LONG</code>	Path in the request was too long.
<code>QMI_ERR_INVALID_ARG</code>	Parameters passed as input were invalid (e.g., an odd length of the path).
<code>QMI_ERR_DEVICE_NOT_READY</code>	Device is not ready.
<code>QMI_ERR_NOT_PROVISIONED</code>	File is not provisioned on the card.

11.2.2.3. Description of QMI_UIM_READ_RECORD REQ/RESP

This function provides read access to a specific record in a linear fixed or cyclic file in the card and provides access by the path.

The response contains the status code received from the card (SW1 and SW2) when the card responded to the read request.

The client can pass a token in the request to receive the result in a subsequent QMI_UIM_READ_RECORD_IND indication. In this case, the immediate response indicates only whether the command was received correctly for processing. The service passes the token back in the immediate response to indicate that it supports the indication. The service then passes the same token in the indication, so that the client can match it with the original request.

A client must be able to accommodate the case where the optional indication is not supported by the service and the return of the complete result in the response, even when the request indicates that the indication is to be used. When multiple records are requested, the QMI UIM stops reading additional records from the file as soon as a Read operation returns an error. The QMI UIM returns only the records that are read successfully from the card without any indication of the specific error for the additional records.

When the records are large, the client must avoid reading too many records at the same time.

Other error codes not listed in the preceding table are handled as `QMI_ERR_INTERNAL`.

11.2.3. QMI_UIM_WRITE_TRANSPARENT

Provides write access to any transparent file in the card and provides access by the path.

- UIM message ID

0x0022

- Version introduced

Major - 1, Minor - 0

11.2.3.1. Request - QMI_UIM_WRITE_TRANSPARENT_REQ

- Message type

Request

- Sender

Control point

- Mandatory TLVs

Name	Version introduced	Version last modified
Session Information	1.0	1.30
File ID	1.0	1.0
Write Transparent	1.0	1.0

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Session Information
Length	Var			2	
Value	→	enum8	session_type	1	Indicates the session type. Valid values: 0 through 24. See Table A-1 for the list of available session types with their values and descriptions.
		uint8	aid_len	1	Number of sets of the following elements: <ul style="list-style-type: none">• aid
		uint8	aid	Var	Application identifier value or channel ID. This value is required for nonprovisioning and for logical channel session types. It is ignored in all other cases.
Type	0x02			1	File ID
Length	Var			2	
Value	→	uint16	file_id	2	File ID.

Field	Field value	Field type	Parameter	Size (byte)	Description
		uint8	path_len	1	Number of sets of the following elements: <ul style="list-style-type: none">• path
		uint8	path	Var	File path. This value must be the complete path of the file, which is a sequence block of 2 bytes (e.g., 0x3F00 0x7FFF).
Type	0x03			1	Write Transparent
Length	Var			2	
Value	→	uint16	offset	2	Offset for the Write operation.
		uint16	data_len	2	Number of sets of the following elements: <ul style="list-style-type: none">• data
		uint8	data	Var	Content to write.

Optional TLVs

Name	Version introduced	Version last modified
Response in Indication	1.12	1.12

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Response in Indication
Length	4			2	
Value	→	uint32	indication_token	4	When this TLV is present, it indicates that the result must be provided in a subsequent indication.

11.2.3.2. Response - QMI_UIM_WRITE_TRANSPARENT_RESP

- Message type

Response

- Sender

Service

- Mandatory TLVs

The Result Code TLV (defined in Section 11.1.3.1) is always present in the response.

Name	Version introduced	Version last modified
Result Code	1.0	1.29

Optional TLVs

Name	Version introduced	Version last modified
Card Result	1.0	1.0
Response in Indication	1.12	1.12

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Card Result
Length	2			2	
Value	→	uint8	sw1	1	SW1 status code received from the card.
		uint8	sw2	1	SW2 status code received from the card.
Type	0x11			1	Response in Indication
Length	4			2	
Value	→	uint32	indication_token	4	When this TLV is present, it indicates that the result is provided in a subsequent indication.

Error codes

<code>QMI_ERR_NONE</code>	No error in the request.
<code>QMI_ERR_INTERNAL</code>	Unexpected error occurred during processing.
<code>QMI_ERR_MALFORMED_MSG</code>	Message was not formulated correctly by the control point or the message was corrupted during transmission.
<code>QMI_ERR_NO_MEMORY</code>	Device could not allocate memory to formulate the response.
<code>QMI_ERR_ACCESS_DENIED</code>	Access to the requested file is denied. This can occur when an attempt is made to access a PIN-protected file.
<code>QMI_ERR_SIM_FILE_NOT_FOUND</code>	File is not present on the card.
<code>QMI_ERR_ARG_TOO_LONG</code>	Path in the request was too long.
<code>QMI_ERR_INVALID_ARG</code>	Parameters passed as input were invalid (e.g., an odd length of the path).
<code>QMI_ERR_DEVICE_NOT_READY</code>	Device is not ready.
<code>QMI_ERR_NOT_PROVISIONED</code>	File is not provisioned on the card.

11.2.3.3. Description of QMI_UIM_WRITE_TRANSPARENT REQ/RESP

This function provides write access to any transparent file in the card and provides access by the path.

The client using this function is able to access only files that have a flat data structure. An error is reported for all other cases.

The response contains the status code received from the card (SW1 and SW2) when the card responded to the write request.

The client can pass a token in the request to receive the result in a subsequent QMI_UIM_WRITE_TRANSPARENT_IND indication. In this case, the immediate response indicates only whether the command was received correctly for processing. The service passes the token back in the immediate response to indicate that it supports the indication. The service then passes the same token in the indication, so that the client can match it with the original request.

A client must be able to accommodate the case where the optional indication is not supported by the service and the return of the complete result in the response, even when the request indicates that the indication is to be used.

Other error codes not listed in the preceding table are handled as QMI_ERR_INTERNAL.

11.2.4. QMI_UIM_WRITE_RECORD

Provides write access to a specific record in a linear fixed or cyclic file in the card and provides access by the path.

- **UIM message ID**

0x0023

- **Version introduced**

Major - 1, Minor - 0

11.2.4.1. Request - QMI_UIM_WRITE_RECORD_REQ

- **Message type**

Request

- **Sender**

Control point

- Mandatory TLVs

Name	Version introduced	Version last modified
Session Information	1.0	1.30
File ID	1.0	1.0
Write Record	1.0	1.0

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Session Information
Length	Var			2	
Value	→	enum8	session_type	1	Indicates the session type. Valid values: 0 through 24. See Table A-1 for the list of available session types with their values and descriptions.
		uint8	aid_len	1	Number of sets of the following elements: <ul style="list-style-type: none">• aid
		uint8	aid	Var	Application identifier value or channel ID. This value is required for nonprovisioning and for logical channel session types. It is ignored in all other cases.
Type	0x02			1	File ID
Length	Var			2	
Value	→	uint16	file_id	2	File ID.

Field	Field value	Field type	Parameter	Size (byte)	Description
		uint8	path_len	1	Number of sets of the following elements: <ul style="list-style-type: none">• path
		uint8	path	Var	File path. This value must be the complete path of the file, which is a sequence block of 2 bytes (e.g., 0x3F00 0x7FFF).
Type	0x03			1	Write Record
Length	Var			2	
Value	→	uint16	record	2	Record number (starting from 1). This field is ignored for cyclic files.
		uint16	data_len	2	Number of sets of the following elements: <ul style="list-style-type: none">• data
		uint8	data	Var	Content to write.

Optional TLVs

Name	Version introduced	Version last modified
Response in Indication	1.12	1.12

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Response in Indication
Length	4			2	
Value	→	uint32	indication_token	4	When this TLV is present, it indicates that the result must be provided in a subsequent indication.

11.2.4.2. Response - QMI_UIM_WRITE_RECORD_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	Version introduced	Version last modified
Result Code	1.0	1.29

Optional TLVs

Name	Version introduced	Version last modified
Card Result	1.0	1.0
Response in Indication	1.12	1.12

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Card Result
Length	2			2	
Value	→	uint8	sw1	1	SW1 status code received from the card.
		uint8	sw2	1	SW2 status code received from the card.
Type	0x11			1	Response in Indication
Length	4			2	
Value	→	uint32	indication_token	4	When this TLV is present, it indicates that the result is provided in a subsequent indication.

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 the response.
QMI_ERR_ACCESS_DENIED	Access to the requested file is denied. This can occur when an attempt is made to access a PIN-protected file.
QMI_ERR_SIM_FILE_NOT_FOUND	File is not present on the card.
QMI_ERR_ARG_TOO_LONG	Path in the request was too long.
QMI_ERR_INVALID_ARG	Parameters passed as input were invalid (e.g., an odd length of the path).
QMI_ERR_DEVICE_NOT_READY	Device is not ready.
QMI_ERR_NOT_PROVISIONED	File is not provisioned on the card.

11.2.4.3. Description of QMI_UIM_WRITE_RECORD REQ/RESP

This function provides write access to a specific record in a linear fixed or cyclic file in the card and provides access by the path.

The response contains the status code received from the card (SW1 and SW2) when the card responded to the write request.

The client can pass a token in the request to receive the result in a subsequent QMI_UIM_WRITE_RECORD_IND indication. In this case, the immediate response indicates only whether the command was received correctly for processing. The service passes the token back in the immediate response to indicate that it supports the indication. The service then passes the same token in the indication, so that the client can match it with the original request.

A client must be able to accommodate the case where the optional indication is not supported by the service and the return of the complete result in the response, even when the request indicates that the indication is to be used.

Other error codes not listed in the preceding table are handled as QMI_ERR_INTERNAL

11.2.5. QMI_UIM_GET_FILE_ATTRIBUTES

Retrieves the file attributes for any EF or DF in the card and provides access by the path.

- **UIM message ID**

0x0024

- **Version introduced**

Major - 1, Minor - 0

11.2.5.1. Request - QMI_UIM_GET_FILE_ATTRIBUTES_REQ

- **Message type**

Request

- **Sender**

Control point

- Mandatory TLVs

Name	Version introduced	Version last modified
Session Information	1.0	1.30
File ID	1.0	1.0

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Session Information
Length	Var			2	
Value	→	enum8	session_type	1	Indicates the session type. Valid values: 0 through 24. See Table A-1 for the list of available session types with their values and descriptions.
		uint8	aid_len	1	Number of sets of the following elements: • aid
		uint8	aid	Var	Application identifier value or channel ID. This value is required for nonprovisioning and for logical channel session types. It is ignored in all other cases.
Type	0x02			1	File ID
Length	Var			2	
Value	→	uint16	file_id	2	File ID.
		uint8	path_len	1	Number of sets of the following elements: • path

Field	Field value	Field type	Parameter	Size (byte)	Description
		uint8	path	Var	File path. This value must be the complete path of the file, which is a sequence block of 2 bytes (e.g., 0x3F00 0x7FFF).

Optional TLVs

Name	Version introduced	Version last modified
Response in Indication	1.12	1.12
UICC Security Attributes	1.30	1.30

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Response in Indication
Length	4			2	
Value	→	uint32	indication_token	4	When this TLV is present, it indicates that the result must be provided in a subsequent indication.
Type	0x11			1	UICC Security Attributes
Length	1			2	
Value	→	boolean	uicc_security_attributes	1	Indicates whether the security attributes are required by the client. Valid values: • 0 – Skip security attributes of the file for the UICC card • 1 – Return security attributes of the file for the UICC card If the UICC Security Attributes TLV is missing, security attributes are returned. This TLV is ignored for ICC cards.

11.2.5.2. Response - QMI_UIM_GET_FILE_ATTRIBUTES_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	Version introduced	Version last modified
Result Code	1.0	1.29

Optional TLVs

Name	Version introduced	Version last modified
Card Result	1.0	1.0
File Attributes	1.0	1.0
Response in Indication	1.12	1.12
File Status	1.32	1.32

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Card Result
Length	2			2	
Value	→	uint8	sw1	1	SW1 status code received from the card.
		uint8	sw2	1	SW2 status code received from the card.
Type	0x11			1	File Attributes
Length	Var			2	
Value	→	uint16	file_size	2	File size.
		uint16	file_id	2	File ID.
	enum8	file_type		1	File type. Valid values: <ul style="list-style-type: none">• UIM_FILE_TYPE_TRANSPARENT (0x00) – Transparent• UIM_FILE_TYPE_CYCLIC (0x01) – Cyclic• UIM_FILE_TYPE_LINEAR_FIXED (0x02) – Linear fixed• UIM_FILE_TYPE_DEDICATED_FILE (0x03) – Dedicated file• UIM_FILE_TYPE_MASTER_FILE (0x04) – Master file
				2	Size of the records (only for cyclic and linear fixed files).
				2	Total number of records (only for linear fixed files).

Field	Field value	Field type	Parameter	Size (byte)	Description
		enum8	sec_read	1	Read security attributes. Valid values: • UIM_SEC_CONDITION_ALWAYS (0x00) – Always • UIM_SEC_CONDITION_NEVER (0x01) – Never • UIM_SEC_CONDITION_AND_COND (0x02) – AND condition • UIM_SEC_CONDITION_OR_COND (0x03) – OR condition • UIM_SEC_CONDITION_SINGLE_COND (0x04) – Single condition
		uint16	sec_read_mask	2	Mask with read security attributes. This field is valid only when required by security attributes. Valid values: • Bit 0 – PIN1 • Bit 1 – PIN2 • Bit 2 – UPIN • Bit 3 – ADM
		enum8	sec_write	1	Write security attributes. Valid values: • UIM_SEC_CONDITION_ALWAYS (0x00) – Always • UIM_SEC_CONDITION_NEVER (0x01) – Never • UIM_SEC_CONDITION_AND_COND (0x02) – AND condition • UIM_SEC_CONDITION_OR_COND (0x03) – OR condition • UIM_SEC_CONDITION_SINGLE_COND (0x04) – Single condition
		uint16	sec_write_mask	2	Mask with write security attributes. This field is valid only when required by security attributes. Valid values: • Bit 0 – PIN1 • Bit 1 – PIN2 • Bit 2 – UPIN • Bit 3 – ADM
		enum8	sec_increase	1	Increase security attributes. Valid values: • UIM_SEC_CONDITION_ALWAYS (0x00) – Always • UIM_SEC_CONDITION_NEVER (0x01) – Never • UIM_SEC_CONDITION_AND_COND (0x02) – AND condition • UIM_SEC_CONDITION_OR_COND (0x03) – OR condition • UIM_SEC_CONDITION_SINGLE_COND (0x04) – Single condition

Field	Field value	Field type	Parameter	Size (byte)	Description
		uint16	sec_increase_mask	2	<p>Mask with increase security attributes. This field is valid only when required by security attributes. Valid values:</p> <ul style="list-style-type: none"> • Bit 0 – PIN1 • Bit 1 – PIN2 • Bit 2 – UPIN • Bit 3 – ADM
		enum8	sec_deactivate	1	<p>Deactivate security attributes. Valid values:</p> <ul style="list-style-type: none"> • UIM_SEC_CONDITION_ALWAYS (0x00) – Always • UIM_SEC_CONDITION_NEVER (0x01) – Never • UIM_SEC_CONDITION_AND_COND (0x02) – AND condition • UIM_SEC_CONDITION_OR_COND (0x03) – OR condition • UIM_SEC_CONDITION_SINGLE_COND (0x04) – Single condition
		uint16	sec_deactivate_mask	2	<p>Mask with deactivate security attributes. This field is valid only when required by security attributes. Valid values:</p> <ul style="list-style-type: none"> • Bit 0 – PIN1 • Bit 1 – PIN2 • Bit 2 – UPIN • Bit 3 – ADM
		enum8	sec_activate	1	<p>Activate security attributes. Valid values:</p> <ul style="list-style-type: none"> • UIM_SEC_CONDITION_ALWAYS (0x00) – Always • UIM_SEC_CONDITION_NEVER (0x01) – Never • UIM_SEC_CONDITION_AND_COND (0x02) – AND condition • UIM_SEC_CONDITION_OR_COND (0x03) – OR condition • UIM_SEC_CONDITION_SINGLE_COND (0x04) – Single condition
		uint16	sec_activate_mask	2	<p>Mask with Activate security attributes. This field is valid only when required by security attributes. Valid values:</p> <ul style="list-style-type: none"> • Bit 0 – PIN1 • Bit 1 – PIN2 • Bit 2 – UPIN • Bit 3 – ADM
		uint16	raw_len	2	<p>Number of sets of the following elements:</p> <ul style="list-style-type: none"> • raw_value

Field	Field value	Field type	Parameter	Size (byte)	Description
		uint8	raw_value	Var	Raw value of file attributes.
Type	0x12			1	Response in Indication
Length	4			2	
Value	→	uint32	indication_token	4	When this TLV is present, it indicates that the result is provided in a subsequent indication.
Type	0x13			1	File Status
Length	4			2	
Value	→	enum	file_status	4	Indicates the status of the file. Valid values: <ul style="list-style-type: none">• UIM_FILE_STATUS_DEACTIVATED (0x00) – File is deactivated.• UIM_FILE_STATUS_ACTIVATED (0x01) – File is activated. Other values are reserved for future use and are to be ignored by the client.

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 the response.
QMI_ERR_ACCESS_DENIED	Access to the requested file is denied. This can occur when an attempt is made to access a PIN-protected file.
QMI_ERR_SIM_FILE_NOT_FOUND	File is not present on the card.
QMI_ERR_ARG_TOO_LONG	Path in the request was too long.
QMI_ERR_INVALID_ARG	Parameters passed as input were invalid (e.g., an odd length of the path).
QMI_ERR_DEVICE_NOT_READY	Device is not ready.
QMI_ERR_NOT_PROVISIONED	File is not provisioned on the card.

11.2.5.3. Description of QMI_UIM_GET_FILE_ATTRIBUTES REQ/RESP

This function retrieves the file attributes for any EF or DF in the card and provides access by the path.

The response contains the status code received from the card (SW1 and SW2) when the card responded to the select request.

The client can pass a token in the request to receive the result in a subsequent QMI_UIM_GET_FILE_ATTRIBUTES_IND indication. In this case, the immediate response indicates only whether the command was received correctly for processing. The service passes the token back in the

immediate response to indicate that it supports the indication. The service then passes the same token in the indication, so that the client can match it with the original request.

A client must be able to accommodate the case where the optional indication is not supported by the service and the return of the complete result in the response, even when the request indicates that the indication is to be used.

To retrieve the attributes for a DF or for the MF, the special value 0xFFFF is used for the file ID in the request.

On a UICC card, the security attributes are normally stored in the Access Rule Reference file (EF-ARR). If a client does not require the security attributes, the client can request to skip them. This improves the performance of the request because the device is not required to read the EF-ARR. In this case, the security attributes in the response indicate that all operations are always permitted.

Other error codes not listed in the preceding table are handled as QMI_ERR_INTERNAL.

11.2.6. QMI_UIM_VERIFY_PIN

Verifies the PIN before the card content is accessed.

- **UIM message ID**

0x0026

- **Version introduced**

Major - 1, Minor - 0

11.2.6.1. Request - QMI_UIM_VERIFY_PIN_REQ

- **Message type**

Request

- **Sender**

Control point

- Mandatory TLVs

Name	Version introduced	Version last modified
Session Information	1.0	1.30
Verify PIN	1.0	1.0

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Session Information
Length	Var			2	
Value	→	enum8	session_type	1	Indicates the session type. Valid values: 0 through 24. See Table A-1 for the list of available session types with their values and descriptions.
		uint8	aid_len	1	Number of sets of the following elements: <ul style="list-style-type: none">• aid
		uint8	aid	Var	Application identifier value or channel ID. This value is required for nonprovisioning and for logical channel session types. It is ignored in all other cases.
Type	0x02			1	Verify PIN
Length	Var			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	enum8	pin_id	1	<p>Indicates the PIN ID to be verified. Valid values:</p> <ul style="list-style-type: none"> • UIM_PIN_ID_PIN_1 (0x01) – PIN1 (also called PIN) • UIM_PIN_ID_PIN_2 (0x02) – PIN2 • UIM_PIN_ID_UNIVERSAL (0x03) – Universal PIN • UIM_PIN_ID_HIDDEN_KEY (0x04) – Hidden key
		uint8	pin_len	1	<p>Number of sets of the following elements:</p> <ul style="list-style-type: none"> • pin_value
		char	pin_value	Var	PIN value. This value is a sequence of ASCII characters.

Optional TLVs

Name	Version introduced	Version last modified
Encrypted Value of PIN1	1.7	1.7
Key Reference ID	1.10	1.10
Response in Indication	1.12	1.12

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Encrypted Value of PIN1 (This field can be used only when the PIN type is PIN1.)
Length	Var			2	
	→	uint8	enc_pin1_len	1	<p>Number of sets of the following elements:</p> <ul style="list-style-type: none"> • enc_pin1_value
Type	0x11		enc_pin1_value	Var	Encrypted PIN1 value.
Length	1			2	
Value	→	enum8	pin_key_ref_id	1	Indicates the PIN key reference ID. Valid values are from 1 to 8, respectively, for application 1 to application 8, as specified in ETSI TS 102 221 . This TLV is used only for PIN1 and PIN2 and is ignored in all other cases.
Type	0x12			1	Response in Indication
Length	4			2	
Value	→	uint32	indication_token	4	When this TLV is present, it indicates that the result must be provided in a subsequent indication.

11.2.6.2. Response - QMI_UIM_VERIFY_PIN_RESP

- **Message type**

Response

- **Sender**

Service

- **Mandatory TLVs**

The Result Code TLV (defined in Section 11.1.3.1) is always present in the response.

Name	Version introduced	Version last modified
Result Code	1.0	1.0

Optional TLVs

Name	Version introduced	Version last modified
Retries Remaining	1.0	1.0
Encrypted PIN1	1.7	1.7
Response in Indication	1.12	1.12
Card Result	1.29	1.29

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Retries Remaining (This value is returned only when the verify operation fails.)
Length	2			2	
Value	→	uint8	verify_left	1	Number of remaining attempts to verify the PIN.
		uint8	unblock_left	1	Number of remaining attempts to unblock the PIN.
Type	0x11			1	Encrypted PIN1 (This value is returned only when PIN1 is verified successfully and the feature is supported.)
Length	Var			2	
Value	→	uint8	enc_pin1_len	1	Number of sets of the following elements: <ul style="list-style-type: none">• enc_pin1_value
		uint8	enc_pin1_value	Var	Encrypted PIN1 value.
Type	0x12			1	Response in Indication
Length	4			2	
Value	→	uint32	indication_token	4	When this TLV is present, it indicates that the result is provided in a subsequent indication.

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x13			1	Card Result
Length	2			2	
Value	→	uint8	sw1	1	SW1 status code received from the card.
		uint8	sw2	1	SW2 status code received from the card.

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 the response.
QMI_ERR_NO_EFFECT	Operation had no effect.
QMI_ERR_ARG_TOO_LONG	Device cannot handle the PIN length in the request.
QMI_ERR_INCORRECT_PIN	PIN in the request is incorrect.
QMI_ERR_PIN_BLOCKED	PIN is blocked. An unblock operation must be issued.
QMI_ERR_PIN_PERM_BLOCKED	PIN is permanently blocked. The SIM is unusable.
QMI_ERR_SIM_NOT_INITIALIZED	PIN is not yet initialized because the SIM initialization has not finished. Try the PIN operation later.
QMI_ERR_OP_DEVICE_UNSUPPORTED	Operation is not supported by the device nor by the SIM card.
QMI_ERR_MISSING_ARG	TLV was missing in the request.
QMI_ERR_INVALID_PINID	PIN in the request is invalid.
QMI_ERR_DEVICE_NOT_READY	Device is not ready.

11.2.6.3. Description of QMI_UIM_VERIFY_PIN REQ/RESP

This function verifies the PIN before the card content is accessed.

The same PIN can be used by multiple sessions (i.e., the PIN is shared between GSM and RUIIM in an ICC card). The PIN is automatically verified for all the sessions when the command is executed.

The hidden key PIN type does not refer to a real PIN of the card, but is used to verify the value entered by the user against the content of the file EF-HIDDENKEY present on the card, as specified in 3GPP TS 31.102. If the hidden key is enabled (file EF-HIDDENKEY has a valid value), the hidden contacts are shown only after the value of this PIN has been successfully verified.

Note: The hidden key PIN type is applicable only to provisioning sessions.

When the optional TLV with an encrypted PIN1 value is present in the request, the modem ignores the clear PIN1 value present in the request and uses the encrypted value only.

The optional TLV with number of retries is not sent for a hidden key PIN type.

The optional TLV with an encrypted PIN1 is only sent after the successful verification of PIN1, when the modem supports the feature and the verification was done using a clear PIN1.

The optional TLV with the PIN key reference ID is used by the client to force a specific key reference to be used in the PIN operation. This TLV is not used in normal conditions, so the appropriate key reference can be selected by the modem.

The client can pass a token in the request to receive the result in a subsequent QMI_UIM_VERIFY_PIN_IND indication. In this case, the immediate response indicates only whether the command was received correctly for processing. The service passes the token back in the immediate response to indicate that it supports the indication. The service then passes the same token in the indication, so that the client can match it with the original request.

A client must be able to accommodate the case where the optional indication is not supported by the service and the return of the complete result in the response, even when the request indicates that the indication is to be used.

Other error codes not listed in the preceding table are handled as QMI_ERR_INTERNAL.

11.2.7. QMI_UIM_EVENT_REG

Registers for event notifications from the card.

- **UIM message ID**

0x002E

- **Version introduced**

Major - 1, Minor - 0

11.2.7.1. Request - QMI_UIM_EVENT_REG_REQ

- **Message type**

Request

- **Sender**

Control point

- Mandatory TLVs

Name	Version introduced	Version last modified
Event Mask	1.0	1.54

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Event Mask
Length	4			2	
Value	→	uint32	event_mask	4	<p>Bitmask of the events to be received by the client. Additional bits are reserved for future use and must be set to zero by the caller.</p> <ul style="list-style-type: none"> • Bit 0 – Card status • Bit 1 – SAP connection • Bit 2 – Extended card status • Bit 3 – Close of provisioning sessions • Bit 4 – Physical slot status • Bit 5 – SIM busy status • Bit 6 – Reduced card status • Bit 7 – Recovery complete • Bit 8 – Supply voltage Vcc status • Bit 9 – Card activation status • Bit 10 – Remote simlock configuration • Bit 11 – Simlock temporary unlock status

- Optional TLVs

None

11.2.7.2. Response - QMI_UIM_EVENT_REG_RESP

- Message type

Response

- Sender

Service

- Mandatory TLVs

The Result Code TLV (defined in Section 11.1.3.1) is always present in the response

Name	Version introduced	Version last modified
Result Code	1.0	1.0

Optional TLVs

Name	Version introduced	Version last modified
Event Registration Result	1.0	1.54

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Event Registration Result
Length	4			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	uint32	event_mask	4	<p>Bitmask of the events that were successfully enabled. This result can be different from the mask used in the request when notifications are not supported. Additional bits are reserved for future use and can be ignored by the caller.</p> <ul style="list-style-type: none"> • Bit 0 – Card status • Bit 1 – SAP connection • Bit 2 – Extended card status • Bit 3 – Close of provisioning sessions • Bit 4 – Physical slot status • Bit 5 – SIM busy status • Bit 6 – Reduced card status • Bit 7 – Recovery complete • Bit 8 – Supply voltage Vcc status • Bit 9 – Card activation status • Bit 10 – Remote simlock configuration • Bit 11 – Simlock temporary unlock status

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 the response.
QMI_ERR_NO_EFFECT	Operation had no effect.

11.2.7.3. Description of QMI_UIM_EVENT_REG REQ/RESP

This function registers for event notifications from the card. The client must verify the mask in the response to determine which events were registered successfully. Events not supported correctly are not registered.

The client can deregister from all event notifications by passing an empty bitmask in the request.

A client can register for both the legacy and the extended card status. In this case, the subsequent indication contains both.

A client can also register for reduced legacy and/or extended card status indications to prevent receiving transient state indications during card initialization.

Other error codes not listed in the preceding table are handled as QMI_ERR_INTERNAL.

11.2.8. QMI_UIM_GET_CARD_STATUS

Retrieves the current status of the card.

- **UIM message ID**

0x002F

- **Version introduced**

Major - 1, Minor - 0

11.2.8.1. Request - QMI_UIM_GET_CARD_STATUS_REQ

- **Message type**

Request

- **Sender**

Control point

- Mandatory TLVs

Name	Version introduced	Version last modified
Extended Card Status	1.30	1.30

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Extended Card Status
Length	1			2	
Value	→	boolean	extended_card_status	1	<p>Indicates whether the extended card status is requested. Valid values:</p> <ul style="list-style-type: none"> • 0 – Legacy card status (up to 2 cards) • 1 – Extended card status <p>If the Extended Card Status TLV is missing, the legacy card status with support for only 2 cards is returned.</p>

- Optional TLVs

None

11.2.8.2. Response - QMI_UIM_GET_CARD_STATUS_RESP

- Message type

Response

- Sender

Service

- Mandatory TLVs

The Result Code TLV (defined in Section 11.1.3.1) is always present in the response.

Name	Version introduced	Version last modified
Result Code	1.0	1.0

Optional TLVs

Name	Version introduced	Version last modified
Card Status	1.0	1.37
Hot-Swap Status	1.13	1.14
Valid Card Status	1.23	1.23
Extended Card Status	1.30	1.37
Extended Hot-Swap Status	1.30	1.30
SIM Busy Status	1.35	1.35

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Card Status
Length	Var			2	
Value	→	uint16	index_gw_pri	2	Index of the primary GW provisioning application. The most significant byte indicates the slot (starting from 0), while the least significant byte indicates the application for that slot (starting from 0). The value 0xFFFF identifies that the session does not exist.
		uint16	index_1x_pri	2	Index of the primary 1X provisioning application. The most significant byte indicates the slot (starting from 0), while the least significant byte indicates the application for that slot (starting from 0). The value 0xFFFF identifies that the session does not exist.

Field	Field value	Field type	Parameter	Size (byte)	Description
		uint16	index_gw_sec	2	Index of the secondary GW provisioning application. The most significant byte indicates the slot (starting from 0), while the least significant byte indicates the application for that slot (starting from 0). The value 0xFFFF identifies that the session does not exist.
		uint16	index_1x_sec	2	Index of the secondary 1X provisioning application. The most significant byte indicates the slot (starting from 0), while the least significant byte indicates the application for that slot (starting from 0). The value 0xFFFF identifies that the session does not exist.
		uint8	num_slot	1	Indicates the number of slots available on the device. The following block is repeated for each slot.
		enum8	card_state	1	Indicates the state of the card. Valid values: <ul style="list-style-type: none">• UIM_CARD_STATE_ABSENT (0x00) – Absent• UIM_CARD_STATE_PRESENT (0x01) – Present• UIM_CARD_STATE_ERROR (0x02) – Error
		enum8	upin_state	1	Indicates the state of UPIN. Valid values: <ul style="list-style-type: none">• UIM_PIN_STATE_UNKNOWN (0x00) – Unknown• UIM_PIN_STATE_ENABLED_NOT_VERIFIED (0x01) – Enabled and not verified• UIM_PIN_STATE_ENABLED_VERIFIED (0x02) – Enabled and verified• UIM_PIN_STATE_DISABLED (0x03) – Disabled• UIM_PIN_STATE_BLOCKED (0x04) – Blocked• UIM_PIN_STATE_PERMANENTLY_BLOCKED (0x05) – Permanently blocked
		uint8	upin_retries	1	Indicates the number of retries remaining to verify the UPIN.
		uint8	upuk_retries	1	Indicates the number of retries remaining to unblock the UPIN.

Field	Field value	Field type	Parameter	Size (byte)	Description
		enum8	error_code	1	<p>Indicates the reason for the card error, and is valid only when the card state is Error. Valid values:</p> <ul style="list-style-type: none"> • UIM_CARD_ERROR_CODE_UNKNOWN (0x00) – Unknown • UIM_CARD_ERROR_CODE_POWER_DOWN (0x01) – Power down • UIM_CARD_ERROR_CODE_POLL_ERROR (0x02) – Poll error • UIM_CARD_ERROR_CODE_NO_ATR_RECEIVED (0x03) – No ATR received • UIM_CARD_ERROR_CODE_VOLT_MISMATCH (0x04) – Volt mismatch • UIM_CARD_ERROR_CODE_PARITY_ERROR (0x05) – Parity error • UIM_CARD_ERROR_CODE_POSSIBLY_REMOVED (0x06) – Unknown, possibly removed • UIM_CARD_ERROR_CODE_SIM_TECHNICAL_PROBLEMS (0x07) – Card returned technical problems • UIM_CARD_ERROR_CODE_NULL_BYTES (0x08) – Card returned NULL bytes • UIM_CARD_ERROR_CODE_SAP_CONNECTED (0x09) – Terminal in SAP mode <p>Other values are possible and reserved for future use. When an unknown value is received, it is to be handled as “Unknown”.</p>

Field	Field value	Field type	Parameter	Size (byte)	Description
	uint8		num_app	1	<p>Number of sets of the following elements:</p> <ul style="list-style-type: none"> • app_type • app_state • perso_state • perso_feature • perso_retries • perso_unblock_retries • aid_len • aid_value • univ_pin • pin_state • pin_retries • puk_retries • pin_state • pin_retries • puk_retries
					<p>Indicates the type of the application. Valid values:</p> <ul style="list-style-type: none"> • UIM_APP_TYPE_UNKNOWN (0x00) – Unknown • UIM_APP_TYPE_SIM (0x01) – SIM card • UIM_APP_TYPE_USIM (0x02) – USIM application • UIM_APP_TYPE_RUIM (0x03) – RUIM card • UIM_APP_TYPE_CSIM (0x04) – CSIM application • UIM_APP_TYPE_ISIM (0x05) – ISIM application <p>Other values are reserved for the future and are to be handled as “Unknown”.</p>

Field	Field value	Field type	Parameter	Size (byte)	Description
		enum8	app_state	1	<p>Indicates the state of the application.</p> <p>Valid values:</p> <ul style="list-style-type: none"> • UIM_APP_STATE_UNKNOWN (0x00) – Unknown • UIM_APP_STATE_DETECTED (0x01) – Detected • UIM_APP_STATE_PIN1_OR_UPIN_REQ (0x02) – PIN1 or UPIN is required • UIM_APP_STATE_PUK1_OR_PUK_REQ (0x03) – PUK1 or PUK for UPIN is required • UIM_APP_STATE_PERSON_CHECK_REQ (0x04) – Personalization state must be checked • UIM_APP_STATE_PIN1_PERM_BLOCKED (0x05) – PIN1 is blocked • UIM_APP_STATE_ILLEGAL (0x06) – Illegal • UIM_APP_STATE_READY (0x07) – Ready
		enum8	perso_state	1	<p>Indicates the state of the personalization for the application. Valid values:</p> <ul style="list-style-type: none"> • UIM_PERSO_STATE_UNKNOWN (0x00) – Unknown • UIM_PERSO_STATE_IN_PROGRESS (0x01) – Personalization operation is in progress • UIM_PERSO_STATE_READY (0x02) – Ready • UIM_PERSO_STATE_CODE_REQ (0x03) – Personalization code is required • UIM_PERSO_STATE_PUK_REQ (0x04) – PUK for personalization code is required • UIM_PERSO_STATE_PERMANENTLY_BLOCKED (0x05) – Permanently blocked

Field	Field value	Field type	Parameter	Size (byte)	Description
				1	<p>Indicates the personalization feature. This applies only when a personalization code is required to deactivate or unblock personalization. Valid values:</p> <ul style="list-style-type: none"> • UIM_PERSO_FEATURE_STATUS_GW_NETWORK (0x00) – GW network personalization • UIM_PERSO_FEATURE_STATUS_GW_NETWORK_SUBSET (0x01) – GW network subset personalization • UIM_PERSO_FEATURE_STATUS_GW_SERVICE_PROVIDER (0x02) – GW service provider personalization • UIM_PERSO_FEATURE_STATUS_GW_CORPORATE (0x03) – GW corporate personalization • UIM_PERSO_FEATURE_STATUS_GW_UIM (0x04) – GW UIM personalization • UIM_PERSO_FEATURE_STATUS_1X_NETWORK_TYPE_1 (0x05) – 1X network type 1 personalization • UIM_PERSO_FEATURE_STATUS_1X_NETWORK_TYPE_2 (0x06) – 1X network type 2 personalization • UIM_PERSO_FEATURE_STATUS_1X_HRPD (0x07) – 1X HRPD personalization • UIM_PERSO_FEATURE_STATUS_1X_SERVICE_PROVIDER (0x08) – 1X service provider personalization • UIM_PERSO_FEATURE_STATUS_1X_CORPORATE (0x09) – 1X corporate personalization • UIM_PERSO_FEATURE_STATUS_1X_RUIM (0x0A) – 1X RUIM personalization • UIM_PERSO_FEATURE_STATUS_UNKNOWN (0x0B) – Unknown
					<p>Indicates the number of retries remaining to disable the personalization. This value is set to 0xFF if the modem configuration allows unlimited retries.</p>
					<p>Indicates the number of retries remaining to unblock the personalization.</p>
					<p>Number of sets of the following elements:</p> <ul style="list-style-type: none"> • aid_value

Field	Field value	Field type	Parameter	Size (byte)	Description
		uint8	aid_value	Var	Application identifier value.
		enum8	univ_pin	1	Indicates whether UPIN replaces PIN1. Valid values: <ul style="list-style-type: none">• UIM_UNIV_PIN_PIN1_USED (0x00) – PIN1 is used• UIM_UNIV_PIN_UPIN_REPLACESPIN1 (0x01) – UPIN replaces PIN1
		enum8	pin1_state	1	Indicates the state of PIN1. Valid values: <ul style="list-style-type: none">• UIM_PIN_STATE_UNKNOWN (0x00) – Unknown• UIM_PIN_STATE_ENABLED_NOT_VERIFIED (0x01) – Enabled and not verified• UIM_PIN_STATE_ENABLED_VERIFIED (0x02) – Enabled and verified• UIM_PIN_STATE_DISABLED (0x03) – Disabled• UIM_PIN_STATE_BLOCKED (0x04) – Blocked• UIM_PIN_STATE_PERMANENTLY_BLOCKED (0x05) – Permanently blocked
		uint8	pin1_retries	1	Indicates the number of retries remaining to verify PIN1.
		uint8	puk1_retries	1	Indicates the number of retries remaining to unblock PIN1.
		enum8	pin2_state	1	Indicates the state of PIN2. Valid values: <ul style="list-style-type: none">• UIM_PIN_STATE_UNKNOWN (0x00) – Unknown• UIM_PIN_STATE_ENABLED_NOT_VERIFIED (0x01) – Enabled and not verified• UIM_PIN_STATE_ENABLED_VERIFIED (0x02) – Enabled and verified• UIM_PIN_STATE_DISABLED (0x03) – Disabled• UIM_PIN_STATE_BLOCKED (0x04) – Blocked• UIM_PIN_STATE_PERMANENTLY_BLOCKED (0x05) – Permanently blocked
		uint8	pin2_retries	1	Indicates the number of retries remaining to verify PIN2.
		uint8	puk2_retries	1	Indicates the number of retries remaining to unblock PIN2.

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x11			1	Hot-Swap Status
Length	Var			2	
Value	→	uint8	hot_swap_len	1	Number of sets of the following elements: • hot_swap
		enum8	hot_swap	Var	Indicates the status of the hot-swap switch. Valid values: • UIM_HOT_SWAP_NOT_SUPPORTED (0x00) – Hot-swap is not supported • UIM_HOT_SWAP_STATUS_NOT_SUPPORTED (0x01) – Hot-swap is supported, but the status of the switch is not supported • UIM_HOT_SWAP_STATUS_PRESENT (0x02) – Switch indicates that the card is present • UIM_HOT_SWAP_STATUS_NOT_PRESENT (0x03) – Switch indicates that the card is not present
Type	0x12			1	Valid Card Status
Length	Var			2	
Value	→	uint8	card_status_validity_len	1	Number of sets of the following elements: • card_status_validity
		boolean	card_status_validity	Var	Indicates whether the reported status of the card on the corresponding slot is valid. Valid values: • 0 – Status of the card is unknown (even when reported as absent) • 1 – Status of the card is valid
Type	0x13			1	Extended Card Status
Length	Var			2	
Value	→	uint8	index_gw_len	1	Number of sets of the following elements: • index_gw
		uint16	index_gw	Var	Index of the GW provisioning applications. The most significant byte indicates the slot (starting from 0), while the least significant byte indicates the application for that slot (starting from 0). The value 0xFFFF identifies that the session does not exist.
		uint8	index_1x_len	1	Number of sets of the following elements: • index_1x

Field	Field value	Field type	Parameter	Size (byte)	Description
		uint16	index_1x	Var	Index of the 1X provisioning applications. The most significant byte indicates the slot (starting from 0), while the least significant byte indicates the application for that slot (starting from 0). The value 0xFFFF identifies that the session does not exist.
		uint8	num_slot	1	<p>Number of sets of the following elements:</p> <ul style="list-style-type: none"> • card_state • pin_state • pin_retries • puk_retries • error_code • app_type • app_state • perso_state • perso_feature • perso_retries • perso_unblock_retries • aid_len • aid_value • univ_pin • pin_state • pin_retries • puk_retries • pin_state • pin_retries • puk_retries
	enum8	card_state		1	<p>Indicates the state of the card. Valid values:</p> <ul style="list-style-type: none"> • UIM_EXTENDED_CARD_STATE_ABSENT (0x00) – Absent • UIM_EXTENDED_CARD_STATE_PRESENT (0x01) – Present • UIM_EXTENDED_CARD_STATE_ERROR (0x02) – Error • UIM_EXTENDED_CARD_STATE_UNKNOWN (0x03) – Unknown

Field	Field value	Field type	Parameter	Size (byte)	Description
		enum8	upin_state	1	<p>Indicates the state of UPIN. Valid values:</p> <ul style="list-style-type: none"> • UIM_PIN_STATE_UNKNOWN (0x00) – Unknown • UIM_PIN_STATE_ENABLED_NOT_VERIFIED (0x01) – Enabled and not verified • UIM_PIN_STATE_ENABLED_VERIFIED (0x02) – Enabled and verified • UIM_PIN_STATE_DISABLED (0x03) – Disabled • UIM_PIN_STATE_BLOCKED (0x04) – Blocked • UIM_PIN_STATE_PERMANENTLY_BLOCKED (0x05) – Permanently blocked
		uint8	upin_retries	1	Indicates the number of retries remaining to verify this UPIN.
		uint8	upuk_retries	1	Indicates the number of retries remaining to unblock UPIN.

Field	Field value	Field type	Parameter	Size (byte)	Description
		enum8	error_code	1	<p>Indicates the reason for the card error, and is valid only when the card state is Error. Valid values:</p> <ul style="list-style-type: none"> • UIM_CARD_ERROR_CODE_UNKNOWN (0x00) – Unknown • UIM_CARD_ERROR_CODE_POWER_DOWN (0x01) – Power down • UIM_CARD_ERROR_CODE_POLL_ERROR (0x02) – Poll error • UIM_CARD_ERROR_CODE_NO_ATR RECEIVED (0x03) – No ATR received • UIM_CARD_ERROR_CODE_VOLT_MISMATCH (0x04) – Volt mismatch • UIM_CARD_ERROR_CODE_PARITY_ERROR (0x05) – Parity error • UIM_CARD_ERROR_CODE_POSSIBLY_REMOVED (0x06) – Unknown, possibly removed • UIM_CARD_ERROR_CODE_SIM_TECHNICAL_PROBLEMS (0x07) – Card returned technical problems • UIM_CARD_ERROR_CODE_NULL_BYTES (0x08) – Card returned NULL bytes • UIM_CARD_ERROR_CODE_SAP_CONNECTED (0x09) – Terminal in SAP mode <p>Other values are possible and reserved for future use. When an unknown value is received, it is to be handled as "Unknown".</p>

Field	Field value	Field type	Parameter	Size (byte)	Description
		uint8	num_app	1	<p>Number of sets of the following elements:</p> <ul style="list-style-type: none"> • app_type • app_state • perso_state • perso_feature • perso_retries • perso_unblock_retries • aid_len • aid_value • univ_pin • pin_state • pin_retries • puk_retries • pin_state • pin_retries • puk_retries
		enum8	app_type	1	<p>Indicates the type of the application. Valid values:</p> <ul style="list-style-type: none"> • UIM_APP_TYPE_UNKNOWN (0x00) – Unknown • UIM_APP_TYPE_SIM (0x01) – SIM card • UIM_APP_TYPE_USIM (0x02) – USIM application • UIM_APP_TYPE_RUIM (0x03) – RUIM card • UIM_APP_TYPE_CSIM (0x04) – CSIM application • UIM_APP_TYPE_ISIM (0x05) – ISIM application <p>Other values are reserved for the future and are to be handled as “Unknown”.</p>

Field	Field value	Field type	Parameter	Size (byte)	Description
		enum8	app_state	1	Indicates the state of the application. Valid values: <ul style="list-style-type: none">• UIM_APP_STATE_UNKNOWN (0x00) – Unknown• UIM_APP_STATE_DETECTED (0x01) – Detected• UIM_APP_STATE_PIN1_OR_UPIN_REQ (0x02) – PIN1 or UPIN is required• UIM_APP_STATE_PUK1_OR_PUK_REQ (0x03) – PUK1 or PUK for UPIN is required• UIM_APP_STATE_PERSON_CHECK_REQ (0x04) – Personalization state must be checked• UIM_APP_STATE_PIN1_PERM_BLOCKED (0x05) – PIN1 is blocked• UIM_APP_STATE_ILLEGAL (0x06) – Illegal• UIM_APP_STATE_READY (0x07) – Ready
					Indicates the state of the personalization for the application. Valid values: <ul style="list-style-type: none">• UIM_PERSO_STATE_UNKNOWN (0x00) – Unknown• UIM_PERSO_STATE_IN_PROGRESS (0x01) – Personalization operation is in progress• UIM_PERSO_STATE_READY (0x02) – Ready• UIM_PERSO_STATE_CODE_REQ (0x03) – Personalization code is required• UIM_PERSO_STATE_PUK_REQ (0x04) – PUK for personalization code is required• UIM_PERSO_STATE_PERMANENTLY_BLOCKED (0x05) – Permanently blocked

Field	Field value	Field type	Parameter	Size (byte)	Description
				1	Indicates the personalization feature. This applies only when a personalization code is required to deactivate or unblock personalization. Valid values: <ul style="list-style-type: none"> • UIM_PERSO_FEATURE_STATUS_GW_NETWORK (0x00) – GW network personalization • UIM_PERSO_FEATURE_STATUS_GW_NETWORK_SUBSET (0x01) – GW network subset personalization • UIM_PERSO_FEATURE_STATUS_GW_SERVICE_PROVIDER (0x02) – GW service provider personalization • UIM_PERSO_FEATURE_STATUS_GW_CORPORATE (0x03) – GW corporate personalization • UIM_PERSO_FEATURE_STATUS_GW_UIM (0x04) – GW UIM personalization • UIM_PERSO_FEATURE_STATUS_1X_NETWORK_TYPE_1 (0x05) – 1X network type 1 personalization • UIM_PERSO_FEATURE_STATUS_1X_NETWORK_TYPE_2 (0x06) – 1X network type 2 personalization • UIM_PERSO_FEATURE_STATUS_1X_HRPD (0x07) – 1X HRPD personalization • UIM_PERSO_FEATURE_STATUS_1X_SERVICE_PROVIDER (0x08) – 1X service provider personalization • UIM_PERSO_FEATURE_STATUS_1X_CORPORATE (0x09) – 1X corporate personalization • UIM_PERSO_FEATURE_STATUS_1X_RUIM (0x0A) – 1X RUIM personalization • UIM_PERSO_FEATURE_STATUS_UNKNOWN (0x0B) – Unknown
					Indicates the number of retries remaining to disable the personalization. This value is set to 0xFF if the modem configuration allows unlimited retries.
					Indicates the number of retries remaining to unblock the personalization.
					Number of sets of the following elements: <ul style="list-style-type: none"> • aid_value

Field	Field value	Field type	Parameter	Size (byte)	Description
		uint8	aid_value	Var	Application identifier value.
		enum8	univ_pin	1	<p>Indicates whether UPIN replaces PIN1. Valid values:</p> <ul style="list-style-type: none"> • UIM_UNIV_PIN_PIN1_USED (0x00) – PIN1 is used • UIM_UNIV_PIN_UPIN_REPLACESPIN1 (0x01) – UPIN replaces PIN1
		enum8	pin1_state	1	<p>Indicates the state of PIN1. Valid values:</p> <ul style="list-style-type: none"> • UIM_PIN_STATE_UNKNOWN (0x00) – Unknown • UIM_PIN_STATE_ENABLED_NOT_VERIFIED (0x01) – Enabled and not verified • UIM_PIN_STATE_ENABLED_VERIFIED (0x02) – Enabled and verified • UIM_PIN_STATE_DISABLED (0x03) – Disabled • UIM_PIN_STATE_BLOCKED (0x04) – Blocked • UIM_PIN_STATE_PERMANENTLY_BLOCKED (0x05) – Permanently blocked
		uint8	pin1_retries	1	Indicates the number of retries remaining to verify PIN1.
		uint8	puk1_retries	1	Indicates the number of retries remaining to unblock PIN1.
		enum8	pin2_state	1	<p>Indicates the state of PIN2. Valid values:</p> <ul style="list-style-type: none"> • UIM_PIN_STATE_UNKNOWN (0x00) – Unknown • UIM_PIN_STATE_ENABLED_NOT_VERIFIED (0x01) – Enabled and not verified • UIM_PIN_STATE_ENABLED_VERIFIED (0x02) – Enabled and verified • UIM_PIN_STATE_DISABLED (0x03) – Disabled • UIM_PIN_STATE_BLOCKED (0x04) – Blocked • UIM_PIN_STATE_PERMANENTLY_BLOCKED (0x05) – Permanently blocked
		uint8	pin2_retries	1	Indicates the number of retries remaining to verify PIN2.
		uint8	puk2_retries	1	Indicates the number of retries remaining to unblock PIN2.

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x14			1	Extended Hot-Swap Status
Length	Var			2	
Value	→	uint8	extended_hot_swap_len	1	Number of sets of the following elements: • extended_hot_swap
		enum8	extended_hot_swap	Var	Indicates the status of the hot-swap switch. Valid values: • UIM_HOT_SWAP_NOT_SUPPORTED (0x00) – Hot-swap is not supported • UIM_HOT_SWAP_STATUS_NOT_SUPPORTED (0x01) – Hot-swap is supported, but the status of the switch is not supported • UIM_HOT_SWAP_STATUS_PRESENT (0x02) – Switch indicates that the card is present • UIM_HOT_SWAP_STATUS_NOT_PRESENT (0x03) – Switch indicates that the card is not present
Type	0x15			1	SIM Busy Status
Length	Var			2	
Value	→	uint8	sim_busy_status_len	1	Number of sets of the following elements: • sim_busy_status
		boolean	sim_busy_status	Var	Indicates whether the card on the corresponding slot is busy. Valid values: • 0 – SIM card is not busy • 1 – SIM card is busy

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 the response.

11.2.8.3. Description of QMI_UIM_GET_CARD_STATUS REQ/RESP

This function retrieves the current status of the cards and the status of all applications available on the cards. The function also returns support information for the hot-swap feature and the status of the switch used to detect a card removal/insertion.

Support of the hot-swap feature is independent of the presence of the optional TLV with the hot-swap switch status.

The result of this function can be used by the client to determine the number of slots supported by the specific target.

During power-up, the status of the card can be unknown for a short period of time. An optional TLV indicates whether the reported status is valid. If the TLV is missing, the client assumes that the status of the card is valid.

The extended card status is used on targets with three cards or more. A client must be able to handle the legacy card status as a response, even if it requested the extended card status. If a session is associated with a card in slot 3 or higher, the legacy card status indicates the session as invalid.

Other error codes not listed in the preceding table are handled as QMI_ERR_INTERNAL.

11.2.9. QMI_UIM_STATUS_CHANGE_IND

Indicates that the status of the card has changed.

- **UIM message ID**

0x0032

- **Version introduced**

Major - 1, Minor - 0

11.2.9.1. Indication - QMI_UIM_STATUS_CHANGE_IND

- **Message type**

Indication

- **Sender**

Service

- **Scope**

Unicast (per control point)

- **Mandatory TLVs**

None

- Optional TLVs

Name	Version introduced	Version last modified
Card Status	1.0	1.37
Valid Card Status	1.25	1.25
Extended Card Status	1.30	1.37

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Card Status
Length	Var			2	
Value	→	uint16	index_gw_pri	2	Index of the primary GW provisioning application. The most significant byte indicates the slot (starting from 0), while the least significant byte indicates the application for that slot (starting from 0). The value 0xFFFF identifies that the session does not exist.

Field	Field value	Field type	Parameter	Size (byte)	Description
		uint16	index_1x_pri	2	Index of the primary 1X provisioning application. The most significant byte indicates the slot (starting from 0), while the least significant byte indicates the application for that slot (starting from 0). The value 0xFFFF identifies that the session does not exist.
		uint16	index_gw_sec	2	Index of the secondary GW provisioning application. The most significant byte indicates the slot (starting from 0), while the least significant byte indicates the application for that slot (starting from 0). The value 0xFFFF identifies that the session does not exist.
		uint16	index_1x_sec	2	Index of the secondary 1X provisioning application. The most significant byte indicates the slot (starting from 0), while the least significant byte indicates the application for that slot (starting from 0). The value 0xFFFF identifies that the session does not exist.
		uint8	num_slot	1	Indicates the number of slots available on the device. The following block is repeated for each slot.
		enum8	card_state	1	Indicates the state of the card. Valid values: <ul style="list-style-type: none">• UIM_CARD_STATE_ABSENT (0x00) – Absent• UIM_CARD_STATE_PRESENT (0x01) – Present• UIM_CARD_STATE_ERROR (0x02) – Error
		enum8	upin_state	1	Indicates the state of UPIN. Valid values: <ul style="list-style-type: none">• UIM_PIN_STATE_UNKNOWN (0x00) – Unknown• UIM_PIN_STATE_ENABLED_NOT_VERIFIED (0x01) – Enabled and not verified• UIM_PIN_STATE_ENABLED_VERIFIED (0x02) – Enabled and verified• UIM_PIN_STATE_DISABLED (0x03) – Disabled• UIM_PIN_STATE_BLOCKED (0x04) – Blocked• UIM_PIN_STATE_PERMANENTLY_BLOCKED (0x05) – Permanently blocked

Field	Field value	Field type	Parameter	Size (byte)	Description
		uint8	upin_retries	1	Indicates the number of retries remaining to verify UPIN.
		uint8	upuk_retries	1	Indicates the number of retries remaining to unlock UPIN.
		enum8	error_code	1	<p>Indicates the reason for the card error, and is valid only when the card state is Error. Valid values:</p> <ul style="list-style-type: none"> • UIM_CARD_ERROR_CODE_UNKNOWN (0x00) – Unknown • UIM_CARD_ERROR_CODE_POWER_DOWN (0x01) – Power down • UIM_CARD_ERROR_CODE_POLL_ERROR (0x02) – Poll error • UIM_CARD_ERROR_CODE_NO_ATR RECEIVED (0x03) – No ATR received • UIM_CARD_ERROR_CODE_VOLT_MISMATCH (0x04) – Volt mismatch • UIM_CARD_ERROR_CODE_PARITY_ERROR (0x05) – Parity error • UIM_CARD_ERROR_CODE_POSSIBLY_REMOVED (0x06) – Unknown, possibly removed • UIM_CARD_ERROR_CODE_SIM_TECHNICAL_PROBLEMS (0x07) – Card returned technical problems • UIM_CARD_ERROR_CODE_NULL_BYTES (0x08) – Card returned NULL bytes • UIM_CARD_ERROR_CODE_SAP_CONNECTED (0x09) – Terminal in SAP mode <p>Other values are possible and reserved for future use. When an unknown value is received, it is to be handled as “Unknown”.</p>

Field	Field value	Field type	Parameter	Size (byte)	Description
		uint8	num_app	1	<p>Number of sets of the following elements:</p> <ul style="list-style-type: none"> • app_type • app_state • perso_state • perso_feature • perso_retries • perso Unblock_retries • aid_len • aid_value • univ_pin • pin_state • pin_retries • puk_retries • pin_state • pin_retries • puk_retries
		enum8	app_type	1	<p>Indicates the type of the application.</p> <p>Valid values:</p> <ul style="list-style-type: none"> • UIM_APP_TYPE_UNKNOWN (0x00) – Unknown • UIM_APP_TYPE_SIM (0x01) – SIM card • UIM_APP_TYPE_USIM (0x02) – USIM application • UIM_APP_TYPE_RUIM (0x03) – RUIM card • UIM_APP_TYPE_CSIM (0x04) – CSIM application • UIM_APP_TYPE_ISIM (0x05) – ISIM application <p>Other values are reserved for the future and are to be handled as “Unknown”.</p>

Field	Field value	Field type	Parameter	Size (byte)	Description
	enum8	app_state		1	Indicates the state of the application. Valid values: <ul style="list-style-type: none">• UIM_APP_STATE_UNKNOWN (0x00) – Unknown• UIM_APP_STATE_DETECTED (0x01) – Detected• UIM_APP_STATE_PIN1_OR_UPIN_REQ (0x02) – PIN1 or UPIN is required• UIM_APP_STATE_PUK1_OR_PUK_REQ (0x03) – PUK1 or PUK for UPIN is required• UIM_APP_STATE_PERSON_CHECK_REQ (0x04) – Personalization state must be checked• UIM_APP_STATE_PIN1_PERM_BLOCKED (0x05) – PIN1 is blocked• UIM_APP_STATE_ILLEGAL (0x06) – Illegal• UIM_APP_STATE_READY (0x07) – Ready
					Indicates the state of the personalization for the application. Valid values: <ul style="list-style-type: none">• UIM_PERSO_STATE_UNKNOWN (0x00) – Unknown• UIM_PERSO_STATE_IN_PROGRESS (0x01) – Personalization operation is in progress• UIM_PERSO_STATE_READY (0x02) – Ready• UIM_PERSO_STATE_CODE_REQ (0x03) – Personalization code is required• UIM_PERSO_STATE_PUK_REQ (0x04) – PUK for personalization code is required• UIM_PERSO_STATE_PERMANENTLY_BLOCKED (0x05) – Permanently blocked

Field	Field value	Field type	Parameter	Size (byte)	Description
		enum8	perso_feature	1	Indicates the personalization feature. This applies only when a personalization code is required to deactivate or unblock personalization. Valid values: <ul style="list-style-type: none"> • UIM_PERSO_FEATURE_STATUS_GW_NETWORK (0x00) – GW network personalization • UIM_PERSO_FEATURE_STATUS_GW_NETWORK_SUBSET (0x01) – GW network subset personalization • UIM_PERSO_FEATURE_STATUS_GW_SERVICE_PROVIDER (0x02) – GW service provider personalization • UIM_PERSO_FEATURE_STATUS_GW_CORPORATE (0x03) – GW corporate personalization • UIM_PERSO_FEATURE_STATUS_GW_UIM (0x04) – GW UIM personalization • UIM_PERSO_FEATURE_STATUS_1X_NETWORK_TYPE_1 (0x05) – 1X network type 1 personalization • UIM_PERSO_FEATURE_STATUS_1X_NETWORK_TYPE_2 (0x06) – 1X network type 2 personalization • UIM_PERSO_FEATURE_STATUS_1X_HRPD (0x07) – 1X HRPD personalization • UIM_PERSO_FEATURE_STATUS_1X_SERVICE_PROVIDER (0x08) – 1X service provider personalization • UIM_PERSO_FEATURE_STATUS_1X_CORPORATE (0x09) – 1X corporate personalization • UIM_PERSO_FEATURE_STATUS_1X_RUIM (0x0A) – 1X RUIM personalization • UIM_PERSO_FEATURE_STATUS_UNKNOWN (0x0B) – Unknown
					Indicates the number of retries remaining to disable the personalization. This value is set to 0xFF if the modem configuration allows unlimited retries.
					Indicates the number of retries remaining to unblock the personalization.
					Number of sets of the following elements: <ul style="list-style-type: none"> • aid_value

Field	Field value	Field type	Parameter	Size (byte)	Description
		uint8	aid_value	Var	Application identifier value.
		enum8	univ_pin	1	Indicates whether UPIN replaces PIN1. Valid values: <ul style="list-style-type: none">• UIM_UNIV_PIN_PIN1_USED (0x00) – PIN1 is used• UIM_UNIV_PIN_UPIN_REPLACESPIN1 (0x01) – UPIN replaces PIN1
		enum8	pin1_state	1	Indicates the state of PIN1. Valid values: <ul style="list-style-type: none">• UIM_PIN_STATE_UNKNOWN (0x00) – Unknown• UIM_PIN_STATE_ENABLED_NOT_VERIFIED (0x01) – Enabled and not verified• UIM_PIN_STATE_ENABLED_VERIFIED (0x02) – Enabled and verified• UIM_PIN_STATE_DISABLED (0x03) – Disabled• UIM_PIN_STATE_BLOCKED (0x04) – Blocked• UIM_PIN_STATE_PERMANENTLY_BLOCKED (0x05) – Permanently blocked
		uint8	pin1_retries	1	Indicates the number of retries remaining to verify PIN1.
		uint8	puk1_retries	1	Indicates the number of retries remaining to unblock PIN1.
		enum8	pin2_state	1	Indicates the state of PIN2. Valid values: <ul style="list-style-type: none">• UIM_PIN_STATE_UNKNOWN (0x00) – Unknown• UIM_PIN_STATE_ENABLED_NOT_VERIFIED (0x01) – Enabled and not verified• UIM_PIN_STATE_ENABLED_VERIFIED (0x02) – Enabled and verified• UIM_PIN_STATE_DISABLED (0x03) – Disabled• UIM_PIN_STATE_BLOCKED (0x04) – Blocked• UIM_PIN_STATE_PERMANENTLY_BLOCKED (0x05) – Permanently blocked
		uint8	pin2_retries	1	Indicates the number of retries remaining to verify PIN2.
		uint8	puk2_retries	1	Indicates the number of retries remaining to unblock PIN2.

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x11			1	Valid Card Status
Length	Var			2	
Value	→	uint8	card_status_validity_len	1	Number of sets of the following elements: • card_status_validity
		boolean	card_status_validity	Var	Indicates whether the reported status of the card on the corresponding slot is valid. Valid values: • 0 – Status of the card is unknown (even if reported as absent) • 1 – Status of the card is valid
Type	0x12			1	Extended Card Status
Length	Var			2	
Value	→	uint8	index_gw_len	1	Number of sets of the following elements: • index_gw
		uint16	index_gw	Var	Index of the GW provisioning applications. The most significant byte indicates the slot (starting from 0), while the least significant byte indicates the application for that slot (starting from 0). The value 0xFFFF identifies that the session does not exist.
		uint8	index_1x_len	1	Number of sets of the following elements: • index_1x
		uint16	index_1x	Var	Index of the 1X provisioning applications. The most significant byte indicates the slot (starting from 0), while the least significant byte indicates the application for that slot (starting from 0). The value 0xFFFF identifies that the session does not exist.

Field	Field value	Field type	Parameter	Size (byte)	Description
		uint8	num_slot	1	<p>Number of sets of the following elements:</p> <ul style="list-style-type: none"> • card_state • pin_state • pin_retries • puk_retries • error_code • app_type • app_state • perso_state • perso_feature • perso_retries • perso_unblock_retries • aid_len • aid_value • univ_pin • pin_state • pin_retries • puk_retries • pin_state • pin_retries • puk_retries
		enum8	card_state	1	<p>Indicates the state of the card. Valid values:</p> <ul style="list-style-type: none"> • UIM_EXTENDED_CARD_STATE_ABSENT (0x00) – Absent • UIM_EXTENDED_CARD_STATE_PRESENT (0x01) – Present • UIM_EXTENDED_CARD_STATE_ERROR (0x02) – Error • UIM_EXTENDED_CARD_STATE_UNKNOWN (0x03) – Unknown
		enum8	upin_state	1	<p>Indicates the state of UPIN. Valid values:</p> <ul style="list-style-type: none"> • UIM_PIN_STATE_UNKNOWN (0x00) – Unknown • UIM_PIN_STATE_ENABLED_NOT_VERIFIED (0x01) – Enabled and not verified • UIM_PIN_STATE_ENABLED_VERIFIED (0x02) – Enabled and verified • UIM_PIN_STATE_DISABLED (0x03) – Disabled • UIM_PIN_STATE_BLOCKED (0x04) – Blocked • UIM_PIN_STATE_PERMANENTLY_BLOCKED (0x05) – Permanently blocked

Field	Field value	Field type	Parameter	Size (byte)	Description
		uint8	upin_retries	1	Indicates the number of retries remaining to verify UPIN.
		uint8	upuk_retries	1	Indicates the number of retries remaining to unblock UPIN.
		enum8	error_code	1	<p>Indicates the reason for the card error, and is valid only when the card state is Error. Valid values:</p> <ul style="list-style-type: none"> • UIM_CARD_ERROR_CODE_UNKNOWN (0x00) – Unknown • UIM_CARD_ERROR_CODE_POWER_DOWN (0x01) – Power down • UIM_CARD_ERROR_CODE_POLL_ERROR (0x02) – Poll error • UIM_CARD_ERROR_CODE_NO_ATR RECEIVED (0x03) – No ATR received • UIM_CARD_ERROR_CODE_VOLT_MISMATCH (0x04) – Volt mismatch • UIM_CARD_ERROR_CODE_PARITY_ERROR (0x05) – Parity error • UIM_CARD_ERROR_CODE_POSSIBLY_REMOVED (0x06) – Unknown, possibly removed • UIM_CARD_ERROR_CODE_SIM_TECHNICAL_PROBLEMS (0x07) – Card returned technical problems • UIM_CARD_ERROR_CODE_NULL_BYTES (0x08) – Card returned NULL bytes • UIM_CARD_ERROR_CODE_SAP_CONNECTED (0x09) – Terminal in SAP mode <p>Other values are possible and reserved for future use. When an unknown value is received, it is to be handled as “Unknown”.</p>

Field	Field value	Field type	Parameter	Size (byte)	Description
		uint8	num_app	1	<p>Number of sets of the following elements:</p> <ul style="list-style-type: none"> • app_type • app_state • perso_state • perso_feature • perso_retries • perso_unblock_retries • aid_len • aid_value • univ_pin • pin_state • pin_retries • puk_retries • pin_state • pin_retries • puk_retries
		enum8	app_type	1	<p>Indicates the type of the application. Valid values:</p> <ul style="list-style-type: none"> • UIM_APP_TYPE_UNKNOWN (0x00) – Unknown • UIM_APP_TYPE_SIM (0x01) – SIM card • UIM_APP_TYPE_USIM (0x02) – USIM application • UIM_APP_TYPE_RUIM (0x03) – RUIM card • UIM_APP_TYPE_CSIM (0x04) – CSIM application • UIM_APP_TYPE_ISIM (0x05) – ISIM application <p>Other values are reserved for the future and are to be handled as “Unknown”.</p>

Field	Field value	Field type	Parameter	Size (byte)	Description
	enum8		app_state	1	<p>Indicates the state of the application.</p> <p>Valid values:</p> <ul style="list-style-type: none"> • UIM_APP_STATE_UNKNOWN (0x00) – Unknown • UIM_APP_STATE_DETECTED (0x01) – Detected • UIM_APP_STATE_PIN1_OR_UPIN_REQ (0x02) – PIN1 or UPIN is required • UIM_APP_STATE_PUK1_OR_PUK_REQ (0x03) – PUK1 or PUK for UPIN is required • UIM_APP_STATE_PERSON_CHECK_REQ (0x04) – Personalization state must be checked • UIM_APP_STATE_PIN1_PERM_BLOCKED (0x05) – PIN1 is blocked • UIM_APP_STATE_ILLEGAL (0x06) – Illegal • UIM_APP_STATE_READY (0x07) – Ready
			perso_state	1	<p>Indicates the state of the personalization for the application. Valid values:</p> <ul style="list-style-type: none"> • UIM_PERSO_STATE_UNKNOWN (0x00) – Unknown • UIM_PERSO_STATE_IN_PROGRESS (0x01) – Personalization operation is in progress • UIM_PERSO_STATE_READY (0x02) – Ready • UIM_PERSO_STATE_CODE_REQ (0x03) – Personalization code is required • UIM_PERSO_STATE_PUK_REQ (0x04) – PUK for personalization code is required • UIM_PERSO_STATE_PERMANENTLY_BLOCKED (0x05) – Permanently blocked

Field	Field value	Field type	Parameter	Size (byte)	Description
		enum8	perso_feature	1	<p>Indicates the personalization feature. This applies only when a personalization code is required to deactivate or unblock personalization. Valid values:</p> <ul style="list-style-type: none"> • UIM_PERSO_FEATURE_STATUS_GW_NETWORK (0x00) – GW network personalization • UIM_PERSO_FEATURE_STATUS_GW_NETWORK_SUBSET (0x01) – GW network subset personalization • UIM_PERSO_FEATURE_STATUS_GW_SERVICE_PROVIDER (0x02) – GW service provider personalization • UIM_PERSO_FEATURE_STATUS_GW_CORPORATE (0x03) – GW corporate personalization • UIM_PERSO_FEATURE_STATUS_GW_UIM (0x04) – GW UIM personalization • UIM_PERSO_FEATURE_STATUS_1X_NETWORK_TYPE_1 (0x05) – 1X network type 1 personalization • UIM_PERSO_FEATURE_STATUS_1X_NETWORK_TYPE_2 (0x06) – 1X network type 2 personalization • UIM_PERSO_FEATURE_STATUS_1X_HRPD (0x07) – 1X HRPD personalization • UIM_PERSO_FEATURE_STATUS_1X_SERVICE_PROVIDER (0x08) – 1X service provider personalization • UIM_PERSO_FEATURE_STATUS_1X_CORPORATE (0x09) – 1X corporate personalization • UIM_PERSO_FEATURE_STATUS_1X_RUIM (0x0A) – 1X RUIM personalization • UIM_PERSO_FEATURE_STATUS_UNKNOWN (0x0B) – Unknown
		uint8	perso_retries	1	<p>Indicates the number of retries remaining to disable the personalization.</p> <p>This value is set to 0xFF if the modem configuration allows unlimited retries.</p>
		uint8	perso_unblock_retries	1	Indicates the number of retries remaining to unblock the personalization.
		uint8	aid_len	1	<p>Number of sets of the following elements:</p> <ul style="list-style-type: none"> • aid_value

Field	Field value	Field type	Parameter	Size (byte)	Description
		uint8	aid_value	Var	Application identifier value.
		enum8	univ_pin	1	Indicates whether UPIN replaces PIN1. Valid values: <ul style="list-style-type: none">• UIM_UNIV_PIN_PIN1_USED (0x00) – PIN1 is used• UIM_UNIV_PIN_UPIN_REPLACESPIN1 (0x01) – UPIN replaces PIN1
		enum8	pin1_state	1	Indicates the state of PIN1. Valid values: <ul style="list-style-type: none">• UIM_PIN_STATE_UNKNOWN (0x00) – Unknown• UIM_PIN_STATE_ENABLED_NOT_VERIFIED (0x01) – Enabled and not verified• UIM_PIN_STATE_ENABLED_VERIFIED (0x02) – Enabled and verified• UIM_PIN_STATE_DISABLED (0x03) – Disabled• UIM_PIN_STATE_BLOCKED (0x04) – Blocked• UIM_PIN_STATE_PERMANENTLY_BLOCKED (0x05) – Permanently blocked
		uint8	pin1_retries	1	Indicates the number of retries remaining to verify PIN1.
		uint8	puk1_retries	1	Indicates the number of retries remaining to unblock PIN1.
		enum8	pin2_state	1	Indicates the state of PIN2. Valid values: <ul style="list-style-type: none">• UIM_PIN_STATE_UNKNOWN (0x00) – Unknown• UIM_PIN_STATE_ENABLED_NOT_VERIFIED (0x01) – Enabled and not verified• UIM_PIN_STATE_ENABLED_VERIFIED (0x02) – Enabled and verified• UIM_PIN_STATE_DISABLED (0x03) – Disabled• UIM_PIN_STATE_BLOCKED (0x04) – Blocked• UIM_PIN_STATE_PERMANENTLY_BLOCKED (0x05) – Permanently blocked
		uint8	pin2_retries	1	Indicates the number of retries remaining to verify PIN2.
		uint8	puk2_retries	1	Indicates the number of retries remaining to unblock PIN2.

11.2.9.2. Description of QMI_UIM_STATUS_CHANGE_IND

This event indicates that the status of the card has changed. The event does not indicate what changed from the previous event. The application is responsible for this.

The indication might not have the TLV with the card status. In this case, the application can retrieve the current card status using the QMI_UIM_GET_CARD_STATUS command.

During power-up, it is possible that the status of the card is unknown for a short period of time. The Valid Card Status optional TLV indicates whether the reported status is valid. If the TLV is missing, the client assumes that the status of the card is valid.

The event is sent to the application only when the application has registered for notifications using the QMI_UIM_EVENT_REG command.

12 Wireless Data Administrative Service (QMI_WDA)

This specification documents the Qualcomm Messaging Interface (QMI) for the Wireless Data Administrative Service (QMI_WDA).

QMI_WDA provides a command set to interface to a wireless mobile station to access wireless data administrative services.

QMI_WDA is a QMI native service, conforming to the generalized behavior defined for QMI services. The QMI_WDA service supports only one client per QMI control channel.

12.1. Theory of Operation

12.1.1. Generalized QMI Service Compliance

The QMI_WDA 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 80-VB816-1. Extensions to the generalized QMI service theory of operation are noted in subsequent sections of this chapter.

12.1.2. WDA Service Type

WDA is assigned QMI service type 0x1A.

12.1.3. Message Definition Template

12.1.3.1. Response Message Result TLV

This Type-Length-Value (TLV) is present in all Response messages defined in this document. It is not present in the Indication messages.

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

12.1.4. QMI_WDA Fundamental Concepts

The QMI_WDA service supports only one control point per QMI control channel. The service allows the control point to perform administrative operations, such as configuring the data format to be used by a data session negotiated on the channel.

For more information about the QOS flow header format referred to in this document, refer to 80-VT270-1.

12.1.5. Service State Variables

12.1.5.1. Shared State Variables

No QMI_WDA state variables are shared across control points.

12.2. QMI_WDA Messages

Command	ID	Description
QMI_WDA_SET_DATA_FORMAT	0x0020	Indicates to the service the data format used by the client. Indicates to the service the data format used by the client.
QMI_WDA_GET_DATA_FORMAT	0x0021	Gets the current data format settings of the client.

12.2.1. QMI_WDA_SET_DATA_FORMAT

Indicates to the service the data format used by the client.

- **WDA message ID**

0x0020

- **Version introduced**

Major - 1, Minor - 0

12.2.1.1. Request - QMI_WDA_SET_DATA_FORMAT_REQ

- **Message type**

Request

- **Sender**

Control point

- **Mandatory TLVs**

None

- Optional TLVs

Name	Version introduced	Version last modified
QOS Data Format	1.0	1.0
Underlying Link Layer Protocol	1.0	1.0
Uplink Data Aggregation Protocol	1.0	1.9
Downlink Data Aggregation Protocol	1.0	1.9
NDP Signature	1.0	1.0
Downlink Data Aggregation Max Datagrams	1.2	1.2
Downlink Data Aggregation Max Size	1.2	1.2
Peripheral End Point ID	1.13	1.13
QOS Header Format	1.14	1.14
QMAP Downlink Minimum Padding	1.15	1.15
Flow Control	1.16	1.16

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	QOS Data Format
Length	1			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	boolean	qos_format	1	Indicates whether the Quality of Service (QOS) data format is used by the client. Values: <ul style="list-style-type: none">• 0 – QOS flow header is not present (Default)• 1 – QOS flow header is present
Type	0x11			1	Underlying Link Layer Protocol
Length	4			2	
Value	→	enum	link_prot	4	Link protocol used by the client: <ul style="list-style-type: none">• 0x01 – 802.3 Ethernet mode (Default)• 0x02 – IP mode
Type	0x12			1	Uplink Data Aggregation Protocol
Length	4			2	
Value	→	enum	ul_data_aggregation_protocol	4	Uplink (UL) data aggregation protocol to be used for uplink data transfer. Values: <ul style="list-style-type: none">• 0x00 – UL data aggregation is disabled (Default)• 0x01 – UL TLP is enabled• 0x02 – UL QC_NCM is enabled• 0x03 – UL MBIM is enabled• 0x04 – UL RNDIS is enabled• 0x05 – UL QMAP is enabled
Type	0x13			1	Downlink Data Aggregation Protocol
Length	4			2	
Value	→	enum	dl_data_aggregation_protocol	4	Downlink (DL) data aggregation protocol to be used for downlink data transfer. Values: <ul style="list-style-type: none">• 0x00 – DL data aggregation is disabled (Default)• 0x01 – DL TLP is enabled• 0x02 – DL QC_NCM is enabled• 0x03 – DL MBIM is enabled• 0x04 – DL RNDIS is enabled• 0x05 – DL QMAP is enabled
Type	0x14			1	NDP Signature
Length	4			2	
Value	→	uint32	ndp_signature	4	NCM Datagram Pointers (NDP) signature.
Type	0x15			1	Downlink Data Aggregation Max Datagrams
Length	4			2	
Value	→	uint32	dl_data_aggregation_max_datagrams	4	Maximum number of datagrams in a single aggregated packet on downlink. The value applies to all downlink data aggregation protocols when downlink data aggregation is enabled. Zero means no limit.

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x16			1	Downlink Data Aggregation Max Size
Length	4			2	
Value	→	uint32	dl_data_aggregation_max_size	4	Maximum size in bytes of a single aggregated packet allowed on downlink. The value applies to all downlink data aggregation protocols when downlink data aggregation is enabled.
Type	0x17			1	Peripheral End Point ID (Peripheral end point on which the data format is set. Default value is the default data channel associated with the QMI control channel from which the request is received.)
Length	8			2	
Value	→	enum	ep_type	4	Peripheral end point type. Values: <ul style="list-style-type: none">• DATA_EP_TYPE_RESERVED (0x00) – Reserved• DATA_EP_TYPE_HSIC (0x01) – HSIC• DATA_EP_TYPE_HSUSB (0x02) – HSUSB• DATA_EP_TYPE_PCIE (0x03) – PCIe• DATA_EP_TYPE_EMBEDDED (0x04) – Embedded All other values are reserved and will be ignored by service or clients.
		uint32	iface_id	4	Peripheral interface number.
Type	0x18			1	QOS Header Format
Length	4			2	
Value	→	enum	qos_header_format	4	QOS header format to be used on the uplink, on all the protocols, if QOS is enabled. Values: <ul style="list-style-type: none">• WDA_QOS_HDR_FORMAT_RESERVED (0x00) – Reserved• WDA_QOS_HDR_FORMAT_6_BYTE (0x01) – QOS 6 byte Default header• WDA_QOS_HDR_FORMAT_8_BYTE (0x02) – QOS 8 byte header
Type	0x19			1	QMAP Downlink Minimum Padding
Length	4			2	
Value	→	uint32	dl_minimum_padding	4	Specifies the minimum padding bytes to be added in between aggregated downlink QMAP packets. Valid values: 0 to 64 bytes; must be 4-byte aligned. Default is 0.
Type	0x1A			1	Flow Control

12.2.1.2. Response - QMI_WDA_SET_DATA_FORMAT_RESP

- **Message type**

Response

- **Sender**

Service

- **Mandatory TLVs**

The Result Code TLV (defined in Section 12.2.3.1) is always present in the response.

- **Optional TLVs**

Name	Version introduced	Version last modified
Configured QOS Data Format	1.0	1.0
Underlying Link Layer Protocol	1.0	1.0
Uplink Data Aggregation Protocol	1.0	1.9
Downlink Data Aggregation Protocol	1.0	1.9
NDP Signature	1.0	1.0
Downlink Data Aggregation Max Datagrams	1.2	1.2
Downlink Data Aggregation Max Size	1.2	1.2
Uplink Data Aggregation Max Datagrams	1.11	1.11
Uplink Data Aggregation Max Size	1.11	1.11
QOS Header Format	1.14	1.14
QMAP Downlink Minimum Padding	1.15	1.15
Flow Control	1.16	1.16

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Configured QOS Data Format
Length	1			2	
Value	→	boolean	qos_format	1	Configured QOS data format. Values: • 0 – QOS flow header is not present (Default) • 1 – QOS flow header is present

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x11			1	Underlying Link Layer Protocol
Length	4			2	
Value	→	enum	link_prot	4	Configured link layer protocol. Values: • 0x01 – 802.3 Ethernet mode (Default) • 0x02 – IP mode
Type	0x12			1	Uplink Data Aggregation Protocol
Length	4			2	
Value	→	enum	ul_data_aggregation_protocol	4	Configured uplink data aggregation protocol. Values: • 0x00 – UL data aggregation is disabled (Default) • 0x01 – UL TLP is enabled • 0x02 – UL QC_NCM is enabled • 0x03 – UL MBIM is enabled • 0x04 – UL RNDIS is enabled • 0x05 – UL QMAP is enabled
Type	0x13			1	Downlink Data Aggregation Protocol
Length	4			2	
Value	→	enum	dl_data_aggregation_protocol	4	Configured downlink data aggregation protocol. Values: • 0x00 – DL data aggregation is disabled (Default) • 0x01 – DL TLP is enabled • 0x02 – DL QC_NCM is enabled • 0x03 – DL MBIM is enabled • 0x04 – DL RNDIS is enabled • 0x05 – DL QMAP is enabled
Type	0x14			1	NDP Signature
Length	4			2	
Value	→	uint32	ndp_signature	4	NDP signature. The default value based on the data aggregation protocol is used.
Type	0x15			1	Downlink Data Aggregation Max Datagrams
Length	4			2	
Value	→	uint32	dl_data_aggregation_max_datagrams	4	Maximum number of datagrams in a single aggregated packet on downlink. The value applies to all downlink data aggregation protocols when downlink data aggregation is enabled. Zero means no limit.
Type	0x16			1	Downlink Data Aggregation Max Size
Length	4			2	
Value	→	uint32	dl_data_aggregation_max_size	4	Maximum size in bytes of a single aggregated packet allowed on downlink. The value applies to all downlink data aggregation protocols when downlink data aggregation is enabled.

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x17			1	Uplink Data Aggregation Max Datagrams
Length	4			2	
Value	→	uint32	ul_data_aggregation_max_datagrams	4	Maximum number of datagrams supported in the modem in a single aggregated packet on uplink for the currently configured aggregation mode. Zero means there is no limit.
Type	0x18			1	Uplink Data Aggregation Max Size
Length	4			2	
Value	→	uint32	ul_data_aggregation_max_size	4	Maximum size in bytes of a single aggregated packet allowed on uplink.
Type	0x19			1	QOS Header Format
Length	4			2	
Value	→	enum	qos_header_format	4	Configured QOS header format used on the uplink, on all the protocols, if QOS is enabled. Values: <ul style="list-style-type: none">• WDA_QOS_HDR_FORMAT_RESERVED (0x00) – Reserved• WDA_QOS_HDR_FORMAT_6_BYTE (0x01) – QOS 6 byte Default header• WDA_QOS_HDR_FORMAT_8_BYTE (0x02) – QOS 8 byte header
Type	0x1A			1	QMAP Downlink Minimum Padding
Length	4			2	
Value	→	uint32	dl_minimum_padding	4	Indicates the minimum padding bytes added in between aggregated downlink QMAP packets.
Type	0x1B			1	Flow Control
Length	1			2	
Value	→	boolean	flow_control	1	Configured flow control. Values: <ul style="list-style-type: none">• 0 – Flow control will not be done by the TE (Default)• 1 – Flow control will be done by the TE

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	Service could not allocate memory to formulate a response
QMI_ERR_INVALID_OPERATION	Invalid operation was performed by the client

12.2.1.3. Description of QMI_WDA_SET_DATA_FORMAT REQ/RESP

When the data format in the QOS Data Format TLV is set to 1, the client must add a 6 or 8 byte QOS flow header to the start of the data packet. This header is useful only if you want to access QOS when the QMI_QOS service is available. In the absence of QMI_QOS, the client is not to use this TLV. If the client requests QOS and QMI_QOS service is not available, the service returns the Configured QOS Data Format TLV set to 0 in the response message. If the TE wishes to perform flow control but does not want to add the QOS flow header, the QOS Data Format TLV must be either missing or set to 0 but the flow_control TLV must be sent and set to 1. The flow_control TLV will be ignored if non zero qos_format TLV is present.

The default underlying link layer protocol is 802.3. To change the protocol to use another mode, such as Raw IP, the optional Underlying Link Layer Protocol TLV in the request must be used to specify the mode supported by the client. The service then chooses the protocol and uses the optional Underlying Link Layer Protocol TLV in the response to indicate the mode to be used by the client.

The default uplink data aggregation protocol setting is disabled by default on the service. The protocol can be set to enable Thin Layer Protocol (TLP, described in 80-VK053-1), QC NCM transfer block (NTB), Mobile Broadband Interface Model (MBIM), Remote Network Driver Interface Specification (RNDIS), or Qualcomm Mux and Aggregation Protocol (QMAP). To change the setting, the optional Uplink Data Aggregation Protocol TLV must be included in the request. The service then uses the configured optional TLV in the response to reflect whether the setting took effect.

The default downlink data aggregation protocol setting is disabled by default on the service. The protocol can be set to enable TLP, QC NTB, MBIM, RNDIS, or QMAP. To change the setting, the optional Downlink Data Aggregation Protocol TLV must be included in the request. The service then uses the configured optional TLV in the response to reflect whether the setting took effect.

The client must specify the same data aggregation protocol in both UL and DL directions. If the client specifies only one optional data aggregation TLV (either Uplink Data Aggregation Protocol or Downlink Data Aggregation Protocol), the service uses the default value of the data aggregation protocol for the other unspecified TLV.

The default NDP signature is based on the set data aggregation protocol. The default signature corresponding to the data aggregation protocol is used if the NDP signature TLV is absent.

By default, the data format is set on the default data channel associated with the QMI control channel from which the request is received. If the Peripheral End Point ID TLV is provided, the data format is set on the peripheral data channel specified by the end point ID.

If the minimum padding value is set, the modem will add minimum specified padding bytes in between the aggregated downlink QMAP packets. More padding bytes may be added to satisfy alignment requirements. A QMI_ERR_INVALID_OPERATION error is returned when multiple clients on two different services try to set the data format (i.e., a client of the QMI_CTL service sets the data format using the

QMI_CTL_SET_DATA_FORMAT command and another client of the QMI_WDA service requests for setting the data format using QMI_WDA_SET_DATA_FORMAT_REQ).

12.2.2. QMI_WDA_GET_DATA_FORMAT

Gets the current data format settings of the client.

- **WDA message ID**

0x0021

- **Version introduced**

Major - 1, Minor - 0

12.2.2.1. Request - QMI_WDA_GET_DATA_FORMAT_REQ

- **Message type**

Request

- **Sender**

Control point

- **Mandatory TLVs**

None

- **Optional TLVs**

Name	Version introduced	Version last modified
Peripheral End Point ID	1.13	1.13

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Peripheral End Point ID The peripheral end point on which the data format is queried. The default value is the default data channel associated with the QMI control channel from which the request is received.
Length	8			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	enum	ep_type	4	Peripheral end point type. Values: • DATA_EP_TYPE_RESERVED (0x00) – Reserved • DATA_EP_TYPE_HSIC (0x01) – HSIC • DATA_EP_TYPE_HSUSB (0x02) – HSUSB • DATA_EP_TYPE_PCIE (0x03) – PCIe • DATA_EP_TYPE_EMBEDDED (0x04) – Embedded All other values are reserved and will be ignored by service or clients.
					Peripheral interface number.

12.2.2.2. Response - QMI_WDA_GET_DATA_FORMAT_RESP

- **Message type**

Response

- **Sender**

Service

- **Mandatory TLVs**

The Result Code TLV (defined in Section 12.12.3.1) is always present in the response.

- **Optional TLVs**

Name	Version introduced	Version last modified
Configured QOS Data Format	1.0	1.0
Underlying Link Layer Protocol	1.0	1.0
Uplink Data Aggregation Protocol	1.0	1.9
Downlink Data Aggregation Protocol	1.0	1.9
NDP Signature	1.0	1.0
Downlink Data Aggregation Max Datagrams	1.2	1.2
Downlink Data Aggregation Max Size	1.2	1.2
Uplink Data Aggregation Max Datagrams	1.11	1.11
Uplink Data Aggregation Max Size	1.11	1.11
QOS Header Format	1.14	1.14
QMAP Downlink Minimum Padding	1.15	1.15
Flow Control	1.16	1.16

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Configured QOS Data Format
Length	1			2	
Value	→	boolean	qos_format	1	Configured QOS data format. Values: • 0 – QOS flow header is not present (Default) • 1 – QOS flow header is present
Type	0x11			1	Underlying Link Layer Protocol
Length	4			2	
Value	→	enum	link_prot	4	Configured link layer protocol. Values: • 0x01 – 802.3 Ethernet mode (Default) • 0x02 – IP mode
Type	0x12			1	Uplink Data Aggregation Protocol
Length	4			2	
Value	→	enum	ul_data_aggregation_protocol	4	Configured uplink data aggregation protocol. Values: • 0x00 – UL data aggregation is disabled (Default) • 0x01 – UL TLP is enabled • 0x02 – UL QC_NCM is enabled • 0x03 – UL MBIM is enabled • 0x04 – UL RNDIS is enabled • 0x05 – UL QMAP is enabled
Type	0x13			1	Downlink Data Aggregation Protocol
Length	4			2	
Value	→	enum	dl_data_aggregation_protocol	4	Configured downlink data aggregation protocol. Values: • 0x00 – DL data aggregation is disabled (Default) • 0x01 – DL TLP is enabled • 0x02 – DL QC_NCM is enabled • 0x03 – DL MBIM is enabled • 0x04 – DL RNDIS is enabled • 0x05 – DL QMAP is enabled
Type	0x14			1	NDP Signature
Length	4			2	
Value	→	uint32	ndp_signature	4	NDP signature. The default value based on the data aggregation protocol is used.
Type	0x15			1	Downlink Data Aggregation Max Datagrams
Length	4			2	
Value	→	uint32	dl_data_aggregation_max_datagrams	4	Maximum number of datagrams in a single aggregated packet on downlink. The value applies to all downlink data aggregation protocols when downlink data aggregation is enabled. Zero means no limit.
Type	0x16			1	Downlink Data Aggregation Max Size

Field	Field value	Field type	Parameter	Size (byte)	Description
Length	4			2	
Value	→	uint32	dl_data_aggregation_max_size	4	Maximum size in bytes of a single aggregated packet allowed on downlink. The value applies to all downlink data aggregation protocols when downlink data aggregation is enabled.
Type	0x17			1	Uplink Data Aggregation Max Datagrams
Length	4			2	
Value	→	uint32	ul_data_aggregation_max_datagrams	4	Maximum number of datagrams supported in the modem in a single aggregated packet on uplink for the currently configured aggregation mode. Zero means there is no limit.
Type	0x18			1	Uplink Data Aggregation Max Size
Length	4			2	
Value	→	uint32	ul_data_aggregation_max_size	4	Maximum size in bytes of a single aggregated packet allowed on uplink.
Type	0x19			1	QOS Header Format
Length	4			2	
Value	→	enum	qos_header_format	4	Configured QOS header format used on the uplink, on all the protocols, if QOS is enabled. Values: <ul style="list-style-type: none">• WDA_QOS_HDR_FORMAT_RESERVED (0x00) – Reserved• WDA_QOS_HDR_FORMAT_6_BYTE (0x01) – QOS 6 byte Default header• WDA_QOS_HDR_FORMAT_8_BYTE (0x02) – QOS 8 byte header
Type	0x1A			1	QMAP Downlink Minimum Padding
Length	4			2	
Value	→	uint32	dl_minimum_padding	4	Indicates the minimum padding bytes added in between aggregated downlink QMAP packets.
Type	0x1B			1	Flow Control
Length	1			2	
Value	→	boolean	flow_control	1	Configured flow control. Values: <ul style="list-style-type: none">• 0 – Flow control will not be done by the TE (Default)• 1 – Flow control will be done by the TE

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	Service could not allocate memory to formulate a response

12.2.2.3. Description of QMI_WDA_GET_DATA_FORMAT REQ/RESP

This command returns the current data format setting for the control point.

13 Appendix A References

Table 8: Reference Documents

SN	Document Name
[1]	80_VB816_1_QMI_Architecture
[2]	80_VB816_3_QMI_CTL_1_11_QMI_CONTROL_SVC_SPEC
[3]	80_NV300_5_QMI WDS 1.117 for MPSS.JO.1.0 QMI Wireless Data Service Spec
[4]	80_VB816_4_T_QMI Device Management Svc Spec
[5]	80_NV300_52_QMI_NAS_EXT_1_4_FOR_MPSS_JO_1_0_QMI_NETWORK_ACCESS_EXTENSION_SERVICE_SPEC
[6]	80_NV300_9_QMI_WMS_1_24_FOR_MPSS_JO_1_0_QMI_WIRELESS_MESSAGE_SERVICE_SPEC
[7]	80-vb816-8_y_qmi_pds_1.35%2C_qmi_position_determination_service_spec
[8]	80_NV300_10_D_QMI VOICE 2.77 for MPSS.JO.1.0
[9]	80_NV300_12_QMI_UIM_1_44_FOR_MPSS_JO_1_0_QMI_USER_IDENTITY_MODULE_SPEC
[10]	80_NV300_26_QMI_WDA_1_16_FOR_MPSS_JO_1_0_QMI_WIRELESS_DATA_ADMINISTRATIVE_SVC_SPEC