

## ACKNOWLEDGEMENT

By utilizing this website and/or documentation, I hereby acknowledge as follows:

Effective October 1, 2012, QUALCOMM Incorporated completed a corporate reorganization in which the assets of certain of its businesses and groups, as well as the stock of certain of its direct and indirect subsidiaries, were contributed to Qualcomm Technologies, Inc. (QTI), a wholly-owned subsidiary of QUALCOMM Incorporated that was created for purposes of the reorganization.

Qualcomm Technology Licensing (QTL), the Company's patent licensing business, continues to be operated by QUALCOMM Incorporated, which continues to own the vast majority of the Company's patent portfolio. Substantially all of the Company's products and services businesses, including QCT, as well as substantially all of the Company's engineering, research and development functions, are now operated by QTI and its direct and indirect subsidiaries<sup>1</sup>. Neither QTI nor any of its subsidiaries has any right, power or authority to grant any licenses or other rights under or to any patents owned by QUALCOMM Incorporated.

No use of this website and/or documentation, including but not limited to the downloading of any software, programs, manuals or other materials of any kind or nature whatsoever, and no purchase or use of any products or services, grants any licenses or other rights, of any kind or nature whatsoever, under or to any patents owned by QUALCOMM Incorporated or any of its subsidiaries. A separate patent license or other similar patent-related agreement from QUALCOMM Incorporated is needed to make, have made, use, sell, import and dispose of any products or services that would infringe any patent owned by QUALCOMM Incorporated in the absence of the grant by QUALCOMM Incorporated of a patent license or other applicable rights under such patent.

Any copyright notice referencing QUALCOMM Incorporated, Qualcomm Incorporated, QUALCOMM Inc., Qualcomm Inc., Qualcomm or similar designation, and which is associated with any of the products or services businesses or the engineering, research or development groups which are now operated by QTI and its direct and indirect subsidiaries, should properly reference, and shall be read to reference, QTI.

---

<sup>1</sup> The products and services businesses, and the engineering, research and development groups, which are now operated by QTI and its subsidiaries include, but are not limited to, QCT, Qualcomm Mobile & Computing (QMC), Qualcomm Atheros (QCA), Qualcomm Internet Services (QIS), Qualcomm Government Technologies (QGOV), Corporate Research & Development, Qualcomm Corporate Engineering Services (QCES), Office of the Chief Technology Officer (OCTO), Office of the Chief Scientist (OCS), Corporate Technical Advisory Group, Global Market Development (GMD), Global Business Operations (GBO), Qualcomm Ventures, Qualcomm Life (QLife), Quest, Qualcomm Labs (QLabs), Snaptracs/QCS, Firethorn, Qualcomm MEMS Technologies (QMT), Pixtronix, Qualcomm Innovation Center (QuIC), Qualcomm iSkoot, Qualcomm Poole and Xiam.



# *QMI QCMAP 1.3 for MPSS.NI.3.0.x*

## *QMI Qualcomm Mobile Access Point Svc Spec*

80-N9986-34 B

September 6, 2012

---

Submit technical questions at:

<https://support.cdmatech.com>

### **Qualcomm Confidential and Proprietary**

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

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

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

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

QUALCOMM is a registered trademark of QUALCOMM Incorporated in the United States and may be registered in other countries. Other product and brand names may be trademarks or registered trademarks of their respective owners. CDMA2000 is a registered certification mark of the Telecommunications Industry Association, used under license. ARM is a registered trademark of ARM Limited.

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

**QUALCOMM Incorporated**  
**5775 Morehouse Drive**  
**San Diego, CA 92121-1714**  
**U.S.A.**  
**Copyright © 2012 QUALCOMM Incorporated.**  
**All rights reserved.**

# Contents

---

<b>1</b>	<b>Introduction</b>	<b>8</b>
1.1	Purpose	8
1.2	Scope	8
1.3	Conventions	8
1.4	References	9
1.5	Technical Assistance	9
1.6	Acronyms	9
<b>2</b>	<b>Theory of Operation</b>	<b>11</b>
2.1	Generalized QMI Service Compliance	11
2.2	QCMAP Service Type	11
2.3	Message Definition Template	11
2.3.1	Response Message Result TLV	11
2.4	QMI_QCMAP Fundamental Concepts	12
2.5	Service State Variables	12
2.5.1	Shared State Variables	12
2.5.2	State Variables Per Control Point	12
<b>3</b>	<b>QMI_QCMAP Messages</b>	<b>13</b>
3.1	QMI_QCMAP_MOBILE_AP_ENABLE	15
3.1.1	Request - QMI_QCMAP_MOBILE_AP_ENABLE_REQ	15
3.1.2	Response - QMI_QCMAP_MOBILE_AP_ENABLE_RESP	17
3.1.3	Description of QMI_QCMAP_MOBILE_AP_ENABLE REQ/RESP	18
3.2	QMI_QCMAP_MOBILE_AP_DISABLE	19
3.2.1	Request - QMI_QCMAP_MOBILE_AP_DISABLE_REQ	19
3.2.2	Response - QMI_QCMAP_DISABLE_RESP	20
3.2.3	Description of QMI_QCMAP_MOBILE_AP_DISABLE REQ/RESP	20
3.3	QMI_QCMAP_BRING_UP_WWAN	21
3.3.1	Request - QMI_QCMAP_BRING_UP_WWAN_REQ	21
3.3.2	Response - QMI_QCMAP_BRING_UP_WWAN_RESP	22
3.3.3	Description of QMI_QCMAP_BRING_UP_WWAN REQ/RESP	22
3.3.4	Indication - QMI_QCMAP_BRING_UP_WWAN_IND	23
3.3.5	Description of QMI_QCMAP_BRING_UP_WWAN_IND	23
3.4	QMI_QCMAP_TEAR_DOWN_WWAN	24
3.4.1	Request - QMI_QCMAP_TEAR_DOWN_WWAN_REQ	24
3.4.2	Response - QMI_QCMAP_TEAR_DOWN_WWAN_RESP	25
3.4.3	Description of QMI_QCMAP_TEAR_DOWN_WWAN REQ/RESP	25
3.4.4	Indication - QMI_QCMAP_TEAR_DOWN_WWAN_IND	26
3.4.5	Description of QMI_QCMAP_TEAR_DOWN_WWAN_IND	26
3.5	QMI_QCMAP_GET_WWAN_STATUS	27

3.5.1	Request - QMI_QCMAP_GET_WWAN_STATUS_REQ	27
3.5.2	Response - QMI_QCMAP_GET_WWAN_STATUS_RESP	28
3.5.3	Description of QMI_QCMAP_GET_WWAN_STATUS_REQ/RESP	29
3.6	QMI_QCMAP_WWAN_STATUS_IND_REG	30
3.6.1	Request - QMI_QCMAP_WWAN_STATUS_IND_REG_REQ	30
3.6.2	Response - QMI_QCMAP_WWAN_STATUS_IND_REG_RESP	31
3.6.3	Description of QMI_QCMAP_WWAN_STATUS_IND_REG_REQ/RESP	31
3.7	QMI_QCMAP_WWAN_STATUS_IND	32
3.7.1	Indication - QMI_QCMAP_WWAN_STATUS_IND	32
3.7.2	Description of QMI_QCMAP_WWAN_STATUS_IND	33
3.8	QMI_QCMAP_SET_IPSEC_VPN_PASS_THROUGH	34
3.8.1	Request - QMI_QCMAP_SET_IPSEC_VPN_PASS_THROUGH_REQ	34
3.8.2	Response - QMI_QCMAP_SET_IPSEC_VPN_PASS_THROUGH_RESP	35
3.8.3	Description of QMI_QCMAP_SET_IPSEC_VPN_PASS_THROUGH_REQ/RESP	35
3.9	QMI_QCMAP_GET_IPSEC_VPN_PASS_THROUGH	36
3.9.1	Request - QMI_QCMAP_GET_IPSEC_VPN_PASS_THROUGH_REQ	36
3.9.2	Response - QMI_QCMAP_GET_IPSEC_VPN_PASS_THROUGH_RESP	37
3.9.3	Description of QMI_QCMAP_GET_IPSEC_VPN_PASS_THROUGH_REQ/RESP	37
3.10	QMI_QCMAP_SET_PPTP_VPN_PASS_THROUGH	38
3.10.1	Request - QMI_QCMAP_SET_PPTP_VPN_PASS_THROUGH_REQ	38
3.10.2	Response - QMI_QCMAP_SET_PPTP_VPN_PASS_THROUGH_RESP	39
3.10.3	Description of QMI_QCMAP_SET_PPTP_VPN_PASS_THROUGH_REQ/RESP	39
3.11	QMI_QCMAP_GET_PPTP_VPN_PASS_THROUGH	40
3.11.1	Request - QMI_QCMAP_GET_PPTP_VPN_PASS_THROUGH_REQ	40
3.11.2	Response - QMI_QCMAP_GET_PPTP_VPN_PASS_THROUGH_RESP	41
3.11.3	Description of QMI_QCMAP_GET_PPTP_VPN_PASS_THROUGH_REQ/RESP	41
3.12	QMI_QCMAP_SET_L2TP_VPN_PASS_THROUGH	42
3.12.1	Request - QMI_QCMAP_SET_L2TP_VPN_PASS_THROUGH_REQ	42
3.12.2	Response - QMI_QCMAP_SET_L2TP_VPN_PASS_THROUGH_RESP	43
3.12.3	Description of QMI_QCMAP_SET_L2TP_VPN_PASS_THROUGH_REQ/RESP	43
3.13	QMI_QCMAP_GET_L2TP_VPN_PASS_THROUGH	44
3.13.1	Request - QMI_QCMAP_GET_L2TP_VPN_PASS_THROUGH_REQ	44
3.13.2	Response - QMI_QCMAP_GET_L2TP_VPN_PASS_THROUGH_RESP	45
3.13.3	Description of QMI_QCMAP_GET_L2TP_VPN_PASS_THROUGH_REQ/RESP	45
3.14	QMI_QCMAP_SET_DYNAMIC_NAT_ENTRY_TIMEOUT	46
3.14.1	Request - QMI_QCMAP_SET_DYNAMIC_NAT_ENTRY_TIMEOUT_REQ	46
3.14.2	Response - QMI_QCMAP_SET_DYNAMIC_NAT_ENTRY_TIMEOUT_RESP	47
3.14.3	Description of QMI_QCMAP_SET_DYNAMIC_NAT_ENTRY_TIMEOUT_REQ/RESP	47
3.15	QMI_QCMAP_GET_DYNAMIC_NAT_ENTRY_TIMEOUT	48
3.15.1	Request - QMI_QCMAP_GET_DYNAMIC_NAT_ENTRY_TIMEOUT_REQ	48
3.15.2	Response - QMI_QCMAP_GET_DYNAMIC_NAT_ENTRY_TIMEOUT_RESP	49
3.15.3	Description of QMI_QCMAP_GET_DYNAMIC_NAT_ENTRY_TIMEOUT_REQ/RESP	49
3.16	QMI_QCMAP_ADD_STATIC_NAT_ENTRY	50
3.16.1	Request - QMI_QCMAP_ADD_STATIC_NAT_ENTRY_REQ	50
3.16.2	Response - QMI_QCMAP_ADD_STATIC_NAT_ENTRY_RESP	51
3.16.3	Description of QMI_QCMAP_ADD_STATIC_NAT_ENTRY_REQ/RESP	51
3.17	QMI_QCMAP_DELETE_STATIC_NAT_ENTRY	52
3.17.1	Request - QMI_QCMAP_DELETE_STATIC_NAT_ENTRY_REQ	52

3.17.2	Response - QMI_QCMAP_DELETE_STATIC_NAT_ENTRY_RESP . . . . .	53
3.17.3	Description of QMI_QCMAP_DELETE_STATIC_NAT_ENTRY REQ/RESP . . . . .	53
3.18	QMI_QCMAP_GET_STATIC_NAT_ENTRIES . . . . .	54
3.18.1	Request - QMI_QCMAP_GET_STATIC_NAT_ENTRIES_REQ . . . . .	54
3.18.2	Response - QMI_QCMAP_GET_STATIC_NAT_ENTRIES_RESP . . . . .	55
3.18.3	Description of QMI_QCMAP_GET_STATIC_NAT_ENTRIES REQ/RESP . . . . .	56
3.19	QMI_QCMAP_SET_DMZ . . . . .	57
3.19.1	Request - QMI_QCMAP_SET_DMZ_REQ . . . . .	57
3.19.2	Response - QMI_QCMAP_SET_DMZ_RESP . . . . .	58
3.19.3	Description of QMI_QCMAP_SET_DMZ REQ/RESP . . . . .	58
3.20	QMI_QCMAP_GET_DMZ . . . . .	59
3.20.1	Request - QMI_QCMAP_GET_DMZ_REQ . . . . .	59
3.20.2	Response - QMI_QCMAP_GET_DMZ_RESP . . . . .	60
3.20.3	Description of QMI_QCMAP_GET_DMZ REQ/RESP . . . . .	60
3.21	QMI_QCMAP_DELETE_DMZ . . . . .	61
3.21.1	Request - QMI_QCMAP_DELETE_DMZ_REQ . . . . .	61
3.21.2	Response - QMI_QCMAP_DELETE_DMZ_RESP . . . . .	62
3.21.3	Description of QMI_QCMAP_DELETE_DMZ REQ/RESP . . . . .	62
3.22	QMI_QCMAP_GET_WWAN_CONFIG . . . . .	63
3.22.1	Request - QMI_QCMAP_GET_WWAN_CONFIG_REQ . . . . .	63
3.22.2	Response - QMI_QCMAP_GET_WWAN_CONFIG_RESP . . . . .	64
3.22.3	Description of QMI_QCMAP_GET_WWAN_CONFIG REQ/RESP . . . . .	65
3.23	QMI_QCMAP_ENABLE_FIREWALL_SETTING . . . . .	66
3.23.1	Request - QMI_QCMAP_ENABLE_FIREWALL_SETTING_REQ . . . . .	66
3.23.2	Response - QMI_QCMAP_ENABLE_FIREWALL_SETTING_RESP . . . . .	67
3.23.3	Description of QMI_QCMAP_ENABLE_FIREWALL_SETTING REQ/RESP . . . . .	67
3.24	QMI_QCMAP_GET_FIREWALL_SETTING . . . . .	68
3.24.1	Request - QMI_QCMAP_GET_FIREWALL_SETTING_REQ . . . . .	68
3.24.2	Response - QMI_QCMAP_GET_FIREWALL_SETTING_RESP . . . . .	69
3.24.3	Description of QMI_QCMAP_GET_FIREWALL_SETTING REQ/RESP . . . . .	70
3.25	QMI_QCMAP_DISABLE_FIREWALL_SETTING . . . . .	71
3.25.1	Request - QMI_QCMAP_DISABLE_FIREWALL_SETTING_REQ . . . . .	71
3.25.2	Response - QMI_QCMAP_DISABLE_FIREWALL_SETTING_RESP . . . . .	72
3.25.3	Description of QMI_QCMAP_DISABLE_FIREWALL_SETTING REQ/RESP . . . . .	72
3.26	QMI_QCMAP_ADD_FIREWALL_CONFIG . . . . .	73
3.26.1	Request - QMI_QCMAP_ADD_FIREWALL_CONFIG_REQ . . . . .	73
3.26.2	Response - QMI_QCMAP_ADD_FIREWALL_CONFIG_RESP . . . . .	74
3.26.3	Description of QMI_QCMAP_ADD_FIREWALL_CONFIG REQ/RESP . . . . .	75
3.27	QMI_QCMAP_DELETE_FIREWALL_CONFIG . . . . .	76
3.27.1	Request - QMI_QCMAP_DELETE_FIREWALL_CONFIG_REQ . . . . .	76
3.27.2	Response - QMI_QCMAP_DELETE_FIREWALL_CONFIG_RESP . . . . .	77
3.27.3	Description of QMI_QCMAP_DELETE_FIREWALL_CONFIG REQ/RESP . . . . .	77
3.28	QMI_QCMAP_GET_FIREWALL_CONFIG . . . . .	78
3.28.1	Request - QMI_QCMAP_GET_FIREWALL_CONFIG_REQ . . . . .	78
3.28.2	Response - QMI_QCMAP_GET_FIREWALL_CONFIG_RESP . . . . .	79
3.28.3	Description of QMI_QCMAP_GET_FIREWALL_CONFIG REQ/RESP . . . . .	80
3.29	QMI_QCMAP_STATION_MODE_ENABLE . . . . .	81
3.29.1	Request - QMI_QCMAP_STATION_MODE_ENABLE_REQ . . . . .	81
3.29.2	Response - QMI_QCMAP_STATION_MODE_ENABLE_RESP . . . . .	82
3.29.3	Description of QMI_QCMAP_STATION_MODE_ENABLE REQ/RESP . . . . .	82

3.30	QMI_QCMAP_STATION_MODE_DISABLE	83
3.30.1	Request - QMI_QCMAP_STATION_MODE_DISABLE_REQ	83
3.30.2	Response - QMI_QCMAP_STATION_MODE_DISABLE_RESP	84
3.30.3	Description of QMI_QCMAP_STATION_MODE_DISABLE REQ/RESP	84
3.31	QMI_QCMAP_GET_STATION_MODE	85
3.31.1	Request - QMI_QCMAP_GET_STATION_MODE_REQ	85
3.31.2	Response - QMI_QCMAP_GET_STATION_MODE_RESP	86
3.31.3	Description of QMI_QCMAP_GET_STATION_MODE REQ/RESP	86
3.32	QMI_QCMAP_ADD_EXTD_FIREWALL_CONFIG	87
3.32.1	Request - QMI_QCMAP_ADD_EXTD_FIREWALL_CONFIG_REQ	87
3.32.2	Response - QMI_QCMAP_ADD_EXTD_FIREWALL_CONFIG_RESP	90
3.32.3	Description of QMI_QCMAP_ADD_EXTD_FIREWALL_CONFIG REQ/RESP	91
3.33	QMI_QCMAP_GET_EXTD_FIREWALL_CONFIG	92
3.33.1	Request - QMI_QCMAP_GET_EXTD_FIREWALL_CONFIG_REQ	92
3.33.2	Response - QMI_QCMAP_GET_EXTD_FIREWALL_CONFIG_RESP	93
3.33.3	Description of QMI_QCMAP_GET_EXTD_FIREWALL_CONFIG REQ/RESP	96
3.34	QMI_QCMAP_GET_FIREWALL_CONFIG_HANDLE_LIST	97
3.34.1	Request - QMI_QCMAP_GET_FIREWALL_CONFIG_HANDLE_LIST_REQ	97
3.34.2	Response - QMI_QCMAP_GET_FIREWALL_CONFIG_HANDLE_LIST_RESP	98
3.34.3	Description of QMI_QCMAP_GET_FIREWALL_CONFIG_HANDLE_LIST REQ/RESP	99
3.35	QMI_QCMAP_CHANGE_NAT_TYPE	100
3.35.1	Request - QMI_QCMAP_CHANGE_NAT_TYPE_REQ	100
3.35.2	Response - QMI_QCMAP_CHANGE_NAT_TYPE_RESP	101
3.35.3	Description of QMI_QCMAP_CHANGE_NAT_TYPE REQ/RESP	101
3.36	QMI_QCMAP_GET_NAT_TYPE	102
3.36.1	Request - QMI_QCMAP_GET_NAT_TYPE_REQ	102
3.36.2	Response - QMI_QCMAP_GET_NAT_TYPE_RESP	103
3.36.3	Description of QMI_QCMAP_GET_NAT_TYPE REQ/RESP	104
<b>A</b>	<b>Call End Reasons</b>	<b>105</b>
A.1	Call End Reasons	105
A.2	Verbose Call End Reasons	107



## List of Tables

1-1	Reference documents and standards	9
1-2	Acronyms	9
3-1	QMI_QCMAP messages	13
A-1	Call end reasons	105
A-2	Verbose call end reasons	107

Qualcomm  
Confidential - May Contain Trade Secrets  
2022-07-21 08:18:44 GMT  
jason1\_gao@askey.com

## Revision History

Revision	Date	Description
A	May 2012	Initial release. Created from 80-VB816-34 A.
B	Sep 2012	Updates for this revision include minor version 2 and minor version 3. Updated sections 2.3.1 and 3.1.3. Added new TLVs: <ul style="list-style-type: none"><li>• SSID2 IP address info</li><li>• NAT type info</li></ul> Added new messages: <ul style="list-style-type: none"><li>• QMI_QCMAP_CHANGE_NAT_TYPE (Section 3.35)</li><li>• QMI_QCMAP_GET_NAT_TYPE (Section 3.36)</li></ul>

Qualcomm  
Confidential - May Contain Trade Secrets  
2022-07-21 08:18:44 GMT  
jason1\_gao@askey.com



# 1 Introduction

---

## 1.1 Purpose

This specification documents Major Version 1 of the Qualcomm Messaging Interface (QMI) for Qualcomm Mobile Access Point Service (QMI\_QCMAP).

QMI\_QCMAP provides a command set to interface with a wireless mobile station to access mobile AP services.

## 1.2 Scope

This document is intended for software developers using QMI\_QCMAP on a host processor and interacting with a Qualcomm MSM™ device for controlling Qualcomm mobile access point functionality.

This document provides the following details about QMI\_QCMAP:

- Theory of operation – Chapter 2 provides the theory of operation of QMI\_QCMAP. The chapter includes messaging conventions, assigned QMI service type, fundamental service concepts, and state variables related to the service.
- Message formats, syntax, and semantics – Chapter 3 provides the specific syntax and semantics of messages included in this version of the QMI\_QCMAP specification.
- Additional information – Appendix A provides tables for call end reasons and verbose call end reasons.

## 1.3 Conventions

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

Parameter types are indicated by arrows:

- Designates an input parameter
- ← Designates an output parameter
- ↔ Designates a parameter used for both input and output

## 1.4 References

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

**Table 1-1 Reference documents and standards**

Ref.	Document	
Qualcomm		
Q1	Application Note: Software Glossary for Customers	CL93-V3077-1
Q2	Qualcomm MSM™ Interface (QMI) Architecture	80-VB816-1
Q3	QMI WDS for MPSS.NI.3.0.x, QMI Wireless Data Svc Spec	80-N9986-5
Standards		
S1	User Datagram Protocol	RFC 768 (Aug 1980)
S2	Internet Protocol DARPA Internet Program Protocol Specification	RFC 791 (Sep 1981)
S3	Internet Control Message Protocol DARPA Internet Program Protocol Specification	RFC 792 (Sep 1981)
S4	Transmission Control Protocol DARPA Internet Program Protocol Specification	RFC 793 (Sep 1981)
S5	Internet Protocol Version 6 (IPv6) Specification	RFC 2460 (Dec 1998)
S6	Internet Protocol Version 6 (IPv6) Addressing Architecture	RFC 3513 (Apr 2003)
S7	IP Encapsulating Security Payload (ESP)	RFC 4303 (Dec 2005)

## 1.5 Technical Assistance

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

If you do not have access to the CDMATEch Support Services website, register for access or send email to [support.cdmatech@qualcomm.com](mailto:support.cdmatech@qualcomm.com).

## 1.6 Acronyms

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

**Table 1-2 Acronyms**

Acronym	Definition
AP	access point
DMZ	DMZ (sometimes referred to as a perimeter network) is a physical or logical subnetwork that contains and exposes an organization's external services to a larger untrusted network, usually the Internet. The purpose of a DMZ is to add an additional layer of security to an organization's LAN.
DNS	domain name service
ESP	Encapsulating Security Payload Protocol
ICMP	Internet Control Message Protocol

Table 1-2 Acronyms (cont.)

Acronym	Definition
IPSec	Internet Protocol security
L2TP	Layer 2 Tunneling Protocol
MIP	Mobile Internet Protocol
NAT	network address translation
PPTP	Point-to-Point Tunneling Protocol
QCMAP	Qualcomm Mobile Access Point Service
QMI	Qualcomm messaging interface
SNAT	static NAT
SSID	service set identifier
STA	station
TCP	Transmission Control Protocol
TE	terminal equipment
TLV	type-length-value
TOS	type of service
UDP	User Datagram Protocol
VPN	virtual private network

## 2 Theory of Operation

---

### 2.1 Generalized QMI Service Compliance

The QMI\_QCMAP 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 [Q2]. Extensions to the generalized QMI service theory of operation are noted in subsequent sections of this chapter.

### 2.2 QCMAP Service Type

QCMAP is assigned QMI service type 0x1E.

### 2.3 Message Definition Template

#### 2.3.1 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 <ul style="list-style-type: none"><li>• QMI_RESULT_SUCCESS</li><li>• QMI_RESULT_FAILURE</li></ul>
		uint16	qmi_error	2	Error code – Possible error code values are described in the error codes section of each message definition

## 2.4 QMI\_QCMAP Fundamental Concepts

QMI\_QCMAP provides a command set to interface with a wireless mobile station to access mobile AP services. The QMI\_QCMAP service supports only one client per QMI control channel.

## 2.5 Service State Variables

### 2.5.1 Shared State Variables

No QMI\_QCMAP state variables are shared across control points.

### 2.5.2 State Variables Per Control Point

Name	Description	Possible values	Default value
register_indication	WWAN status indication registration per mobile AP handle	<ul style="list-style-type: none"><li>• FALSE</li><li>• TRUE</li></ul>	FALSE

### 3 QMI\_QCMAP Messages

---

**Table 3-1 QMI\_QCMAP messages**

Command	ID	Description
QMI_QCMAP_MOBILE_AP_ENABLE	0x0020	Enables the mobile AP functionality via a single mobile AP instance on the modem.
QMI_QCMAP_MOBILE_AP_DISABLE	0x0021	Disables the mobile AP functionality for a mobile AP instance on the modem.
QMI_QCMAP_BRING_UP_WWAN	0x0022	Invokes bringing up the WWAN from the mobile AP.
QMI_QCMAP_BRING_UP_WWAN_IND	0x0022 indication	Indicates the completion of processing a QMI_QCMAP_BRING_UP_WWAN_REQ.
QMI_QCMAP_TEAR_DOWN_WWAN	0x0023	Tears down the WWAN.
QMI_QCMAP_TEAR_DOWN_WWAN_IND	0x0023 indication	Indicates the completion of processing a QMI_QCMAP_TEAR_DOWN_WWAN_REQ.
QMI_QCMAP_GET_WWAN_STATUS	0x0024	Queries the current WWAN status.
QMI_QCMAP_WWAN_STATUS_IND_REG	0x003A	Registers/deregisters the control point to receive QMI_QCMAP_WWAN_STATUS_IND.
QMI_QCMAP_WWAN_STATUS_IND	0x003E	Indicates a change in the current mobile AP WWAN connection status.
QMI_QCMAP_SET_IPSEC_VPN_PASS_THROUGH	0x0026	Configures the Internet Protocol security (IPSec) Virtual Private Network (VPN) passthrough setting.
QMI_QCMAP_GET_IPSEC_VPN_PASS_THROUGH	0x0025	Queries the IPSec VPN passthrough setting.
QMI_QCMAP_SET_PPTP_VPN_PASS_THROUGH	0x0028	Configures the Point-to-Point Tunneling Protocol (PPTP) VPN passthrough setting.
QMI_QCMAP_GET_PPTP_VPN_PASS_THROUGH	0x0027	Queries the PPTP VPN passthrough setting.
QMI_QCMAP_SET_L2TP_VPN_PASS_THROUGH	0x002A	Configures the Layer 2 Tunneling Protocol (L2TP) VPN passthrough setting.
QMI_QCMAP_GET_L2TP_VPN_PASS_THROUGH	0x0029	Queries the L2TP VPN passthrough setting.

Table 3-1 QMI\_QCMAP messages (cont.)

Command	ID	Description
QMI_QCMAP_SET_DYNAMIC_NAT_ENTRY_TIMEOUT	0x002C	Sets the Network Address Translation (NAT) entry timeout.
QMI_QCMAP_GET_DYNAMIC_NAT_ENTRY_TIMEOUT	0x002B	Queries the NAT entry timeout.
QMI_QCMAP_ADD_STATIC_NAT_ENTRY	0x002D	Adds a static NAT entry.
QMI_QCMAP_DELETE_STATIC_NAT_ENTRY	0x002E	Deletes a static NAT entry.
QMI_QCMAP_GET_STATIC_NAT_ENTRIES	0x002F	Queries all static NAT entries.
QMI_QCMAP_SET_DMZ	0x0030	Sets the DMZ (perimeter network) IP address for the mobile AP.
QMI_QCMAP_GET_DMZ	0x0032	Queries the DMZ IP address on the mobile AP.
QMI_QCMAP_DELETE_DMZ	0x0031	Deletes the DMZ entry or DMZ IP address.
QMI_QCMAP_GET_WWAN_CONFIG	0x0033	Queries the WWAN IP configuration.
QMI_QCMAP_ENABLE_FIREWALL_SETTING	0x0034	Enables the firewall setting.
QMI_QCMAP_GET_FIREWALL_SETTING	0x0035	Queries the firewall setting.
QMI_QCMAP_DISABLE_FIREWALL_SETTING	0x0036	Disables the firewall setting.
QMI_QCMAP_ADD_FIREWALL_CONFIG	0x0037	Adds a firewall configuration rule.
QMI_QCMAP_DELETE_FIREWALL_CONFIG	0x0039	Deletes a firewall configuration rule.
QMI_QCMAP_GET_FIREWALL_CONFIG	0x0038	Queries the firewall configuration rules.
QMI_QCMAP_STATION_MODE_ENABLE	0x003B	Enables Station (STA) mode functionality for a mobile AP instance on the modem.
QMI_QCMAP_STATION_MODE_DISABLE	0x003C	Disables STA mode functionality for a mobile AP instance on the modem.
QMI_QCMAP_GET_STATION_MODE	0x003D	Queries the STA mode functionality for a mobile AP instance on the modem.
QMI_QCMAP_ADD_EXTD_FIREWALL_CONFIG	0x003F	Adds IP filter-based firewall rules (extended firewall).
QMI_QCMAP_GET_EXTD_FIREWALL_CONFIG	0x0040	Gets the firewall rules.
QMI_QCMAP_GET_FIREWALL_CONFIG_HANDLE_LIST	0x0041	Gets the handles of all the firewall rules.
QMI_QCMAP_CHANGE_NAT_TYPE	0x0042	Changes the currently existing NAT type.
QMI_QCMAP_GET_NAT_TYPE	0x0043	Gets the currently enabled NAT type.



### 3.1 QMI\_QCMAP\_MOBILE\_AP\_ENABLE

Enables the mobile AP functionality via a single mobile AP instance on the modem.

#### QCMAP message ID

0x0020

#### Version introduced

Major - 1, Minor - 0

#### 3.1.1 Request - QMI\_QCMAP\_MOBILE\_AP\_ENABLE\_REQ

##### Message type

Request

##### Sender

Control point

##### Mandatory TLVs

Name	Version introduced	Version last modified
IP Family	1.0	1.0

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	IP Family
Length	4			2	
Value	→	enum	ip_family	4	Determines whether mobile AP IPv4 or IPv6 must be enabled. Values: • 4 – IPv4 • 6 – IPv6

##### Optional TLVs

Name	Version introduced	Version last modified
IP Address	1.0	1.0
Network Policy	1.0	1.0
SSID2 IP Address Info	1.2	1.2
NAT Type Info	1.3	1.3

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	IP Address
Length	28			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	uint32	subnet_mask	4	Subnet mask.
		uint32	nat_ip_addr	4	NAT IP address.
		uint32	nat_dns_addr	4	NAT Domain Name Service (DNS) address.
		uint32	usb_rmnet_ip_addr	4	RmNet USB Terminal Equipment (TE) address.
		uint32	usb_rmnet_gateway_addr	4	RmNet USB gateway address.
		uint32	apps_rmnet_ip_addr	4	RmNet applications IP address.
		uint32	apps_rmnet_gateway_addr	4	RmNet applications gateway address.
Type	0x11			1	Network Policy
Length	10			2	
Value	→	mask	tech_pref	8	Bitmap indicating the technology preference. A single connection is attempted using the following specified technology preferences: <ul style="list-style-type: none"> <li>• Bit 0 – 3GPP</li> <li>• Bit 1 – 3GPP2</li> </ul> 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 the technology to be used from those specified.
		uint8	profile_id_3gpp2	1	CDMA profile ID.
		uint8	profile_id_3gpp	1	UMTS profile ID.
				1	SSID2 IP Address Info
Type	0x12			1	
Length	8			2	
Value	→	uint32	addr	4	IPv4 address as specified in the IPv4 protocol specification (RFC 791 [S2]).
		uint32	subnet_mask	4	IPv4 subnet mask as specified in the IPv4 protocol specification (RFC 791 [S2]).
Type	0x13			1	NAT Type Info
Length	4			2	
Value	→	enum	qcmmap_nat_type_info	4	NAT type specified during mobile AP enable. Values: <ul style="list-style-type: none"> <li>• 0x00 – QCMAP_NAT_TYPE_SYMMETRIC – Symmetric NAT</li> <li>• 0x01 – QCMAP_NAT_TYPE_PORT_RESTRICTED_CONE – Port restricted cone NAT</li> </ul>

### 3.1.2 Response - QMI\_QCMAP\_MOBILE\_AP\_ENABLE\_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

Name	Version introduced	Version last modified
Mobile AP Handle	1.0	1.0

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Mobile AP Handle
Length	4			2	
Value	→	uint32	mobile_ap_handle	4	Handle identifying the mobile AP call instance. The mobile AP handle must be retained by the control point and specified in all mobile AP-specific QCMAP messages. For example, QMI_QCMAP_DISABLE, QMI_QCMAP_BRING_UP_WWAN, etc.

#### 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	Some TLV was missing
QMI_ERR_NOT_SUPPORTED	Operation is not supported
QMI_ERR_NO_EFFECT	Mobile AP instance is already enabled

### 3.1.3 Description of QMI\_QCMAP\_MOBILE\_AP\_ENABLE REQ/RESP

This command enables the mobile AP functionality at the modem. The control point passes the network policy that is used to bring up the WWAN when QMI\_QCMAP\_BRING\_UP\_WWAN is called. After QMI\_QCMAP\_MOBILE\_AP\_ENABLE is successfully processed, any subsequent RmNet call using the same network policy is brought up in the Mobile AP mode. If the IP family is QCMAP\_IP\_V4, the control point must fill in the optional IP Address TLV.

The control point is expected to store the mobile AP handle that is returned and to pass it in all mobile AP-specific messages.

The Network Policy TLV provides the network policy that is used by the mobile AP to select the WWAN network. If this value is not specified, the default WWAN network is selected.

The IP Address TLV is required when the mobile AP IPv4 is enabled. The value is ignored when the mobile AP IPv6 is enabled. If the TLV is not specified when enabling the mobile AP IPv4, a QMI\_ERR\_MISSING\_ARG error is returned.

The SSID2 IP Address Info TLV is required when the mobile AP IPv4 Service Set Identifier 2 (SSID2) is enabled. The value is ignored when the mobile AP IPv6 is enabled. If this TLV is not specified when enabling the mobile AP IPv4, it is assumed that SSID2 is not enabled.

The mobile AP instance enabled by this command remains enabled until the control point or client issues a QMI\_QCMAP\_MOBILE\_AP\_DISABLE\_REQ request or until the control point disassociates from the service.

## 3.2 QMI\_QCMAP\_MOBILE\_AP\_DISABLE

Disables the mobile AP functionality for a mobile AP instance on the modem.

### QCMAP message ID

0x0021

### Version introduced

Major - 1, Minor - 0

### 3.2.1 Request - QMI\_QCMAP\_MOBILE\_AP\_DISABLE\_REQ

#### Message type

Request

#### Sender

Control point

#### Mandatory TLVs

Name	Version introduced	Version last modified
Mobile AP Handle	1.0	1.0

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Mobile AP Handle
Length	4			2	
Value	→	uint32	mobile_ap_handle	4	Handle identifying the mobile AP call instance. The value must be the handle previously returned by QMI_QCMAP_MOBILE_AP_ENABLE_REQ.

#### Optional TLVs

None

### 3.2.2 Response - QMI\_QCMAP\_DISABLE\_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	Some TLV was missing
QMI_ERR_INVALID_HANDLE	Mobile AP handle provided in the request is not valid, i.e., it is not assigned to the control point
QMI_ERR_NO_EFFECT	WWAN is connected or in a transient state; the control point must terminate the WWAN connection using QMI_QCMAP_TEAR_DOWN_WWAN_REQ and wait for the final WWAN status before disabling the mobile AP

### 3.2.3 Description of QMI\_QCMAP\_MOBILE\_AP\_DISABLE REQ/RESP

This command disables the mobile AP functionality at the modem for a single mobile AP instance. After the request is successfully processed, the ongoing RmNet and WWAN calls (if any) are torn down and subsequent RmNet calls are brought up in the non-Mobile AP mode. If the RmNet call is up in the Mobile AP mode at the time this command is sent, the control point considers that the packet data connection state is unchanged until notified of a state change via QMI\_WDS\_PKT\_SRVC\_STATUS\_IND (refer to Q3) for the RmNet session. If the WWAN call is active, the mobile AP is not disabled and a QMI\_ERR\_NO\_EFFECT error is returned.

The mobile AP instance associated with the control point can be disabled using either this command or when the control point disconnects from the QMI\_QCMAP service. Qualcomm recommends that the client disable the mobile AP instances specifically using this command and then proceed by disconnecting from the service.

All NAT-specific functionalities associated with this mobile AP instance are disabled when the command is used or when the control point disassociates from the QMI\_QCMAP service. The control point must reactivate or set functionalities such as the DMZ, VPN passthrough, static NAT, and the firewall after enabling the mobile AP again.

### 3.3 QMI\_QCMAP\_BRING\_UP\_WWAN

Invokes bringing up the WWAN from the mobile AP.

#### QCMAP message ID

0x0022

#### Version introduced

Major - 1, Minor - 0

#### 3.3.1 Request - QMI\_QCMAP\_BRING\_UP\_WWAN\_REQ

##### Message type

Request

##### Sender

Control point

##### Mandatory TLVs

Name	Version introduced	Version last modified
Mobile AP Handle	1.0	1.0

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Mobile AP Handle
Length	4			2	
Value	→	uint32	mobile_ap_handle	4	Handle identifying the mobile AP call instance. The value must be the handle previously returned by QMI_QCMAP_MOBILE_AP_ENABLE_REQ.

##### Optional TLVs

None



### 3.3.2 Response - QMI\_QCMAP\_BRING\_UP\_WWAN\_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	Some TLV was missing
QMI_ERR_INVALID_HANDLE	Mobile AP handle provided in the request is not valid, i.e., it is not assigned to the control point
QMI_ERR_NO_EFFECT	WWAN is already up or a previous request is still in process (WWAN is connecting)

### 3.3.3 Description of QMI\_QCMAP\_BRING\_UP\_WWAN REQ/RESP

This command brings up the WWAN connection. The call is established using the stored network policy that enabled the mobile AP via QMI\_QCMAP\_MOBILE\_AP\_ENABLE\_REQ.

If the response returned is SUCCESS, the corresponding QMI\_QCMAP\_BRING\_UP\_WWAN\_IND indication determines that the request has been completely processed by the modem.

The WWAN status can be queried using QMI\_QCMAP\_GET\_WWAN\_STATUS or sent as an indication for registered clients. See QMI\_QCMAP\_WWAN\_STATUS\_IND\_REG (Section 3.6) for information on registration.

If the control point issues multiple requests in short intervals, a QMI\_ERR\_NO\_EFFECT error is returned indicating that the previous request is still in process.

### 3.3.4 Indication - QMI\_QCMAP\_BRING\_UP\_WWAN\_IND

#### Message type

Indication

#### Sender

Service

#### Indication scope

Unicast

#### Mandatory TLVs

Name	Version introduced	Version last modified
Mobile AP Handle	1.0	1.0
IP Family	1.0	1.0

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Mobile AP Handle
Length	4			2	
Value	→	uint32	mobile_ap_handle	4	Handle identifying the mobile AP call instance.
Type	0x02			1	IP Family
Length	4			2	
Value	→	enum	ip_family	4	Determines whether the mobile AP is IPv4 or IPv6. Values: <ul style="list-style-type: none"> <li>• 4 – IPv4</li> <li>• 6 – IPv6</li> </ul>

#### Optional TLVs

None

### 3.3.5 Description of QMI\_QCMAP\_BRING\_UP\_WWAN\_IND

This indication communicates the completion of processing a QMI\_QCMAP\_BRING\_UP\_WWAN\_REQ received from the control point. If the client registered for the QMI\_QCMAP\_WWAN\_STATUS\_IND indication, it receives the corresponding event indication that reports the WWAN status. Alternatively, the control point can issue QMI\_QCMAP\_GET\_WWAN\_STATUS\_REQ to query the current WWAN status.

## 3.4 QMI\_QCMAP\_TEAR\_DOWN\_WWAN

Tears down the WWAN.

### QCMAP message ID

0x0023

### Version introduced

Major - 1, Minor - 0

### 3.4.1 Request - QMI\_QCMAP\_TEAR\_DOWN\_WWAN\_REQ

#### Message type

Request

#### Sender

Control point

#### Mandatory TLVs

Name	Version introduced	Version last modified
Mobile AP Handle	1.0	1.0

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Mobile AP Handle
Length	4			2	
Value	→	uint32	mobile_ap_handle	4	Handle identifying the mobile AP call instance. The value must be the handle previously returned by QMI_QCMAP_MOBILE_AP_ENABLE_REQ.

#### Optional TLVs

None

### 3.4.2 Response - QMI\_QCMAP\_TEAR\_DOWN\_WWAN\_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	Some TLV was missing
QMI_ERR_INVALID_HANDLE	Mobile AP handle provided in the request is not valid, i.e., it is not assigned to the control point
QMI_ERR_NO_EFFECT	WWAN is already down or a previous request is still in process (WWAN is disconnecting)

### 3.4.3 Description of QMI\_QCMAP\_TEAR\_DOWN\_WWAN\_REQ/RESP

This command tears down the mobile AP WWAN interface that was brought up via QMI\_QCMAP\_BRING\_UP\_WWAN.

If the response returned is SUCCESS, the corresponding QMI\_QCMAP\_TEAR\_DOWN\_WWAN\_IND indication determines that the request has been completely processed by the modem.

The WWAN status can be queried using QMI\_QCMAP\_GET\_WWAN\_STATUS or sent as an indication for registered clients. See QMI\_QCMAP\_WWAN\_STATUS\_IND\_REG (Section 3.6) for information on registration.

If the control point issues multiple requests in short intervals, a QMI\_ERR\_NO\_EFFECT error is returned indicating that the previous request is still in process.

### 3.4.4 Indication - QMI\_QCMAP\_TEAR\_DOWN\_WWAN\_IND

#### Message type

Indication

#### Sender

Service

#### Indication scope

Unicast

#### Mandatory TLVs

Name	Version introduced	Version last modified
Mobile AP Handle	1.0	1.0
IP Family	1.0	1.0

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Mobile AP Handle
Length	4			2	
Value	→	uint32	mobile_ap_handle	4	Handle identifying the mobile AP call instance.
Type	0x02			1	IP Family
Length	4			2	
Value	→	enum	ip_family	4	Determines whether the mobile AP is IPv4 or IPv6. Values: <ul style="list-style-type: none"> <li>• 4 – IPv4</li> <li>• 6 – IPv6</li> </ul>

#### Optional TLVs

None

### 3.4.5 Description of QMI\_QCMAP\_TEAR\_DOWN\_WWAN\_IND

This indication communicates the completion of processing a QMI\_QCMAP\_TEAR\_DOWN\_WWAN\_REQ received from the control point. If the client registered for the QMI\_QCMAP\_WWAN\_STATUS\_IND indication, it receives the corresponding event indication that reports the WWAN status. Alternatively, the control point can issue QMI\_QCMAP\_GET\_WWAN\_STATUS\_REQ to query the current WWAN status.

## 3.5 QMI\_QCMAP\_GET\_WWAN\_STATUS

Queries the current WWAN status.

### QCMAP message ID

0x0024

### Version introduced

Major - 1, Minor - 0

### 3.5.1 Request - QMI\_QCMAP\_GET\_WWAN\_STATUS\_REQ

#### Message type

Request

#### Sender

Control point

#### Mandatory TLVs

Name	Version introduced	Version last modified
Mobile AP Handle	1.0	1.0

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Mobile AP Handle
Length	4			2	
Value	→	uint32	mobile_ap_handle	4	Handle identifying the mobile AP call instance. The value must be the handle previously returned by QMI_QCMAP_MOBILE_AP_ENABLE_REQ.

#### Optional TLVs

None

### 3.5.2 Response - QMI\_QCMAP\_GET\_WWAN\_STATUS\_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

Name	Version introduced	Version last modified
Call End Reason	1.0	1.0
Verbose Call End Reason	1.0	1.0
Packet Service Status	1.0	1.0

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Call End Reason
Length	4			2	
Value	→	enum	call_end_reason	4	Reason the call ended; see Table A-1 for the definition of these values.
Type	0x11			1	Verbose Call End Reason
Length	4			2	
Value	→	enum	verbose_call_end_reason	4	Reason the call ended (verbose); see Table A-2 for the definition of these values.
Type	0x12			1	Packet Service Status
Length	4			2	
Value	→	enum	wwan_status	4	If the response is QMI_ERR_NONE, this indicates the WWAN status. Values: <ul style="list-style-type: none"> <li>• 1 – Connecting</li> <li>• 2 – Connected</li> <li>• 3 – Disconnecting</li> <li>• 4 – Disconnected</li> </ul>



**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_INVALID_HANDLE	Mobile AP handle provided in the request is not valid, i.e., it is not assigned to the control point
QMI_ERR_NO_MEMORY	Device could not allocate memory to formulate a response
QMI_ERR_MISSING_ARG	One or more mandatory TLVs are missing

**3.5.3 Description of QMI\_QCMAP\_GET\_WWAN\_STATUS REQ/RESP**

This command queries the state of the WWAN instantaneously corresponding to the mobile AP handle. The WWAN state could have changed for the following reasons:

- The WWAN state was earlier changed via QMI\_QCMAP\_BRING\_UP\_WWAN or QMI\_QCMAP\_TEAR\_DOWN\_WWAN
- If the network-initiated call status changes

## 3.6 QMI\_QCMAP\_WWAN\_STATUS\_IND\_REG

Registers/deregisters the control point to receive QMI\_QCMAP\_WWAN\_STATUS\_IND.

### QCMAP message ID

0x003A

### Version introduced

Major - 1, Minor - 0

### 3.6.1 Request - QMI\_QCMAP\_WWAN\_STATUS\_IND\_REG\_REQ

#### Message type

Request

#### Sender

Control point

#### Mandatory TLVs

Name	Version introduced	Version last modified
Mobile AP Handle	1.0	1.0
Register Indication	1.0	1.0

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Mobile AP Handle
Length	4			2	
Value	→	uint32	mobile_ap_handle	4	Handle identifying the mobile AP call instance. The value must be the handle previously returned by QMI_QCMAP_MOBILE_AP_ENABLE_REQ.
Type	0x02			1	Register Indication
Length	1			2	
Value	→	boolean	register_indication	1	Specifies the registration. Values: • 0 – Do not register or deregister if already registered • 1 – Register for the indication; ignore if already registered

**Optional TLVs**

None

**3.6.2 Response - QMI\_QCMAP\_WWAN\_STATUS\_IND\_REG\_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_INVALID_HANDLE	Mobile AP handle provided in the request is not valid, i.e., it is not assigned to the control point
QMI_ERR_NO_MEMORY	Device could not allocate memory to formulate a response
QMI_ERR_MISSING_ARG	One or more of mandatory TLVs are missing

**3.6.3 Description of QMI\_QCMAP\_WWAN\_STATUS\_IND\_REG REQ/RESP**

This command registers/deregisters the control point to receive the QMI\_QCMAP\_WWAN\_STATUS\_IND indication.

## 3.7 QMI\_QCMAP\_WWAN\_STATUS\_IND

Indicates a change in the current mobile AP WWAN connection status.

### QCMAP message ID

0x003E

### Version introduced

Major - 1, Minor - 0

### 3.7.1 Indication - QMI\_QCMAP\_WWAN\_STATUS\_IND

#### Message type

Indication

#### Sender

Service

#### Indication scope

Unicast

#### Mandatory TLVs

Name	Version introduced	Version last modified
Mobile AP Handle	1.0	1.0
IP Family	1.0	1.0
Packet Service Status	1.0	1.0
Reconfiguration Required	1.0	1.0

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Mobile AP Handle
Length	4			2	
Value	→	uint32	mobile_ap_handle	4	Handle identifying the mobile AP call instance.
Type	0x02			1	IP Family
Length	4			2	
Value	→	enum	ip_family	4	Determines whether the mobile AP is IPv4 or IPv6. Value: <ul style="list-style-type: none"> <li>• 4 – IPv4</li> <li>• 6 – IPv6</li> </ul>
Type	0x03			1	Packet Service Status
Length	4			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	enum	wwan_status	4	Indicates the WWAN status. Values: <ul style="list-style-type: none"> <li>• 1 – Connecting</li> <li>• 2 – Connected</li> <li>• 3 – Disconnecting</li> <li>• 4 – Disconnected</li> </ul>
Type	0x04			1	Reconfiguration Required
Length	1			2	
Value	→	uint8	reconfig_required	1	Indicates whether the IP reconfiguration is required by the control point.

### Optional TLVs

Name	Version introduced	Version last modified
Call End Reason	1.0	1.0
Verbose Call End Reason	1.0	1.0

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Call End Reason
Length	4			2	
Value	→	enum	call_end_reason	4	Reason the call ended; see Table A-1 for the definition of these values.
Type	0x11			1	Verbose Call End Reason
Length	4			2	
Value	→	enum	verbose_call_end_reason	4	Reason the call ended (verbose); see Table A-2 for the definition of these values.

### 3.7.2 Description of QMI\_QCMAP\_WWAN\_STATUS\_IND

This indication communicates changes in the WWAN state.

The indication is also sent when the WWAN technology changes after a handoff is performed on the modem. The Reconfiguration Required TLV value will be set to indicate that an IP address reconfiguration is required by the control point.

If the indication is sent due to a WWAN Down state, the optional Call End Reason TLV and optional Verbose Call End Reason TLV are included and will contain the reason the call was terminated. These reasons include network and user-generated reasons. See Table A-1 for the call end reasons. See Table A-2 for the verbose call end reasons.

The Call End Reason TLV has been kept for backward compatibility. All new QMI clients must use the Verbose Call End Reason TLV. Any new call end reason will be added to the Verbose Call End Reason TLV.

## 3.8 QMI\_QCMAP\_SET\_IPSEC\_VPN\_PASS\_THROUGH

Configures the Internet Protocol security (IPSec) Virtual Private Network (VPN) passthrough setting.

### QCMAP message ID

0x0026

### Version introduced

Major - 1, Minor - 0

### 3.8.1 Request - QMI\_QCMAP\_SET\_IPSEC\_VPN\_PASS\_THROUGH\_REQ

#### Message type

Request

#### Sender

Control point

#### Mandatory TLVs

Name	Version introduced	Version last modified
Mobile AP Handle	1.0	1.0
VPN Passthrough Value	1.0	1.0

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Mobile AP Handle
Length	4			2	
Value	→	uint32	mobile_ap_handle	4	Handle identifying the mobile AP call instance. The value must be the handle previously returned by QMI_QCMAP_MOBILE_AP_ENABLE_REQ.
Type	0x02			1	VPN Passthrough Value
Length	1			2	
Value	→	boolean	vpn_pass_through_value	1	Indicates whether an IPSec VPN passthrough is allowed; boolean value.

**Optional TLVs**

None

**3.8.2 Response - QMI\_QCMAP\_SET\_IPSEC\_VPN\_PASS\_THROUGH\_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	Some TLV was missing
QMI_ERR_INVALID_HANDLE	Mobile AP handle provided in the request is not valid, i.e., it is not assigned to the control point
QMI_ERR_INVALID_ARG	Argument is not correct
QMI_ERR_NOT_SUPPORTED	Operation is not supported

**3.8.3 Description of QMI\_QCMAP\_SET\_IPSEC\_VPN\_PASS\_THROUGH\_REQ/RESP**

This command sets the IPsec VPN passthrough on the device. The command handler overwrites any previously configured value with the current value.



## 3.9 QMI\_QCMAP\_GET\_IPSEC\_VPN\_PASS\_THROUGH

Queries the IPsec VPN passthrough setting.

### QCMAP message ID

0x0025

### Version introduced

Major - 1, Minor - 0

### 3.9.1 Request - QMI\_QCMAP\_GET\_IPSEC\_VPN\_PASS\_THROUGH\_REQ

#### Message type

Request

#### Sender

Control point

#### Mandatory TLVs

Name	Version introduced	Version last modified
Mobile AP Handle	1.0	1.0

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Mobile AP Handle
Length	4			2	
Value	→	uint32	mobile_ap_handle	4	Handle identifying the mobile AP call instance. The value must be the handle previously returned by QMI_QCMAP_MOBILE_AP_ENABLE_REQ.

#### Optional TLVs

None

### 3.9.2 Response - QMI\_QCMAP\_GET\_IPSEC\_VPN\_PASS\_THROUGH\_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

Name	Version introduced	Version last modified
VPN Passthrough Value	1.0	1.0

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	VPN Passthrough Value
Length	1			2	
Value	→	boolean	vpn_pass_through_value	1	Indicates whether an IPsec VPN passthrough is allowed; boolean value.

#### Error codes

QMI_ERR_NONE	No error in the request
QMI_ERR_INTERNAL	Unexpected error occurred during processing
QMI_ERR_MALFORMED_MSG	Message was not formulated correctly by the control point or the message was corrupted during transmission
QMI_ERR_MISSING_ARG	Some TLV was missing
QMI_ERR_INVALID_HANDLE	Mobile AP handle provided in the request is not valid, i.e., it is not assigned to the control point
QMI_ERR_NOT_SUPPORTED	Operation is not supported

### 3.9.3 Description of QMI\_QCMAP\_GET\_IPSEC\_VPN\_PASS\_THROUGH\_REQ/RESP

This command queries the IPsec VPN passthrough value on the device.

## 3.10 QMI\_QCMAP\_SET\_PPTP\_VPN\_PASS\_THROUGH

Configures the Point-to-Point Tunneling Protocol (PPTP) VPN passthrough setting.

### QCMAP message ID

0x0028

### Version introduced

Major - 1, Minor - 0

### 3.10.1 Request - QMI\_QCMAP\_SET\_PPTP\_VPN\_PASS\_THROUGH\_REQ

#### Message type

Request

#### Sender

Control point

#### Mandatory TLVs

Name	Version introduced	Version last modified
Mobile AP Handle	1.0	1.0
VPN Passthrough Value	1.0	1.0

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Mobile AP Handle
Length	4			2	
Value	→	uint32	mobile_ap_handle	4	Handle identifying the mobile AP call instance. The value must be the handle previously returned by QMI_QCMAP_MOBILE_AP_ENABLE_REQ.
Type	0x02			1	VPN Passthrough Value
Length	1			2	
Value	→	boolean	vpn_pass_through_value	1	Indicates whether an IPSec VPN passthrough is allowed; boolean value.

**Optional TLVs**

None

**3.10.2 Response - QMI\_QCMAP\_SET\_PPTP\_VPN\_PASS\_THROUGH\_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	Some TLV was missing
QMI_ERR_INVALID_HANDLE	Mobile AP handle provided in the request is not valid, i.e., it is not assigned to the control point
QMI_ERR_INVALID_ARG	Argument is not correct
QMI_ERR_NOT_SUPPORTED	Operation is not supported

**3.10.3 Description of QMI\_QCMAP\_SET\_PPTP\_VPN\_PASS\_THROUGH\_REQ/RESP**

This command sets the PPTP VPN passthrough on the device. The command handler overwrites any previously configured value with the current value.

## 3.11 QMI\_QCMAP\_GET\_PPTP\_VPN\_PASS\_THROUGH

Queries the PPTP VPN passthrough setting.

### QCMAP message ID

0x0027

### Version introduced

Major - 1, Minor - 0

### 3.11.1 Request - QMI\_QCMAP\_GET\_PPTP\_VPN\_PASS\_THROUGH\_REQ

#### Message type

Request

#### Sender

Control point

#### Mandatory TLVs

Name	Version introduced	Version last modified
Mobile AP Handle	1.0	1.0

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Mobile AP Handle
Length	4			2	
Value	→	uint32	mobile_ap_handle	4	Handle identifying the mobile AP call instance. The value must be the handle previously returned by QMI_QCMAP_MOBILE_AP_ENABLE_REQ.

#### Optional TLVs

None

### 3.11.2 Response - QMI\_QCMAP\_GET\_PPTP\_VPN\_PASS\_THROUGH\_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

Name	Version introduced	Version last modified
VPN Passthrough Value	1.0	1.0

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	VPN Passthrough Value
Length	1			2	
Value	→	boolean	vpn_pass_through_value	1	Indicates whether an IPsec VPN passthrough is allowed; boolean value.

#### Error codes

QMI_ERR_NONE	No error in the request
QMI_ERR_INTERNAL	Unexpected error occurred during processing
QMI_ERR_MALFORMED_MSG	Message was not formulated correctly by the control point or the message was corrupted during transmission
QMI_ERR_MISSING_ARG	Some TLV was missing
QMI_ERR_INVALID_HANDLE	Mobile AP handle provided in the request is not valid, i.e., it is not assigned to the control point
QMI_ERR_NOT_SUPPORTED	Operation is not supported

### 3.11.3 Description of QMI\_QCMAP\_GET\_PPTP\_VPN\_PASS\_THROUGH\_REQ/RESP

This command queries the PPTP VPN passthrough value on the device.

## 3.12 QMI\_QCMAP\_SET\_L2TP\_VPN\_PASS\_THROUGH

Configures the Layer 2 Tunneling Protocol (L2TP) VPN passthrough setting.

### QCMAP message ID

0x002A

### Version introduced

Major - 1, Minor - 0

### 3.12.1 Request - QMI\_QCMAP\_SET\_L2TP\_VPN\_PASS\_THROUGH\_REQ

#### Message type

Request

#### Sender

Control point

#### Mandatory TLVs

Name	Version introduced	Version last modified
Mobile AP Handle	1.0	1.0
VPN Passthrough Value	1.0	1.0

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Mobile AP Handle
Length	4			2	
Value	→	uint32	mobile_ap_handle	4	Handle identifying the mobile AP call instance. The value must be the handle previously returned by QMI_QCMAP_MOBILE_AP_ENABLE_REQ.
Type	0x02			1	VPN Passthrough Value
Length	1			2	
Value	→	boolean	vpn_pass_through_value	1	Indicates whether an IPSec VPN passthrough is allowed; boolean value.

**Optional TLVs**

None

**3.12.2 Response - QMI\_QCMAP\_SET\_L2TP\_VPN\_PASS\_THROUGH\_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	Some TLV was missing
QMI_ERR_INVALID_HANDLE	Mobile AP handle provided in the request is not valid, i.e., it is not assigned to the control point
QMI_ERR_INVALID_ARG	Argument is not correct
QMI_ERR_NOT_SUPPORTED	Operation is not supported

**3.12.3 Description of QMI\_QCMAP\_SET\_L2TP\_VPN\_PASS\_THROUGH\_REQ/RESP**

This command sets the L2TP VPN passthrough on the device. The command handler overwrites any previously configured value with the current value.



### 3.13 QMI\_QCMAP\_GET\_L2TP\_VPN\_PASS\_THROUGH

Queries the L2TP VPN passthrough setting.

#### QCMAP message ID

0x0029

#### Version introduced

Major - 1, Minor - 0

#### 3.13.1 Request - QMI\_QCMAP\_GET\_L2TP\_VPN\_PASS\_THROUGH\_REQ

##### Message type

Request

##### Sender

Control point

##### Mandatory TLVs

Name	Version introduced	Version last modified
Mobile AP Handle	1.0	1.0

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Mobile AP Handle
Length	4			2	
Value	→	uint32	mobile_ap_handle	4	Handle identifying the mobile AP call instance. The value must be the handle previously returned by QMI_QCMAP_MOBILE_AP_ENABLE_REQ.

##### Optional TLVs

None

### 3.13.2 Response - QMI\_QCMAP\_GET\_L2TP\_VPN\_PASS\_THROUGH\_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

Name	Version introduced	Version last modified
VPN Passthrough Value	1.0	1.0

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	VPN Passthrough Value
Length	1			2	
Value	→	boolean	vpn_pass_through_value	1	Indicates whether an IPsec VPN passthrough is allowed; boolean value.

#### Error codes

QMI_ERR_NONE	No error in the request
QMI_ERR_INTERNAL	Unexpected error occurred during processing
QMI_ERR_MALFORMED_MSG	Message was not formulated correctly by the control point or the message was corrupted during transmission
QMI_ERR_MISSING_ARG	Some TLV was missing
QMI_ERR_INVALID_HANDLE	Mobile AP handle provided in the request is not valid, i.e., it is not assigned to the control point
QMI_ERR_NOT_SUPPORTED	Operation is not supported

### 3.13.3 Description of QMI\_QCMAP\_GET\_L2TP\_VPN\_PASS\_THROUGH\_REQ/RESP

This command queries the L2TP VPN passthrough value on the device.

## 3.14 QMI\_QCMAP\_SET\_DYNAMIC\_NAT\_ENTRY\_TIMEOUT

Sets the Network Address Translation (NAT) entry timeout.

### QCMAP message ID

0x002C

### Version introduced

Major - 1, Minor - 0

### 3.14.1 Request - QMI\_QCMAP\_SET\_DYNAMIC\_NAT\_ENTRY\_TIMEOUT\_REQ

#### Message type

Request

#### Sender

Control point

#### Mandatory TLVs

Name	Version introduced	Version last modified
Mobile AP Handle	1.0	1.0
Timeout	1.0	1.0

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Mobile AP Handle
Length	4			2	
Value	→	uint32	mobile_ap_handle	4	Handle identifying the mobile AP call instance. The value must be the handle previously returned by QMI_QCMAP_MOBILE_AP_ENABLE_REQ.
Type	0x02			1	Timeout
Length	2			2	
Value	→	uint16	timeout	2	NAT entry timeout.

**Optional TLVs**

None

**3.14.2 Response - QMI\_QCMAP\_SET\_DYNAMIC\_NAT\_ENTRY\_-  
TIMEOUT\_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	Some TLV was missing
QMI_ERR_INVALID_HANDLE	Mobile AP handle provided in the request is not valid, i.e., it is not assigned to the control point
QMI_ERR_INVALID_ARG	Argument is not correct
QMI_ERR_NOT_SUPPORTED	Operation is not supported

**3.14.3 Description of QMI\_QCMAP\_SET\_DYNAMIC\_NAT\_ENTRY\_-  
TIMEOUT\_REQ/RESP**

This command sets the NAT entry timeout on the device.

## 3.15 QMI\_QCMAP\_GET\_DYNAMIC\_NAT\_ENTRY\_TIMEOUT

Queries the NAT entry timeout.

### QCMAP message ID

0x002B

### Version introduced

Major - 1, Minor - 0

### 3.15.1 Request - QMI\_QCMAP\_GET\_DYNAMIC\_NAT\_ENTRY\_TIMEOUT\_REQ

#### Message type

Request

#### Sender

Control point

#### Mandatory TLVs

Name	Version introduced	Version last modified
Mobile AP Handle	1.0	1.0

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Mobile AP Handle
Length	4			2	
Value	→	uint32	mobile_ap_handle	4	Handle identifying the mobile AP call instance. The value must be the handle previously returned by QMI_QCMAP_MOBILE_AP_ENABLE_REQ.

#### Optional TLVs

None

### 3.15.2 Response - QMI\_QCMAP\_GET\_DYNAMIC\_NAT\_ENTRY\_- TIMEOUT\_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

Name	Version introduced	Version last modified
Timeout	1.0	1.0

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Timeout
Length	2			2	
Value	→	uint16	timeout	2	Dynamic NAT entry timeout.

#### 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	Some TLV was missing
QMI_ERR_INVALID_HANDLE	Mobile AP handle provided in the request is not valid, i.e., it is not assigned to the control point
QMI_ERR_NOT_SUPPORTED	Operation is not supported

### 3.15.3 Description of QMI\_QCMAP\_GET\_DYNAMIC\_NAT\_ENTRY\_- TIMEOUT REQ/RESP

This command queries the NAT entry timeout on the device.

## 3.16 QMI\_QCMAP\_ADD\_STATIC\_NAT\_ENTRY

Adds a static NAT entry.

### QCMAP message ID

0x002D

### Version introduced

Major - 1, Minor - 0

### 3.16.1 Request - QMI\_QCMAP\_ADD\_STATIC\_NAT\_ENTRY\_REQ

#### Message type

Request

#### Sender

Control point

#### Mandatory TLVs

Name	Version introduced	Version last modified
Mobile AP Handle	1.0	1.0
SNAT Entry Configuration	1.0	1.0

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Mobile AP Handle
Length	4			2	
Value	→	uint32	mobile_ap_handle	4	Handle identifying the mobile AP call instance. The value must be the handle previously returned by QMI_QCMAP_MOBILE_AP_ENABLE_REQ.
Type	0x02			1	SNAT Entry Configuration
Length	9			2	
Value	→	uint32	private_ip_addr	4	Private IP address.
		uint16	private_port	2	Private port.
		uint16	global_port	2	Global port.
		uint8	protocol	1	Protocol.

**Optional TLVs**

None

**3.16.2 Response - QMI\_QCMAP\_ADD\_STATIC\_NAT\_ENTRY\_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	Some TLV was missing
QMI_ERR_INVALID_HANDLE	Mobile AP handle provided in the request is not valid, i.e., it is not assigned to the control point
QMI_ERR_INVALID_ARG	Argument is not correct
QMI_ERR_NOT_SUPPORTED	Operation is not supported
QMI_ERR_MAX_LIMIT_REACHED	Maximum limit was reached for the static NAT entry
QMI_ERR_DUPLICATE_ENTRY	Entry already exists

**3.16.3 Description of QMI\_QCMAP\_ADD\_STATIC\_NAT\_ENTRY REQ/RESP**

This command adds a static NAT entry.



## 3.17 QMI\_QCMAP\_DELETE\_STATIC\_NAT\_ENTRY

Deletes a static NAT entry.

### QCMAP message ID

0x002E

### Version introduced

Major - 1, Minor - 0

### 3.17.1 Request - QMI\_QCMAP\_DELETE\_STATIC\_NAT\_ENTRY\_REQ

#### Message type

Request

#### Sender

Control point

#### Mandatory TLVs

Name	Version introduced	Version last modified
Mobile AP Handle	1.0	1.0
SNAT Entry Configuration	1.0	1.0

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Mobile AP Handle
Length	4			2	
Value	→	uint32	mobile_ap_handle	4	Handle identifying the mobile AP call instance. The value must be the handle previously returned by QMI_QCMAP_MOBILE_AP_ENABLE_REQ.
Type	0x02			1	SNAT Entry Configuration
Length	9			2	
Value	→	uint32	private_ip_addr	4	Private IP address.
		uint16	private_port	2	Private port.
		uint16	global_port	2	Global port.
		uint8	protocol	1	Protocol.

**Optional TLVs**

None

**3.17.2 Response - QMI\_QCMAP\_DELETE\_STATIC\_NAT\_ENTRY\_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	Some TLV was missing
QMI_ERR_INVALID_HANDLE	Mobile AP handle provided in the request is not valid, i.e., it is not assigned to the control point
QMI_ERR_INVALID_ARG	Argument is not correct
QMI_ERR_NOT_SUPPORTED	Operation is not supported
QMI_ERR_NO_ENTRY	Entry was not found

**3.17.3 Description of QMI\_QCMAP\_DELETE\_STATIC\_NAT\_ENTRY REQ/RESP**

This command deletes a static NAT entry.

## 3.18 QMI\_QCMAP\_GET\_STATIC\_NAT\_ENTRIES

Queries all static NAT entries.

### QCMAP message ID

0x002F

### Version introduced

Major - 1, Minor - 0

### 3.18.1 Request - QMI\_QCMAP\_GET\_STATIC\_NAT\_ENTRIES\_REQ

#### Message type

Request

#### Sender

Control point

#### Mandatory TLVs

Name	Version introduced	Version last modified
Mobile AP Handle	1.0	1.0

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Mobile AP Handle
Length	4			2	
Value	→	uint32	mobile_ap_handle	4	Handle identifying the mobile AP call instance. The value must be the handle previously returned by QMI_QCMAP_MOBILE_AP_ENABLE_REQ.

#### Optional TLVs

None

### 3.18.2 Response - QMI\_QCMAP\_GET\_STATIC\_NAT\_ENTRIES\_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

Name	Version introduced	Version last modified
SNAT Configuration	1.0	1.0

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	SNAT Configuration
Length	Var			2	
Value	→	uint8	snat_config_len	1	Number of sets of the following elements: • private_ip_addr • private_port • global_port • protocol
		uint32	private_ip_addr	4	Private IP address.
		uint16	private_port	2	Private port.
		uint16	global_port	2	Global port.
		uint8	protocol	1	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_MISSING_ARG	Some TLV was missing
QMI_ERR_INVALID_HANDLE	Mobile AP handle provided in the request is not valid, i.e., it is not assigned to the control point
QMI_ERR_INVALID_ARG	Argument is not correct
QMI_ERR_NOT_SUPPORTED	Operation is not supported

### 3.18.3 Description of QMI\_QCMAP\_GET\_STATIC\_NAT\_ENTRIES REQ/RESP

This command queries all static NAT entries. The response message contains the number of entries followed by the value of these entries sequentially.

Qualcomm  
Confidential - May Contain Trade Secrets  
2022-07-21 08:18:44 GMT  
jason1\_gao@askey.com

## 3.19 QMI\_QCMAP\_SET\_DMZ

Sets the DMZ (perimeter network) IP address for the mobile AP.

### QCMAP message ID

0x0030

### Version introduced

Major - 1, Minor - 0

### 3.19.1 Request - QMI\_QCMAP\_SET\_DMZ\_REQ

#### Message type

Request

#### Sender

Control point

#### Mandatory TLVs

Name	Version introduced	Version last modified
Mobile AP Handle	1.0	1.0
DMZ IP Address	1.0	1.0

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Mobile AP Handle
Length	4			2	
Value	→	uint32	mobile_ap_handle	4	Handle identifying the mobile AP call instance. The value must be the handle previously returned by QMI_QCMAP_MOBILE_AP_ENABLE_REQ.
Type	0x02			1	DMZ IP Address
Length	4			2	
Value	→	uint32	dmz_ip_addr	4	DMZ IP address.

#### Optional TLVs

None

### 3.19.2 Response - QMI\_QCMAP\_SET\_DMZ\_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	Some TLV was missing
QMI_ERR_INVALID_HANDLE	Mobile AP handle provided in the request is not valid, i.e., it is not assigned to the control point
QMI_ERR_INVALID_ARG	Argument is not correct
QMI_ERR_NOT_SUPPORTED	Operation is not supported

### 3.19.3 Description of QMI\_QCMAP\_SET\_DMZ REQ/RESP

This command sets the DMZ IP address for the mobile AP.

## 3.20 QMI\_QCMAP\_GET\_DMZ

Queries the DMZ IP address on the mobile AP.

### QCMAP message ID

0x0032

### Version introduced

Major - 1, Minor - 0

### 3.20.1 Request - QMI\_QCMAP\_GET\_DMZ\_REQ

#### Message type

Request

#### Sender

Control point

#### Mandatory TLVs

Name	Version introduced	Version last modified
Mobile AP Handle	1.0	1.0

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Mobile AP Handle
Length	4			2	
Value	→	uint32	mobile_ap_handle	4	Handle identifying the mobile AP call instance. The value must be the handle previously returned by QMI_QCMAP_MOBILE_AP_ENABLE_REQ.

#### Optional TLVs

None



### 3.20.2 Response - QMI\_QCMAP\_GET\_DMZ\_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

Name	Version introduced	Version last modified
DMZ IP Address	1.0	1.0

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	DMZ IP Address
Length	4			2	
Value	→	uint32	dmz_ip_addr	4	DMZ IP address.

#### 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	Some TLV was missing
QMI_ERR_INVALID_HANDLE	Mobile AP handle provided in the request is not valid, i.e., it is not assigned to the control point
QMI_ERR_INVALID_ARG	Argument is not correct
QMI_ERR_NOT_SUPPORTED	Operation is not supported

### 3.20.3 Description of QMI\_QCMAP\_GET\_DMZ REQ/RESP

This command queries the DMZ entry that was previously set via QMI\_QCMAP\_SET\_DMZ.

If no DMZ is set at the modem, an IP address of 0.0.0.0 is returned.

## 3.21 QMI\_QCMAP\_DELETE\_DMZ

Deletes the DMZ entry or DMZ IP address.

### QCMAP message ID

0x0031

### Version introduced

Major - 1, Minor - 0

### 3.21.1 Request - QMI\_QCMAP\_DELETE\_DMZ\_REQ

#### Message type

Request

#### Sender

Control point

#### Mandatory TLVs

Name	Version introduced	Version last modified
Mobile AP Handle	1.0	1.0

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Mobile AP Handle
Length	4			2	
Value	→	uint32	mobile_ap_handle	4	Handle identifying the mobile AP call instance. The value must be the handle previously returned by QMI_QCMAP_MOBILE_AP_ENABLE_REQ.

#### Optional TLVs

None

### 3.21.2 Response - QMI\_QCMAP\_DELETE\_DMZ\_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	Some TLV was missing
QMI_ERR_INVALID_HANDLE	Mobile AP handle provided in the request is not valid, i.e., it is not assigned to the control point
QMI_ERR_INVALID_ARG	Argument is not correct
QMI_ERR_NOT_SUPPORTED	Operation is not supported

### 3.21.3 Description of QMI\_QCMAP\_DELETE\_DMZ REQ/RESP

This command deletes the DMZ entry that was previously set via QMI\_QCMAP\_SET\_DMZ.

## 3.22 QMI\_QCMAP\_GET\_WWAN\_CONFIG

Queries the WWAN IP configuration.

### QCMAP message ID

0x0033

### Version introduced

Major - 1, Minor - 0

### 3.22.1 Request - QMI\_QCMAP\_GET\_WWAN\_CONFIG\_REQ

#### Message type

Request

#### Sender

Control point

#### Mandatory TLVs

Name	Version introduced	Version last modified
Mobile AP Handle	1.0	1.0
Address Type	1.0	1.0

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Mobile AP Handle
Length	4			2	
Value	→	uint32	mobile_ap_handle	4	Handle identifying the mobile AP call instance. The value must be the handle previously returned by QMI_QCMAP_MOBILE_AP_ENABLE_REQ.
Type	0x02			1	Address Type
Length	8			2	
Value	→	mask	addr_type_op	8	WWAN configuration mask values: <ul style="list-style-type: none"> <li>• 1 – IPv4 address</li> <li>• 2 – IPv6 address</li> <li>• 4 – IPv4 DNS address</li> <li>• 8 – IPv6 DNS address</li> </ul>

**Optional TLVs**

None

**3.22.2 Response - QMI\_QCMAP\_GET\_WWAN\_CONFIG\_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**

Name	Version introduced	Version last modified
IPv4 Address	1.0	1.0
IPv6 Address	1.0	1.0
IPv4 Primary DNS Address	1.0	1.0
IPv4 Secondary DNS Address	1.0	1.0
IPv6 Primary DNS Address	1.0	1.0
IPv6 Secondary DNS Address	1.0	1.0

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	IPv4 Address
Length	4			2	
Value	→	uint32	v4_addr	4	IPv4 address.
Type	0x11			1	IPv6 Address
Length	16			2	
Value	→	uint8	v6_addr	16	IPv6 address.
Type	0x12			1	IPv4 Primary DNS Address
Length	4			2	
Value	→	uint32	v4_prim_dns_addr	4	IPv4 primary DNS address.
Type	0x13			1	IPv4 Secondary DNS Address
Length	4			2	
Value	→	uint32	v4_sec_dns_addr	4	IPv4 secondary DNS address.
Type	0x14			1	IPv6 Primary DNS Address
Length	16			2	
Value	→	uint8	v6_prim_dns_addr	16	IPv6 primary DNS address.
Type	0x15			1	IPv6 Secondary DNS Address
Length	16			2	
Value	→	uint8	v6_sec_dns_addr	16	IPv6 secondary DNS address.

**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	Some TLV was missing
QMI_ERR_INVALID_HANDLE	Mobile AP handle provided in the request is not valid, i.e., it is not assigned to the control point
QMI_ERR_INVALID_ARG	Argument is not correct
QMI_ERR_NOT_SUPPORTED	Operation is not supported

**3.22.3 Description of QMI\_QCMAP\_GET\_WWAN\_CONFIG REQ/RESP**

This command queries the WWAN IP configuration for the mobile AP. The command must be issued by the control point after QCMAP\_WWAN\_STATUS\_IND has indicated a successful WWAN bringup, otherwise a QMI\_ERR\_INTERNAL error is returned.

Qualcomm  
Confidential - May Contain Trade Secrets  
2022-07-21 08:18:44 GMT  
jason1\_gao@askey.com

## 3.23 QMI\_QCMAP\_ENABLE\_FIREWALL\_SETTING

Enables the firewall setting.

### QCMAP message ID

0x0034

### Version introduced

Major - 1, Minor - 0

### 3.23.1 Request - QMI\_QCMAP\_ENABLE\_FIREWALL\_SETTING\_REQ

#### Message type

Request

#### Sender

Control point

#### Mandatory TLVs

Name	Version introduced	Version last modified
Mobile AP Handle	1.0	1.0
Packets Allowed	1.0	1.0

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Mobile AP Handle
Length	4			2	
Value	→	uint32	mobile_ap_handle	4	Handle identifying the mobile AP call instance. The value must be the handle previously returned by QMI_QCMAP_MOBILE_AP_ENABLE_REQ.
Type	0x02			1	Packets Allowed
Length	1			2	
Value	→	boolean	pkts_allowed	1	Packets allowed operation. Values: • TRUE – Packets matching the firewall rule are allowed • FALSE – Packets matching the firewall rule are dropped

**Optional TLVs**

None

**3.23.2 Response - QMI\_QCMAP\_ENABLE\_FIREWALL\_SETTING\_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	Some TLV was missing
QMI_ERR_INVALID_HANDLE	Mobile AP handle provided in the request is not valid, i.e., it is not assigned to the control point
QMI_ERR_INVALID_ARG	Argument is not correct
QMI_ERR_NOT_SUPPORTED	Operation is not supported

**3.23.3 Description of QMI\_QCMAP\_ENABLE\_FIREWALL\_SETTING REQ/RESP**

This command enables the firewall and sets the condition whether the packets matching the firewall rule are to be allowed or dropped.



## 3.24 QMI\_QCMAP\_GET\_FIREWALL\_SETTING

Queries the firewall setting.

### QCMAP message ID

0x0035

### Version introduced

Major - 1, Minor - 0

### 3.24.1 Request - QMI\_QCMAP\_GET\_FIREWALL\_SETTING\_REQ

#### Message type

Request

#### Sender

Control point

#### Mandatory TLVs

Name	Version introduced	Version last modified
Mobile AP Handle	1.0	1.0

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Mobile AP Handle
Length	4			2	
Value	→	uint32	mobile_ap_handle	4	Handle identifying the mobile AP call instance. The value must be the handle previously returned by QMI_QCMAP_MOBILE_AP_ENABLE_REQ.

#### Optional TLVs

None

### 3.24.2 Response - QMI\_QCMAP\_GET\_FIREWALL\_SETTING\_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

Name	Version introduced	Version last modified
Firewall Enabled	1.0	1.0
Packets Allowed	1.0	1.0

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Firewall Enabled
Length	1			2	
Value	→	boolean	firewall_enabled	1	Whether the firewall is enabled; boolean value.
Type	0x11			1	Packets Allowed
Length	1			2	
Value	→	boolean	pkts_allowed	1	Whether packets are allowed; boolean value.

#### Error codes

QMI_ERR_NONE	No error in the request
QMI_ERR_INTERNAL	Unexpected error occurred during processing
QMI_ERR_MALFORMED_MSG	Message was not formulated correctly by the control point or the message was corrupted during transmission
QMI_ERR_MISSING_ARG	Some TLV was missing
QMI_ERR_INVALID_HANDLE	Mobile AP handle provided in the request is not valid, i.e., it is not assigned to the control point
QMI_ERR_NOT_SUPPORTED	Operation is not supported

### 3.24.3 Description of QMI\_QCMAP\_GET\_FIREWALL\_SETTING REQ/RESP

This command queries the firewall setting.

Qualcomm  
Confidential - May Contain Trade Secrets  
2022-07-21 08:18:44 GMT  
jason1\_gao@askey.com

## 3.25 QMI\_QCMAP\_DISABLE\_FIREWALL\_SETTING

Disables the firewall setting.

### QCMAP message ID

0x0036

### Version introduced

Major - 1, Minor - 0

### 3.25.1 Request - QMI\_QCMAP\_DISABLE\_FIREWALL\_SETTING\_REQ

#### Message type

Request

#### Sender

Control point

#### Mandatory TLVs

Name	Version introduced	Version last modified
Mobile AP Handle	1.0	1.0

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Mobile AP Handle
Length	4			2	
Value	→	uint32	mobile_ap_handle	4	Handle identifying the mobile AP call instance. The value must be the handle previously returned by QMI_QCMAP_MOBILE_AP_ENABLE_REQ.

#### Optional TLVs

None

### 3.25.2 Response - QMI\_QCMAP\_DISABLE\_FIREWALL\_SETTING\_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	Some TLV was missing
QMI_ERR_INVALID_HANDLE	Mobile AP handle provided in the request is not valid, i.e., it is not assigned to the control point
QMI_ERR_NOT_SUPPORTED	Operation is not supported

### 3.25.3 Description of QMI\_QCMAP\_DISABLE\_FIREWALL\_SETTING\_REQ/RESP

This command disables the firewall setting.

## 3.26 QMI\_QCMAP\_ADD\_FIREWALL\_CONFIG

Adds a firewall configuration rule.

### QCMAP message ID

0x0037

### Version introduced

Major - 1, Minor - 0

### 3.26.1 Request - QMI\_QCMAP\_ADD\_FIREWALL\_CONFIG\_REQ

#### Message type

Request

#### Sender

Control point

#### Mandatory TLVs

Name	Version introduced	Version last modified
Mobile AP Handle	1.0	1.0
Firewall Configuration	1.0	1.0

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Mobile AP Handle
Length	4			2	
Value	→	uint32	mobile_ap_handle	4	Handle identifying the mobile AP call instance. The value must be the handle previously returned by QMI_QCMAP_MOBILE_AP_ENABLE_REQ.
Type	0x02			1	Firewall Configuration
Length	5			2	
Value	→	uint16	start_dest_port	2	Start value of the destination port range.
		uint16	end_dest_port	2	End value of the destination port range.
		uint8	protocol	1	Protocol value.

**Optional TLVs**

None

**3.26.2 Response - QMI\_QCMAP\_ADD\_FIREWALL\_CONFIG\_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**

Name	Version introduced	Version last modified
Firewall Handle	1.0	1.0

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Firewall Handle
Length	4			2	
Value	→	uint32	firewall_handle	4	Handle identifying the firewall rule.

**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	Some TLV was missing
QMI_ERR_INVALID_HANDLE	Mobile AP handle provided in the request is not valid, i.e., it is not assigned to the control point
QMI_ERR_INVALID_ARG	Argument is not correct
QMI_ERR_NOT_SUPPORTED	Operation is not supported
QMI_ERR_NO_MEMORY	Maximum number of supported firewall rules was exceeded; cannot add any more firewall rules

### 3.26.3 Description of QMI\_QCMAP\_ADD\_FIREWALL\_CONFIG REQ/RESP

This command adds a firewall configuration rule.

Qualcomm  
Confidential - May Contain Trade Secrets  
2022-07-21 08:18:44 GMT  
jason1\_gao@askey.com



## 3.27 QMI\_QCMAP\_DELETE\_FIREWALL\_CONFIG

Deletes a firewall configuration rule.

### QCMAP message ID

0x0039

### Version introduced

Major - 1, Minor - 0

### 3.27.1 Request - QMI\_QCMAP\_DELETE\_FIREWALL\_CONFIG\_REQ

#### Message type

Request

#### Sender

Control point

#### Mandatory TLVs

Name	Version introduced	Version last modified
Mobile AP Handle	1.0	1.0
Firewall Handle	1.0	1.0

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Mobile AP Handle
Length	4			2	
Value	→	uint32	mobile_ap_handle	4	Handle identifying the mobile AP call instance. The value must be the handle previously returned by QMI_QCMAP_MOBILE_AP_ENABLE_REQ.
Type	0x02			1	Firewall Handle
Length	4			2	
Value	→	uint32	firewall_handle	4	Handle identifying the firewall entry. The value must be the handle previously returned by QMI_QCMAP_ADD_FIREWALL_CONFIG_RESP or QMI_QCMAP_GET_FIREWALL_CONFIG_RESP.

**Optional TLVs**

None

**3.27.2 Response - QMI\_QCMAP\_DELETE\_FIREWALL\_CONFIG\_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	Some TLV was missing
QMI_ERR_INVALID_HANDLE	Mobile AP handle provided in the request is not valid, i.e., it is not assigned to the control point
QMI_ERR_INVALID_ARG	Argument is not correct
QMI_ERR_NOT_SUPPORTED	Operation is not supported

**3.27.3 Description of QMI\_QCMAP\_DELETE\_FIREWALL\_CONFIG REQ/RESP**

This command deletes a firewall rule.

## 3.28 QMI\_QCMAP\_GET\_FIREWALL\_CONFIG

Queries the firewall configuration rules.

### QCMAP message ID

0x0038

### Version introduced

Major - 1, Minor - 0

### 3.28.1 Request - QMI\_QCMAP\_GET\_FIREWALL\_CONFIG\_REQ

#### Message type

Request

#### Sender

Control point

#### Mandatory TLVs

Name	Version introduced	Version last modified
Mobile AP Handle	1.0	1.0

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Mobile AP Handle
Length	4			2	
Value	→	uint32	mobile_ap_handle	4	Handle identifying the mobile AP call instance. The value must be the handle previously returned by QMI_QCMAP_MOBILE_AP_ENABLE_REQ.

#### Optional TLVs

None

### 3.28.2 Response - QMI\_QCMAP\_GET\_FIREWALL\_CONFIG\_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

Name	Version introduced	Version last modified
Firewall Configuration	1.0	1.0

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Firewall Configuration
Length	Var			2	
Value	→	uint8	firewall_config_len	1	Number of sets of the following elements: <ul style="list-style-type: none"> <li>• firewall_handle</li> <li>• start_dest_port</li> <li>• end_dest_port</li> <li>• protocol</li> </ul>
		uint32	firewall_handle	4	Handle identifying the firewall rule.
		uint16	start_dest_port	2	Start value of the destination port range.
		uint16	end_dest_port	2	End value of the destination port range.
		uint8	protocol	1	Protocol value.

#### Error codes

QMI_ERR_NONE	No error in the request
QMI_ERR_INTERNAL	Unexpected error occurred during processing
QMI_ERR_MALFORMED_MSG	Message was not formulated correctly by the control point or the message was corrupted during transmission
QMI_ERR_MISSING_ARG	Some TLV was missing
QMI_ERR_INVALID_HANDLE	Mobile AP handle provided in the request is not valid, i.e., it is not assigned to the control point
QMI_ERR_NOT_SUPPORTED	Operation is not supported

### 3.28.3 Description of QMI\_QCMAP\_GET\_FIREWALL\_CONFIG REQ/RESP

This command queries all the firewall entries. The response message contains the number of entries followed by the value of these entries sequentially.

Qualcomm  
Confidential - May Contain Trade Secrets  
2022-07-21 08:18:44 GMT  
jason1\_gao@askey.com

## 3.29 QMI\_QCMAP\_STATION\_MODE\_ENABLE

Enables Station (STA) mode functionality for a mobile AP instance on the modem.

### QCMAP message ID

0x003B

### Version introduced

Major - 1, Minor - 0

### 3.29.1 Request - QMI\_QCMAP\_STATION\_MODE\_ENABLE\_REQ

#### Message type

Request

#### Sender

Control point

#### Mandatory TLVs

Name	Version introduced	Version last modified
Mobile AP Handle	1.0	1.0

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Mobile AP Handle
Length	4			2	
Value	→	uint32	mobile_ap_handle	4	Handle identifying the mobile AP call instance. The value must be the handle previously returned by QMI_QCMAP_MOBILE_AP_ENABLE_REQ.

#### Optional TLVs

None

### 3.29.2 Response - QMI\_QCMAP\_STATION\_MODE\_ENABLE\_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	Some TLV was missing
QMI_ERR_INVALID_HANDLE	Mobile AP handle provided in the request is not valid, i.e., it is not assigned to the control point
QMI_ERR_NOT_SUPPORTED	Operation is not supported

### 3.29.3 Description of QMI\_QCMAP\_STATION\_MODE\_ENABLE\_REQ/RESP

This command enables STA mode functionality at the modem for a single mobile AP instance.

After this request is successfully processed, all packet connectivity to an outside network occurs through the WLAN station. The modem routing engine appropriately handles the packet routing into and out of the modem.

### 3.30 QMI\_QCMAP\_STATION\_MODE\_DISABLE

Disables STA mode functionality for a mobile AP instance on the modem.

#### QCMAP message ID

0x003C

#### Version introduced

Major - 1, Minor - 0

#### 3.30.1 Request - QMI\_QCMAP\_STATION\_MODE\_DISABLE\_REQ

##### Message type

Request

##### Sender

Control point

##### Mandatory TLVs

Name	Version introduced	Version last modified
Mobile AP Handle	1.0	1.0

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Mobile AP Handle
Length	4			2	
Value	→	uint32	mobile_ap_handle	4	Handle identifying the mobile AP call instance. The value must be the handle previously returned by QMI_QCMAP_MOBILE_AP_ENABLE_REQ.

##### Optional TLVs

None



### 3.30.2 Response - QMI\_QCMAP\_STATION\_MODE\_DISABLE\_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	Some TLV was missing
QMI_ERR_INVALID_HANDLE	Mobile AP handle provided in the request is not valid, i.e., it is not assigned to the control point
QMI_ERR_NOT_SUPPORTED	Operation is not supported

### 3.30.3 Description of QMI\_QCMAP\_STATION\_MODE\_DISABLE\_REQ/RESP

This command disables STA mode functionality at the modem for a single mobile AP instance. When this request has been successfully processed, the control point invokes bringing up the WWAN from the mobile AP.

### 3.31 QMI\_QCMAP\_GET\_STATION\_MODE

Queries the STA mode functionality for a mobile AP instance on the modem.

#### QCMAP message ID

0x003D

#### Version introduced

Major - 1, Minor - 0

#### 3.31.1 Request - QMI\_QCMAP\_GET\_STATION\_MODE\_REQ

##### Message type

Request

##### Sender

Control point

##### Mandatory TLVs

Name	Version introduced	Version last modified
Mobile AP Handle	1.0	1.0

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Mobile AP Handle
Length	4			2	
Value	→	uint32	mobile_ap_handle	4	Handle identifying the mobile AP call instance. The value must be the handle previously returned by QMI_QCMAP_MOBILE_AP_ENABLE_REQ.

##### Optional TLVs

None

### 3.31.2 Response - QMI\_QCMAP\_GET\_STATION\_MODE\_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

Name	Version introduced	Version last modified
Station Mode	1.0	1.0

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Station Mode
Length	1			2	
Value	→	boolean	station_mode	1	Whether STA mode has been enabled; boolean value.

#### Error codes

QMI_ERR_NONE	No error in the request
QMI_ERR_INTERNAL	Unexpected error occurred during processing
QMI_ERR_MALFORMED_MSG	Message was not formulated correctly by the control point or the message was corrupted during transmission
QMI_ERR_MISSING_ARG	Some TLV was missing
QMI_ERR_INVALID_HANDLE	Mobile AP handle provided in the request is not valid, i.e., it is not assigned to the control point
QMI_ERR_NOT_SUPPORTED	Operation is not supported

### 3.31.3 Description of QMI\_QCMAP\_GET\_STATION\_MODE REQ/RESP

This command queries the STA mode functionality at the modem for a single mobile AP instance.

## 3.32 QMI\_QCMAP\_ADD\_EXTD\_FIREWALL\_CONFIG

Adds IP filter-based firewall rules (extended firewall).

### QCMAP message ID

0x003F

### Version introduced

Major - 1, Minor - 1

### 3.32.1 Request - QMI\_QCMAP\_ADD\_EXTD\_FIREWALL\_CONFIG\_REQ

#### Message type

Request

#### Sender

Control point

#### Mandatory TLVs

Name	Version introduced	Version last modified
Mobile AP Handle	1.1	1.1
Next Header Protocol	1.1	1.1

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Mobile AP Handle
Length	4			2	
Value	→	uint32	mobile_ap_handle	4	Handle identifying the mobile AP call instance. The value must be the handle previously returned by QMI_QCMAP_MOBILE_AP_ENABLE_REQ.
Type	0x02			1	Next Header Protocol
Length	4			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	enum	next_hdr_prot	4	IPv4/IPv6 next header protocol after the IP header. Values: <ul style="list-style-type: none"> <li>• 0x01 – QCMAP_EXTD_FIREWALL_PROTO_TCP – Transmission Control Protocol</li> <li>• 0x02 – QCMAP_EXTD_FIREWALL_PROTO_UDP – User Datagram Protocol</li> <li>• 0x03 – QCMAP_EXTD_FIREWALL_PROTO_ICMP – Internet Control Message Protocol</li> <li>• 0x04 – QCMAP_EXTD_FIREWALL_PROTO_ICMP6 – Internet Control Message Protocol version 6</li> <li>• 0x05 – QCMAP_EXTD_FIREWALL_PROTO_ESP – Encapsulating Security Payload Protocol</li> <li>• 0x06 – QCMAP_EXTD_FIREWALL_PROTO_TCP_UDP – Transmission Control Protocol/User Datagram Protocol</li> </ul>

## Optional TLVs

Name	Version introduced	Version last modified
TCP/UDP Source	1.1	1.1
TCP/UDP Destination	1.1	1.1
ICMP Type	1.1	1.1
ICMP Code	1.1	1.1
ESP Security Parameters Index	1.1	1.1
IPv4 Source Address	1.1	1.1
IPv4 Destination Address	1.1	1.1
IPv4 TOS	1.1	1.1
IPv6 Source Address	1.1	1.1
IPv6 Destination Address	1.1	1.1
IPv6 Traffic Class	1.1	1.1

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	TCP/UDP Source
Length	4			2	
Value	→	uint16	port	2	TCP/UDP port as specified in the TCP/UDP protocol (RFC 793 [S4] and RFC 768 [S1]).
		uint16	range	2	TCP/UDP port range as specified in the TCP/UDP protocol (RFC 793 [S4] and RFC 768 [S1]).

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x11			1	TCP/UDP Destination
Length	4			2	
Value	→	uint16	port	2	TCP/UDP port as specified in the TCP/UDP protocol (RFC 793 [S4] and RFC 768 [S1]).
		uint16	range	2	TCP/UDP port range as specified in the TCP/UDP protocol (RFC 793 [S4] and RFC 768 [S1]).
Type	0x12			1	ICMP Type
Length	1			2	
Value	→	uint8	icmp_type	1	ICMP type as specified in the ICMP specification (RFC 792 [S3]).
Type	0x13			1	ICMP Code
Length	1			2	
Value	→	uint8	icmp_code	1	ICMP code as specified in the ICMP specification (RFC 792 [S3]).
Type	0x14			1	ESP Security Parameters Index
Length	4			2	
Value	→	uint32	esp_spi	4	Security parameters index as specified in the ESP protocol (RFC 4303 [S7]).
Type	0x15			1	IPv4 Source Address
Length	8			2	
Value	→	uint32	addr	4	IPv4 address as specified in the IPv4 protocol specification (RFC 791 [S2]).
		uint32	subnet_mask	4	IPv4 subnet mask as specified in the IPv4 protocol specification (RFC 791 [S2]).
Type	0x16			1	IPv4 Destination Address
Length	8			2	
Value	→	uint32	addr	4	IPv4 address as specified in the IPv4 protocol specification (RFC 791 [S2]).
		uint32	subnet_mask	4	IPv4 subnet mask as specified in the IPv4 protocol specification (RFC 791 [S2]).
Type	0x17			1	IPv4 TOS
Length	2			2	
Value	→	uint8	value	1	TOS value as specified in the IPv4 protocol specification (RFC 791 [S2]).
		uint8	mask	1	IPv4 TOS mask.
Type	0x18			1	IPv6 Source Address
Length	17			2	
Value	→	uint8	addr	16	IPv6 address as specified in the IPv6 protocol specification (RFC 2460 [S5]).
		uint8	prefix_len	1	IPv6 prefix length as specified in the IPv6 protocol addressing architecture specification (RFC 3513 [S6]).
Type	0x19			1	IPv6 Destination Address
Length	17			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	uint8	addr	16	IPv6 address as specified in the IPv6 protocol specification (RFC 2460 [S5]).
		uint8	prefix_len	1	IPv6 prefix length as specified in the IPv6 protocol addressing architecture specification (RFC 3513 [S6]).
Type	0x1A			1	IPv6 Traffic Class
Length	2			2	
Value	→	uint8	value	1	IPv6 traffic class value as specified in the IPv6 protocol specification (RFC 2460 [S5]).
		uint8	mask	1	IPv6 traffic class mask.

### 3.32.2 Response - QMI\_QCMAP\_ADD\_EXTD\_FIREWALL\_CONFIG\_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

Name	Version introduced	Version last modified
Firewall handle	1.1	1.1

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Firewall handle
Length	4			2	
Value	→	uint32	firewall_handle	4	Handle identifying the added firewall rule.

#### 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_MISSING_ARG	Some TLV was missing

QMI_ERR_INVALID_HANDLE	Mobile AP handle provided in the request is not valid, i.e., it is not assigned to the control point
QMI_ERR_INVALID_ARG	Argument is not correct
QMI_ERR_NOT_SUPPORTED	Operation is not supported
QMI_ERR_NO_MEMORY	Maximum number of supported firewall rules was exceeded; cannot add any more firewall rules

### 3.32.3 Description of QMI\_QCMAP\_ADD\_EXTD\_FIREWALL\_CONFIG REQ/RESP

This command adds a single IP filter-based firewall rule. The control point must specify the source/destination port and range when the value of the Next Header Protocol TLV is TCP/UDP. Otherwise, a QMI\_ERR\_MISSING\_ARG error is returned.

Qualcomm  
Confidential - May Contain Trade Secrets  
2022-07-21 08:18:44 GMT  
jason1\_gao@askey.com



### 3.33 QMI\_QCMAP\_GET\_EXTD\_FIREWALL\_CONFIG

Gets the firewall rules.

**QCMAP message ID**

0x0040

**Version introduced**

Major - 1, Minor - 1

#### 3.33.1 Request - QMI\_QCMAP\_GET\_EXTD\_FIREWALL\_CONFIG\_REQ

**Message type**

Request

**Sender**

Control point

**Mandatory TLVs**

Name	Version introduced	Version last modified
Firewall Handle	1.1	1.1

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Mobile AP handle
Length	4			2	
Value	→	uint32	mobile_ap_handle	4	Handle identifying the mobile AP call instance. The value must be the handle previously returned by QMI_QCMAP_MOBILE_AP_ENABLE_REQ.
Type	0x02			1	Firewall Handle
Length	4			2	
Value	→	uint32	firewall_handle	4	Handle identifying the firewall entry. The value must be the handle previously returned by one of the following: <ul style="list-style-type: none"> <li>• QMI_QCMAP_ADD_FIREWALL_CONFIG_RESP</li> <li>• QMI_QCMAP_GET_FIREWALL_CONFIG_RESP</li> <li>• QMI_QCMAP_ADD_EXTD_FIREWALL_CONFIG_RESP</li> <li>• QMI_QCMAP_GET_FIREWALL_CONFIG_HANDLE_LIST_RESP</li> </ul>

**Optional TLVs**

None

**3.33.2 Response - QMI\_QCMAP\_GET\_EXTD\_FIREWALL\_CONFIG\_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**

Name	Version introduced	Version last modified
Next Header Protocol	1.1	1.1
TCP/UDP Source	1.1	1.1
TCP/UDP Destination	1.1	1.1
ICMP Type	1.1	1.1
ICMP Code	1.1	1.1
ESP Security Parameters Index	1.1	1.1
IPv4 Source Address	1.1	1.1
IPv4 Destination Address	1.1	1.1
IPv4 TOS	1.1	1.1
IPv6 Source Address	1.1	1.1
IPv6 Destination Address	1.1	1.1
IPv6 Traffic Class	1.1	1.1

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Next Header Protocol
Length	4			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	enum	next_hdr_prot	4	IPv4/IPv6 next header protocol after the IP header. Values: <ul style="list-style-type: none"> <li>• 0x01 – QCMAP_EXTD_FIREWALL_PROTO_TCP – Transmission Control Protocol</li> <li>• 0x02 – QCMAP_EXTD_FIREWALL_PROTO_UDP – User Datagram Protocol</li> <li>• 0x03 – QCMAP_EXTD_FIREWALL_PROTO_ICMP – Internet Control Message Protocol</li> <li>• 0x04 – QCMAP_EXTD_FIREWALL_PROTO_ICMP6 – Internet Control Message Protocol for IPv6</li> <li>• 0x05 – QCMAP_EXTD_FIREWALL_PROTO_ESP – Encapsulating Security Payload Protocol</li> <li>• 0x06 – QCMAP_EXTD_FIREWALL_PROTO_TCP_UDP – Transmission Control Protocol/User Datagram Protocol</li> </ul>
Type	0x11			1	TCP/UDP Source
Length	4			2	
Value	→	uint16	port	2	TCP/UDP port as specified in the TCP/UDP protocol (RFC 793 [S4] and RFC 768 [S1]).
		uint16	range	2	TCP/UDP port range as specified in the TCP/UDP protocol (RFC 793 [S4] and RFC 768 [S1]).
Type	0x12			1	TCP/UDP Destination
Length	4			2	
Value	→	uint16	port	2	TCP/UDP port as specified in the TCP/UDP protocol (RFC 793 [S4] and RFC 768 [S1]).
		uint16	range	2	TCP/UDP port range as specified in the TCP/UDP protocol (RFC 793 [S4] and RFC 768 [S1]).
Type	0x13			1	ICMP Type
Length	1			2	
Value	→	uint8	icmp_type	1	ICMP type as specified in the ICMP specification (RFC 792 [S3]).
Type	0x14			1	ICMP Code
Length	1			2	
Value	→	uint8	icmp_code	1	ICMP code as specified in the ICMP specification (RFC 792 [S3]).
Type	0x15			1	ESP Security Parameters Index
Length	4			2	

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	uint32	esp_spi	4	Security parameters index as specified in the ESP protocol (RFC 4303 [S7]).
Type	0x16			1	IPv4 Source Address
Length	8			2	
Value	→	uint32	addr	4	IPv4 address as specified in the IPv4 protocol specification (RFC 791 [S2]).
		uint32	subnet_mask	4	IPv4 subnet mask as specified in the IPv4 protocol specification (RFC 791 [S2]).
Type	0x17			1	IPv4 Destination Address
Length	8			2	
Value	→	uint32	addr	4	IPv4 address as specified in the IPv4 protocol specification (RFC 791 [S2]).
		uint32	subnet_mask	4	IPv4 subnet mask as specified in the IPv4 protocol specification (RFC 791 [S2]).
Type	0x18			1	IPv4 TOS
Length	2			2	
Value	→	uint8	value	1	TOS value as specified in the IPv4 protocol specification (RFC 791 [S2]).
		uint8	mask	1	IPv4 TOS mask.
Type	0x19			1	IPv6 Source Address
Length	17			2	
Value	→	uint8	addr	16	IPv6 address as specified in the IPv6 protocol specification (RFC 2460 [S5]).
		uint8	prefix_len	1	IPv6 prefix length as specified in the IPv6 protocol addressing architecture specification (RFC 3513 [S6]).
Type	0x1A			1	IPv6 Destination Address
Length	17			2	
Value	→	uint8	addr	16	IPv6 address as specified in the IPv6 protocol specification (RFC 2460 [S5]).
		uint8	prefix_len	1	IPv6 prefix length as specified in the IPv6 protocol addressing architecture specification (RFC 3513 [S6]).
Type	0x1B			1	IPv6 Traffic Class
Length	2			2	
Value	→	uint8	value	1	IPv6 traffic class value as specified in the IPv6 protocol specification (RFC 2460 [S5]).
		uint8	mask	1	IPv6 traffic class mask.

**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_MISSING_ARG	Some TLV was missing
QMI_ERR_INVALID_HANDLE	Mobile AP handle provided in the request is not valid, i.e., it is not assigned to the control point
QMI_ERR_NOT_SUPPORTED	Operation is not supported

**3.33.3 Description of QMI\_QCMAP\_GET\_EXTD\_FIREWALL\_CONFIG REQ/RESP**

This command gets a firewall rule associated with a single firewall handle.

Qualcomm  
Confidential - May Contain Trade Secrets  
2022-07-21 08:18:44 GMT  
jason1\_gao@askey.com

### 3.34 QMI\_QCMAP\_GET\_FIREWALL\_CONFIG\_HANDLE\_LIST

Gets the handles of all the firewall rules.

#### QCMAP message ID

0x0041

#### Version introduced

Major - 1, Minor - 1

#### 3.34.1 Request - QMI\_QCMAP\_GET\_FIREWALL\_CONFIG\_HANDLE\_LIST\_REQ

##### Message type

Request

##### Sender

Control point

##### Mandatory TLVs

Name	Version introduced	Version last modified
Mobile AP handle	1.1	1.1

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Mobile AP handle
Length	4			2	
Value	→	uint32	mobile_ap_handle	4	Handle identifying the mobile AP call instance. The value must be the handle previously returned by QMI_QCMAP_MOBILE_AP_ENABLE_REQ.

##### Optional TLVs

None

### 3.34.2 Response - QMI\_QCMAP\_GET\_FIREWALL\_CONFIG\_HANDLE\_LIST\_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

Name	Version introduced	Version last modified
Firewall Handle List	1.1	1.1

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Firewall Handle List
Length	Var			2	
Value	→	uint8	firewall_handle_list_len	1	Number of sets of the following elements: • firewall_handle_list
		uint32	firewall_handle_list	Var	Firewall handle list.

#### 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_MISSING_ARG	Some TLV was missing
QMI_ERR_INVALID_HANDLE	Mobile AP handle provided in the request is not valid, i.e., it is not assigned to the control point
QMI_ERR_NOT_SUPPORTED	Operation is not supported

### 3.34.3 Description of QMI\_QCMAP\_GET\_FIREWALL\_CONFIG\_- HANDLE\_LIST REQ/RESP

This command gets all the firewall handles associated with a single mobile AP instance.

Qualcomm  
Confidential - May Contain Trade Secrets  
2022-07-21 08:18:44 GMT  
jason1\_gao@askey.com



## 3.35 QMI\_QCMAP\_CHANGE\_NAT\_TYPE

Changes the currently existing NAT type.

### QCMAP message ID

0x0042

### Version introduced

Major - 1, Minor - 3

### 3.35.1 Request - QMI\_QCMAP\_CHANGE\_NAT\_TYPE\_REQ

#### Message type

Request

#### Sender

Control point

#### Mandatory TLVs

Name	Version introduced	Version last modified
Mobile AP handle	1.3	1.3

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Mobile AP handle
Length	4			2	
Value	→	uint32	mobile_ap_handle	4	Handle identifying the mobile AP call instance. The value must be the handle previously returned by QMI_QCMAP_MOBILE_AP_ENABLE_REQ.

#### Optional TLVs

Name	Version introduced	Version last modified
NAT Type Option	1.3	1.3

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

Field	Field value	Field type	Parameter	Size (byte)	Description
Value	→	enum	nat_type_option	4	NAT type specified for the NAT type change. Values: <ul style="list-style-type: none"> <li>• 0x00 – QCMAP_NAT_TYPE_SYMMETRIC – Symmetric NAT</li> <li>• 0x01 – QCMAP_NAT_TYPE_PORT_RESTRICTED_CONE – Port restricted cone NAT</li> </ul>

### 3.35.2 Response - QMI\_QCMAP\_CHANGE\_NAT\_TYPE\_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 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	Some TLV was missing
QMI_ERR_INVALID_HANDLE	Mobile AP handle provided in the request is not valid, i.e., it is not assigned to the control point
QMI_ERR_NOT_SUPPORTED	Operation is not supported

### 3.35.3 Description of QMI\_QCMAP\_CHANGE\_NAT\_TYPE REQ/RESP

This command changes the NAT type associated with a mobile AP instance. When the NAT type is changed, the old NAT table data is cleaned and all existing connections are lost. If an IPv6 handle is passed, a QMI\_ERR\_INVALID\_HANDLE error is returned.

## 3.36 QMI\_QCMAP\_GET\_NAT\_TYPE

Gets the currently enabled NAT type.

### QCMAP message ID

0x0043

### Version introduced

Major - 1, Minor - 3

### 3.36.1 Request - QMI\_QCMAP\_GET\_NAT\_TYPE\_REQ

#### Message type

Request

#### Sender

Control point

#### Mandatory TLVs

Name	Version introduced	Version last modified
Mobile AP handle	1.3	1.3

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x01			1	Mobile AP handle
Length	4			2	
Value	→	uint32	mobile_ap_handle	4	Handle identifying the mobile AP call instance. The value must be the handle previously returned by QMI_QCMAP_MOBILE_AP_ENABLE_REQ.

#### Optional TLVs

None

### 3.36.2 Response - QMI\_QCMAP\_GET\_NAT\_TYPE\_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

Name	Version introduced	Version last modified
Current NAT Type	1.3	1.3

Field	Field value	Field type	Parameter	Size (byte)	Description
Type	0x10			1	Current NAT Type
Length	4			2	
Value	→	enum	nat_type_option	4	NAT type currently on the modem. Values: • 0x00 – QCMAP_NAT_TYPE_SYMMETRIC – Symmetric NAT • 0x01 – QCMAP_NAT_TYPE_PORT_RESTRICTED_CONE – Port restricted cone NAT

#### 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_MISSING_ARG	Some TLV was missing
QMI_ERR_INVALID_HANDLE	Mobile AP handle provided in the request is not valid, i.e., it is not assigned to the control point
QMI_ERR_NOT_SUPPORTED	Operation is not supported

### 3.36.3 Description of QMI\_QCMAP\_GET\_NAT\_TYPE REQ/RESP

This command gets the currently enabled NAT type associated with a mobile AP instance. If an IPv6 handle is passed, a QMI\_ERR\_INVALID\_HANDLE is returned.

Qualcomm  
Confidential - May Contain Trade Secrets  
2022-07-21 08:18:44 GMT  
jason1\_gao@askey.com

# A Call End Reasons

## A.1 Call End Reasons

Table A-1 lists the call end reasons.

Table A-1 Call end reasons

Value	Name
1	QCMAP_CER_UNSPECIFIED
2	QCMAP_CER_CLIENT_END
3	QCMAP_CER_NO_SRV
4	QCMAP_CER_FADE
5	QCMAP_CER_REL_NORMAL
6	QCMAP_CER_ACC_IN_PROG
7	QCMAP_CER_ACC_FAIL
8	QCMAP_CER_REDIR_OR_HANDOFF
9	QCMAP_CER_CLOSE_IN_PROGRESS
10	QCMAP_CER_AUTH_FAILED
11	QCMAP_CER_INTERNAL_CALL_END
500	QCMAP_CER_CDMA_LOCK
501	QCMAP_CER_INTERCEPT
502	QCMAP_CER_REORDER
503	QCMAP_CER_REL_SO_REJ
504	QCMAP_CER_INCOM_CALL
505	QCMAP_CER_ALERT_STOP
506	QCMAP_CER_ACTIVATION
507	QCMAP_CER_MAX_ACCESS_PROBE
508	QCMAP_CER_CCS_NOT_SUPP_BY_BS
509	QCMAP_CER_NO_RESPONSE_FROM_BS
510	QCMAP_CER_REJECTED_BY_BS
511	QCMAP_CER_INCOMPATIBLE
512	QCMAP_CER_ALREADY_IN_TC
513	QCMAP_CER_USER_CALL_ORIG_DURING_GPS
514	QCMAP_CER_USER_CALL_ORIG_DURING_SMS
515	QCMAP_CER_NO_CDMA_SRV
1000	QCMAP_CER_CONF_FAILED
1001	QCMAP_CER_INCOM_REJ
1002	QCMAP_CER_NO_GW_SRV
1003	QCMAP_CER_NETWORK_END
1004	QCMAP_CER_LLC_SNDP_FAILURE
1005	QCMAP_CER_INSUFFICIENT_RESOURCES

Table A-1 Call end reasons (cont.)

Value	Name
1006	QCMAP_CER_OPTION_TEMP_OOO
1007	QCMAP_CER_NSAPI_ALREADY_USED
1008	QCMAP_CER_REGULAR_DEACTIVATION
1009	QCMAP_CER_NETWORK_FAILURE
1010	QCMAP_CER_UMTS_REATTACH_REQ
1011	QCMAP_CER_PROTOCOL_ERROR
1012	QCMAP_CER_OPERATOR_DETERMINED_BARRING x
1013	QCMAP_CER_UNKNOWN_APN
1014	QCMAP_CER_UNKNOWN_PDP
1015	QCMAP_CER_GGSN_REJECT
1016	QCMAP_CER_ACTIVATION_REJECT
1017	QCMAP_CER_OPTION_NOT_SUPP
1018	QCMAP_CER_OPTION_UNSUBSCRIBED
1019	QCMAP_CER_QOS_NOT_ACCEPTED
1020	QCMAP_CER_TFT_SEMANTIC_ERROR
1021	QCMAP_CER_TFT_SYNTAX_ERROR
1022	QCMAP_CER_UNKNOWN_PDP_CONTEXT
1023	QCMAP_CER_FILTER_SEMANTIC_ERROR
1024	QCMAP_CER_FILTER_SYNTAX_ERROR
1025	QCMAP_CER_PDP_WITHOUT_ACTIVE_TFT
1026	QCMAP_CER_INVALID_TRANSACTION_ID
1027	QCMAP_CER_MESSAGE_INCORRECT_SEMANTIC
1028	QCMAP_CER_INVALID_MANDATORY_INFO
1029	QCMAP_CER_MESSAGE_TYPE_UNSUPPORTED
1030	QCMAP_CER_MSG_TYPE_NONCOMPATIBLE_STATE
1031	QCMAP_CER_UNKNOWN_INFO_ELEMENT
1032	QCMAP_CER_CONDITIONAL_IE_ERROR
1033	QCMAP_CER_MSG_AND_PROTOCOL_STATE_UNCOMPATIBLE
1034	QCMAP_CER_APN_TYPE_CONFLICT
1035	QCMAP_CER_NO_GPRS_CONTEXT
1036	QCMAP_CER_FEATURE_NOT_SUPPORTED
1500	QCMAP_CER_CD_GEN_OR_BUSY
1501	QCMAP_CER_CD_BILL_OR_AUTH
1502	QCMAP_CER_CHG_HDR
1503	QCMAP_CER_EXIT_HDR
1504	QCMAP_CER_HDR_NO_SESSION
1505	QCMAP_CER_HDR_ORIG_DURING_GPS_FIX
1506	QCMAP_CER_HDR_CS_TIMEOUT
1507	QCMAP_CER_HDR_RELEASED_BY_CM

## A.2 Verbose Call End Reasons

Table A-2 lists the verbose call end reasons.

**Table A-2 Verbose call end reasons**

Value	Name
0	QCMAP_VCER_UNSPECIFIED
<b>MIP</b>	
65600	QCMAP_VCER_MIP_FA_REASON_UNSPECIFIED
65601	QCMAP_VCER_MIP_FA_ADMIN_PROHIBITED
65602	QCMAP_VCER_MIP_FA_INSUFFICIENT_RESOURCES
65603	QCMAP_VCER_MIP_FA_MOBILE_NODE_AUTH_FAILURE
65604	QCMAP_VCER_MIP_FA_HA_AUTH_FAILURE
65605	QCMAP_VCER_MIP_FA_REQ_LIFETIME_TOO_LONG
65606	QCMAP_VCER_MIP_FA_MALFORMED_REQUEST
65607	QCMAP_VCER_MIP_FA_MALFORMED_REPLY
65608	QCMAP_VCER_MIP_FA_ENCAPSULATION_UNAVAILABLE
65609	QCMAP_VCER_MIP_FA_VJHC_UNAVAILABLE
65610	QCMAP_VCER_MIP_FA_REV_TUNNEL_UNAVAILABLE
65611	QCMAP_VCER_MIP_FA_REV_TUNNEL_IS_MAND_AND_T_BIT_NOT_SET
65615	QCMAP_VCER_MIP_FA_DELIVERY_STYLE_NOT_SUPP
65633	QCMAP_VCER_MIP_FA_MISSING_NAI
65634	QCMAP_VCER_MIP_FA_MISSING_HA
65635	QCMAP_VCER_MIP_FA_MISSING_HOME_ADDR
65640	QCMAP_VCER_MIP_FA_UNKNOWN_CHALLENGE
65641	QCMAP_VCER_MIP_FA_MISSING_CHALLENGE
65642	QCMAP_VCER_MIP_FA_STALE_CHALLENGE
65664	QCMAP_VCER_MIP_HA_REASON_UNSPECIFIED
65665	QCMAP_VCER_MIP_HA_ADMIN_PROHIBITED
65666	QCMAP_VCER_MIP_HA_INSUFFICIENT_RESOURCES
65667	QCMAP_VCER_MIP_HA_MOBILE_NODE_AUTH_FAILURE
65668	QCMAP_VCER_MIP_HA_FA_AUTH_FAILURE
65669	QCMAP_VCER_MIP_HA_REGISTRATION_ID_MISMATCH
65670	QCMAP_VCER_MIP_HA_MALFORMED_REQUEST
65672	QCMAP_VCER_MIP_HA_UNKNOWN_HA_ADDR
65673	QCMAP_VCER_MIP_HA_REV_TUNNEL_UNAVAILABLE
65674	QCMAP_VCER_MIP_HA_REV_TUNNEL_IS_MAND_AND_T_BIT_NOT_SET
65675	QCMAP_VCER_MIP_HA_ENCAPSULATION_UNAVAILABLE
131071	QCMAP_VCER_MIP_HA_REASON_UNKNOWN
<b>Internal</b>	
131273	QCMAP_VCER_INTERNAL_INTERNAL_ERROR
131274	QCMAP_VCER_INTERNAL_CALL_ENDED
131275	QCMAP_VCER_INTERNAL_INTERNAL_UNKNOWN_CAUSE_CODE
131276	QCMAP_VCER_INTERNAL_UNKNOWN_CAUSE_CODE
131277	QCMAP_VCER_INTERNAL_CLOSE_IN_PROGRESS
131278	QCMAP_VCER_INTERNAL_NW_INITIATED_TERMINATION
131279	QCMAP_VCER_INTERNAL_APP_PREEMPTED



Table A-2 Verbose call end reasons (cont.)

Value	Name
<b>Call manager</b>	
197108	QCMAP_VCER_CM_CDMA_LOCK
197109	QCMAP_VCER_CM_INTERCEPT
197110	QCMAP_VCER_CM_REORDER
197111	QCMAP_VCER_CM_REL_SO_REJ
197112	QCMAP_VCER_CM_INCOM_CALL
197113	QCMAP_VCER_CM_ALERT_STOP
197114	QCMAP_VCER_CM_ACTIVATION
197115	QCMAP_VCER_CM_MAX_ACCESS_PROBE
197116	QCMAP_VCER_CM_CCS_NOT_SUPP_BY_BS
197117	QCMAP_VCER_CM_NO_RESPONSE_FROM_BS
197118	QCMAP_VCER_CM_REJECTED_BY_BS
197119	QCMAP_VCER_CM_INCOMPATIBLE
197120	QCMAP_VCER_CM_ALREADY_IN_TC
197121	QCMAP_VCER_CM_USER_CALL_ORIG_DURING_GPS
197122	QCMAP_VCER_CM_USER_CALL_ORIG_DURING_SMS
197123	QCMAP_VCER_CM_NO_CDMA_SRV
197127	QCMAP_VCER_CM_RETRY_ORDER
197608	QCMAP_VCER_CM_CONF_FAILED
197609	QCMAP_VCER_CM_INCOM_REJ
197616	QCMAP_VCER_CM_NO_GW_SERV
197617	QCMAP_VCER_CM_NO_GPRS_CONTEXT
197618	QCMAP_VCER_CM_ILLEGAL_MS
197619	QCMAP_VCER_CM_ILLEGAL_ME
197620	QCMAP_VCER_CM_GPRS_SERV_AND_NON_GPRS_SERV_NOT_ALLOWED
197621	QCMAP_VCER_CM_GPRS_SERV_NOT_ALLOWED
197622	QCMAP_VCER_CM_MS_IDENTITY_CANNOT_BE_DERIVED_BY_THE_NETWORK
197623	QCMAP_VCER_CM_IMPLICITLY_DETACHED
197624	QCMAP_VCER_CM_PLMN_NOT_ALLOWED
197625	QCMAP_VCER_CM_LA_NOT_ALLOWED
197626	QCMAP_VCER_CM_GPRS_SERV_NOT_ALLOWED_IN_THIS_PLMN
197627	QCMAP_VCER_CM_PDP_DUPLICATE
197628	QCMAP_VCER_CM_UE_RAT_CHANGE
197629	QCMAP_VCER_CM_CONGESTION
197630	QCMAP_VCER_CM_NO_PDP_CONTEXT_ACTIVATED
197631	QCMAP_VCER_CM_ACCESS_CLASS_DSAC_REJECTION
198108	QCMAP_VCER_CM_CD_GEN_OR_BUSY
198109	QCMAP_VCER_CM_CD_BILL_OR_AUTH
198110	QCMAP_VCER_CM_CHG_HDR
198111	QCMAP_VCER_CM_EXIT_HDR
198112	QCMAP_VCER_CM_HDR_NO_SESSION
198113	QCMAP_VCER_CM_HDR_ORIG_DURING_GPS_FIX
198114	QCMAP_VCER_CM_HDR_CS_TIMEOUT
198115	QCMAP_VCER_CM_HDR_RELEASED_BY_CM
198118	QCMAP_VCER_CM_NO_HYBR_HDR_SRV

Table A-2 Verbose call end reasons (cont.)

Value	Name
198608	QCMAP_VCER_CM_CLIENT_END
198609	QCMAP_VCER_CM_NO_SRV
198610	QCMAP_VCER_CM_FADE
198611	QCMAP_VCER_CM_REL_NORMAL
198612	QCMAP_VCER_CM_ACC_IN_PROG
198613	QCMAP_VCER_CM_ACC_FAIL
198614	QCMAP_VCER_CM_REDIR_OR_HANDOFF
<b>3GPP specification</b>	
393224	QCMAP_VCER_3GPP_OPERATOR_DETERMINED_BARRING
393241	QCMAP_VCER_3GPP_LLC_SNDP_FAILURE
393242	QCMAP_VCER_3GPP_INSUFFICIENT_RESOURCES
393243	QCMAP_VCER_3GPP_UNKNOWN_APN
393244	QCMAP_VCER_3GPP_UNKNOWN_PDP
393245	QCMAP_VCER_3GPP_AUTH_FAILED
393246	QCMAP_VCER_3GPP_GGSN_REJECT
393247	QCMAP_VCER_3GPP_ACTIVATION_REJECT
393248	QCMAP_VCER_3GPP_OPTION_NOT_SUPPORTED
393249	QCMAP_VCER_3GPP_OPTION_UNSUBSCRIBED
393250	QCMAP_VCER_3GPP_OPTION_TEMP_OOO
393251	QCMAP_VCER_3GPP_NSAPI_ALREADY_USED
393252	QCMAP_VCER_3GPP_REGULAR_DEACTIVATION
393253	QCMAP_VCER_3GPP_QOS_NOT_ACCEPTED
393254	QCMAP_VCER_3GPP_NETWORK_FAILURE
393255	QCMAP_VCER_3GPP_UMTS_REACTIVATION_REQ
393256	QCMAP_VCER_3GPP_FEATURE_NOT_SUPP
393257	QCMAP_VCER_3GPP_TFT_SEMANTIC_ERROR
393258	QCMAP_VCER_3GPP_TFT_SYNTAX_ERROR
393259	QCMAP_VCER_3GPP_UNKNOWN_PDP_CONTEXT
393260	QCMAP_VCER_3GPP_FILTER_SEMANTIC_ERROR
393261	QCMAP_VCER_3GPP_FILTER_SYNTAX_ERROR
393262	QCMAP_VCER_3GPP_PDP_WITHOUT_ACTIVE_TFT
393297	QCMAP_VCER_3GPP_INVALID_TRANSACTION_ID
393311	QCMAP_VCER_3GPP_MESSAGE_INCORRECT_SEMANTIC
393312	QCMAP_VCER_3GPP_INVALID_MANDATORY_INFO
393313	QCMAP_VCER_3GPP_MESSAGE_TYPE_UNSUPPORTED
393314	QCMAP_VCER_3GPP_MSG_TYPE_NONCOMPATIBLE_STATE
393315	QCMAP_VCER_3GPP_UNKNOWN_INFO_ELEMENT
393316	QCMAP_VCER_3GPP_CONDITIONAL_IE_ERROR
393317	QCMAP_VCER_3GPP_MSG_AND_PROTOCOL_STATE_UNCOMPATIBLE
393327	QCMAP_VCER_3GPP_PROTOCOL_ERROR
393328	QCMAP_VCER_3GPP_APN_TYPE_CONFLICT

Table A-2 Verbose call end reasons (cont.)

Value	Name
<b>Point-to-Point Protocol</b>	
458753	QCMAP_VCER_PPP_TIMEOUT
458754	QCMAP_VCER_PPP_AUTH_FAILURE
458755	QCMAP_VCER_PPP_OPTION_MISMATCH
458783	QCMAP_VCER_PPP_PAP_FAILURE
458784	QCMAP_VCER_PPP_CHAP_FAILURE
524287	QCMAP_VCER_PPP_UNKNOWN
<b>eHRPD</b>	
524289	QCMAP_VCER_EHRPD_SUBS_LIMITED_TO_V4
524290	QCMAP_VCER_EHRPD_SUBS_LIMITED_TO_V6
524292	QCMAP_VCER_EHRPD_VSNCP_TIMEOUT
524293	QCMAP_VCER_EHRPD_VSNCP_FAILURE
524294	QCMAP_VCER_EHRPD_VSNCP_3GPP2I_GEN_ERROR
524295	QCMAP_VCER_EHRPD_VSNCP_3GPP2I_UNAUTH_APN
524296	QCMAP_VCER_EHRPD_VSNCP_3GPP2I_PDN_LIMIT_EXCEED
524297	QCMAP_VCER_EHRPD_VSNCP_3GPP2I_NO_PDN_GW
524298	QCMAP_VCER_EHRPD_VSNCP_3GPP2I_PDN_GW_UNREACH
524299	QCMAP_VCER_EHRPD_VSNCP_3GPP2I_PDN_GW_REJ
524300	QCMAP_VCER_EHRPD_VSNCP_3GPP2I_INSUFF_PARAM
524301	QCMAP_VCER_EHRPD_VSNCP_3GPP2I_RESOURCE_UNAVAIL
524302	QCMAP_VCER_EHRPD_VSNCP_3GPP2I_ADMIN_PROHIBIT
524303	QCMAP_VCER_EHRPD_VSNCP_3GPP2I_PDN_ID_IN_USE
524304	QCMAP_VCER_EHRPD_VSNCP_3GPP2I_SUBSCR_LIMITATION
524305	QCMAP_VCER_EHRPD_VSNCP_3GPP2I_PDN_EXISTS_FOR_THIS_APN
<b>IPv6</b>	
589825	QCMAP_VCER_IPV6_PREFIX_UNAVAILABLE