

LinuxQMISDK  
SLQS03.03.12

Generated by Doxygen 1.8.6

Wed Oct 28 2015 00:40:12



# Contents

<b>1</b>	<b>Welcome to the Sierra Wireless Linux QMI SDK API Reference Guide</b>	<b>1</b>
1.1	Important Notice . . . . .	1
1.2	Limitation of Liability . . . . .	1
1.3	Patents . . . . .	1
1.4	Copyright . . . . .	2
1.5	Trademarks . . . . .	2
1.6	Contact Information . . . . .	2
<b>2</b>	<b>Module Index</b>	<b>3</b>
2.1	Modules . . . . .	3
<b>3</b>	<b>Namespace Index</b>	<b>5</b>
3.1	Namespace List . . . . .	5
<b>4</b>	<b>Data Structure Index</b>	<b>7</b>
4.1	Data Structures . . . . .	7
<b>5</b>	<b>File Index</b>	<b>19</b>
5.1	File List . . . . .	19
<b>6</b>	<b>Module Documentation</b>	<b>21</b>
6.1	Device Connectivity Service (DCS) . . . . .	21
6.1.1	Detailed Description . . . . .	21
6.2	Wireless Data Service (WDS) . . . . .	22
6.2.1	Detailed Description . . . . .	22
6.3	Device Management Service (DMS) . . . . .	23
6.3.1	Detailed Description . . . . .	23
6.4	Network Access Service (NAS) . . . . .	24
6.4.1	Detailed Description . . . . .	24
6.5	CallBack registration (CBK) . . . . .	25
6.5.1	Detailed Description . . . . .	25
6.6	Short Message Service (SMS) . . . . .	26
6.6.1	Detailed Description . . . . .	26

6.7	Position Determination Service (PDS)	27
6.7.1	Detailed Description	27
6.8	Card Application Toolkit (CAT)	28
6.8.1	Detailed Description	28
6.9	Remote Management Service (RMS)	29
6.9.1	Detailed Description	29
6.10	Firmware Management Service (FMS)	30
6.10.1	Detailed Description	30
6.11	Open Mobile Alliance Service (OMA)	31
6.11.1	Detailed Description	31
6.12	Specific Absorption Rate (SAR)	32
6.12.1	Detailed Description	32
6.13	SWI Open Mobile Alliance Service (SWIOMA)	33
6.13.1	Detailed Description	33
6.14	Voice Service (VOICE)	34
6.14.1	Detailed Description	34
6.15	Non-service specific APIs (SWI)	35
6.15.1	Detailed Description	35
6.16	User Identity Module Service (UIM)	36
6.16.1	Detailed Description	36
6.17	Audio Service (AUDIO)	37
6.17.1	Detailed Description	37
6.18	Quality of Service (QOS)	38
6.18.1	Detailed Description	38
6.19	IMS Service (IMS)	39
6.19.1	Detailed Description	39
6.20	SWI Audio Service(SWIAUDIO)	40
6.20.1	Detailed Description	40
6.21	Location Service(LOC)	41
6.21.1	Detailed Description	41
<b>7</b>	<b>Namespace Documentation</b>	<b>43</b>
7.1	Tables Namespace Reference	43
7.1.1	Detailed Description	43
<b>8</b>	<b>Data Structure Documentation</b>	<b>45</b>
8.1	_getIndicationRegResp Struct Reference	45
8.1.1	Detailed Description	45
8.1.2	Field Documentation	46
8.1.2.1	pRegCallStatInfoEvt	46
8.1.2.2	pRegTransLayerInfoEvt	46

8.1.2.3	pRegTransNWRegInfoEvt	46
8.2	_GetProfileSettingIn Struct Reference	46
8.2.1	Detailed Description	46
8.2.2	Field Documentation	47
8.2.2.1	ProfileID	47
8.2.2.2	ProfileType	47
8.3	_GetProfileSettingOut Struct Reference	47
8.3.1	Detailed Description	47
8.3.2	Field Documentation	47
8.3.2.1	curProfile	47
8.3.2.2	pExtErrCode	47
8.4	_getTransLayerInfoResp Struct Reference	47
8.4.1	Detailed Description	48
8.4.2	Field Documentation	49
8.4.2.1	pRegInd	49
8.4.2.2	pTransLayerInfo	49
8.5	_getTransNWRegInfoResp Struct Reference	49
8.5.1	Detailed Description	49
8.5.2	Field Documentation	50
8.5.2.1	pRegStatus	50
8.6	_modemTempNotification Struct Reference	50
8.6.1	Detailed Description	50
8.6.2	Field Documentation	50
8.6.2.1	ModemTemperature	50
8.6.2.2	ModemTempState	50
8.7	_packetSrvStatus Struct Reference	50
8.7.1	Detailed Description	51
8.7.2	Field Documentation	52
8.7.2.1	bearerID	52
8.7.2.2	connStatus	52
8.7.2.3	ipFamily	52
8.7.2.4	pQmiInterfaceInfo	52
8.7.2.5	reconfigReqd	52
8.7.2.6	sessionEndReason	52
8.7.2.7	techName	52
8.7.2.8	verboseSessnEndReason	52
8.7.2.9	verboseSessnEndReasonType	52
8.8	_qaQmi3GPP2BroadcastCfgInfo Struct Reference	53
8.8.1	Detailed Description	53
8.8.2	Field Documentation	53

8.8.2.1	activated_ind . . . . .	53
8.8.2.2	CDMABroadcastConfig . . . . .	53
8.8.2.3	num_instances . . . . .	53
8.9	_qaQmi3GPPBroadcastCfgInfo Struct Reference . . . . .	53
8.9.1	Detailed Description . . . . .	54
8.9.2	Field Documentation . . . . .	55
8.9.2.1	activated_ind . . . . .	55
8.9.2.2	broadcastConfig . . . . .	55
8.9.2.3	num_instances . . . . .	55
8.10	_setIndicationRegReq Struct Reference . . . . .	55
8.10.1	Detailed Description . . . . .	55
8.10.2	Field Documentation . . . . .	56
8.10.2.1	pRegCallStatInfoEvt . . . . .	56
8.10.2.2	pRegTransLayerInfoEvt . . . . .	56
8.10.2.3	pRegTransNWRegInfoEvt . . . . .	56
8.11	_slqs3GPPConfigItem Struct Reference . . . . .	56
8.11.1	Detailed Description . . . . .	57
8.11.2	Field Documentation . . . . .	59
8.11.2.1	LTEAttachProfileListLen . . . . .	59
8.11.2.2	p3gppRelease . . . . .	59
8.11.2.3	pDefaultPDNEnabled . . . . .	59
8.11.2.4	pLTEAttachProfile . . . . .	59
8.11.2.5	pLTEAttachProfileList . . . . .	59
8.11.2.6	pProfileList . . . . .	59
8.12	_slqsNetworkScanInfo Struct Reference . . . . .	59
8.12.1	Detailed Description . . . . .	59
8.12.2	Field Documentation . . . . .	60
8.12.2.1	pNetworkInfo . . . . .	60
8.12.2.2	pNetworkInfoInstances . . . . .	60
8.12.2.3	pPCSDigitInfo . . . . .	60
8.12.2.4	pPCSDigitInstances . . . . .	60
8.12.2.5	pRATInfo . . . . .	60
8.12.2.6	pRATInstances . . . . .	60
8.12.2.7	pScanResult . . . . .	60
8.13	_SLQSOMADMSessionInfo Struct Reference . . . . .	61
8.13.1	Detailed Description . . . . .	61
8.13.2	Field Documentation . . . . .	63
8.13.2.1	pDate . . . . .	63
8.13.2.2	pDateLength . . . . .	63
8.13.2.3	pPkgDescLength . . . . .	63

8.13.2.4	pPkgDescription	63
8.13.2.5	pPkgName	63
8.13.2.6	pPkgNameLength	63
8.13.2.7	pRetryCount	63
8.13.2.8	pSessionState	63
8.13.2.9	pSessionType	63
8.13.2.10	pSeverity	63
8.13.2.11	pSource	63
8.13.2.12	pSourceLength	63
8.13.2.13	pStatus	63
8.13.2.14	pTime	64
8.13.2.15	pTimeLength	64
8.13.2.16	pUpdateCompleteStatus	64
8.14	_SLQSOMADMSettings Struct Reference	64
8.14.1	Detailed Description	64
8.14.2	Field Documentation	65
8.14.2.1	pAutosdm	65
8.14.2.2	pFOTAdownload	65
8.14.2.3	pFOTAUpdate	65
8.14.2.4	pFwAutoCheck	65
8.14.2.5	pOMADMEEnabled	65
8.15	_SLQSOMADMSettingsReqParams Struct Reference	65
8.15.1	Detailed Description	66
8.15.2	Field Documentation	66
8.15.2.1	FOTAdownload	66
8.15.2.2	FOTAUpdate	66
8.15.2.3	pAutosdm	66
8.16	_SLQSOMADMSettingsReqParams3 Struct Reference	66
8.16.1	Detailed Description	67
8.16.2	Field Documentation	67
8.16.2.1	FOTAdownload	67
8.16.2.2	FOTAUpdate	67
8.16.2.3	pAutosdm	67
8.16.2.4	pFwAutoCheck	67
8.17	_SLQSSwiGetHostDevInfoParams Struct Reference	67
8.17.1	Field Documentation	68
8.17.1.1	bManSize	68
8.17.1.2	bModelSize	68
8.17.1.3	bPlasmaIDSize	68
8.17.1.4	bSWVerSize	68

8.17.1.5	pManString . . . . .	68
8.17.1.6	pModelString . . . . .	68
8.17.1.7	pPlasmaIDString . . . . .	68
8.17.1.8	pSWVerString . . . . .	68
8.18	_SLQSSwiGetOSInfoParams Struct Reference . . . . .	68
8.18.1	Field Documentation . . . . .	68
8.18.1.1	bNameSize . . . . .	68
8.18.1.2	bVersionSize . . . . .	68
8.18.1.3	pNameString . . . . .	68
8.18.1.4	pVersionString . . . . .	68
8.19	_SLQSSwiGetSerialNoExtParams Struct Reference . . . . .	68
8.19.1	Detailed Description . . . . .	69
8.19.2	Field Documentation . . . . .	69
8.19.2.1	meidLength . . . . .	69
8.19.2.2	pMeidString . . . . .	69
8.20	_SLQSSwiSetHostDevInfoParams Struct Reference . . . . .	69
8.20.1	Field Documentation . . . . .	69
8.20.1.1	bManSize . . . . .	69
8.20.1.2	bModelSize . . . . .	69
8.20.1.3	bPlasmaIDSize . . . . .	69
8.20.1.4	bSWVerSize . . . . .	69
8.20.1.5	pManString . . . . .	69
8.20.1.6	pModelString . . . . .	69
8.20.1.7	pPlasmaIDString . . . . .	69
8.20.1.8	pSWVerString . . . . .	69
8.21	_SLQSSwiSetOSInfoParams Struct Reference . . . . .	69
8.21.1	Field Documentation . . . . .	70
8.21.1.1	bNameSize . . . . .	70
8.21.1.2	bVersionSize . . . . .	70
8.21.1.3	pNameString . . . . .	70
8.21.1.4	pVersionString . . . . .	70
8.22	_sysSelectPrefInfo Struct Reference . . . . .	70
8.22.1	Detailed Description . . . . .	70
8.22.2	Field Documentation . . . . .	73
8.22.2.1	pBandPref . . . . .	73
8.22.2.2	pEmerMode . . . . .	73
8.22.2.3	pGWAcqOrderPref . . . . .	73
8.22.2.4	pLTEBandPref . . . . .	73
8.22.2.5	pModePref . . . . .	73
8.22.2.6	pNetSelPref . . . . .	73



8.22.2.7	pPRLPref . . . . .	74
8.22.2.8	pRoamPref . . . . .	74
8.22.2.9	pSrvDomainPref . . . . .	74
8.23	_sysSelectPrefParams Struct Reference . . . . .	74
8.23.1	Detailed Description . . . . .	74
8.23.2	Field Documentation . . . . .	78
8.23.2.1	pAcqOrderPref . . . . .	78
8.23.2.2	pBandPref . . . . .	78
8.23.2.3	pChgDuration . . . . .	78
8.23.2.4	pCSGID . . . . .	78
8.23.2.5	pEmerMode . . . . .	78
8.23.2.6	pGWAcqOrderPref . . . . .	78
8.23.2.7	pLTEBandPref . . . . .	78
8.23.2.8	pMNCIncPCSDigStat . . . . .	78
8.23.2.9	pModePref . . . . .	78
8.23.2.10	pNetSelPref . . . . .	78
8.23.2.11	pPRLPref . . . . .	78
8.23.2.12	pRAT . . . . .	78
8.23.2.13	pRoamPref . . . . .	78
8.23.2.14	pSrvDomainPref . . . . .	78
8.23.2.15	pSrvRegRestriction . . . . .	78
8.23.2.16	pTdsdmaBandPref . . . . .	78
8.24	_transLayerinfo Struct Reference . . . . .	78
8.24.1	Detailed Description . . . . .	79
8.24.2	Field Documentation . . . . .	80
8.24.2.1	TransCap . . . . .	80
8.24.2.2	TransType . . . . .	80
8.25	_transLayerInfoNotification Struct Reference . . . . .	80
8.25.1	Detailed Description . . . . .	80
8.25.2	Field Documentation . . . . .	81
8.25.2.1	pTransLayerInfo . . . . .	81
8.25.2.2	regInd . . . . .	81
8.26	_transNWRegInfoNotification Struct Reference . . . . .	81
8.26.1	Detailed Description . . . . .	81
8.26.2	Field Documentation . . . . .	81
8.26.2.1	NWRegStat . . . . .	81
8.27	accelAcceptReady_s Struct Reference . . . . .	81
8.27.1	Detailed Description . . . . .	82
8.27.2	Field Documentation . . . . .	82
8.27.2.1	batchPerSec . . . . .	82

8.27.2.2	injectEnable	82
8.27.2.3	samplesPerBatch	82
8.28	accelTempAcceptReady_s Struct Reference	82
8.28.1	Detailed Description	82
8.28.2	Field Documentation	83
8.28.2.1	batchPerSec	83
8.28.2.2	injectEnable	83
8.28.2.3	samplesPerBatch	83
8.29	acqOrderPref Struct Reference	83
8.29.1	Detailed Description	83
8.29.2	Field Documentation	84
8.29.2.1	acqOrdeLen	84
8.29.2.2	pAcqOrder	84
8.30	ActPilotPNElement Struct Reference	84
8.30.1	Detailed Description	84
8.30.2	Field Documentation	84
8.30.2.1	ActSetPilotPN	84
8.30.2.2	ActSetPilotPNStrength	84
8.31	AddCDMASysInfo Struct Reference	84
8.31.1	Detailed Description	85
8.31.2	Field Documentation	85
8.31.2.1	geoSysIdx	85
8.31.2.2	regPrd	85
8.32	AddSysInfo Struct Reference	85
8.32.1	Detailed Description	85
8.32.2	Field Documentation	86
8.32.2.1	cellBroadcastCap	86
8.32.2.2	geoSysIdx	86
8.33	airTimer Struct Reference	86
8.33.1	Detailed Description	86
8.33.2	Field Documentation	86
8.33.2.1	airTimerValue	86
8.33.2.2	namID	86
8.34	allCallsAlphaIDInfo Struct Reference	87
8.34.1	Detailed Description	87
8.34.2	Field Documentation	87
8.34.2.1	AlphaIDInfo	87
8.34.2.2	callID	87
8.35	allCallsDiagInfo Struct Reference	87
8.35.1	Detailed Description	87

8.35.2	Field Documentation	87
8.35.2.1	callID	87
8.35.2.2	DiagInfo	87
8.36	allCallsUUSInfo Struct Reference	88
8.36.1	Detailed Description	88
8.36.2	Field Documentation	88
8.36.2.1	callID	88
8.36.2.2	uusInfo	88
8.37	alphaIDInfo Struct Reference	88
8.37.1	Detailed Description	88
8.37.2	Field Documentation	89
8.37.2.1	alphaDcs	89
8.37.2.2	alphaLen	89
8.37.2.3	alphaText	89
8.38	appStatus Struct Reference	89
8.38.1	Detailed Description	89
8.38.2	Field Documentation	92
8.38.2.1	aidLength	92
8.38.2.2	aidVal	92
8.38.2.3	appState	92
8.38.2.4	appType	92
8.38.2.5	persoFeature	92
8.38.2.6	persoRetries	92
8.38.2.7	persoState	92
8.38.2.8	persoUnblockRetries	92
8.38.2.9	pin1Retries	92
8.38.2.10	pin1State	92
8.38.2.11	pin2Retries	92
8.38.2.12	pin2State	92
8.38.2.13	puk1Retries	92
8.38.2.14	puk2Retries	93
8.38.2.15	univPin	93
8.39	arrAlertingPattern Struct Reference	93
8.39.1	Detailed Description	93
8.39.2	Field Documentation	93
8.39.2.1	alertingPattern	93
8.39.2.2	callID	93
8.39.2.3	numInstances	93
8.40	arrAlertingType Struct Reference	93
8.40.1	Detailed Description	94

8.40.2	Field Documentation	94
8.40.2.1	AlertingType	94
8.40.2.2	callID	94
8.40.2.3	numInstances	94
8.41	arrAlphaID Struct Reference	94
8.41.1	Detailed Description	94
8.41.2	Field Documentation	95
8.41.2.1	allCallsAlphaIDInfoArr	95
8.41.2.2	numInstances	95
8.42	arrCalledPartyNum Struct Reference	95
8.42.1	Detailed Description	95
8.42.2	Field Documentation	95
8.42.2.1	CalledPartyNum	95
8.42.2.2	numInstances	95
8.43	arrCallEndReason Struct Reference	95
8.43.1	Detailed Description	95
8.43.2	Field Documentation	96
8.43.2.1	callEndReason	96
8.43.2.2	callID	96
8.43.2.3	numInstances	96
8.44	arrCallInfo Struct Reference	96
8.44.1	Detailed Description	96
8.44.2	Field Documentation	96
8.44.2.1	getAllCallInfo	96
8.44.2.2	numInstances	96
8.45	arrConnectPartyNum Struct Reference	97
8.45.1	Detailed Description	97
8.45.2	Field Documentation	97
8.45.2.1	ConnectedPartyNum	97
8.45.2.2	numInstances	97
8.46	arrDiagInfo Struct Reference	97
8.46.1	Detailed Description	97
8.46.2	Field Documentation	98
8.46.2.1	DiagInfo	98
8.46.2.2	numInstances	98
8.47	arrRedirPartyNum Struct Reference	98
8.47.1	Detailed Description	98
8.47.2	Field Documentation	98
8.47.2.1	numInstances	98
8.47.2.2	RedirPartyNum	98

8.48 arrRemotePartyName Struct Reference . . . . .	98
8.48.1 Detailed Description . . . . .	98
8.48.2 Field Documentation . . . . .	99
8.48.2.1 GetAllCallRmtPtyName . . . . .	99
8.48.2.2 numInstances . . . . .	99
8.49 arrRemotePartyNum Struct Reference . . . . .	99
8.49.1 Detailed Description . . . . .	99
8.49.2 Field Documentation . . . . .	99
8.49.2.1 numInstances . . . . .	99
8.49.2.2 RmtPtyNum . . . . .	99
8.50 arrSvcOption Struct Reference . . . . .	99
8.50.1 Detailed Description . . . . .	100
8.50.2 Field Documentation . . . . .	100
8.50.2.1 callID . . . . .	100
8.50.2.2 numInstances . . . . .	100
8.50.2.3 srvOption . . . . .	100
8.51 arrUUSInfo Struct Reference . . . . .	100
8.51.1 Detailed Description . . . . .	100
8.51.2 Field Documentation . . . . .	101
8.51.2.1 AllCallsUUSInfo . . . . .	101
8.51.2.2 numInstances . . . . .	101
8.52 authenticateResult Struct Reference . . . . .	101
8.52.1 Detailed Description . . . . .	101
8.52.2 Field Documentation . . . . .	101
8.52.2.1 content . . . . .	101
8.52.2.2 contentLen . . . . .	101
8.53 authenticationData Struct Reference . . . . .	101
8.53.1 Detailed Description . . . . .	101
8.53.2 Field Documentation . . . . .	102
8.53.2.1 context . . . . .	102
8.53.2.2 data . . . . .	102
8.53.2.3 dataLen . . . . .	102
8.54 BdsSV Struct Reference . . . . .	102
8.54.1 Detailed Description . . . . .	103
8.54.2 Field Documentation . . . . .	103
8.54.2.1 id . . . . .	103
8.54.2.2 mask . . . . .	103
8.55 BdsSVInfo Struct Reference . . . . .	103
8.55.1 Detailed Description . . . . .	103
8.55.2 Field Documentation . . . . .	104

8.55.2.1	len	104
8.55.2.2	pSV	104
8.56	BroadcastConfig Struct Reference	104
8.56.1	Detailed Description	104
8.56.2	Field Documentation	104
8.56.2.1	fromServiceId	104
8.56.2.2	selected	104
8.56.2.3	toServiceId	104
8.57	burstDTMFInfo Struct Reference	104
8.57.1	Detailed Description	105
8.57.2	Field Documentation	105
8.57.2.1	digitCnt	105
8.57.2.2	pCallID	105
8.57.2.3	pDigitBuff	105
8.58	CallBarringSysInfo Struct Reference	105
8.58.1	Detailed Description	105
8.58.2	Field Documentation	106
8.58.2.1	csBarStatus	106
8.58.2.2	psBarStatus	106
8.59	callBarStatus Struct Reference	106
8.59.1	Detailed Description	106
8.59.2	Field Documentation	107
8.59.2.1	csBarStatus	107
8.59.2.2	psBarStatus	107
8.60	calledPartyInfo Struct Reference	107
8.60.1	Detailed Description	107
8.60.2	Field Documentation	109
8.60.2.1	number	109
8.60.2.2	numLen	109
8.60.2.3	numPlan	109
8.60.2.4	numType	109
8.60.2.5	PI	109
8.60.2.6	SI	109
8.61	calledPartySubAdd Struct Reference	109
8.61.1	Detailed Description	109
8.61.2	Field Documentation	110
8.61.2.1	extBit	110
8.61.2.2	oddEvenInd	110
8.61.2.3	subAddr	110
8.61.2.4	subAddrLen	110

8.61.2.5	subAddrType	110
8.62	callerIDInfo Struct Reference	110
8.62.1	Detailed Description	110
8.62.2	Field Documentation	111
8.62.2.1	callerID	111
8.62.2.2	callerIDLen	111
8.62.2.3	PI	111
8.63	callFwdTypeAndPlan Struct Reference	111
8.63.1	Detailed Description	111
8.63.2	Field Documentation	112
8.63.2.1	numberPlan	112
8.63.2.2	numberType	112
8.64	callFWExtInfo Struct Reference	112
8.64.1	Detailed Description	113
8.64.2	Field Documentation	115
8.64.2.1	noReplyTimer	115
8.64.2.2	number	115
8.64.2.3	numLen	115
8.64.2.4	numPlan	115
8.64.2.5	numType	115
8.64.2.6	PI	115
8.64.2.7	SI	115
8.64.2.8	SvcClass	115
8.64.2.9	SvcStatus	115
8.65	callFWInfo Struct Reference	115
8.65.1	Detailed Description	116
8.65.2	Field Documentation	116
8.65.2.1	noReplyTimer	116
8.65.2.2	number	116
8.65.2.3	numLen	116
8.65.2.4	SvcClass	116
8.65.2.5	SvcStatus	116
8.66	callInfo Struct Reference	116
8.66.1	Detailed Description	117
8.66.2	Field Documentation	118
8.66.2.1	callID	118
8.66.2.2	callState	118
8.66.2.3	callType	118
8.66.2.4	direction	118
8.66.2.5	mode	118

8.67	callingPartyInfo Struct Reference	118
8.67.1	Detailed Description	119
8.67.2	Field Documentation	120
8.67.2.1	number	120
8.67.2.2	numLen	120
8.67.2.3	numPlan	120
8.67.2.4	numType	120
8.67.2.5	PI	120
8.67.2.6	SI	120
8.68	cardResult Struct Reference	120
8.68.1	Detailed Description	120
8.68.2	Field Documentation	121
8.68.2.1	sw1	121
8.68.2.2	sw2	121
8.69	cardStatus Struct Reference	121
8.69.1	Detailed Description	121
8.69.2	Field Documentation	122
8.69.2.1	index1xPri	122
8.69.2.2	index1xSec	122
8.69.2.3	indexGwPri	122
8.69.2.4	indexGwSec	122
8.69.2.5	numSlot	122
8.69.2.6	SlotInfo	122
8.70	CatAlPhalIdentifierTlv Struct Reference	122
8.70.1	Detailed Description	122
8.70.2	Field Documentation	123
8.70.2.1	AlphaID	123
8.70.2.2	AlphaIDLength	123
8.70.2.3	ReferenceID	123
8.71	CatCommonEventTlv Struct Reference	123
8.71.1	Field Documentation	123
8.71.1.1	CatEvent	123
8.71.1.2	EventID	123
8.71.1.3	EventLength	123
8.71.1.4	TlvPresent	123
8.72	CatEndProactiveSessionTlv Struct Reference	123
8.72.1	Detailed Description	123
8.72.2	Field Documentation	124
8.72.2.1	EndProactiveSession	124
8.73	CATEventDataType Struct Reference	124



8.73.1	Field Documentation	124
8.73.1.1	eventMask	124
8.73.1.2	pErrorMask	124
8.74	CatEventIDDataTlv Struct Reference	124
8.74.1	Detailed Description	124
8.74.2	Field Documentation	124
8.74.2.1	Data	124
8.74.2.2	DataLength	124
8.74.2.3	ReferenceID	124
8.75	CatEventListTlv Struct Reference	124
8.75.1	Detailed Description	124
8.75.2	Field Documentation	125
8.75.2.1	SetupEventList	125
8.76	CatRefreshTlv Struct Reference	125
8.76.1	Detailed Description	125
8.76.2	Field Documentation	125
8.76.2.1	RefreshMode	125
8.76.2.2	RefreshStage	125
8.77	ccSUPSType Struct Reference	125
8.77.1	Detailed Description	125
8.77.2	Field Documentation	126
8.77.2.1	reason	126
8.77.2.2	svcType	126
8.78	CDMABroadcastConfig Struct Reference	126
8.78.1	Detailed Description	126
8.78.2	Field Documentation	127
8.78.2.1	language	127
8.78.2.2	selected	127
8.78.2.3	serviceCategory	127
8.79	CDMAChannel Struct Reference	127
8.79.1	Detailed Description	127
8.79.2	Field Documentation	128
8.79.2.1	priChA	128
8.79.2.2	priChB	128
8.79.2.3	secChA	128
8.79.2.4	secChB	128
8.80	CDMAECIOThresh Struct Reference	128
8.80.1	Detailed Description	128
8.80.2	Field Documentation	128
8.80.2.1	CDMAECIOThreshListLen	128

8.80.2.2	pCDMAECIOThreshList	128
8.81	CDMAInfo Struct Reference	128
8.81.1	Detailed Description	129
8.81.2	Field Documentation	129
8.81.2.1	baseId	129
8.81.2.2	baseLat	129
8.81.2.3	baseLong	129
8.81.2.4	nid	129
8.81.2.5	refpn	129
8.81.2.6	sid	130
8.82	cdmaMsgDecodingParams Struct Reference	130
8.82.1	Detailed Description	130
8.82.2	Field Documentation	132
8.82.2.1	absoluteValidity	132
8.82.2.2	mcTimeStamp	132
8.82.2.3	messageLength	132
8.82.2.4	pAlertPriority	132
8.82.2.5	pCallbkAddr	132
8.82.2.6	pCallbkAddrLength	132
8.82.2.7	pDisplayMode	132
8.82.2.8	pLanguage	132
8.82.2.9	pMessage	132
8.82.2.10	pMessageID	132
8.82.2.11	pPriority	132
8.82.2.12	pPrivacy	133
8.82.2.13	pReadAcknowledgementReq	133
8.82.2.14	pRelativeValidity	133
8.82.2.15	pSenderAddr	133
8.82.2.16	pSenderAddrLength	133
8.82.2.17	pTextMsg	133
8.82.2.18	pTextMsgLength	133
8.82.2.19	pUserAcknowledgementReq	133
8.83	cdmaMsgEncodingParams Struct Reference	133
8.83.1	Detailed Description	133
8.83.2	Field Documentation	134
8.83.2.1	messageld	134
8.83.2.2	pCallbackAddr	134
8.83.2.3	pDestAddr	134
8.83.2.4	pEncodingAlphabet	134
8.83.2.5	pMessage	134

8.83.2.6	pMessageSize	134
8.83.2.7	pPriority	134
8.83.2.8	pRelValidity	134
8.83.2.9	pTextMsg	135
8.83.2.10	textMsgLength	135
8.84	CDMARSSIThresh Struct Reference	135
8.84.1	Detailed Description	135
8.84.2	Field Documentation	135
8.84.2.1	CDMARSSIThreshListLen	135
8.84.2.2	pCDMARSSIThreshList	135
8.85	CDMASSInfo Struct Reference	135
8.85.1	Detailed Description	135
8.85.2	Field Documentation	136
8.85.2.1	ecio	136
8.85.2.2	rsi	136
8.86	CDMASysInfo Struct Reference	136
8.86.1	Detailed Description	136
8.86.2	Field Documentation	139
8.86.2.1	baseId	139
8.86.2.2	baseLat	140
8.86.2.3	baseLong	140
8.86.2.4	bsInfoValid	140
8.86.2.5	bsPRev	140
8.86.2.6	bsPRevValid	140
8.86.2.7	ccsSupported	140
8.86.2.8	ccsSupportedValid	140
8.86.2.9	cdmaSysIdValid	140
8.86.2.10	isSysPrIMatch	140
8.86.2.11	isSysPrIMatchValid	140
8.86.2.12	MCC	140
8.86.2.13	MNC	140
8.86.2.14	networkID	140
8.86.2.15	networkIdValid	140
8.86.2.16	packetZone	140
8.86.2.17	packetZoneValid	140
8.86.2.18	pRevInUse	140
8.86.2.19	pRevInUseValid	140
8.86.2.20	sysInfoCDMA	140
8.86.2.21	systemID	140
8.87	CDMASysInfoExt Struct Reference	140

8.87.1 Detailed Description . . . . .	140
8.87.2 Field Documentation . . . . .	141
8.87.2.1 imsi_11_12 . . . . .	141
8.87.2.2 MCC . . . . .	141
8.88 CellDb Struct Reference . . . . .	141
8.88.1 Detailed Description . . . . .	141
8.88.2 Field Documentation . . . . .	141
8.88.2.1 mask . . . . .	141
8.89 cellParams Struct Reference . . . . .	141
8.89.1 Detailed Description . . . . .	142
8.89.2 Field Documentation . . . . .	142
8.89.2.1 pci . . . . .	142
8.89.2.2 rsrp . . . . .	142
8.89.2.3 rsrq . . . . .	142
8.89.2.4 rssi . . . . .	142
8.89.2.5 srxlev . . . . .	142
8.90 changeUIMPIN Struct Reference . . . . .	142
8.90.1 Detailed Description . . . . .	143
8.90.2 Field Documentation . . . . .	143
8.90.2.1 oldPINLen . . . . .	143
8.90.2.2 oldPINVal . . . . .	143
8.90.2.3 pinID . . . . .	143
8.90.2.4 pinLen . . . . .	143
8.90.2.5 pinVal . . . . .	143
8.91 channelRate Struct Reference . . . . .	143
8.91.1 Detailed Description . . . . .	144
8.91.2 Field Documentation . . . . .	144
8.91.2.1 CurrChanRxRate . . . . .	144
8.91.2.2 CurrChanTxRate . . . . .	144
8.92 ChannelRate Struct Reference . . . . .	144
8.92.1 Detailed Description . . . . .	144
8.92.2 Field Documentation . . . . .	145
8.92.2.1 CurrChanRxRate . . . . .	145
8.92.2.2 CurrChanTxRate . . . . .	145
8.92.2.3 MaxChanRxRate . . . . .	145
8.92.2.4 MaxChanTxRate . . . . .	145
8.93 CLIPResp Struct Reference . . . . .	145
8.93.1 Detailed Description . . . . .	145
8.93.2 Field Documentation . . . . .	146
8.93.2.1 ActiveStatus . . . . .	146

8.93.2.2 ProvisionStatus . . . . .	146
8.94 CLIRResp Struct Reference . . . . .	146
8.94.1 Detailed Description . . . . .	146
8.94.2 Field Documentation . . . . .	146
8.94.2.1 ActiveStatus . . . . .	146
8.94.2.2 ProvisionStatus . . . . .	146
8.95 ClkInfo Struct Reference . . . . .	147
8.95.1 Detailed Description . . . . .	147
8.95.2 Field Documentation . . . . .	148
8.95.2.1 mask . . . . .	149
8.96 CNAPResp Struct Reference . . . . .	149
8.96.1 Detailed Description . . . . .	149
8.96.2 Field Documentation . . . . .	149
8.96.2.1 ActiveStatus . . . . .	149
8.96.2.2 ProvisionStatus . . . . .	149
8.97 COLPResp Struct Reference . . . . .	149
8.97.1 Detailed Description . . . . .	149
8.97.2 Field Documentation . . . . .	150
8.97.2.1 ActiveStatus . . . . .	150
8.97.2.2 ProvisionStatus . . . . .	150
8.98 COLRResp Struct Reference . . . . .	150
8.98.1 Detailed Description . . . . .	150
8.98.2 Field Documentation . . . . .	151
8.98.2.1 ActiveStatus . . . . .	151
8.98.2.2 ProvisionStatus . . . . .	151
8.99 CommInfo Struct Reference . . . . .	151
8.99.1 Detailed Description . . . . .	151
8.99.2 Field Documentation . . . . .	153
8.99.2.1 imsRegState . . . . .	153
8.99.2.2 modemMode . . . . .	153
8.99.2.3 psState . . . . .	153
8.99.2.4 systemMode . . . . .	153
8.99.2.5 temperature . . . . .	153
8.100ConnectionStatus Struct Reference . . . . .	153
8.100.1 Detailed Description . . . . .	153
8.100.2 Field Documentation . . . . .	154
8.100.2.1 MDMCallDuration . . . . .	154
8.100.2.2 MDMConnStatus . . . . .	154
8.101connectNumInfo Struct Reference . . . . .	154
8.101.1 Detailed Description . . . . .	154

8.101.2 Field Documentation	156
8.101.2.1 callerID	156
8.101.2.2 callerIDLen	156
8.101.2.3 numPlan	156
8.101.2.4 numPresInd	156
8.101.2.5 numType	156
8.101.2.6 screeningInd	156
8.102 CrashInfo Struct Reference	156
8.102.1 Detailed Description	157
8.102.2 Field Documentation	158
8.102.2.1 crashData	158
8.102.2.2 crashId	158
8.102.2.3 crashStrLen	158
8.102.2.4 gcDumpStrLen	158
8.102.2.5 numCrashes	158
8.102.2.6 pCrashString	158
8.102.2.7 pGCDumpString	158
8.103 CrashInfoParams Struct Reference	158
8.103.1 Detailed Description	158
8.103.2 Field Documentation	159
8.103.2.1 pCrashInfo	159
8.103.2.2 pDevCrashStatus	159
8.104 CreateProfileIn Struct Reference	159
8.104.1 Detailed Description	159
8.104.2 Field Documentation	160
8.104.2.1 curProfile	160
8.104.2.2 pProfileID	160
8.104.2.3 pProfileType	160
8.105 CreateProfileOut Struct Reference	160
8.105.1 Detailed Description	160
8.105.2 Field Documentation	160
8.105.2.1 pExtErrorCode	160
8.105.2.2 pProfileIndex	160
8.105.2.3 pProfileType	160
8.106 CSGID Struct Reference	160
8.106.1 Detailed Description	161
8.106.2 Field Documentation	161
8.106.2.1 id	161
8.106.2.2 mcc	161
8.106.2.3 mnc	161

8.106.2.4 mncPcsDigits . . . . .	161
8.106.2.5 rat . . . . .	161
8.107CUGInfo Struct Reference . . . . .	161
8.107.1 Detailed Description . . . . .	162
8.107.2 Field Documentation . . . . .	162
8.107.2.1 CUGIndex . . . . .	162
8.107.2.2 SuppOA . . . . .	162
8.107.2.3 SuppPrefCUG . . . . .	162
8.108curAMRConfig Struct Reference . . . . .	162
8.108.1 Detailed Description . . . . .	162
8.108.2 Field Documentation . . . . .	163
8.108.2.1 gsmAmrStat . . . . .	163
8.108.2.2 wcdmaAmrStat . . . . .	163
8.109CurrDataSysStat Struct Reference . . . . .	163
8.109.1 Detailed Description . . . . .	163
8.109.2 Field Documentation . . . . .	164
8.109.2.1 pCurrNetworkInfo . . . . .	164
8.109.2.2 pNetworkInfoLen . . . . .	164
8.109.2.3 pPrefNetwork . . . . .	164
8.110currentCatEvent Union Reference . . . . .	164
8.110.1 Detailed Description . . . . .	164
8.110.2 Field Documentation . . . . .	165
8.110.2.1 CatAlphaldtfr . . . . .	165
8.110.2.2 CatEndPS . . . . .	165
8.110.2.3 CatEventLst . . . . .	165
8.110.2.4 CatEvIDData . . . . .	165
8.110.2.5 CatRefresh . . . . .	165
8.111CurrentImgLst Struct Reference . . . . .	165
8.111.1 Detailed Description . . . . .	165
8.111.2 Field Documentation . . . . .	165
8.111.2.1 carrier . . . . .	165
8.111.2.2 fwvers . . . . .	166
8.111.2.3 numEntries . . . . .	166
8.111.2.4 pCurrImglInfo . . . . .	166
8.111.2.5 pkgver . . . . .	166
8.111.2.6 priver . . . . .	166
8.112currentPLMN Struct Reference . . . . .	166
8.112.1 Detailed Description . . . . .	166
8.112.2 Field Documentation . . . . .	166
8.112.2.1 MCC . . . . .	167

8.112.2.2 MNC . . . . .	167
8.112.2.3 netDescr . . . . .	167
8.112.2.4 netDescrLength . . . . .	167
8.113CurrImageInfo Struct Reference . . . . .	167
8.113.1 Detailed Description . . . . .	167
8.113.2 Field Documentation . . . . .	167
8.113.2.1 buildID . . . . .	167
8.113.2.2 buildIDLen . . . . .	167
8.113.2.3 imageType . . . . .	167
8.113.2.4 uniqueID . . . . .	167
8.114CurrNetworkInfo Struct Reference . . . . .	168
8.114.1 Detailed Description . . . . .	168
8.114.2 Field Documentation . . . . .	170
8.114.2.1 NetworkType . . . . .	170
8.114.2.2 RATMask . . . . .	170
8.114.2.3 SOMask . . . . .	170
8.115custFeaturesInfo Struct Reference . . . . .	170
8.115.1 Detailed Description . . . . .	170
8.115.2 Field Documentation . . . . .	172
8.115.2.1 GpsEnable . . . . .	172
8.115.2.2 pDHCPRelayEnabled . . . . .	172
8.115.2.3 pDisableIMSI . . . . .	172
8.115.2.4 pGPSLPM . . . . .	172
8.115.2.5 pGPSSel . . . . .	172
8.115.2.6 pIPFamSupport . . . . .	172
8.115.2.7 plsVoiceEnabled . . . . .	172
8.115.2.8 pRMAutoConnect . . . . .	172
8.115.2.9 pSMSSupport . . . . .	172
8.116custFeaturesSetting Struct Reference . . . . .	172
8.116.1 Detailed Description . . . . .	173
8.116.2 Field Documentation . . . . .	174
8.116.2.1 pDHCPRelayEnabled . . . . .	174
8.116.2.2 pGPSEnable . . . . .	174
8.116.2.3 pGPSLPM . . . . .	174
8.116.2.4 pGPSSel . . . . .	174
8.116.2.5 plsVoiceEnabled . . . . .	174
8.117custSettingInfo Struct Reference . . . . .	174
8.117.1 Detailed Description . . . . .	175
8.117.2 Field Documentation . . . . .	175
8.117.2.1 cust_attr . . . . .	175



8.117.2.2 cust_id . . . . .	175
8.117.2.3 cust_value . . . . .	175
8.117.2.4 id_length . . . . .	175
8.117.2.5 value_length . . . . .	175
8.118custSettingList Struct Reference . . . . .	175
8.118.1 Detailed Description . . . . .	176
8.118.2 Field Documentation . . . . .	176
8.118.2.1 custSetting . . . . .	176
8.118.2.2 list_type . . . . .	176
8.118.2.3 num_instances . . . . .	176
8.119dataBearers Struct Reference . . . . .	176
8.119.1 Detailed Description . . . . .	176
8.119.2 Field Documentation . . . . .	177
8.119.2.1 dataBearerMask . . . . .	177
8.119.2.2 pCurDataBearerTechnology . . . . .	177
8.119.2.3 pLastCallDataBearerTechnology . . . . .	177
8.120DataBearerTech Struct Reference . . . . .	177
8.120.1 Detailed Description . . . . .	177
8.120.2 Field Documentation . . . . .	179
8.120.2.1 ratValue . . . . .	179
8.120.2.2 soMask . . . . .	179
8.120.2.3 techType . . . . .	179
8.121DataBearerTechExt Struct Reference . . . . .	179
8.121.1 Detailed Description . . . . .	179
8.121.2 Field Documentation . . . . .	179
8.121.2.1 pBearerTech . . . . .	179
8.121.2.2 pLastBearerTech . . . . .	179
8.122dataBearerTechnology Struct Reference . . . . .	179
8.122.1 Detailed Description . . . . .	179
8.122.2 Field Documentation . . . . .	181
8.122.2.1 currentNetwork . . . . .	181
8.122.2.2 ratMask . . . . .	181
8.122.2.3 soMask . . . . .	181
8.123dataRate Struct Reference . . . . .	181
8.123.1 Detailed Description . . . . .	181
8.123.2 Field Documentation . . . . .	181
8.123.2.1 dataRateMax . . . . .	181
8.123.2.2 guaranteedRate . . . . .	181
8.124dataSrvCapabilities Struct Reference . . . . .	181
8.124.1 Detailed Description . . . . .	182

8.124.2 Field Documentation	182
8.124.2.1 dataCapabilities	182
8.124.2.2 dataCapabilitiesLen	182
8.125DataStatusDetail Struct Reference	182
8.125.1 Detailed Description	182
8.125.2 Field Documentation	184
8.125.2.1 IPAddress	184
8.125.2.2 LastErrCode	184
8.126DataULongLongTlv Struct Reference	184
8.126.1 Field Documentation	184
8.126.1.1 TlvPresent	184
8.126.1.2 ullData	184
8.127DataULongTlv Struct Reference	184
8.127.1 Field Documentation	184
8.127.1.1 TlvPresent	184
8.127.1.2 ulData	184
8.128DcsUsbPortNames Struct Reference	184
8.128.1 Field Documentation	184
8.128.1.1 AtCmdPort	184
8.128.1.2 DmPort	184
8.128.1.3 NmeaPort	184
8.129delAssistDataStatus Struct Reference	184
8.129.1 Detailed Description	185
8.129.2 Field Documentation	185
8.129.2.1 status	185
8.130depersonalizationInformation Struct Reference	185
8.130.1 Detailed Description	186
8.130.2 Field Documentation	187
8.130.2.1 ckLen	187
8.130.2.2 ckVal	187
8.130.2.3 feature	187
8.130.2.4 operation	187
8.131detailSvcInfo Struct Reference	187
8.131.1 Detailed Description	188
8.131.2 Field Documentation	189
8.131.2.1 hdrHybrid	189
8.131.2.2 hdrSrvStatus	189
8.131.2.3 isSysForbidden	189
8.131.2.4 srvCapability	189
8.131.2.5 srvStatus	189

8.132DeviceConfigDetail Struct Reference . . . . .	189
8.132.1 Detailed Description . . . . .	189
8.132.2 Field Documentation . . . . .	190
8.132.2.1 Chipset . . . . .	190
8.132.2.2 HWVersion . . . . .	190
8.132.2.3 QLIC . . . . .	190
8.132.2.4 Technology . . . . .	190
8.133diagInfo Struct Reference . . . . .	190
8.133.1 Detailed Description . . . . .	191
8.133.2 Field Documentation . . . . .	191
8.133.2.1 diagInfoLen . . . . .	191
8.133.2.2 diagnosticInfo . . . . .	191
8.134dirNum Struct Reference . . . . .	191
8.134.1 Detailed Description . . . . .	191
8.134.2 Field Documentation . . . . .	191
8.134.2.1 dirNum . . . . .	191
8.134.2.2 dirNumLen . . . . .	191
8.135dmsCurrentPRLInfo Struct Reference . . . . .	191
8.135.1 Detailed Description . . . . .	192
8.135.2 Field Documentation . . . . .	192
8.135.2.1 pPRLPreference . . . . .	192
8.135.2.2 pPRLVersion . . . . .	192
8.136Domain Struct Reference . . . . .	192
8.136.1 Detailed Description . . . . .	192
8.136.2 Field Documentation . . . . .	192
8.136.2.1 domainLen . . . . .	192
8.136.2.2 domainName . . . . .	192
8.137DomainNameList Struct Reference . . . . .	193
8.137.1 Detailed Description . . . . .	193
8.137.2 Field Documentation . . . . .	193
8.137.2.1 domain . . . . .	193
8.137.2.2 numInstances . . . . .	193
8.138DRCPParams Struct Reference . . . . .	193
8.138.1 Detailed Description . . . . .	193
8.138.2 Field Documentation . . . . .	193
8.138.2.1 DRCCover . . . . .	193
8.138.2.2 DRCValue . . . . .	193
8.139DTMFInfo Struct Reference . . . . .	194
8.139.1 Detailed Description . . . . .	194
8.139.2 Field Documentation . . . . .	194

8.139.2.1 callID . . . . .	194
8.139.2.2 digitBuff . . . . .	194
8.139.2.3 digitCnt . . . . .	194
8.139.2.4 DTMFEvent . . . . .	194
8.140DTMFLengths Struct Reference . . . . .	194
8.140.1 Detailed Description . . . . .	195
8.140.2 Field Documentation . . . . .	195
8.140.2.1 DTMFInterdigitInterval . . . . .	195
8.140.2.2 DTMFPulseWidth . . . . .	195
8.141DUNCallInfoInd Struct Reference . . . . .	195
8.141.1 Field Documentation . . . . .	195
8.141.1.1 CallEndReason . . . . .	195
8.141.1.2 ChannelRate . . . . .	196
8.141.1.3 DataBearerTech . . . . .	196
8.141.1.4 DormancyStatus . . . . .	196
8.141.1.5 MdmConnStatus . . . . .	196
8.141.1.6 RXOKBytesCount . . . . .	196
8.141.1.7 TXOKBytesCount . . . . .	196
8.142ecioListElement Struct Reference . . . . .	196
8.142.1 Detailed Description . . . . .	196
8.142.2 Field Documentation . . . . .	196
8.142.2.1 ecio . . . . .	196
8.142.2.2 radiolf . . . . .	196
8.143ECIOThresh Struct Reference . . . . .	196
8.143.1 Detailed Description . . . . .	197
8.143.2 Field Documentation . . . . .	198
8.143.2.1 ECIOThresListLen . . . . .	198
8.143.2.2 pECIOThresList . . . . .	198
8.144ECTNum Struct Reference . . . . .	198
8.144.1 Detailed Description . . . . .	198
8.144.2 Field Documentation . . . . .	199
8.144.2.1 ECTCallState . . . . .	199
8.144.2.2 number . . . . .	199
8.144.2.3 presentationInd . . . . .	199
8.145encryptedPIN1 Struct Reference . . . . .	199
8.145.1 Detailed Description . . . . .	199
8.145.2 Field Documentation . . . . .	199
8.145.2.1 pin1Len . . . . .	199
8.145.2.2 pin1Val . . . . .	199
8.146ERIFileparams Struct Reference . . . . .	200

8.146.1 Detailed Description . . . . .	200
8.146.2 Field Documentation . . . . .	200
8.146.2.1 pFile . . . . .	200
8.146.2.2 pFileSize . . . . .	200
8.147errorRateListElement Struct Reference . . . . .	200
8.147.1 Detailed Description . . . . .	200
8.147.2 Field Documentation . . . . .	201
8.147.2.1 errorRate . . . . .	201
8.147.2.2 radiolf . . . . .	201
8.148extDispRecInfo Struct Reference . . . . .	201
8.148.1 Detailed Description . . . . .	202
8.148.2 Field Documentation . . . . .	203
8.148.2.1 dispType . . . . .	203
8.148.2.2 extDisplInfo . . . . .	203
8.148.2.3 extDisplInfoLen . . . . .	203
8.149FactorySequenceNumber Struct Reference . . . . .	203
8.149.1 Detailed Description . . . . .	203
8.149.2 Field Documentation . . . . .	203
8.149.2.1 FSNumber . . . . .	203
8.150fileAttributes Struct Reference . . . . .	203
8.150.1 Detailed Description . . . . .	204
8.150.2 Field Documentation . . . . .	208
8.150.2.1 fileID . . . . .	208
8.150.2.2 fileSize . . . . .	208
8.150.2.3 fileType . . . . .	208
8.150.2.4 rawLen . . . . .	208
8.150.2.5 rawValue . . . . .	208
8.150.2.6 recordCount . . . . .	208
8.150.2.7 recordSize . . . . .	208
8.150.2.8 secActivate . . . . .	208
8.150.2.9 secActivateMask . . . . .	208
8.150.2.10secDeactivate . . . . .	208
8.150.2.11secDeactivateMask . . . . .	208
8.150.2.12secIncrease . . . . .	208
8.150.2.13secIncreaseMask . . . . .	208
8.150.2.14secRead . . . . .	208
8.150.2.15secReadMask . . . . .	208
8.150.2.16secWrite . . . . .	208
8.150.2.17secWriteMask . . . . .	208
8.151fileInfo Struct Reference . . . . .	208

8.151.1 Detailed Description . . . . .	209
8.151.2 Field Documentation . . . . .	210
8.151.2.1 fileID . . . . .	210
8.151.2.2 path . . . . .	210
8.151.2.3 pathLen . . . . .	210
8.152FirmwareUpdatStat Struct Reference . . . . .	210
8.152.1 Detailed Description . . . . .	210
8.152.2 Field Documentation . . . . .	211
8.152.2.1 plmgType . . . . .	211
8.152.2.2 pRefData . . . . .	211
8.152.2.3 pRefString . . . . .	211
8.152.2.4 pRefStringLength . . . . .	211
8.152.2.5 ResCode . . . . .	211
8.153fwinfo_s Struct Reference . . . . .	212
8.153.1 Detailed Description . . . . .	212
8.153.2 Field Documentation . . . . .	212
8.153.2.1 Carrier . . . . .	212
8.153.2.2 FirmwareID . . . . .	212
8.153.2.3 GPSCapability . . . . .	212
8.153.2.4 Region . . . . .	212
8.153.2.5 Technology . . . . .	212
8.154GERANInfo Struct Reference . . . . .	212
8.154.1 Detailed Description . . . . .	213
8.154.2 Field Documentation . . . . .	214
8.154.2.1 arfcn . . . . .	214
8.154.2.2 bsic . . . . .	214
8.154.2.3 cellID . . . . .	214
8.154.2.4 insNmrCellInfo . . . . .	214
8.154.2.5 lac . . . . .	214
8.154.2.6 nmInst . . . . .	214
8.154.2.7 plmn . . . . .	214
8.154.2.8 rxLev . . . . .	214
8.154.2.9 timingAdvance . . . . .	214
8.155geranInstInfo Struct Reference . . . . .	214
8.155.1 Detailed Description . . . . .	215
8.155.2 Field Documentation . . . . .	216
8.155.2.1 geranArfcn . . . . .	216
8.155.2.2 geranBsicBcc . . . . .	216
8.155.2.3 geranBsicNcc . . . . .	216
8.155.2.4 geranRssi . . . . .	216

8.156getAllCallInformation Struct Reference	216
8.156.1 Detailed Description	216
8.156.2 Field Documentation	217
8.156.2.1 ALS	217
8.156.2.2 Callinfo	217
8.156.2.3 isEmpty	217
8.157getAllCallRmtPtyName Struct Reference	217
8.157.1 Detailed Description	217
8.157.2 Field Documentation	217
8.157.2.1 callID	217
8.157.2.2 RemotePartyName	217
8.158getAllCallRmtPtyNum Struct Reference	217
8.158.1 Detailed Description	218
8.158.2 Field Documentation	219
8.158.2.1 callID	219
8.158.2.2 RemotePartyNum	219
8.159GetAudioPathConfigReq Struct Reference	219
8.159.1 Detailed Description	219
8.159.2 Field Documentation	220
8.159.2.1 Item	220
8.159.2.2 Profile	220
8.160GetAudioPathConfigResp Struct Reference	220
8.160.1 Detailed Description	220
8.160.2 Field Documentation	221
8.160.2.1 pCodecSTGain	221
8.160.2.2 pDTMFTXGain	222
8.160.2.3 pECMode	222
8.160.2.4 pMICGainSelect	222
8.160.2.5 pNSEnable	222
8.160.2.6 pRXAGCList	222
8.160.2.7 pRXAVCAGCSwitch	222
8.160.2.8 pRXAVCList	222
8.160.2.9 pRXPCMIIRFtr	222
8.160.2.10pTXAGCList	222
8.160.2.11pTXAVCSwitch	222
8.160.2.12pTXGain	222
8.160.2.13pTXPCMIIRFtr	222
8.161GetAudioProfileReq Struct Reference	222
8.161.1 Detailed Description	222
8.161.2 Field Documentation	222

8.161.2.1 Generator . . . . .	222
8.162GetAudioProfileResp Struct Reference . . . . .	222
8.162.1 Detailed Description . . . . .	223
8.162.2 Field Documentation . . . . .	223
8.162.2.1 EarMute . . . . .	223
8.162.2.2 MicMute . . . . .	223
8.162.2.3 Profile . . . . .	223
8.162.2.4 Volume . . . . .	224
8.163GetAudioVolTLBConfigReq Struct Reference . . . . .	224
8.163.1 Detailed Description . . . . .	224
8.163.2 Field Documentation . . . . .	224
8.163.2.1 Generator . . . . .	224
8.163.2.2 Item . . . . .	224
8.163.2.3 Profile . . . . .	224
8.163.2.4 Volume . . . . .	224
8.164GetAudioVolTLBConfigResp Struct Reference . . . . .	224
8.164.1 Detailed Description . . . . .	225
8.164.2 Field Documentation . . . . .	225
8.164.2.1 ResCode . . . . .	225
8.165getCallFWExtInfo Struct Reference . . . . .	225
8.165.1 Detailed Description . . . . .	225
8.165.2 Field Documentation . . . . .	225
8.165.2.1 CallFWExtInfo . . . . .	225
8.165.2.2 numInstances . . . . .	225
8.166getCallFWInfo Struct Reference . . . . .	225
8.166.1 Detailed Description . . . . .	226
8.166.2 Field Documentation . . . . .	226
8.166.2.1 CallFWInfo . . . . .	226
8.166.2.2 numInstances . . . . .	226
8.167getCustomFeatureV2 Struct Reference . . . . .	226
8.167.1 Detailed Description . . . . .	226
8.167.2 Field Documentation . . . . .	226
8.167.2.1 pCustSettingInfo . . . . .	227
8.167.2.2 pCustSettingList . . . . .	227
8.167.2.3 pGetCustomInput . . . . .	227
8.168getCustomInput Struct Reference . . . . .	227
8.168.1 Detailed Description . . . . .	227
8.168.2 Field Documentation . . . . .	227
8.168.2.1 cust_id . . . . .	227
8.168.2.2 list_type . . . . .	227



8.169getDUNCallInfoReq Struct Reference . . . . .	227
8.169.1 Detailed Description . . . . .	227
8.169.2 Field Documentation . . . . .	228
8.169.2.1 Mask . . . . .	228
8.169.2.2 pReportChannelRate . . . . .	229
8.169.2.3 pReportConnStatus . . . . .	229
8.169.2.4 pReportDataBearerTech . . . . .	229
8.169.2.5 pReportDormStatus . . . . .	229
8.169.2.6 pTransferStatInd . . . . .	229
8.170getDUNCallInfoResp Struct Reference . . . . .	229
8.170.1 Detailed Description . . . . .	229
8.170.2 Field Documentation . . . . .	231
8.170.2.1 pCallEndReason . . . . .	231
8.170.2.2 pChannelRate . . . . .	231
8.170.2.3 pConnectionStatus . . . . .	231
8.170.2.4 pDataBearerTech . . . . .	231
8.170.2.5 pDormancyStatus . . . . .	231
8.170.2.6 pLastCallDataBearerTech . . . . .	231
8.170.2.7 pLastCallRXOKBytesCnt . . . . .	231
8.170.2.8 pLastCallTXOKBytesCnt . . . . .	231
8.170.2.9 pMdmCallDurationActive . . . . .	231
8.170.2.10pRXOKBytesCount . . . . .	231
8.170.2.11pTXOKBytesCount . . . . .	231
8.171GetErrRateResp Struct Reference . . . . .	231
8.171.1 Detailed Description . . . . .	231
8.171.2 Field Documentation . . . . .	232
8.171.2.1 pCDMAFrameErrRate . . . . .	232
8.171.2.2 pGSMBER . . . . .	232
8.171.2.3 pHDRPackErrRate . . . . .	232
8.171.2.4 pWCDMABER . . . . .	232
8.172GetHRPDStatsResp Struct Reference . . . . .	232
8.172.1 Detailed Description . . . . .	232
8.172.2 Field Documentation . . . . .	233
8.172.2.1 pDRCPParams . . . . .	233
8.172.2.2 pPilotSetData . . . . .	233
8.172.2.3 pUATI . . . . .	233
8.173GetIMSSMSConfigParams Struct Reference . . . . .	233
8.173.1 Detailed Description . . . . .	233
8.173.2 Field Documentation . . . . .	234
8.173.2.1 pPhoneCtxtURI . . . . .	234

8.173.2.2 pPhoneCtxtURLen . . . . .	234
8.173.2.3 pSettingResp . . . . .	234
8.173.2.4 pSMSFormat . . . . .	234
8.173.2.5 pSMSOverIPNwInd . . . . .	234
8.174 GetIMSUserConfigParams Struct Reference . . . . .	234
8.174.1 Detailed Description . . . . .	234
8.174.2 Field Documentation . . . . .	234
8.174.2.1 pIMSDomain . . . . .	234
8.174.2.2 pIMSDomainLen . . . . .	234
8.174.2.3 pSettingResp . . . . .	235
8.175 GetIMSVoIPConfigResp Struct Reference . . . . .	235
8.175.1 Detailed Description . . . . .	235
8.175.2 Field Documentation . . . . .	237
8.175.2.1 pAmrMode . . . . .	237
8.175.2.2 pAmrOctetAligned . . . . .	237
8.175.2.3 pAmrWbEnable . . . . .	237
8.175.2.4 pAmrWBMode . . . . .	237
8.175.2.5 pAmrWBOctetAligned . . . . .	237
8.175.2.6 pMinSessionExpiryTimer . . . . .	237
8.175.2.7 pRingBackTimer . . . . .	237
8.175.2.8 pRingingTimer . . . . .	237
8.175.2.9 pRTPRTCPInactTimer . . . . .	237
8.175.2.10 pScrAmrEnable . . . . .	237
8.175.2.11 pScrAmrWbEnable . . . . .	237
8.175.2.12 pSessionExpiryTimer . . . . .	238
8.175.2.13 pSettingResp . . . . .	238
8.176 GetInstIDResp Struct Reference . . . . .	238
8.176.1 Field Documentation . . . . .	238
8.176.1.1 pInstanceId . . . . .	238
8.176.1.2 pIPFamily . . . . .	238
8.177 GetM2MAudioProfileReq Struct Reference . . . . .	238
8.177.1 Detailed Description . . . . .	238
8.177.2 Field Documentation . . . . .	238
8.177.2.1 pGenerator . . . . .	238
8.178 GetM2MAudioProfileResp Struct Reference . . . . .	238
8.178.1 Detailed Description . . . . .	239
8.178.2 Field Documentation . . . . .	240
8.178.2.1 CwtMute . . . . .	240
8.178.2.2 EarMute . . . . .	240
8.178.2.3 Generator . . . . .	240

8.178.2.4 MicMute . . . . .	240
8.178.2.5 Profile . . . . .	240
8.178.2.6 Volume . . . . .	240
8.179GetM2MAudioVolumeReq Struct Reference . . . . .	241
8.179.1 Detailed Description . . . . .	241
8.179.2 Field Documentation . . . . .	241
8.179.2.1 Generator . . . . .	241
8.179.2.2 Profile . . . . .	241
8.180GetM2MAudioVolumeResp Struct Reference . . . . .	241
8.180.1 Detailed Description . . . . .	241
8.180.2 Field Documentation . . . . .	241
8.180.2.1 Level . . . . .	241
8.181GetM2MAVMuteReq Struct Reference . . . . .	242
8.181.1 Detailed Description . . . . .	242
8.181.2 Field Documentation . . . . .	242
8.181.2.1 Profile . . . . .	242
8.182GetM2MAVMuteResp Struct Reference . . . . .	242
8.182.1 Detailed Description . . . . .	242
8.182.2 Field Documentation . . . . .	243
8.182.2.1 CwtMute . . . . .	243
8.182.2.2 EarMute . . . . .	243
8.182.2.3 MicMute . . . . .	243
8.183GetM2MSpkrGainReq Struct Reference . . . . .	243
8.183.1 Detailed Description . . . . .	243
8.183.2 Field Documentation . . . . .	243
8.183.2.1 Profile . . . . .	243
8.184GetM2MSpkrGainResp Struct Reference . . . . .	243
8.184.1 Detailed Description . . . . .	243
8.184.2 Field Documentation . . . . .	244
8.184.2.1 Value . . . . .	244
8.185getMsgWaitingInfo Struct Reference . . . . .	244
8.185.1 Detailed Description . . . . .	244
8.185.2 Field Documentation . . . . .	244
8.185.2.1 msgWaitInfo . . . . .	244
8.185.2.2 numInstances . . . . .	244
8.186GetRegMgrConfigParams Struct Reference . . . . .	244
8.186.1 Detailed Description . . . . .	245
8.186.2 Field Documentation . . . . .	246
8.186.2.1 pIMSTestMode . . . . .	246
8.186.2.2 pPCSCFPort . . . . .	246

8.186.2.3 pPriCSCFPortName . . . . .	246
8.186.2.4 pPriCSCFPortNameLen . . . . .	246
8.186.2.5 pSettingResp . . . . .	246
8.187GetSessionIDResp Struct Reference . . . . .	246
8.187.1 Field Documentation . . . . .	246
8.187.1.1 pSessionIDv4 . . . . .	246
8.187.1.2 pSessionIDv6 . . . . .	246
8.188GetSIPConfigResp Struct Reference . . . . .	247
8.188.1 Detailed Description . . . . .	247
8.188.2 Field Documentation . . . . .	247
8.188.2.1 pSettingResp . . . . .	247
8.188.2.2 pSigCompEnabled . . . . .	248
8.188.2.3 pSIPLocalPort . . . . .	248
8.188.2.4 pSubscribeTimer . . . . .	248
8.188.2.5 pTimerSIPReg . . . . .	248
8.188.2.6 pTimerT1 . . . . .	248
8.188.2.7 pTimerT2 . . . . .	248
8.188.2.8 pTimerTf . . . . .	248
8.189GnssData Struct Reference . . . . .	248
8.189.1 Detailed Description . . . . .	248
8.189.2 Field Documentation . . . . .	250
8.189.2.1 mask . . . . .	250
8.190gnssSvInfoNotification Struct Reference . . . . .	250
8.190.1 Detailed Description . . . . .	250
8.190.2 Field Documentation . . . . .	250
8.190.2.1 bAltitudeAssumed . . . . .	250
8.190.2.2 pSatelliteInfo . . . . .	250
8.191GPRSQoS Struct Reference . . . . .	250
8.191.1 Detailed Description . . . . .	250
8.191.2 Field Documentation . . . . .	251
8.191.2.1 delayClass . . . . .	251
8.191.2.2 meanThroughputClass . . . . .	251
8.191.2.3 peakThroughputClass . . . . .	251
8.191.2.4 precedenceClass . . . . .	251
8.191.2.5 reliabilityClass . . . . .	251
8.192GPRSRequestedQoS Struct Reference . . . . .	251
8.192.1 Detailed Description . . . . .	251
8.192.2 Field Documentation . . . . .	252
8.192.2.1 delayClass . . . . .	252
8.192.2.2 meanThroughputClass . . . . .	252

8.192.2.3 peakThroughputClass . . . . .	252
8.192.2.4 precedenceClass . . . . .	252
8.192.2.5 reliabilityClass . . . . .	252
8.193GPSStateInfo Struct Reference . . . . .	252
8.193.1 Detailed Description . . . . .	253
8.193.2 Field Documentation . . . . .	255
8.193.2.1 Altitude . . . . .	255
8.193.2.2 EngineState . . . . .	256
8.193.2.3 glo_almanac_sv_msk . . . . .	256
8.193.2.4 glo_ephemeris_sv_msk . . . . .	256
8.193.2.5 glo_health_sv_msk . . . . .	256
8.193.2.6 glo_visible_sv_msk . . . . .	256
8.193.2.7 gps_almanac_sv_msk . . . . .	256
8.193.2.8 gps_ephemeris_sv_msk . . . . .	256
8.193.2.9 gps_health_sv_msk . . . . .	256
8.193.2.10gps_visible_sv_msk . . . . .	256
8.193.2.11HorizontalUncertainty . . . . .	256
8.193.2.12iono_valid . . . . .	256
8.193.2.13Latitude . . . . .	256
8.193.2.14Longitude . . . . .	256
8.193.2.15sbas_almanac_sv_msk . . . . .	256
8.193.2.16sbas_ephemeris_sv_msk . . . . .	256
8.193.2.17sbas_health_sv_msk . . . . .	256
8.193.2.18sbas_visible_sv_msk . . . . .	256
8.193.2.19Time_uncert_ms . . . . .	256
8.193.2.20TimeStmp_gps_week . . . . .	256
8.193.2.21TimeStmp_tow_ms . . . . .	256
8.193.2.22ValidMask . . . . .	256
8.193.2.23VerticalUncertainty . . . . .	256
8.193.2.24xtra_start_gps_minutes . . . . .	256
8.193.2.25xtra_start_gps_week . . . . .	256
8.193.2.26xtra_valid_duration_hours . . . . .	256
8.194gpsTime_s Struct Reference . . . . .	256
8.194.1 Detailed Description . . . . .	257
8.194.2 Field Documentation . . . . .	257
8.194.2.1 gpsTimeOfWeekMs . . . . .	257
8.194.2.2 gpsWeek . . . . .	257
8.195gsmCellInfo Struct Reference . . . . .	257
8.195.1 Detailed Description . . . . .	257
8.195.2 Field Documentation . . . . .	258

8.195.2.1 arfcn . . . . .	258
8.195.2.2 band1900 . . . . .	258
8.195.2.3 bsicld . . . . .	258
8.195.2.4 cellldValid . . . . .	258
8.195.2.5 rssi . . . . .	258
8.195.2.6 srxlev . . . . .	258
8.196GSMRSSIThresh Struct Reference . . . . .	258
8.196.1 Detailed Description . . . . .	258
8.196.2 Field Documentation . . . . .	259
8.196.2.1 GSMRSSIThreshListLen . . . . .	259
8.196.2.2 pGSMRSSIThreshList . . . . .	259
8.197GSMSrvStatusInfo Struct Reference . . . . .	259
8.197.1 Detailed Description . . . . .	259
8.197.2 Field Documentation . . . . .	260
8.197.2.1 isPrefDataPath . . . . .	260
8.197.2.2 srvStatus . . . . .	260
8.197.2.3 trueSrvStatus . . . . .	260
8.198GSMSysInfo Struct Reference . . . . .	260
8.198.1 Detailed Description . . . . .	260
8.198.2 Field Documentation . . . . .	263
8.198.2.1 cellld . . . . .	263
8.198.2.2 cellldValid . . . . .	263
8.198.2.3 dtmSupp . . . . .	263
8.198.2.4 dtmSuppValid . . . . .	263
8.198.2.5 egprsSupp . . . . .	263
8.198.2.6 egprsSuppValid . . . . .	263
8.198.2.7 lac . . . . .	263
8.198.2.8 lacValid . . . . .	263
8.198.2.9 MCC . . . . .	263
8.198.2.10MNC . . . . .	263
8.198.2.11networkldValid . . . . .	263
8.198.2.12regRejectInfoValid . . . . .	263
8.198.2.13rejCause . . . . .	263
8.198.2.14rejectSrvDomain . . . . .	263
8.198.2.15sysInfoGSM . . . . .	263
8.199gyroAcceptReady_s Struct Reference . . . . .	263
8.199.1 Detailed Description . . . . .	263
8.199.2 Field Documentation . . . . .	264
8.199.2.1 batchPerSec . . . . .	264
8.199.2.2 injectEnable . . . . .	264

8.199.2.3 samplesPerBatch . . . . .	264
8.200gyroTempAcceptReady_s Struct Reference . . . . .	264
8.200.1 Detailed Description . . . . .	264
8.200.2 Field Documentation . . . . .	265
8.200.2.1 batchPerSec . . . . .	265
8.200.2.2 injectEnable . . . . .	265
8.200.2.3 samplesPerBatch . . . . .	265
8.201HDRECIOThresh Struct Reference . . . . .	265
8.201.1 Detailed Description . . . . .	265
8.201.2 Field Documentation . . . . .	266
8.201.2.1 HDRECIOThreshListLen . . . . .	266
8.201.2.2 pHDECIOThreshList . . . . .	266
8.202HDRIOTresh Struct Reference . . . . .	266
8.202.1 Detailed Description . . . . .	266
8.202.2 Field Documentation . . . . .	266
8.202.2.1 HDRIOTreshListLen . . . . .	266
8.202.2.2 pHDRIOThreshList . . . . .	266
8.203HDRPersonalityInd Struct Reference . . . . .	266
8.203.1 Field Documentation . . . . .	266
8.203.1.1 pCurrentPersonality . . . . .	266
8.203.1.2 pPersonalityListLength . . . . .	266
8.203.1.3 pProtocolSubtypeElement . . . . .	266
8.204HDRPersonalityResp Struct Reference . . . . .	267
8.204.1 Detailed Description . . . . .	267
8.204.2 Field Documentation . . . . .	267
8.204.2.1 pCurrentPersonality . . . . .	267
8.204.2.2 pPersonalityListLength . . . . .	267
8.204.2.3 pProtocolSubtypeElement . . . . .	267
8.205HDRProtSubtypResp Struct Reference . . . . .	267
8.205.1 Detailed Description . . . . .	267
8.205.2 Field Documentation . . . . .	268
8.205.2.1 pAppSubType . . . . .	268
8.205.2.2 pCurrentPrsnlty . . . . .	268
8.205.2.3 pPersonalityListLength . . . . .	268
8.205.2.4 pProtoSubTypElmnt . . . . .	268
8.206HRRSSIThresh Struct Reference . . . . .	268
8.206.1 Detailed Description . . . . .	268
8.206.2 Field Documentation . . . . .	268
8.206.2.1 HRRSSIThreshListLen . . . . .	268
8.206.2.2 pHDRRSSIThreshList . . . . .	268

8.207HDRSINRThresh Struct Reference . . . . .	269
8.207.1 Detailed Description . . . . .	269
8.207.2 Field Documentation . . . . .	269
8.207.2.1 HDRSINRThresListLen . . . . .	269
8.207.2.2 pHDRSINRThresList . . . . .	269
8.208HDRSINRThreshold Struct Reference . . . . .	269
8.208.1 Detailed Description . . . . .	269
8.208.2 Field Documentation . . . . .	270
8.208.2.1 HDRSINRThreshListLen . . . . .	270
8.208.2.2 pHDRSINRThreshList . . . . .	270
8.209HDRSSInfo Struct Reference . . . . .	270
8.209.1 Detailed Description . . . . .	270
8.209.2 Field Documentation . . . . .	271
8.209.2.1 ecio . . . . .	271
8.209.2.2 io . . . . .	271
8.209.2.3 rssi . . . . .	271
8.209.2.4 sinr . . . . .	271
8.210HDRSysInfo Struct Reference . . . . .	271
8.210.1 Detailed Description . . . . .	272
8.210.2 Field Documentation . . . . .	274
8.210.2.1 hdrActiveProt . . . . .	274
8.210.2.2 hdrActiveProtValid . . . . .	274
8.210.2.3 hdrPersonality . . . . .	274
8.210.2.4 hdrPersonalityValid . . . . .	274
8.210.2.5 is856SysId . . . . .	274
8.210.2.6 is856SysIdValid . . . . .	274
8.210.2.7 isSysPrIMatch . . . . .	274
8.210.2.8 isSysPrIMatchValid . . . . .	274
8.210.2.9 sysInfoHDR . . . . .	274
8.211homeSIDNID Struct Reference . . . . .	275
8.211.1 Detailed Description . . . . .	275
8.211.2 Field Documentation . . . . .	275
8.211.2.1 numInstances . . . . .	275
8.211.2.2 SidNid . . . . .	275
8.212hotSwapStatus Struct Reference . . . . .	275
8.212.1 Detailed Description . . . . .	275
8.212.2 Field Documentation . . . . .	276
8.212.2.1 hotSwap . . . . .	276
8.212.2.2 hotSwapLength . . . . .	276
8.213ImageElement Struct Reference . . . . .	276



8.213.1 Detailed Description . . . . .	276
8.213.2 Field Documentation . . . . .	276
8.213.2.1 buildId . . . . .	277
8.213.2.2 buildIdLength . . . . .	277
8.213.2.3 imageId . . . . .	277
8.213.2.4 imageType . . . . .	277
8.214ImageIdElement Struct Reference . . . . .	277
8.214.1 Detailed Description . . . . .	277
8.214.2 Field Documentation . . . . .	277
8.214.2.1 buildID . . . . .	277
8.214.2.2 buildIDLength . . . . .	277
8.214.2.3 failureCount . . . . .	277
8.214.2.4 imageID . . . . .	277
8.214.2.5 storageIndex . . . . .	278
8.215ImageIDEntries Struct Reference . . . . .	278
8.215.1 Detailed Description . . . . .	278
8.215.2 Field Documentation . . . . .	278
8.215.2.1 executingImage . . . . .	278
8.215.2.2 imageIDElement . . . . .	278
8.215.2.3 imageIDSize . . . . .	278
8.215.2.4 imageType . . . . .	278
8.215.2.5 maxImages . . . . .	278
8.216ImageList Struct Reference . . . . .	278
8.216.1 Detailed Description . . . . .	279
8.216.2 Field Documentation . . . . .	279
8.216.2.1 imageIDEntries . . . . .	279
8.216.2.2 listSize . . . . .	279
8.217IMSAIndRegisterInfo Struct Reference . . . . .	279
8.217.1 Detailed Description . . . . .	279
8.217.2 Field Documentation . . . . .	280
8.217.2.1 pPdpStatusConfig . . . . .	280
8.217.2.2 pRatHandoverStatusConfig . . . . .	280
8.217.2.3 pRegStatusConfig . . . . .	280
8.217.2.4 pServiceStatusConfig . . . . .	280
8.218imsaPdpStatusInfo Struct Reference . . . . .	280
8.218.1 Detailed Description . . . . .	281
8.218.2 Field Documentation . . . . .	281
8.218.2.1 connetionState . . . . .	281
8.218.2.2 pFailErrorCode . . . . .	281
8.219imsaRatStatusInfo Struct Reference . . . . .	281

8.219.1 Detailed Description . . . . .	281
8.219.2 Field Documentation . . . . .	282
8.219.2.1 pErrorCodeStr . . . . .	282
8.219.2.2 pRATStatus . . . . .	282
8.219.2.3 pSrcRAT . . . . .	282
8.219.2.4 pTgtRAT . . . . .	282
8.220 IMSARegistrationStatus Struct Reference . . . . .	282
8.220.1 Detailed Description . . . . .	282
8.220.2 Field Documentation . . . . .	283
8.220.2.1 plmsRegErrCode . . . . .	283
8.220.2.2 plmsRegStatus . . . . .	283
8.220.2.3 pNewlmsRegStatus . . . . .	283
8.221 imsaRegStatusInfo Struct Reference . . . . .	283
8.221.1 Detailed Description . . . . .	283
8.221.2 Field Documentation . . . . .	284
8.221.2.1 pbIMSRegistered . . . . .	284
8.221.2.2 plmsRegStatus . . . . .	284
8.221.2.3 pRegStatusErrorCode . . . . .	284
8.222 IMSAServiceStatus Struct Reference . . . . .	284
8.222.1 Detailed Description . . . . .	284
8.222.2 Field Documentation . . . . .	286
8.222.2.1 pSmsServiceRat . . . . .	286
8.222.2.2 pSmsServiceStatus . . . . .	286
8.222.2.3 pUtServiceRat . . . . .	286
8.222.2.4 pUtServiceStatus . . . . .	286
8.222.2.5 pVoipServiceRat . . . . .	286
8.222.2.6 pVoipServiceStatus . . . . .	286
8.222.2.7 pVsServiceRat . . . . .	286
8.222.2.8 pVsServiceStatus . . . . .	286
8.222.2.9 pVtServiceRat . . . . .	287
8.222.2.10 pVtServiceStatus . . . . .	287
8.223 IMSASupportedFieldsResp Struct Reference . . . . .	287
8.223.1 Detailed Description . . . . .	287
8.223.2 Field Documentation . . . . .	287
8.223.2.1 pIndFieldsList . . . . .	287
8.223.2.2 pReqFieldsList . . . . .	287
8.223.2.3 pRespFieldsList . . . . .	287
8.224 IMSASupportedMsgInfo Struct Reference . . . . .	287
8.224.1 Detailed Description . . . . .	287
8.224.2 Field Documentation . . . . .	288

8.224.2.1 pSupportedMsgList . . . . .	288
8.225imsaSvcStatusInfo Struct Reference . . . . .	288
8.225.1 Detailed Description . . . . .	288
8.225.2 Field Documentation . . . . .	288
8.225.2.1 pSMSSvcRAT . . . . .	288
8.225.2.2 pSMSSvcStatus . . . . .	288
8.225.2.3 pUTSvcRAT . . . . .	288
8.225.2.4 pUTSvcStatus . . . . .	288
8.225.2.5 pVOIPSvcRAT . . . . .	289
8.225.2.6 pVOIPSvcStatus . . . . .	289
8.225.2.7 pVTSvcRAT . . . . .	289
8.225.2.8 pVTSvcStatus . . . . .	289
8.226imsCfgIndRegisterInfo Struct Reference . . . . .	289
8.226.1 Detailed Description . . . . .	289
8.226.2 Field Documentation . . . . .	290
8.226.2.1 pRegMgrConfigEvents . . . . .	290
8.226.2.2 pSIPConfigEvents . . . . .	290
8.226.2.3 pSMSConfigEvents . . . . .	290
8.226.2.4 pUserConfigEvents . . . . .	290
8.226.2.5 pVoIPConfigEvents . . . . .	290
8.227imsRegMgrConfigInfo Struct Reference . . . . .	290
8.227.1 Detailed Description . . . . .	291
8.227.2 Field Documentation . . . . .	292
8.227.2.1 pCSCFPortName . . . . .	292
8.227.2.2 pIMSTestMode . . . . .	292
8.227.2.3 pPriCSCFPort . . . . .	292
8.228imsSIPConfigInfo Struct Reference . . . . .	292
8.228.1 Detailed Description . . . . .	292
8.228.2 Field Documentation . . . . .	293
8.228.2.1 pSigCompEnabled . . . . .	293
8.228.2.2 pSIPLocalPort . . . . .	293
8.228.2.3 pSubscribeTimer . . . . .	293
8.228.2.4 pTimerSIPReg . . . . .	293
8.228.2.5 pTimerT1 . . . . .	293
8.228.2.6 pTimerT2 . . . . .	293
8.228.2.7 pTimerTf . . . . .	293
8.229imsSMSConfigInfo Struct Reference . . . . .	293
8.229.1 Detailed Description . . . . .	293
8.229.2 Field Documentation . . . . .	294
8.229.2.1 pPhoneCtxURI . . . . .	294

8.229.2.2 pSMSFormat . . . . .	294
8.229.2.3 pSMSOverIPNwInd . . . . .	294
8.230imsUserConfigInfo Struct Reference . . . . .	294
8.230.1 Detailed Description . . . . .	294
8.230.2 Field Documentation . . . . .	294
8.230.2.1 pIMSDomain . . . . .	294
8.231imsVoIPConfigInfo Struct Reference . . . . .	294
8.231.1 Detailed Description . . . . .	295
8.231.2 Field Documentation . . . . .	297
8.231.2.1 pAmrMode . . . . .	297
8.231.2.2 pAmrOctetAligned . . . . .	297
8.231.2.3 pAmrWbEnable . . . . .	297
8.231.2.4 pAmrWBMode . . . . .	297
8.231.2.5 pAmrWBOctetAligned . . . . .	297
8.231.2.6 pMinSessionExpiryTimer . . . . .	297
8.231.2.7 pRingBackTimer . . . . .	297
8.231.2.8 pRingingTimer . . . . .	297
8.231.2.9 pRTPRTCPInactTimer . . . . .	297
8.231.2.10pScrAmrEnable . . . . .	297
8.231.2.11pScrAmrWbEnable . . . . .	297
8.231.2.12pSessionExpiryTimer . . . . .	297
8.232IndFieldsList Struct Reference . . . . .	297
8.232.1 Detailed Description . . . . .	297
8.232.2 Field Documentation . . . . .	298
8.232.2.1 indicationFields . . . . .	298
8.232.2.2 indicationFieldsLen . . . . .	298
8.233infoInterFreq Struct Reference . . . . .	298
8.233.1 Detailed Description . . . . .	298
8.233.2 Field Documentation . . . . .	299
8.233.2.1 cell_resel_priority . . . . .	299
8.233.2.2 cellInterFreqParams . . . . .	299
8.233.2.3 cells_len . . . . .	299
8.233.2.4 earfcn . . . . .	299
8.233.2.5 threshXHigh . . . . .	299
8.233.2.6 threshXLow . . . . .	299
8.234IOThresh Struct Reference . . . . .	299
8.234.1 Detailed Description . . . . .	299
8.234.2 Field Documentation . . . . .	300
8.234.2.1 IOThresListLen . . . . .	300
8.234.2.2 pIOThresList . . . . .	300

8.235IPv4Addr Struct Reference . . . . .	300
8.235.1 Detailed Description . . . . .	300
8.235.2 Field Documentation . . . . .	300
8.235.2.1 addr . . . . .	300
8.235.2.2 subnetMask . . . . .	300
8.236IPv6Addr Struct Reference . . . . .	300
8.236.1 Detailed Description . . . . .	301
8.236.2 Field Documentation . . . . .	302
8.236.2.1 addr . . . . .	302
8.236.2.2 prefixLen . . . . .	302
8.237IPv6AddressInfo Struct Reference . . . . .	302
8.237.1 Detailed Description . . . . .	302
8.237.2 Field Documentation . . . . .	302
8.237.2.1 IPAddressV6 . . . . .	302
8.237.2.2 IPV6PrefixLen . . . . .	302
8.238IPv6GWAddressInfo Struct Reference . . . . .	302
8.238.1 Detailed Description . . . . .	302
8.238.2 Field Documentation . . . . .	303
8.238.2.1 gwAddressV6 . . . . .	303
8.238.2.2 gwV6PrefixLen . . . . .	303
8.239IPv6TrafCls Struct Reference . . . . .	303
8.239.1 Detailed Description . . . . .	303
8.239.2 Field Documentation . . . . .	303
8.239.2.1 mask . . . . .	303
8.239.2.2 val . . . . .	303
8.240lineCtrlInfo Struct Reference . . . . .	303
8.240.1 Detailed Description . . . . .	304
8.240.2 Field Documentation . . . . .	304
8.240.2.1 polarityIncluded . . . . .	304
8.240.2.2 pwrDenialTime . . . . .	304
8.240.2.3 revPolarity . . . . .	304
8.240.2.4 toggleMode . . . . .	304
8.241LocApplicationInfo Struct Reference . . . . .	304
8.241.1 Detailed Description . . . . .	304
8.241.2 Field Documentation . . . . .	305
8.241.2.1 appNameLength . . . . .	305
8.241.2.2 appProviderLength . . . . .	305
8.241.2.3 appVersionLength . . . . .	305
8.241.2.4 appVersionValid . . . . .	305
8.241.2.5 pAppName . . . . .	305

8.241.2.6 pAppProvider . . . . .	305
8.241.2.7 pAppVersion . . . . .	305
8.242LocDelAssDataReq Struct Reference . . . . .	305
8.242.1 Detailed Description . . . . .	306
8.242.2 Field Documentation . . . . .	306
8.242.2.1 pBdsSVInfo . . . . .	306
8.242.2.2 pCellDb . . . . .	306
8.242.2.3 pClkInfo . . . . .	306
8.242.2.4 pGnssData . . . . .	306
8.242.2.5 pSVInfo . . . . .	306
8.243LOCEventRegisterReqResp Struct Reference . . . . .	306
8.243.1 Detailed Description . . . . .	306
8.243.2 Field Documentation . . . . .	308
8.243.2.1 eventRegister . . . . .	308
8.244LOCExtPowerStateReqResp Struct Reference . . . . .	308
8.244.1 Detailed Description . . . . .	308
8.244.2 Field Documentation . . . . .	308
8.244.2.1 extPowerState . . . . .	308
8.245LOCStartReq Struct Reference . . . . .	308
8.245.1 Detailed Description . . . . .	308
8.245.2 Field Documentation . . . . .	310
8.245.2.1 pApplicationInfo . . . . .	310
8.245.2.2 pConfigAltitudeAssumed . . . . .	310
8.245.2.3 pHorizontalAccuracyLvl . . . . .	310
8.245.2.4 pIntermediateReportState . . . . .	310
8.245.2.5 pMinIntervalTime . . . . .	310
8.245.2.6 pRecurrenceType . . . . .	310
8.245.2.7 SessionId . . . . .	310
8.246LOCStopReq Struct Reference . . . . .	310
8.246.1 Detailed Description . . . . .	310
8.246.2 Field Documentation . . . . .	310
8.246.2.1 sessionId . . . . .	310
8.247LteCQIParm Struct Reference . . . . .	310
8.247.1 Detailed Description . . . . .	310
8.247.2 Field Documentation . . . . .	311
8.247.2.1 CQIValueCW0 . . . . .	311
8.247.2.2 CQIValueCW1 . . . . .	311
8.247.2.3 ValidityCW0 . . . . .	311
8.247.2.4 ValidityCW1 . . . . .	311
8.248lteGsmCellInfo Struct Reference . . . . .	311

8.248.1 Detailed Description . . . . .	311
8.248.2 Field Documentation . . . . .	312
8.248.2.1 cellReselPriority . . . . .	312
8.248.2.2 cells_len . . . . .	312
8.248.2.3 GsmCellInfo . . . . .	312
8.248.2.4 nccPermitted . . . . .	312
8.248.2.5 threshGsmHigh . . . . .	312
8.248.2.6 threshGsmLow . . . . .	312
8.249LTEInfo Struct Reference . . . . .	312
8.249.1 Detailed Description . . . . .	313
8.249.2 Field Documentation . . . . .	315
8.249.2.1 band . . . . .	315
8.249.2.2 bandwidth . . . . .	315
8.249.2.3 emmConnState . . . . .	315
8.249.2.4 emmState . . . . .	315
8.249.2.5 emmSubState . . . . .	315
8.249.2.6 RXChan . . . . .	315
8.249.2.7 TXChan . . . . .	315
8.250LTEInfoInterfreq Struct Reference . . . . .	315
8.250.1 Detailed Description . . . . .	315
8.250.2 Field Documentation . . . . .	316
8.250.2.1 freqsLen . . . . .	316
8.250.2.2 InfoInterfreq . . . . .	316
8.250.2.3 ueInIdle . . . . .	316
8.251LTEInfoIntrafreq Struct Reference . . . . .	316
8.251.1 Detailed Description . . . . .	316
8.251.2 Field Documentation . . . . .	318
8.251.2.1 CellParams . . . . .	318
8.251.2.2 cellReselPriority . . . . .	318
8.251.2.3 cellsLen . . . . .	318
8.251.2.4 earfcn . . . . .	318
8.251.2.5 globalCellId . . . . .	318
8.251.2.6 plmn . . . . .	318
8.251.2.7 servingCellId . . . . .	318
8.251.2.8 sIntraSearch . . . . .	318
8.251.2.9 sNonIntraSearch . . . . .	318
8.251.2.10tac . . . . .	318
8.251.2.11threshServingLow . . . . .	318
8.251.2.12ueInIdle . . . . .	318
8.252LTEInfoNeighboringGSM Struct Reference . . . . .	319

8.252.1 Detailed Description . . . . .	319
8.252.2 Field Documentation . . . . .	319
8.252.2.1 freqsLen . . . . .	319
8.252.2.2 LteGsmCellInfo . . . . .	319
8.252.2.3 ueInIdle . . . . .	319
8.253LteInfoNeighboringWCDMA Struct Reference . . . . .	319
8.253.1 Detailed Description . . . . .	319
8.253.2 Field Documentation . . . . .	320
8.253.2.1 freqsLen . . . . .	320
8.253.2.2 LTEWCDMACellInfo . . . . .	320
8.253.2.3 ueInIdle . . . . .	320
8.254LteNasReleaseInfo_s Struct Reference . . . . .	320
8.254.1 Detailed Description . . . . .	320
8.254.2 Field Documentation . . . . .	320
8.254.2.1 nas_major . . . . .	320
8.254.2.2 nas_minor . . . . .	320
8.254.2.3 nas_release . . . . .	321
8.255LteRsrpInformation Struct Reference . . . . .	321
8.255.1 Detailed Description . . . . .	321
8.255.2 Field Documentation . . . . .	321
8.255.2.1 rsrplevel . . . . .	321
8.256LTERSRPThresh Struct Reference . . . . .	321
8.256.1 Detailed Description . . . . .	321
8.256.2 Field Documentation . . . . .	321
8.256.2.1 LTERSRPThreshListLen . . . . .	321
8.256.2.2 pLTERSRPThreshList . . . . .	321
8.257LTERSRQThresh Struct Reference . . . . .	322
8.257.1 Detailed Description . . . . .	322
8.257.2 Field Documentation . . . . .	322
8.257.2.1 LTERSRQThreshListLen . . . . .	322
8.257.2.2 pLTERSRQThreshList . . . . .	322
8.258LTERSSIThresh Struct Reference . . . . .	322
8.258.1 Detailed Description . . . . .	322
8.258.2 Field Documentation . . . . .	323
8.258.2.1 LTERSSIThreshListLen . . . . .	323
8.258.2.2 pLTERSSIThreshList . . . . .	323
8.259LTESigRptCfg Struct Reference . . . . .	323
8.259.1 Detailed Description . . . . .	323
8.259.2 Field Documentation . . . . .	323
8.259.2.1 avgPeriod . . . . .	324



8.259.2.2 rptRate . . . . .	324
8.260LTESigRptConfig Struct Reference . . . . .	324
8.260.1 Detailed Description . . . . .	324
8.260.2 Field Documentation . . . . .	324
8.260.2.1 avgPeriod . . . . .	324
8.260.2.2 rptRate . . . . .	325
8.261lteSnrinformation Struct Reference . . . . .	325
8.261.1 Detailed Description . . . . .	325
8.261.2 Field Documentation . . . . .	325
8.261.2.1 snrlevel . . . . .	325
8.262LTESNRThresh Struct Reference . . . . .	325
8.262.1 Detailed Description . . . . .	325
8.262.2 Field Documentation . . . . .	326
8.262.2.1 LTESNRThresListLen . . . . .	326
8.262.2.2 pLTESNRThresList . . . . .	326
8.263LTESNRThreshold Struct Reference . . . . .	326
8.263.1 Detailed Description . . . . .	326
8.263.2 Field Documentation . . . . .	326
8.263.2.1 LTESNRThreshListLen . . . . .	326
8.263.2.2 pLTESNRThreshList . . . . .	326
8.264LTESInfo Struct Reference . . . . .	326
8.264.1 Detailed Description . . . . .	326
8.264.2 Field Documentation . . . . .	327
8.264.2.1 rsrp . . . . .	327
8.264.2.2 rsrq . . . . .	327
8.264.2.3 rssi . . . . .	327
8.264.2.4 snr . . . . .	327
8.265LTESysInfo Struct Reference . . . . .	327
8.265.1 Detailed Description . . . . .	328
8.265.2 Field Documentation . . . . .	330
8.265.2.1 cellId . . . . .	330
8.265.2.2 cellIdValid . . . . .	330
8.265.2.3 lac . . . . .	330
8.265.2.4 lacValid . . . . .	330
8.265.2.5 MCC . . . . .	330
8.265.2.6 MNC . . . . .	330
8.265.2.7 networkIdValid . . . . .	330
8.265.2.8 regRejectInfoValid . . . . .	330
8.265.2.9 rejCause . . . . .	330
8.265.2.10rejectSrvDomain . . . . .	331

8.265.2.1 sysInfoLTE . . . . .	331
8.265.2.2 tac . . . . .	331
8.265.2.3 tacValid . . . . .	331
8.266 lteWcdmaCellInfo Struct Reference . . . . .	331
8.266.1 Detailed Description . . . . .	331
8.266.2 Field Documentation . . . . .	332
8.266.2.1 cellReselPriority . . . . .	332
8.266.2.2 cellsLen . . . . .	332
8.266.2.3 threshXhigh . . . . .	332
8.266.2.4 threshXlow . . . . .	332
8.266.2.5 uarfcn . . . . .	332
8.266.2.6 WCDMACellInfo . . . . .	332
8.267 messageWaitingInfoContent Struct Reference . . . . .	332
8.267.1 Detailed Description . . . . .	332
8.267.2 Field Documentation . . . . .	332
8.267.2.1 activeInd . . . . .	332
8.267.2.2 msgCount . . . . .	333
8.267.2.3 msgType . . . . .	333
8.268 minBasedIMSI Struct Reference . . . . .	333
8.268.1 Detailed Description . . . . .	333
8.268.2 Field Documentation . . . . .	333
8.268.2.1 imsiM1112 . . . . .	333
8.268.2.2 imsiMS1 . . . . .	333
8.268.2.3 imsiMS2 . . . . .	333
8.268.2.4 mccM . . . . .	333
8.269 MNRInfo Struct Reference . . . . .	333
8.269.1 Detailed Description . . . . .	334
8.269.2 Field Documentation . . . . .	335
8.269.2.1 mcc . . . . .	335
8.269.2.2 mnc . . . . .	335
8.269.2.3 rat . . . . .	335
8.270 ModifyProfileIn Struct Reference . . . . .	335
8.270.1 Detailed Description . . . . .	335
8.270.2 Field Documentation . . . . .	336
8.270.2.1 curProfile . . . . .	336
8.270.2.2 pProfileID . . . . .	336
8.270.2.3 pProfileType . . . . .	336
8.271 ModifyProfileOut Struct Reference . . . . .	336
8.271.1 Detailed Description . . . . .	336
8.271.2 Field Documentation . . . . .	336

8.271.2.1 pExtErrorCode . . . . .	336
8.272msgWaitingInfo Struct Reference . . . . .	336
8.272.1 Detailed Description . . . . .	336
8.272.2 Field Documentation . . . . .	337
8.272.2.1 msgWaitInfo . . . . .	337
8.272.2.2 numInstances . . . . .	337
8.273namName Struct Reference . . . . .	337
8.273.1 Detailed Description . . . . .	337
8.273.2 Field Documentation . . . . .	337
8.273.2.1 namName . . . . .	337
8.273.2.2 namNameLen . . . . .	337
8.274nasCellLocationInfoResp Struct Reference . . . . .	337
8.274.1 Detailed Description . . . . .	338
8.274.2 Field Documentation . . . . .	338
8.274.2.1 pCDMAInfo . . . . .	338
8.274.2.2 pGERANInfo . . . . .	338
8.274.2.3 pLTEInfoInterfreq . . . . .	338
8.274.2.4 pLTEInfoIntrafreq . . . . .	338
8.274.2.5 pLTEInfoNeighboringGSM . . . . .	339
8.274.2.6 pLTEInfoNeighboringWCDMA . . . . .	339
8.274.2.7 pUMTSCellID . . . . .	339
8.274.2.8 pUMTSInfo . . . . .	339
8.274.2.9 pWCDMAInfoLTENeighborCell . . . . .	339
8.275nasGet3GPP2SubscriptionInfoReq Struct Reference . . . . .	339
8.275.1 Detailed Description . . . . .	339
8.275.2 Field Documentation . . . . .	339
8.275.2.1 namID . . . . .	339
8.276nasGet3GPP2SubscriptionInfoResp Struct Reference . . . . .	339
8.276.1 Detailed Description . . . . .	339
8.276.2 Field Documentation . . . . .	340
8.276.2.1 pCDMAChannel . . . . .	340
8.276.2.2 pDirNum . . . . .	340
8.276.2.3 pHomeSIDNID . . . . .	340
8.276.2.4 pMinBasedIMSI . . . . .	340
8.276.2.5 pNAMNameInfo . . . . .	340
8.276.2.6 pTrueIMSI . . . . .	340
8.277nasGetHDRColorCodeResp Struct Reference . . . . .	340
8.277.1 Detailed Description . . . . .	340
8.277.2 Field Documentation . . . . .	341
8.277.2.1 pColorCode . . . . .	341

8.278nasGetLTECphyCa Struct Reference . . . . .	341
8.278.1 Field Documentation . . . . .	341
8.278.1.1 sPhyCaAggPcellInfo . . . . .	341
8.278.1.2 sPhyCaAggScellIDBw . . . . .	341
8.278.1.3 sPhyCaAggScellIndex . . . . .	341
8.278.1.4 sPhyCaAggScellIndType . . . . .	341
8.278.1.5 sPhyCaAggScellInfo . . . . .	341
8.279nasGetLTECphyCaResp Struct Reference . . . . .	341
8.279.1 Field Documentation . . . . .	341
8.279.1.1 pPhyCaAggPcellInfo . . . . .	341
8.279.1.2 pPhyCaAggScellIDBw . . . . .	341
8.279.1.3 pPhyCaAggScellIndex . . . . .	342
8.279.1.4 pPhyCaAggScellIndType . . . . .	342
8.279.1.5 pPhyCaAggScellInfo . . . . .	342
8.280nasGetSigInfoResp Struct Reference . . . . .	342
8.280.1 Detailed Description . . . . .	342
8.280.2 Field Documentation . . . . .	342
8.280.2.1 pCDMASSInfo . . . . .	342
8.280.2.2 pGSMSSInfo . . . . .	342
8.280.2.3 pHDRSSInfo . . . . .	342
8.280.2.4 pLTESInfo . . . . .	342
8.280.2.5 pWCDMASSInfo . . . . .	342
8.281nasGetSysInfoResp Struct Reference . . . . .	343
8.281.1 Detailed Description . . . . .	343
8.281.2 Field Documentation . . . . .	345
8.281.2.1 pAddCDMASysInfo . . . . .	345
8.281.2.2 pAddGSMSysInfo . . . . .	345
8.281.2.3 pAddHDRSysInfo . . . . .	345
8.281.2.4 pAddLTESysInfo . . . . .	345
8.281.2.5 pAddWCDMASysInfo . . . . .	345
8.281.2.6 pCDMASrvStatusInfo . . . . .	345
8.281.2.7 pCDMASysInfo . . . . .	345
8.281.2.8 pGSMCalBarringSysInfo . . . . .	345
8.281.2.9 pGSMCipherDomainSysInfo . . . . .	345
8.281.2.10pGSMSSrvStatusInfo . . . . .	345
8.281.2.11pGSMSSysInfo . . . . .	345
8.281.2.12pHDRSrvStatusInfo . . . . .	345
8.281.2.13pHDRSysInfo . . . . .	345
8.281.2.14pLTESrvStatusInfo . . . . .	345
8.281.2.15pLTESysInfo . . . . .	345

8.281.2.10pLTEVoiceSupportSysInfo . . . . .	345
8.281.2.17pWCDMACallBarringSysInfo . . . . .	345
8.281.2.18pWCDMACipherDomainSysInfo . . . . .	345
8.281.2.19pWCDMASrvStatusInfo . . . . .	345
8.281.2.20pWCDMASysInfo . . . . .	345
8.282nasGetTxRxInfoReq Struct Reference . . . . .	345
8.282.1 Detailed Description . . . . .	346
8.282.2 Field Documentation . . . . .	346
8.282.2.1 radio_if . . . . .	346
8.283nasGetTxRxInfoResp Struct Reference . . . . .	346
8.283.1 Detailed Description . . . . .	346
8.283.2 Field Documentation . . . . .	346
8.283.2.1 pRXChain0Info . . . . .	346
8.283.2.2 pRXChain1Info . . . . .	347
8.283.2.3 pTXInfo . . . . .	347
8.284nasIndicationRegisterReq Struct Reference . . . . .	347
8.284.1 Detailed Description . . . . .	347
8.284.2 Field Documentation . . . . .	350
8.284.2.1 pDDTMInd . . . . .	350
8.284.2.2 pDualStandByPrefInd . . . . .	350
8.284.2.3 pErrorRateInd . . . . .	350
8.284.2.4 pHDRNewUATIAssInd . . . . .	350
8.284.2.5 pHDRSessionCloseInd . . . . .	350
8.284.2.6 pLTECphyCa . . . . .	350
8.284.2.7 pManagedRoamingInd . . . . .	350
8.284.2.8 pNetworkTimeInd . . . . .	350
8.284.2.9 pServingSystemInd . . . . .	350
8.284.2.10pSignalStrengthInd . . . . .	350
8.284.2.11pSubscriptionInfoInd . . . . .	350
8.284.2.12pSysInfoInd . . . . .	350
8.284.2.13pSystemSelectionInd . . . . .	350
8.285nasInitNetworkReg Struct Reference . . . . .	350
8.285.1 Detailed Description . . . . .	350
8.285.2 Field Documentation . . . . .	351
8.285.2.1 pChangeDuration . . . . .	351
8.285.2.2 pMncPcsDigitStatus . . . . .	351
8.285.2.3 pMNRInfo . . . . .	351
8.285.2.4 regAction . . . . .	351
8.286nasNetworkTime Struct Reference . . . . .	351
8.286.1 Detailed Description . . . . .	351

8.286.2 Field Documentation	352
8.286.2.1 pDayltSavAdj	352
8.286.2.2 pTimeZone	352
8.286.2.3 universalTime	352
8.287nasOperatorNameResp Struct Reference	352
8.287.1 Detailed Description	352
8.287.2 Field Documentation	353
8.287.2.1 pNITZInformation	353
8.287.2.2 pOperatorNameString	353
8.287.2.3 pOperatorPLMNList	353
8.287.2.4 pPLMNNetworkName	353
8.287.2.5 pSvcProviderName	353
8.288nasPLMNNameReq Struct Reference	353
8.288.1 Detailed Description	353
8.288.2 Field Documentation	354
8.288.2.1 mcc	354
8.288.2.2 mnc	354
8.289nasPLMNNameResp Struct Reference	354
8.289.1 Detailed Description	354
8.289.2 Field Documentation	356
8.289.2.1 longName	356
8.289.2.2 longNameCI	356
8.289.2.3 longNameEn	356
8.289.2.4 longNameLen	356
8.289.2.5 longNameSB	356
8.289.2.6 shortName	356
8.289.2.7 shortNameCI	356
8.289.2.8 shortNameEn	356
8.289.2.9 shortNameLen	356
8.289.2.10shortNameSB	357
8.289.2.11spn	357
8.289.2.12spnEncoding	357
8.289.2.13spnLength	357
8.290nasSigInfo Struct Reference	357
8.290.1 Detailed Description	357
8.290.2 Field Documentation	357
8.290.2.1 pCDMASigInfo	358
8.290.2.2 pGSMSigInfo	358
8.290.2.3 pHDRSigInfo	358
8.290.2.4 pLTESigInfo	358

8.290.2.5 pRscp . . . . .	358
8.290.2.6 pTDSCDMASigInfoExt . . . . .	358
8.290.2.7 pWCDMASigInfo . . . . .	358
8.291 NasSwIndReg Struct Reference . . . . .	358
8.291.1 Detailed Description . . . . .	358
8.291.2 Field Documentation . . . . .	359
8.291.2.1 gsmUmtsDI . . . . .	359
8.291.2.2 gsmUmtsUI . . . . .	359
8.291.2.3 lteEmmDI . . . . .	359
8.291.2.4 lteEmmUI . . . . .	359
8.291.2.5 lteEsmDI . . . . .	359
8.291.2.6 lteEsmUI . . . . .	359
8.292 nasSysInfo Struct Reference . . . . .	359
8.292.1 Detailed Description . . . . .	359
8.292.2 Field Documentation . . . . .	361
8.292.2.1 pAddCDMASysInfo . . . . .	361
8.292.2.2 pAddGSMSysInfo . . . . .	361
8.292.2.3 pAddHDRSysInfo . . . . .	361
8.292.2.4 pAddLTESysInfo . . . . .	361
8.292.2.5 pAddWCDMASysInfo . . . . .	361
8.292.2.6 pCDMASrvStatusInfo . . . . .	361
8.292.2.7 pCDMASysInfo . . . . .	362
8.292.2.8 pGSMCallBarringSysInfo . . . . .	362
8.292.2.9 pGSMCipherDomainSysInfo . . . . .	362
8.292.2.10 pGSMSrvStatusInfo . . . . .	362
8.292.2.11 pGSMSysInfo . . . . .	362
8.292.2.12 pHDRSrvStatusInfo . . . . .	362
8.292.2.13 pHDRSysInfo . . . . .	362
8.292.2.14 pLTESrvStatusInfo . . . . .	362
8.292.2.15 pLTESysInfo . . . . .	362
8.292.2.16 pLTEVoiceSupportSysInfo . . . . .	362
8.292.2.17 pSysInfoNoChange . . . . .	362
8.292.2.18 pWCDMACallBarringSysInfo . . . . .	362
8.292.2.19 pWCDMACipherDomainSysInfo . . . . .	362
8.292.2.20 pWCDMASrvStatusInfo . . . . .	362
8.292.2.21 pWCDMASysInfo . . . . .	362
8.293 netSelectionPref Struct Reference . . . . .	362
8.293.1 Detailed Description . . . . .	362
8.293.2 Field Documentation . . . . .	363
8.293.2.1 mcc . . . . .	363

8.293.2.2 mnc . . . . .	363
8.293.2.3 netReg . . . . .	363
8.294NetStats Struct Reference . . . . .	363
8.294.1 Detailed Description . . . . .	363
8.294.2 Field Documentation . . . . .	364
8.294.2.1 rx_bytes . . . . .	364
8.294.2.2 rx_errors . . . . .	364
8.294.2.3 rx_overflows . . . . .	364
8.294.2.4 rx_packets . . . . .	364
8.294.2.5 tx_bytes . . . . .	364
8.294.2.6 tx_errors . . . . .	364
8.294.2.7 tx_overflows . . . . .	364
8.294.2.8 tx_packets . . . . .	364
8.295NetworkDebugResp Struct Reference . . . . .	364
8.295.1 Detailed Description . . . . .	365
8.295.2 Field Documentation . . . . .	366
8.295.2.1 pDataStatusDetail . . . . .	366
8.295.2.2 pDeviceConfigDetail . . . . .	366
8.295.2.3 pNetworkStat1x . . . . .	366
8.295.2.4 pNetworkStatEVDO . . . . .	366
8.295.2.5 pObjectVer . . . . .	366
8.296NetworkStat1x Struct Reference . . . . .	366
8.296.1 Detailed Description . . . . .	366
8.296.2 Field Documentation . . . . .	369
8.296.2.1 ActSetCnt . . . . .	369
8.296.2.2 NeighborSetCnt . . . . .	369
8.296.2.3 pActPilotPNElements . . . . .	369
8.296.2.4 pNeighborSetPilotPN . . . . .	369
8.296.2.5 RX_EC_IO . . . . .	369
8.296.2.6 RX_PWR . . . . .	369
8.296.2.7 SO . . . . .	369
8.296.2.8 State . . . . .	369
8.296.2.9 TX_PWR . . . . .	369
8.297NetworkStatEVDO Struct Reference . . . . .	369
8.297.1 Detailed Description . . . . .	369
8.297.2 Field Documentation . . . . .	371
8.297.2.1 MACIndex . . . . .	371
8.297.2.2 PER . . . . .	371
8.297.2.3 PilotEnergy . . . . .	371
8.297.2.4 pSectorID . . . . .	371



8.297.2.5 RX_PWR . . . . .	371
8.297.2.6 SectorIDLen . . . . .	371
8.297.2.7 SNR . . . . .	371
8.297.2.8 State . . . . .	371
8.298newPwdData Struct Reference . . . . .	371
8.298.1 Detailed Description . . . . .	371
8.298.2 Field Documentation . . . . .	372
8.298.2.1 newPwd . . . . .	372
8.298.2.2 newPwdAgain . . . . .	372
8.299nmrCellInfo Struct Reference . . . . .	372
8.299.1 Detailed Description . . . . .	372
8.299.2 Field Documentation . . . . .	373
8.299.2.1 nmrArfcn . . . . .	373
8.299.2.2 nmrBsic . . . . .	373
8.299.2.3 nmrCellID . . . . .	373
8.299.2.4 nmrLac . . . . .	373
8.299.2.5 nmrPlmn . . . . .	373
8.299.2.6 nmrRxLev . . . . .	373
8.300NSSAudioCtrl Struct Reference . . . . .	373
8.300.1 Detailed Description . . . . .	374
8.300.2 Field Documentation . . . . .	374
8.300.2.1 downLink . . . . .	374
8.300.2.2 upLink . . . . .	374
8.301NWProfile Struct Reference . . . . .	374
8.301.1 Detailed Description . . . . .	374
8.301.2 Field Documentation . . . . .	374
8.301.2.1 pProfSz . . . . .	374
8.301.2.2 pProfValues . . . . .	374
8.301.2.3 tech . . . . .	374
8.302omaDmConfigTlv Struct Reference . . . . .	374
8.302.1 Detailed Description . . . . .	375
8.302.2 Field Documentation . . . . .	375
8.302.2.1 alertmsg . . . . .	375
8.302.2.2 alertmsglength . . . . .	375
8.302.2.3 state . . . . .	375
8.302.2.4 userInputReq . . . . .	375
8.302.2.5 userInputTimeout . . . . .	375
8.303omaDmConfigTlvExt Struct Reference . . . . .	375
8.303.1 Detailed Description . . . . .	376
8.303.2 Field Documentation . . . . .	378

8.303.2.1 alertmsg . . . . .	378
8.303.2.2 alertmsglength . . . . .	378
8.303.2.3 state . . . . .	378
8.303.2.4 userInputReq . . . . .	378
8.303.2.5 userInputTimeout . . . . .	378
8.304omaDmFotaTlv Struct Reference . . . . .	378
8.304.1 Detailed Description . . . . .	378
8.304.2 Field Documentation . . . . .	380
8.304.2.1 description . . . . .	380
8.304.2.2 descriptionlength . . . . .	380
8.304.2.3 fwdloadsize . . . . .	380
8.304.2.4 fwloadComplete . . . . .	380
8.304.2.5 namelength . . . . .	380
8.304.2.6 package_name . . . . .	380
8.304.2.7 sessionType . . . . .	380
8.304.2.8 severity . . . . .	380
8.304.2.9 state . . . . .	380
8.304.2.10updateCompleteStatus . . . . .	380
8.304.2.11userInputReq . . . . .	380
8.304.2.12userInputTimeout . . . . .	380
8.304.2.13version . . . . .	380
8.304.2.14versionlength . . . . .	380
8.305omaDmFotaTlvExt Struct Reference . . . . .	380
8.305.1 Detailed Description . . . . .	381
8.305.2 Field Documentation . . . . .	382
8.305.2.1 description . . . . .	382
8.305.2.2 descriptionlength . . . . .	382
8.305.2.3 fumoResultCode . . . . .	382
8.305.2.4 namelength . . . . .	382
8.305.2.5 package_name . . . . .	382
8.305.2.6 packageSize . . . . .	382
8.305.2.7 receivedBytes . . . . .	382
8.305.2.8 reserved . . . . .	383
8.305.2.9 state . . . . .	383
8.305.2.10userInputTimeout . . . . .	383
8.305.2.11version . . . . .	383
8.305.2.12versionlength . . . . .	383
8.306omaDmNotificationsTlv Struct Reference . . . . .	383
8.306.1 Field Documentation . . . . .	383
8.306.1.1 notification . . . . .	383

8.306.1.2 sessionStatus . . . . .	383
8.307operatorNameString Struct Reference . . . . .	383
8.307.1 Detailed Description . . . . .	383
8.307.2 Field Documentation . . . . .	383
8.307.2.1 PLMNName . . . . .	383
8.308OperatorPLMNData Struct Reference . . . . .	383
8.308.1 Detailed Description . . . . .	384
8.308.2 Field Documentation . . . . .	384
8.308.2.1 lac1 . . . . .	384
8.308.2.2 lac2 . . . . .	384
8.308.2.3 mcc . . . . .	384
8.308.2.4 mnc . . . . .	384
8.308.2.5 PLMNRecID . . . . .	384
8.309operatorPLMNList Struct Reference . . . . .	384
8.309.1 Detailed Description . . . . .	384
8.309.2 Field Documentation . . . . .	385
8.309.2.1 numInstance . . . . .	385
8.309.2.2 PLMNData . . . . .	385
8.310PCMparams Struct Reference . . . . .	385
8.310.1 Detailed Description . . . . .	385
8.310.2 Field Documentation . . . . .	385
8.310.2.1 iFaceTab . . . . .	385
8.310.2.2 iFaceTabLen . . . . .	385
8.311PCSCFFQDNAddress Struct Reference . . . . .	385
8.311.1 Detailed Description . . . . .	386
8.311.2 Field Documentation . . . . .	387
8.311.2.1 fqdnAddr . . . . .	387
8.311.2.2 fqdnLen . . . . .	387
8.312PCSCFFQDNAddressList Struct Reference . . . . .	387
8.312.1 Detailed Description . . . . .	387
8.312.2 Field Documentation . . . . .	387
8.312.2.1 numInstances . . . . .	387
8.312.2.2 pcsfQDNAddress . . . . .	387
8.313PCSCFIPv4ServerAddressList Struct Reference . . . . .	387
8.313.1 Detailed Description . . . . .	387
8.313.2 Field Documentation . . . . .	388
8.313.2.1 numInstances . . . . .	388
8.313.2.2 pcsfIPv4Addr . . . . .	388
8.314PDSPositionData Struct Reference . . . . .	388
8.314.1 Detailed Description . . . . .	388

8.314.2 Field Documentation	390
8.314.2.1 pAltitudeWrtEllipsoid	390
8.314.2.2 pAltitudeWrtSealevel	390
8.314.2.3 pHorizontalConfidence	390
8.314.2.4 pHorizontalUncCircular	390
8.314.2.5 pLatitude	390
8.314.2.6 pLongitude	390
8.314.2.7 pPositionSource	390
8.314.2.8 pTimeStamp	390
8.314.2.9 pTimeType	390
8.314.2.10 pVerticalConfidence	390
8.314.2.11 pVerticalUnc	390
8.315 PDSPosMethodStateReq Struct Reference	390
8.315.1 Detailed Description	390
8.315.2 Field Documentation	391
8.315.2.1 pWifiState	391
8.315.2.2 pXtraDataState	391
8.315.2.3 pXtraTimeState	391
8.316 peerNumberInfo Struct Reference	391
8.316.1 Detailed Description	391
8.316.2 Field Documentation	393
8.316.2.1 callID	393
8.316.2.2 number	393
8.316.2.3 numLen	393
8.316.2.4 numPI	393
8.316.2.5 numPlan	393
8.316.2.6 numSI	393
8.316.2.7 numType	393
8.317 PhyCaAggPcellInfo Struct Reference	393
8.317.1 Detailed Description	393
8.317.2 Field Documentation	394
8.317.2.1 dl_bw_value	394
8.317.2.2 freq	394
8.317.2.3 iLTEbandValue	394
8.317.2.4 pci	394
8.317.2.5 TlvPresent	394
8.318 PhyCaAggScellIDIBw Struct Reference	394
8.318.1 Detailed Description	394
8.318.2 Field Documentation	395
8.318.2.1 dl_bw_value	395

8.318.2.2 TlvPresent . . . . .	395
8.319PhyCaAggScellIndex Struct Reference . . . . .	395
8.319.1 Detailed Description . . . . .	395
8.319.2 Field Documentation . . . . .	395
8.319.2.1 scell_idx . . . . .	395
8.319.2.2 TlvPresent . . . . .	395
8.320PhyCaAggScellIndType Struct Reference . . . . .	395
8.320.1 Detailed Description . . . . .	395
8.320.2 Field Documentation . . . . .	396
8.320.2.1 freq . . . . .	396
8.320.2.2 pci . . . . .	396
8.320.2.3 scell_state . . . . .	396
8.320.2.4 TlvPresent . . . . .	396
8.321PhyCaAggScellInfo Struct Reference . . . . .	396
8.321.1 Detailed Description . . . . .	396
8.321.2 Field Documentation . . . . .	397
8.321.2.1 dl_bw_value . . . . .	397
8.321.2.2 freq . . . . .	397
8.321.2.3 iLTEbandValue . . . . .	397
8.321.2.4 pci . . . . .	397
8.321.2.5 scell_state . . . . .	397
8.321.2.6 TlvPresent . . . . .	397
8.322PilotSetData Struct Reference . . . . .	397
8.322.1 Detailed Description . . . . .	397
8.322.2 Field Documentation . . . . .	398
8.322.2.1 NumPilots . . . . .	398
8.322.2.2 pPilotSetInfo . . . . .	398
8.323PilotSetParams Struct Reference . . . . .	398
8.323.1 Detailed Description . . . . .	398
8.323.2 Field Documentation . . . . .	398
8.323.2.1 PilotPN . . . . .	398
8.323.2.2 PilotStrength . . . . .	399
8.323.2.3 PilotType . . . . .	399
8.324pktErrRate Struct Reference . . . . .	399
8.324.1 Detailed Description . . . . .	399
8.324.2 Field Documentation . . . . .	399
8.324.2.1 exponent . . . . .	399
8.324.2.2 multiplier . . . . .	399
8.325PLMNNetworkName Struct Reference . . . . .	399
8.325.1 Detailed Description . . . . .	399

8.325.2 Field Documentation . . . . .	399
8.325.2.1 numInstance . . . . .	399
8.325.2.2 PLMNNetName . . . . .	399
8.326PLMNNetworkNameData Struct Reference . . . . .	400
8.326.1 Detailed Description . . . . .	400
8.326.2 Field Documentation . . . . .	402
8.326.2.1 codingScheme . . . . .	402
8.326.2.2 countryInitials . . . . .	402
8.326.2.3 longName . . . . .	402
8.326.2.4 longNameLen . . . . .	402
8.326.2.5 longNameSpareBits . . . . .	402
8.326.2.6 shortName . . . . .	402
8.326.2.7 shortNameLen . . . . .	402
8.326.2.8 shortNameSpareBits . . . . .	402
8.327Port Struct Reference . . . . .	402
8.327.1 Detailed Description . . . . .	402
8.327.2 Field Documentation . . . . .	402
8.327.2.1 port . . . . .	402
8.327.2.2 range . . . . .	403
8.328precisionDilution_s Struct Reference . . . . .	403
8.328.1 Detailed Description . . . . .	403
8.328.2 Field Documentation . . . . .	403
8.328.2.1 HDOP . . . . .	403
8.328.2.2 PDOP . . . . .	403
8.328.2.3 VDOP . . . . .	403
8.329PrefImageList Struct Reference . . . . .	403
8.329.1 Detailed Description . . . . .	403
8.329.2 Field Documentation . . . . .	404
8.329.2.1 listEntries . . . . .	404
8.329.2.2 listSize . . . . .	404
8.330prefVoiceSO Struct Reference . . . . .	404
8.330.1 Detailed Description . . . . .	404
8.330.2 Field Documentation . . . . .	406
8.330.2.1 evrcCapability . . . . .	406
8.330.2.2 homeOrigVoiceSO . . . . .	406
8.330.2.3 homePageVoiceSO . . . . .	406
8.330.2.4 namID . . . . .	406
8.330.2.5 roamOrigVoiceSO . . . . .	406
8.331Profile3GPP Struct Reference . . . . .	406
8.331.1 Detailed Description . . . . .	407

8.331.2 Field Documentation	411
8.331.2.1 pAddrAllocPref	411
8.331.2.2 pAPNClass	411
8.331.2.3 pAPNDisabledFlag	411
8.331.2.4 pAPNName	411
8.331.2.5 pAPNnameSize	411
8.331.2.6 pAuthenticationPref	411
8.331.2.7 pGPRSMinimumQoS	411
8.331.2.8 pGPRSRequestedQoS	411
8.331.2.9 plmCnFlag	411
8.331.2.10 pIPv4AddrPref	412
8.331.2.11 pIPv6AddPref	412
8.331.2.12 pPassword	412
8.331.2.13 pPasswordSize	412
8.331.2.14 pPcscfAddrUsingDhcp	412
8.331.2.15 pPcscfAddrUsingPCO	412
8.331.2.16 pPDNInactivTimeout	412
8.331.2.17 pPdpAccessConFlag	412
8.331.2.18 pPdpContext	412
8.331.2.19 pPdpDataCompType	412
8.331.2.20 pPdpHdrCompType	412
8.331.2.21 pPDPtype	412
8.331.2.22 pPriDNSIPv4AddPref	412
8.331.2.23 pPriDNSIPv6addpref	412
8.331.2.24 pPrimaryID	412
8.331.2.25 pProfilename	412
8.331.2.26 pProfilenameSize	412
8.331.2.27 pQoSClassID	412
8.331.2.28 pSecDNSIPv4AddPref	412
8.331.2.29 pSecDNSIPv6addpref	412
8.331.2.30 pSecondaryFlag	412
8.331.2.31 pTFTID1Params	412
8.331.2.32 pTFTID2Params	412
8.331.2.33 pUMTSMinQoS	412
8.331.2.34 pUMTSMinQoSSigInd	412
8.331.2.35 pUMTSReqQoS	412
8.331.2.36 pUMTSReqQoSSigInd	412
8.331.2.37 pUsername	412
8.331.2.38 pUsernameSize	413
8.332 Profile3GPP2 Struct Reference	413

8.332.1 Detailed Description . . . . .	413
8.332.2 Field Documentation . . . . .	418
8.332.2.1 pAllowLinger . . . . .	418
8.332.2.2 pAPNClass3GPP2 . . . . .	418
8.332.2.3 pAPNEnabled3GPP2 . . . . .	418
8.332.2.4 pApnString . . . . .	418
8.332.2.5 pApnStringSize . . . . .	418
8.332.2.6 pAppPriority . . . . .	418
8.332.2.7 pAppType . . . . .	418
8.332.2.8 pAuthPassword . . . . .	418
8.332.2.9 pAuthPasswordSize . . . . .	418
8.332.2.10 pAuthProtocol . . . . .	418
8.332.2.11 pAuthRetryCount . . . . .	418
8.332.2.12 pAuthTimeout . . . . .	418
8.332.2.13 pDataMode . . . . .	418
8.332.2.14 pDataRate . . . . .	418
8.332.2.15 pIpcpAckTimeout . . . . .	418
8.332.2.16 pIpcpCreqRetryCount . . . . .	418
8.332.2.17 pIsPcscfAddressNedded . . . . .	418
8.332.2.18 pLcpAckTimeout . . . . .	418
8.332.2.19 pLcpCreqRetryCount . . . . .	418
8.332.2.20 pNegoDnsSrvrPref . . . . .	418
8.332.2.21 pPDNInactivTimeout3GPP2 . . . . .	418
8.332.2.22 pPdnType . . . . .	418
8.332.2.23 pPppSessCloseTimer1x . . . . .	419
8.332.2.24 pPppSessCloseTimerDO . . . . .	419
8.332.2.25 pPrimaryV4DnsAddress . . . . .	419
8.332.2.26 pPriV6DnsAddress . . . . .	419
8.332.2.27 pRATType . . . . .	419
8.332.2.28 pSecondaryV4DnsAddress . . . . .	419
8.332.2.29 pSecV6DnsAddress . . . . .	419
8.332.2.30 pUserId . . . . .	419
8.332.2.31 pUserIdSize . . . . .	419
8.333 ProfileIdentifier Struct Reference . . . . .	419
8.333.1 Detailed Description . . . . .	419
8.333.2 Field Documentation . . . . .	419
8.333.2.1 profileIndex . . . . .	419
8.333.2.2 profileType . . . . .	419
8.334 protocolSubtypeElement Struct Reference . . . . .	419
8.334.1 Detailed Description . . . . .	420



8.334.2 Field Documentation	421
8.334.2.1 AccessMac	421
8.334.2.2 AuthProt	421
8.334.2.3 ControlMac	421
8.334.2.4 EncryptProt	421
8.334.2.5 ForwardMac	421
8.334.2.6 IdleState	421
8.334.2.7 KeyExchange	421
8.334.2.8 MultDisc	421
8.334.2.9 PhysicalLayer	421
8.334.2.10ReverseMac	421
8.334.2.11SecProt	421
8.334.2.12VirtStream	421
8.335PSDetachReq Struct Reference	421
8.335.1 Detailed Description	421
8.335.2 Field Documentation	421
8.335.2.1 pDetachAction	421
8.336qaQmi3Gpp2TimeZone Struct Reference	422
8.336.1 Detailed Description	422
8.336.2 Field Documentation	422
8.336.2.1 daylightSavings	422
8.336.2.2 leapSeconds	422
8.336.2.3 localTimeOffset	422
8.337qaQmiInterfaceInfo Struct Reference	422
8.337.1 Detailed Description	422
8.337.2 Field Documentation	423
8.337.2.1 qaQmiinstanceid	423
8.337.2.2 qaQmisvctype	423
8.337.2.3 v4sessionId	423
8.337.2.4 v6sessionId	423
8.338qaQmiServingSystemParam Struct Reference	423
8.338.1 Detailed Description	424
8.338.2 Field Documentation	426
8.338.2.1 BasestationID	426
8.338.2.2 BasestationLatitude	426
8.338.2.3 BasestationLongitude	426
8.338.2.4 CallBarStatus	426
8.338.2.5 CDMA_P_Rev	427
8.338.2.6 CDMASystemInfoExt	427
8.338.2.7 CellID	427

8.338.2.8 concSvcInfo . . . . .	427
8.338.2.9 CurrentPLMN . . . . .	427
8.338.2.10DataSrvCapabilities . . . . .	427
8.338.2.11defaultRoamInd . . . . .	427
8.338.2.12DetailedSvcInfo . . . . .	427
8.338.2.13DTMInd . . . . .	427
8.338.2.14Gpp2TimeZone . . . . .	427
8.338.2.15GppNetworkDSTAdjustment . . . . .	427
8.338.2.16GppTimeZone . . . . .	427
8.338.2.17hdrPersonality . . . . .	427
8.338.2.18Lac . . . . .	427
8.338.2.19NetworkID . . . . .	427
8.338.2.20PRLInd . . . . .	427
8.338.2.21roamIndicatorVal . . . . .	427
8.338.2.22RoamingIndicatorList . . . . .	427
8.338.2.23ServingSystem . . . . .	427
8.338.2.24SystemID . . . . .	427
8.338.2.25TrackAreaCode . . . . .	427
8.339QmiCbkCatEventStatusReportInd Struct Reference . . . . .	427
8.339.1 Field Documentation . . . . .	427
8.339.1.1 CCETiv . . . . .	427
8.339.1.2 event_Index . . . . .	427
8.340QmiCbkLocCradleMountInd Struct Reference . . . . .	428
8.340.1 Detailed Description . . . . .	428
8.340.2 Field Documentation . . . . .	428
8.340.2.1 cradleMountConfigStatus . . . . .	428
8.341QmiCbkLocEventTimeSyncInd Struct Reference . . . . .	428
8.341.1 Detailed Description . . . . .	428
8.341.2 Field Documentation . . . . .	429
8.341.2.1 timeSyncRefCounter . . . . .	429
8.342QmiCbkLocInjectSensorDataInd Struct Reference . . . . .	429
8.342.1 Detailed Description . . . . .	429
8.342.2 Field Documentation . . . . .	430
8.342.2.1 injectSensorDataStatus . . . . .	430
8.342.2.2 pAccelSamplesAccepted . . . . .	430
8.342.2.3 pAccelTempSamplesAccepted . . . . .	430
8.342.2.4 pGyroSamplesAccepted . . . . .	430
8.342.2.5 pGyroTempSamplesAccepted . . . . .	430
8.342.2.6 pOpaqueIdentifier . . . . .	430
8.343QmiCbkLocInjectTimeInd Struct Reference . . . . .	430

8.343.1 Detailed Description . . . . .	430
8.343.2 Field Documentation . . . . .	431
8.343.2.1 injectTimeSyncStatus . . . . .	431
8.344QmiCbkLocPositionReportInd Struct Reference . . . . .	431
8.344.1 Detailed Description . . . . .	432
8.344.2 Field Documentation . . . . .	436
8.344.2.1 pAltitudeAssumed . . . . .	436
8.344.2.2 pAltitudeWrtEllipsoid . . . . .	436
8.344.2.3 pAltitudeWrtMeanSeaLevel . . . . .	436
8.344.2.4 pFixId . . . . .	436
8.344.2.5 pGpsTime . . . . .	436
8.344.2.6 pHeading . . . . .	436
8.344.2.7 pHeadingUnc . . . . .	436
8.344.2.8 pHorConfidence . . . . .	436
8.344.2.9 pHorReliability . . . . .	436
8.344.2.10pHorUncCircular . . . . .	436
8.344.2.11pHorUncEllipseOrientAzimuth . . . . .	436
8.344.2.12pHorUncEllipseSemiMajor . . . . .	436
8.344.2.13pHorUncEllipseSemiMinor . . . . .	436
8.344.2.14pLatitude . . . . .	436
8.344.2.15pLeapSeconds . . . . .	436
8.344.2.16pLongitude . . . . .	436
8.344.2.17pMagneticDeviation . . . . .	436
8.344.2.18pPrecisionDilution . . . . .	437
8.344.2.19pSensorDataUsage . . . . .	437
8.344.2.20pSpeedHorizontal . . . . .	437
8.344.2.21pSpeedUnc . . . . .	437
8.344.2.22pSpeedVertical . . . . .	437
8.344.2.23pSvUsedforFix . . . . .	437
8.344.2.24pTechnologyMask . . . . .	437
8.344.2.25pTimeSrc . . . . .	437
8.344.2.26pTimestampUtc . . . . .	437
8.344.2.27pTimeUnc . . . . .	437
8.344.2.28pVertConfidence . . . . .	437
8.344.2.29pVertReliability . . . . .	437
8.344.2.30pVertUnc . . . . .	437
8.344.2.31sessionId . . . . .	437
8.344.2.32sessionStatus . . . . .	437
8.345QmiCbkLocSensorStreamingInd Struct Reference . . . . .	437
8.345.1 Detailed Description . . . . .	437

8.345.2 Field Documentation	438
8.345.2.1 pAccelAcceptReady	438
8.345.2.2 pAccelTempAcceptReady	438
8.345.2.3 pGyroAcceptReady	438
8.345.2.4 pGyroTempAcceptReady	438
8.346QmiCbkNasLTECphyCalInfo Struct Reference	438
8.346.1 Detailed Description	438
8.346.2 Field Documentation	438
8.346.2.1 sPhyCaAggPcellInfo	439
8.346.2.2 sPhyCaAggScellIDBw	439
8.346.2.3 sPhyCaAggScellIndex	439
8.346.2.4 sPhyCaAggScellIndType	439
8.346.2.5 sPhyCaAggScellInfo	439
8.347QmiCbkSwiOmaDmEventStatusReportInd Struct Reference	439
8.347.1 Field Documentation	439
8.347.1.1 SITlv	439
8.348QmiCbkSwiOmaDmEventStatusReportIndExt Struct Reference	439
8.348.1 Field Documentation	439
8.348.1.1 SITlv	439
8.349QmiCbkWdsStatisticsIndState Struct Reference	439
8.349.1 Detailed Description	439
8.349.2 Field Documentation	440
8.349.2.1 RxDropConutTlv	440
8.349.2.2 RxOkByteCountTlv	440
8.349.2.3 RxOkConutTlv	440
8.349.2.4 TxDropConutTlv	440
8.349.2.5 TxOkByteCountTlv	440
8.349.2.6 TxOkConutTlv	440
8.350qmiwinfo_s Struct Reference	440
8.350.1 Detailed Description	440
8.350.2 Field Documentation	441
8.350.2.1 dev	441
8.350.2.2 g	441
8.350.2.3 s	441
8.351QmiNas3GppNetworkInfo Struct Reference	441
8.351.1 Detailed Description	441
8.351.2 Field Documentation	442
8.351.2.1 pDesription	442
8.351.2.2 pForbidden	442
8.351.2.3 plnUse	442

8.351.2.4 pMCC . . . . .	442
8.351.2.5 pMNC . . . . .	442
8.351.2.6 pPreferred . . . . .	442
8.351.2.7 pRoaming . . . . .	443
8.352QmiNasGetRFBandInfoResp Struct Reference . . . . .	443
8.352.1 Field Documentation . . . . .	443
8.352.1.1 plnInstancesSize . . . . .	443
8.352.1.2 pRFBandInfoElements . . . . .	443
8.352.1.3 results . . . . .	443
8.353QmiNasPerformNetworkScanResp Struct Reference . . . . .	443
8.353.1 Field Documentation . . . . .	443
8.353.1.1 plnInstances . . . . .	443
8.353.1.2 plnInstanceSize . . . . .	443
8.353.1.3 results . . . . .	443
8.354QmiWdsIpAddressInfo Struct Reference . . . . .	443
8.354.1 Detailed Description . . . . .	444
8.354.2 Field Documentation . . . . .	444
8.354.2.1 pIPAddressV4 . . . . .	444
8.354.2.2 pIPAddressV6 . . . . .	444
8.354.2.3 pIPv6prefixlen . . . . .	444
8.355qmiWdsRunTimeSettings Struct Reference . . . . .	444
8.355.1 Detailed Description . . . . .	445
8.355.2 Field Documentation . . . . .	447
8.355.2.1 pAPNName . . . . .	447
8.355.2.2 pAuthentication . . . . .	447
8.355.2.3 pDomainList . . . . .	447
8.355.2.4 pGPRSGrantedQoS . . . . .	447
8.355.2.5 pGWAddressV4 . . . . .	447
8.355.2.6 pIMCNflag . . . . .	447
8.355.2.7 pIPAddressV4 . . . . .	447
8.355.2.8 pIPFamilyPreference . . . . .	447
8.355.2.9 pIPv6AddrInfo . . . . .	448
8.355.2.10pIPv6GWAddrInfo . . . . .	448
8.355.2.11pMtu . . . . .	448
8.355.2.12pPCSCFAddrPCO . . . . .	448
8.355.2.13pPCSCFFQDAddrList . . . . .	448
8.355.2.14pPDPTtype . . . . .	448
8.355.2.15pPrimaryDNSV4 . . . . .	448
8.355.2.16pPrimaryDNSV6 . . . . .	448
8.355.2.17pProfileID . . . . .	448

8.355.2.18pProfileName . . . . .	448
8.355.2.19pSecondaryDNSV4 . . . . .	448
8.355.2.20pSecondaryDNSV6 . . . . .	448
8.355.2.21pServerAddrList . . . . .	448
8.355.2.22pSubnetMaskV4 . . . . .	448
8.355.2.23pTechnology . . . . .	448
8.355.2.24pUMTSGrantedQoS . . . . .	448
8.355.2.25pUsername . . . . .	448
8.356QosClassID Struct Reference . . . . .	448
8.356.1 Detailed Description . . . . .	448
8.356.2 Field Documentation . . . . .	449
8.356.2.1 gDIBitRate . . . . .	449
8.356.2.2 gUIBitRate . . . . .	449
8.356.2.3 maxDIBitRate . . . . .	449
8.356.2.4 maxUIBitRate . . . . .	449
8.356.2.5 QCI . . . . .	449
8.357QosEventInfo Struct Reference . . . . .	449
8.357.1 Detailed Description . . . . .	449
8.357.2 Field Documentation . . . . .	450
8.357.2.1 pDataBearer . . . . .	451
8.357.2.2 pPacketsCountRX . . . . .	451
8.357.2.3 pPacketsCountTX . . . . .	451
8.357.2.4 pTotalBytesRX . . . . .	451
8.357.2.5 pTotalBytesTX . . . . .	451
8.358QosFlowInfo Struct Reference . . . . .	451
8.358.1 Detailed Description . . . . .	451
8.358.2 Field Documentation . . . . .	451
8.358.2.1 pBearerID . . . . .	452
8.358.2.2 pQFlowState . . . . .	452
8.358.2.3 pRxQFilter . . . . .	452
8.358.2.4 pRxQFlowGranted . . . . .	452
8.358.2.5 pTxQFilter . . . . .	452
8.358.2.6 pTxQFlowGranted . . . . .	452
8.359QosFlowInfoState Struct Reference . . . . .	452
8.359.1 Detailed Description . . . . .	452
8.359.2 Field Documentation . . . . .	452
8.359.2.1 id . . . . .	452
8.359.2.2 isNewFlow . . . . .	452
8.359.2.3 state . . . . .	452
8.360QosMap Struct Reference . . . . .	452

8.360.1 Detailed Description	453
8.360.2 Field Documentation	453
8.360.2.1 dscp	453
8.360.2.2 qos_id	453
8.360.2.3 state	453
8.361 readResult Struct Reference	453
8.361.1 Detailed Description	453
8.361.2 Field Documentation	454
8.361.2.1 content	454
8.361.2.2 contentLen	454
8.362 readTransparentInfo Struct Reference	454
8.362.1 Detailed Description	454
8.362.2 Field Documentation	454
8.362.2.1 length	454
8.362.2.2 offset	454
8.363 redirNumInfo Struct Reference	454
8.363.1 Detailed Description	454
8.363.2 Field Documentation	456
8.363.2.1 number	456
8.363.2.2 numLen	456
8.363.2.3 numPlan	456
8.363.2.4 numType	456
8.363.2.5 PI	456
8.363.2.6 reason	456
8.363.2.7 SI	456
8.364 registerRefresh Struct Reference	456
8.364.1 Detailed Description	457
8.364.2 Field Documentation	458
8.364.2.1 arrfileInfo	458
8.364.2.2 numFiles	458
8.364.2.3 registerFlag	458
8.364.2.4 voteForInit	458
8.365 remainingRetries Struct Reference	458
8.365.1 Detailed Description	458
8.365.2 Field Documentation	459
8.365.2.1 unblockLeft	459
8.365.2.2 verifyLeft	459
8.366 remotePartyName Struct Reference	459
8.366.1 Detailed Description	459
8.366.2 Field Documentation	460

8.366.2.1 callerName . . . . .	460
8.366.2.2 codingScheme . . . . .	460
8.366.2.3 nameLen . . . . .	460
8.366.2.4 namePI . . . . .	460
8.367remotePartyNum Struct Reference . . . . .	460
8.367.1 Detailed Description . . . . .	460
8.367.2 Field Documentation . . . . .	461
8.367.2.1 numLen . . . . .	461
8.367.2.2 presentationInd . . . . .	461
8.367.2.3 remPartyNumber . . . . .	461
8.368ReqFieldsList Struct Reference . . . . .	461
8.368.1 Detailed Description . . . . .	461
8.368.2 Field Documentation . . . . .	462
8.368.2.1 requestFields . . . . .	462
8.368.2.2 requestFieldsLen . . . . .	462
8.369RespFieldsList Struct Reference . . . . .	462
8.369.1 Detailed Description . . . . .	462
8.369.2 Field Documentation . . . . .	462
8.369.2.1 responseFields . . . . .	462
8.369.2.2 responseFieldsLen . . . . .	462
8.370RFBandInfoElements Struct Reference . . . . .	462
8.370.1 Detailed Description . . . . .	462
8.370.2 Field Documentation . . . . .	463
8.370.2.1 activeBandClass . . . . .	463
8.370.2.2 activeChannel . . . . .	463
8.370.2.3 radiolInterface . . . . .	463
8.371roamIndList Struct Reference . . . . .	463
8.371.1 Detailed Description . . . . .	463
8.371.2 Field Documentation . . . . .	464
8.371.2.1 numInstances . . . . .	464
8.371.2.2 radiolInterface . . . . .	464
8.371.2.3 roamIndicator . . . . .	464
8.372RoamingInfo Struct Reference . . . . .	464
8.372.1 Field Documentation . . . . .	464
8.372.1.1 roaming_ind . . . . .	464
8.372.1.2 TlvPresent . . . . .	464
8.373roamTimer Struct Reference . . . . .	464
8.373.1 Detailed Description . . . . .	465
8.373.2 Field Documentation . . . . .	466
8.373.2.1 namID . . . . .	466



8.373.2.2 roamTimerValue . . . . .	466
8.374RSRPThresh Struct Reference . . . . .	466
8.374.1 Detailed Description . . . . .	466
8.374.2 Field Documentation . . . . .	467
8.374.2.1 pRSRPThresList . . . . .	467
8.374.2.2 RSRPThresListLen . . . . .	467
8.375rsrqInformation Struct Reference . . . . .	467
8.375.1 Detailed Description . . . . .	467
8.375.2 Field Documentation . . . . .	467
8.375.2.1 radiolf . . . . .	467
8.375.2.2 rsrq . . . . .	467
8.376RSRQThresh Struct Reference . . . . .	467
8.376.1 Detailed Description . . . . .	467
8.376.2 Field Documentation . . . . .	468
8.376.2.1 pRSRQThresList . . . . .	468
8.376.2.2 RSRQThresListLen . . . . .	468
8.377RSSIThresh Struct Reference . . . . .	468
8.377.1 Detailed Description . . . . .	468
8.377.2 Field Documentation . . . . .	469
8.377.2.1 pRSSIThresList . . . . .	469
8.377.2.2 RSSIThresListLen . . . . .	469
8.378RXAGCList Struct Reference . . . . .	469
8.378.1 Detailed Description . . . . .	469
8.378.2 Field Documentation . . . . .	469
8.378.2.1 pRXAIG . . . . .	469
8.378.2.2 pRXComprSlope . . . . .	469
8.378.2.3 pRXComprThres . . . . .	469
8.378.2.4 pRXExpSlope . . . . .	469
8.378.2.5 pRXExpThres . . . . .	470
8.378.2.6 pRXStaticGain . . . . .	470
8.379RXAVCList Struct Reference . . . . .	470
8.379.1 Detailed Description . . . . .	470
8.379.2 Field Documentation . . . . .	470
8.379.2.1 pAVRXAVCHheadroom . . . . .	470
8.379.2.2 pAVRXAVCSens . . . . .	470
8.380rxInfo Struct Reference . . . . .	470
8.380.1 Detailed Description . . . . .	470
8.380.2 Field Documentation . . . . .	471
8.380.2.1 ecio . . . . .	471
8.380.2.2 isRadioTuned . . . . .	471

8.380.2.3 phase . . . . .	471
8.380.2.4 rscp . . . . .	471
8.380.2.5 rsrp . . . . .	471
8.380.2.6 rxPower . . . . .	472
8.381RXPCMIIRFitr Struct Reference . . . . .	472
8.381.1 Detailed Description . . . . .	472
8.381.2 Field Documentation . . . . .	473
8.381.2.1 pFlag . . . . .	473
8.381.2.2 pStage0Val . . . . .	473
8.381.2.3 pStage1Val . . . . .	473
8.381.2.4 pStage2Val . . . . .	473
8.381.2.5 pStage3Val . . . . .	473
8.381.2.6 pStage4Val . . . . .	474
8.381.2.7 pStageCnt . . . . .	474
8.382rxSignalStrengthListElement Struct Reference . . . . .	474
8.382.1 Detailed Description . . . . .	474
8.382.2 Field Documentation . . . . .	474
8.382.2.1 radiolf . . . . .	474
8.382.2.2 rxSignalStrength . . . . .	474
8.383sApnExtraParams Struct Reference . . . . .	474
8.383.1 Detailed Description . . . . .	475
8.383.2 Field Documentation . . . . .	475
8.383.2.1 ambr_dl . . . . .	475
8.383.2.2 ambr_dl_ext . . . . .	475
8.383.2.3 ambr_dl_ext2 . . . . .	475
8.383.2.4 ambr_ul . . . . .	476
8.383.2.5 ambr_ul_ext . . . . .	476
8.383.2.6 ambr_ul_ext2 . . . . .	476
8.383.2.7 apnId . . . . .	476
8.384satelliteInfo Struct Reference . . . . .	476
8.384.1 Detailed Description . . . . .	476
8.384.2 Field Documentation . . . . .	478
8.384.2.1 azimuth . . . . .	478
8.384.2.2 elevation . . . . .	478
8.384.2.3 gnssSvId . . . . .	478
8.384.2.4 healthStatus . . . . .	478
8.384.2.5 snr . . . . .	478
8.384.2.6 svInfoMask . . . . .	478
8.384.2.7 svListLen . . . . .	478
8.384.2.8 svStatus . . . . .	478

8.384.2.9 system . . . . .	478
8.384.2.10 validMask . . . . .	478
8.385 sensorDataUsage_s Struct Reference . . . . .	478
8.385.1 Detailed Description . . . . .	478
8.385.2 Field Documentation . . . . .	479
8.385.2.1 aidingIndicatorMask . . . . .	479
8.385.2.2 usageMask . . . . .	479
8.386 serialNumbersInfo Struct Reference . . . . .	479
8.386.1 Detailed Description . . . . .	479
8.386.2 Field Documentation . . . . .	480
8.386.2.1 esnSize . . . . .	480
8.386.2.2 imeiSize . . . . .	480
8.386.2.3 imeiSvnSize . . . . .	480
8.386.2.4 meidSize . . . . .	480
8.386.2.5 pESNString . . . . .	480
8.386.2.6 pIMEIString . . . . .	480
8.386.2.7 plmeiSvnString . . . . .	480
8.386.2.8 pMEIDString . . . . .	480
8.387 serviceProviderName Struct Reference . . . . .	480
8.387.1 Detailed Description . . . . .	480
8.387.2 Field Documentation . . . . .	481
8.387.2.1 displayCondition . . . . .	481
8.387.2.2 spn . . . . .	481
8.387.2.3 spnLength . . . . .	481
8.388 ServingSystemInfo Struct Reference . . . . .	481
8.388.1 Detailed Description . . . . .	481
8.388.2 Field Documentation . . . . .	482
8.388.2.1 csAttachState . . . . .	482
8.388.2.2 hdrPersonality . . . . .	482
8.388.2.3 psAttachState . . . . .	482
8.388.2.4 radiolInterfaceList . . . . .	482
8.388.2.5 radiolInterfaceNo . . . . .	483
8.388.2.6 registrationState . . . . .	483
8.388.2.7 selectedNetwork . . . . .	483
8.389 servSystem Struct Reference . . . . .	483
8.389.1 Detailed Description . . . . .	483
8.389.2 Field Documentation . . . . .	484
8.389.2.1 csAttachState . . . . .	484
8.389.2.2 numRadioInterfaces . . . . .	485
8.389.2.3 psAttachState . . . . .	485

8.389.2.4 radiolInterface . . . . .	485
8.389.2.5 regState . . . . .	485
8.389.2.6 selNetwork . . . . .	485
8.390sessionInfo Union Reference . . . . .	485
8.390.1 Detailed Description . . . . .	485
8.390.2 Field Documentation . . . . .	485
8.390.2.1 omaDmConfig . . . . .	485
8.390.2.2 omaDmFota . . . . .	485
8.390.2.3 omaDmNotifications . . . . .	485
8.391sessionInfoExt Union Reference . . . . .	485
8.391.1 Detailed Description . . . . .	485
8.391.2 Field Documentation . . . . .	485
8.391.2.1 omaDmConfig . . . . .	485
8.391.2.2 omaDmFota . . . . .	485
8.392sessionInfoTlv Struct Reference . . . . .	485
8.392.1 Detailed Description . . . . .	486
8.392.2 Field Documentation . . . . .	486
8.392.2.1 sessionInfo . . . . .	486
8.392.2.2 sessionType . . . . .	486
8.392.2.3 TlvPresent . . . . .	486
8.393sessionInfoTlvExt Struct Reference . . . . .	486
8.393.1 Detailed Description . . . . .	486
8.393.2 Field Documentation . . . . .	486
8.393.2.1 sessionInfo . . . . .	486
8.393.2.2 sessionType . . . . .	486
8.393.2.3 TlvPresent . . . . .	486
8.394SetAudioPathConfigReq Struct Reference . . . . .	486
8.394.1 Detailed Description . . . . .	487
8.394.2 Field Documentation . . . . .	488
8.394.2.1 pCodecSTGain . . . . .	488
8.394.2.2 pDTMFTXGain . . . . .	488
8.394.2.3 pECMode . . . . .	488
8.394.2.4 pNSEnable . . . . .	488
8.394.2.5 Profile . . . . .	488
8.394.2.6 pRXAGCList . . . . .	488
8.394.2.7 pRXAVCAGCSwitch . . . . .	488
8.394.2.8 pRXAVCList . . . . .	488
8.394.2.9 pRXPCMIIRFtr . . . . .	489
8.394.2.10pTXAGCList . . . . .	489
8.394.2.11pTXAVCSwitch . . . . .	489

8.394.2.12pTXGain . . . . .	489
8.394.2.13pTXPCMIIRFiltr . . . . .	489
8.395SetAudioProfileReq Struct Reference . . . . .	489
8.395.1 Detailed Description . . . . .	489
8.395.2 Field Documentation . . . . .	490
8.395.2.1 EarMute . . . . .	490
8.395.2.2 Generator . . . . .	490
8.395.2.3 MicMute . . . . .	490
8.395.2.4 Profile . . . . .	491
8.395.2.5 Volume . . . . .	491
8.396SetAudioVolTLBConfigReq Struct Reference . . . . .	491
8.396.1 Detailed Description . . . . .	491
8.396.2 Field Documentation . . . . .	492
8.396.2.1 Generator . . . . .	492
8.396.2.2 Item . . . . .	492
8.396.2.3 Profile . . . . .	492
8.396.2.4 Volume . . . . .	492
8.396.2.5 VolValue . . . . .	492
8.397SetAudioVolTLBConfigResp Struct Reference . . . . .	492
8.397.1 Detailed Description . . . . .	492
8.397.2 Field Documentation . . . . .	492
8.397.2.1 ResCode . . . . .	492
8.398setCustomSettingV2 Struct Reference . . . . .	492
8.398.1 Detailed Description . . . . .	492
8.398.2 Field Documentation . . . . .	493
8.398.2.1 cust_id . . . . .	493
8.398.2.2 cust_value . . . . .	493
8.398.2.3 value_length . . . . .	493
8.399SetIMSSMSConfigReq Struct Reference . . . . .	493
8.399.1 Detailed Description . . . . .	493
8.399.2 Field Documentation . . . . .	494
8.399.2.1 pPhoneCtxtURI . . . . .	494
8.399.2.2 pPhoneCtxtURILen . . . . .	494
8.399.2.3 pSMSFormat . . . . .	494
8.399.2.4 pSMSOverIPNwInd . . . . .	494
8.400SetIMSSMSConfigResp Struct Reference . . . . .	494
8.400.1 Detailed Description . . . . .	494
8.400.2 Field Documentation . . . . .	494
8.400.2.1 pSettingResp . . . . .	494
8.401SetIMSUserConfigReq Struct Reference . . . . .	494

8.401.1 Detailed Description . . . . .	494
8.401.2 Field Documentation . . . . .	495
8.401.2.1 pIMSDomain . . . . .	495
8.401.2.2 pIMSDomainLen . . . . .	495
8.402SetIMSUserConfigResp Struct Reference . . . . .	495
8.402.1 Detailed Description . . . . .	495
8.402.2 Field Documentation . . . . .	495
8.402.2.1 pSettingResp . . . . .	495
8.403SetIMSVoIPConfigReq Struct Reference . . . . .	495
8.403.1 Detailed Description . . . . .	496
8.403.2 Field Documentation . . . . .	498
8.403.2.1 pAmrMode . . . . .	498
8.403.2.2 pAmrOctetAligned . . . . .	498
8.403.2.3 pAmrWbEnable . . . . .	499
8.403.2.4 pAmrWBMode . . . . .	499
8.403.2.5 pAmrWBOctetAligned . . . . .	499
8.403.2.6 pMinSessionExpiryTimer . . . . .	499
8.403.2.7 pRingBackTimer . . . . .	499
8.403.2.8 pRingingTimer . . . . .	499
8.403.2.9 pRTPRTCPInactTimer . . . . .	499
8.403.2.10pScrAmrEnable . . . . .	499
8.403.2.11pScrAmrWbEnable . . . . .	499
8.403.2.12pSessionExpiryTimer . . . . .	499
8.404SetIMSVoIPConfigResp Struct Reference . . . . .	499
8.404.1 Detailed Description . . . . .	499
8.404.2 Field Documentation . . . . .	499
8.404.2.1 pSettingResp . . . . .	499
8.405SetM2MAudioAVCFGRReq Struct Reference . . . . .	499
8.405.1 Detailed Description . . . . .	499
8.405.2 Field Documentation . . . . .	500
8.405.2.1 Device . . . . .	500
8.405.2.2 PIFACEId . . . . .	500
8.405.2.3 pPCMPParams . . . . .	500
8.405.2.4 Profile . . . . .	500
8.406SetM2MAudioLPBKReq Struct Reference . . . . .	500
8.406.1 Detailed Description . . . . .	500
8.406.2 Field Documentation . . . . .	501
8.406.2.1 Enable . . . . .	501
8.407SetM2MAudioProfileReq Struct Reference . . . . .	501
8.407.1 Detailed Description . . . . .	501

8.407.2 Field Documentation	502
8.407.2.1 pCwtMute	502
8.407.2.2 pEarMute	502
8.407.2.3 pGenerator	502
8.407.2.4 pMicMute	502
8.407.2.5 Profile	502
8.407.2.6 pVolume	502
8.408SetM2MAudioVolumeReq Struct Reference	502
8.408.1 Detailed Description	502
8.408.2 Field Documentation	503
8.408.2.1 Generator	503
8.408.2.2 Level	503
8.408.2.3 Profile	503
8.409SetM2MAVMuteReq Struct Reference	503
8.409.1 Detailed Description	503
8.409.2 Field Documentation	503
8.409.2.1 EarMute	504
8.409.2.2 MicMute	504
8.409.2.3 pCwtMute	504
8.409.2.4 Profile	504
8.410SetM2MSpkrGainReq Struct Reference	504
8.410.1 Detailed Description	504
8.410.2 Field Documentation	504
8.410.2.1 Profile	504
8.410.2.2 Value	504
8.411setPINProtection Struct Reference	504
8.411.1 Detailed Description	504
8.411.2 Field Documentation	505
8.411.2.1 pinID	505
8.411.2.2 pinLength	505
8.411.2.3 pinOperation	505
8.411.2.4 pinValue	505
8.412SetRegMgrConfigReq Struct Reference	505
8.412.1 Detailed Description	505
8.412.2 Field Documentation	506
8.412.2.1 pCSCFPortName	506
8.412.2.2 pCSCFPortNameLen	506
8.412.2.3 pIMSTestMode	506
8.412.2.4 pPriCSCFPort	506
8.413SetRegMgrConfigResp Struct Reference	506

8.413.1 Detailed Description . . . . .	506
8.413.2 Field Documentation . . . . .	506
8.413.2.1 pSettingResp . . . . .	506
8.414 setSignalStrengthInfo Struct Reference . . . . .	507
8.414.1 Detailed Description . . . . .	507
8.414.2 Field Documentation . . . . .	511
8.414.2.1 pCDMAECIODelta . . . . .	511
8.414.2.2 pCDMAECIOThresh . . . . .	511
8.414.2.3 pCDMARSSIDelta . . . . .	511
8.414.2.4 pCDMARSSIThresh . . . . .	511
8.414.2.5 pGSMRSSIDelta . . . . .	511
8.414.2.6 pGSMRSSIThresh . . . . .	511
8.414.2.7 pHDRECIODelta . . . . .	511
8.414.2.8 pHDRECIOThresh . . . . .	511
8.414.2.9 pHDRIODelta . . . . .	511
8.414.2.10 pHDRIOThresh . . . . .	511
8.414.2.11 pHDRRSSIDelta . . . . .	511
8.414.2.12 pHDRRSSIThresh . . . . .	511
8.414.2.13 pHDRSINRDelta . . . . .	511
8.414.2.14 pHDRSINRThresh . . . . .	511
8.414.2.15 pLTERSRPDelta . . . . .	511
8.414.2.16 pLTERSRPThresh . . . . .	511
8.414.2.17 pLTERSRQDelta . . . . .	511
8.414.2.18 pLTERSRQThresh . . . . .	511
8.414.2.19 pLTERSSIDelta . . . . .	511
8.414.2.20 pLTERSSIThresh . . . . .	511
8.414.2.21 pLTERSigRptConfig . . . . .	511
8.414.2.22 pLTESNRDelta . . . . .	511
8.414.2.23 pLTESNRThresh . . . . .	511
8.414.2.24 pTDSCDMAECIODelta . . . . .	511
8.414.2.25 pTDSCDMAECIOThresh . . . . .	511
8.414.2.26 pTDSCDMARSCPDelta . . . . .	511
8.414.2.27 pTDSCDMARSCPThresh . . . . .	511
8.414.2.28 pTDSCDMARSSIDelta . . . . .	512
8.414.2.29 pTDSCDMARSSIThresh . . . . .	512
8.414.2.30 pTDSCDMASINRDelta . . . . .	512
8.414.2.31 pTDSCDMASINRThresh . . . . .	512
8.414.2.32 pWCDMAECIODelta . . . . .	512
8.414.2.33 pWCDMAECIOThresh . . . . .	512
8.414.2.34 pWCDMARSSIDelta . . . . .	512



8.414.2.35pWCDMARSSIThresh . . . . .	512
8.415SetSIPConfigReq Struct Reference . . . . .	512
8.415.1 Detailed Description . . . . .	512
8.415.2 Field Documentation . . . . .	513
8.415.2.1 pSigCompEnabled . . . . .	513
8.415.2.2 pSIPLocalPort . . . . .	513
8.415.2.3 pSubscribeTimer . . . . .	513
8.415.2.4 pTimerSIPReg . . . . .	513
8.415.2.5 pTimerT1 . . . . .	513
8.415.2.6 pTimerT2 . . . . .	513
8.415.2.7 pTimerTf . . . . .	513
8.416SetSIPConfigResp Struct Reference . . . . .	513
8.416.1 Detailed Description . . . . .	513
8.416.2 Field Documentation . . . . .	513
8.416.2.1 pSettingResp . . . . .	513
8.417sGetDeviceSeriesResult Struct Reference . . . . .	513
8.417.1 Detailed Description . . . . .	514
8.417.2 Field Documentation . . . . .	514
8.417.2.1 eDevice . . . . .	514
8.417.2.2 uResult . . . . .	514
8.418sidNid Struct Reference . . . . .	514
8.418.1 Detailed Description . . . . .	514
8.418.2 Field Documentation . . . . .	514
8.418.2.1 nid . . . . .	514
8.418.2.2 sid . . . . .	514
8.419sigInfo Struct Reference . . . . .	514
8.419.1 Detailed Description . . . . .	515
8.419.2 Field Documentation . . . . .	515
8.419.2.1 pECIOThresh . . . . .	515
8.419.2.2 pHDRSINRThresh . . . . .	516
8.419.2.3 pLOThresh . . . . .	516
8.419.2.4 pLTESigRptCfg . . . . .	516
8.419.2.5 pLTESNRThresh . . . . .	516
8.419.2.6 pRSRPThresh . . . . .	516
8.419.2.7 pRSRQThresh . . . . .	516
8.419.2.8 pRSSIThresh . . . . .	516
8.420signalInfo Struct Reference . . . . .	516
8.420.1 Detailed Description . . . . .	516
8.420.2 Field Documentation . . . . .	516
8.420.2.1 alertPitch . . . . .	516

8.420.2.2 signal . . . . .	516
8.420.2.3 signalType . . . . .	516
8.421 SignalStrengthDataType Struct Reference . . . . .	516
8.421.1 Field Documentation . . . . .	517
8.421.1.1 thresholds . . . . .	517
8.421.1.2 thresholdsSize . . . . .	517
8.422 slotInfo Struct Reference . . . . .	517
8.422.1 Detailed Description . . . . .	517
8.422.2 Field Documentation . . . . .	518
8.422.2.1 AppStatus . . . . .	518
8.422.2.2 cardState . . . . .	518
8.422.2.3 errorState . . . . .	518
8.422.2.4 numApp . . . . .	518
8.422.2.5 upinRetries . . . . .	518
8.422.2.6 upinState . . . . .	518
8.422.2.7 upukRetries . . . . .	518
8.423 slqsautoconnect Struct Reference . . . . .	519
8.423.1 Detailed Description . . . . .	519
8.423.2 Field Documentation . . . . .	519
8.423.2.1 acroamsetting . . . . .	519
8.423.2.2 acsetting . . . . .	519
8.423.2.3 action . . . . .	519
8.424 SLQSDeleteProfileParams Struct Reference . . . . .	519
8.424.1 Detailed Description . . . . .	519
8.424.2 Field Documentation . . . . .	520
8.424.2.1 profileIndex . . . . .	520
8.424.2.2 profileType . . . . .	520
8.425 slqsfwinfo_s Struct Reference . . . . .	520
8.425.1 Detailed Description . . . . .	520
8.425.2 Field Documentation . . . . .	521
8.425.2.1 appversion_str . . . . .	521
8.425.2.2 bootversion_str . . . . .	521
8.425.2.3 carrier_str . . . . .	521
8.425.2.4 cur_carr_name . . . . .	521
8.425.2.5 cur_carr_rev . . . . .	521
8.425.2.6 modelid_str . . . . .	521
8.425.2.7 packageid_str . . . . .	521
8.425.2.8 priversion_str . . . . .	521
8.425.2.9 sku_str . . . . .	521
8.426 SlqsNas3GppNetworkInfo Struct Reference . . . . .	521

8.426.1 Detailed Description	522
8.426.2 Field Documentation	523
8.426.2.1 Description	523
8.426.2.2 Forbidden	523
8.426.2.3 InUse	523
8.426.2.4 MCC	524
8.426.2.5 MNC	524
8.426.2.6 Preferred	524
8.426.2.7 Roaming	524
8.427SlqsNas3GppNetworkRAT Struct Reference	524
8.427.1 Detailed Description	524
8.427.2 Field Documentation	524
8.427.2.1 MCC	524
8.427.2.2 MNC	524
8.427.2.3 RAT	524
8.428SlqsNasPcsDigit Struct Reference	524
8.428.1 Detailed Description	525
8.428.2 Field Documentation	526
8.428.2.1 includes_pcs_digit	526
8.428.2.2 MCC	526
8.428.2.3 MNC	526
8.429slqssendasynsmsparams_s Struct Reference	526
8.429.1 Detailed Description	526
8.429.2 Field Documentation	528
8.429.2.1 messageFormat	528
8.429.2.2 messageSize	528
8.429.2.3 pFollowOnDC	528
8.429.2.4 pForceOnDC	528
8.429.2.5 pLinktimer	528
8.429.2.6 pMessage	528
8.429.2.7 pRetryMessage	528
8.429.2.8 pRetryMessageld	528
8.429.2.9 pServiceOption	529
8.429.2.10pSmsOnlms	529
8.429.2.11pUserData	529
8.430slqssendsmsparams_s Struct Reference	529
8.430.1 Detailed Description	529
8.430.2 Field Documentation	529
8.430.2.1 messageFailureCode	529
8.430.2.2 messageFormat	529

8.430.2.3 messageID . . . . .	530
8.430.2.4 messageSize . . . . .	530
8.430.2.5 pLinktimer . . . . .	530
8.430.2.6 pMessage . . . . .	530
8.431slqsSessionStateInfo Struct Reference . . . . .	530
8.431.1 Detailed Description . . . . .	530
8.431.2 Field Documentation . . . . .	530
8.431.2.1 pQmiInterfaceInfo . . . . .	530
8.431.2.2 reconfiguration_required . . . . .	530
8.431.2.3 sessionEndReason . . . . .	530
8.431.2.4 state . . . . .	531
8.432slqsSignalStrengthInfo Struct Reference . . . . .	531
8.432.1 Detailed Description . . . . .	531
8.432.2 Field Documentation . . . . .	533
8.432.2.1 ecioList . . . . .	533
8.432.2.2 ecioListLen . . . . .	533
8.432.2.3 errorRateList . . . . .	533
8.432.2.4 errorRateListLen . . . . .	533
8.432.2.5 lo . . . . .	533
8.432.2.6 ltersrp . . . . .	533
8.432.2.7 ltesnr . . . . .	533
8.432.2.8 rsrqInfo . . . . .	533
8.432.2.9 rxSignalStrengthList . . . . .	534
8.432.2.10rxSignalStrengthListLen . . . . .	534
8.432.2.11signalStrengthReqMask . . . . .	534
8.432.2.12sinr . . . . .	534
8.433SLQSSignalStrengthsIndReq Struct Reference . . . . .	534
8.433.1 Detailed Description . . . . .	534
8.433.2 Field Documentation . . . . .	535
8.433.2.1 ecioDelta . . . . .	535
8.433.2.2 ecioThresholdList . . . . .	535
8.433.2.3 ecioThresholdListLen . . . . .	535
8.433.2.4 ioDelta . . . . .	535
8.433.2.5 lteRsrpDelta . . . . .	535
8.433.2.6 lteSnrDelta . . . . .	535
8.433.2.7 rsrqDelta . . . . .	536
8.433.2.8 rxSignalStrengthDelta . . . . .	536
8.433.2.9 sinrDelta . . . . .	536
8.433.2.10sinrThresholdList . . . . .	536
8.433.2.11sinrThresholdListLen . . . . .	536

8.434SLQSSignalStrengthsInformation Struct Reference	536
8.434.1 Detailed Description	536
8.434.2 Field Documentation	537
8.434.2.1 eciInfo	537
8.434.2.2 errorRateInfo	537
8.434.2.3 io	537
8.434.2.4 lteRsrpinfo	537
8.434.2.5 lteSnrinfo	537
8.434.2.6 rsrqInfo	537
8.434.2.7 rxSignalStrengthInfo	537
8.434.2.8 sinr	537
8.435slqsWdsEventInfo Struct Reference	537
8.435.1 Detailed Description	537
8.435.2 Field Documentation	539
8.435.2.1 pDataBearer	539
8.435.2.2 pDormancyStatus	539
8.435.2.3 pPacketsCountRX	539
8.435.2.4 pPacketsCountTX	539
8.435.2.5 pQmiInterfaceInfo	539
8.435.2.6 pTotalBytesRX	539
8.435.2.7 pTotalBytesTX	539
8.436SMSAsyncRawSend_s Struct Reference	539
8.436.1 Detailed Description	540
8.436.2 Field Documentation	542
8.436.2.1 alphaIDLen	542
8.436.2.2 causeCode	542
8.436.2.3 errorClass	542
8.436.2.4 messageId	542
8.436.2.5 msgDelFailureCause	542
8.436.2.6 msgDelFailureType	542
8.436.2.7 pAlphaID	542
8.436.2.8 RPCause	542
8.436.2.9 sendStatus	542
8.436.2.10TPCause	542
8.436.2.11userData	542
8.437SMSCAddress Struct Reference	542
8.437.1 Detailed Description	542
8.437.2 Field Documentation	543
8.437.2.1 data	543
8.437.2.2 length	543

8.438SMSEtwsMessage Struct Reference . . . . .	543
8.438.1 Detailed Description . . . . .	543
8.438.2 Field Documentation . . . . .	543
8.438.2.1 data . . . . .	543
8.438.2.2 length . . . . .	543
8.438.2.3 notificationType . . . . .	543
8.439SMSEtwsPlmn Struct Reference . . . . .	543
8.439.1 Detailed Description . . . . .	544
8.439.2 Field Documentation . . . . .	545
8.439.2.1 mobileCountryCode . . . . .	545
8.439.2.2 mobileNetworkCode . . . . .	545
8.440SMSEventInfo_s Struct Reference . . . . .	545
8.440.1 Detailed Description . . . . .	545
8.440.2 Field Documentation . . . . .	546
8.440.2.1 pEtwsMessageInfo . . . . .	546
8.440.2.2 pEtwsPlmnInfo . . . . .	546
8.440.2.3 pMessageModelInfo . . . . .	546
8.440.2.4 pMTMessageInfo . . . . .	546
8.440.2.5 pSMSCAddressInfo . . . . .	546
8.440.2.6 pSMSOnIMSInfo . . . . .	546
8.440.2.7 pTransferRouteMTMessageInfo . . . . .	546
8.440.2.8 smsEventType . . . . .	546
8.441smsMaxStorageSizeReq Struct Reference . . . . .	546
8.441.1 Detailed Description . . . . .	546
8.441.2 Field Documentation . . . . .	547
8.441.2.1 pMessageMode . . . . .	547
8.441.2.2 storageType . . . . .	547
8.442smsMaxStorageSizeResp Struct Reference . . . . .	547
8.442.1 Detailed Description . . . . .	547
8.442.2 Field Documentation . . . . .	547
8.442.2.1 freeSlots . . . . .	547
8.442.2.2 maxStorageSize . . . . .	547
8.443SMSMemoryInfo Struct Reference . . . . .	548
8.443.1 Detailed Description . . . . .	548
8.443.2 Field Documentation . . . . .	548
8.443.2.1 messageMode . . . . .	548
8.443.2.2 storageType . . . . .	548
8.444SMSMessageMode Struct Reference . . . . .	548
8.444.1 Detailed Description . . . . .	548
8.444.2 Field Documentation . . . . .	548

8.444.2.1 messageMode . . . . .	548
8.445smsMsgprotocolResp Struct Reference . . . . .	548
8.445.1 Detailed Description . . . . .	549
8.445.2 Field Documentation . . . . .	549
8.445.2.1 msgProtocol . . . . .	549
8.446SMSMTMessage Struct Reference . . . . .	549
8.446.1 Detailed Description . . . . .	549
8.446.2 Field Documentation . . . . .	549
8.446.2.1 messageIndex . . . . .	549
8.446.2.2 storageType . . . . .	549
8.447SMSOnIMS Struct Reference . . . . .	549
8.447.1 Detailed Description . . . . .	550
8.447.2 Field Documentation . . . . .	551
8.447.2.1 smsOnIMS . . . . .	551
8.448smsRouteEntry Struct Reference . . . . .	551
8.448.1 Detailed Description . . . . .	551
8.448.2 Field Documentation . . . . .	552
8.448.2.1 messageClass . . . . .	552
8.448.2.2 messageType . . . . .	552
8.448.2.3 receiptAction . . . . .	552
8.448.2.4 routeStorage . . . . .	552
8.449smsSetRoutesReq Struct Reference . . . . .	553
8.449.1 Detailed Description . . . . .	553
8.449.2 Field Documentation . . . . .	553
8.449.2.1 numOfRoutes . . . . .	553
8.449.2.2 pTransferStatusReport . . . . .	553
8.449.2.3 routeList . . . . .	553
8.450SMSTransferRouteMTMessage Struct Reference . . . . .	553
8.450.1 Detailed Description . . . . .	553
8.450.2 Field Documentation . . . . .	554
8.450.2.1 ackIndicator . . . . .	554
8.450.2.2 data . . . . .	554
8.450.2.3 format . . . . .	554
8.450.2.4 length . . . . .	554
8.450.2.5 transactionID . . . . .	554
8.451sQosFlowStat Struct Reference . . . . .	554
8.451.1 Detailed Description . . . . .	554
8.451.2 Field Documentation . . . . .	555
8.451.2.1 bearerId . . . . .	555
8.451.2.2 tx_bytes . . . . .	555

8.451.2.3 tx_bytes_drp . . . . .	555
8.451.2.4 tx_pkt . . . . .	555
8.451.2.5 tx_pkt_drp . . . . .	555
8.452sQosStat Struct Reference . . . . .	555
8.452.1 Detailed Description . . . . .	555
8.452.2 Field Documentation . . . . .	556
8.452.2.1 apnId . . . . .	556
8.452.2.2 numQosFlow . . . . .	556
8.452.2.3 qosFlow . . . . .	556
8.452.2.4 total_rx_bytes . . . . .	556
8.452.2.5 total_rx_pkt . . . . .	556
8.452.2.6 total_tx_bytes . . . . .	556
8.452.2.7 total_tx_bytes_drp . . . . .	556
8.452.2.8 total_tx_pkt . . . . .	556
8.452.2.9 total_tx_pkt_drp . . . . .	556
8.453SrvStatusInfo Struct Reference . . . . .	557
8.453.1 Detailed Description . . . . .	557
8.453.2 Field Documentation . . . . .	557
8.453.2.1 isPrefDataPath . . . . .	557
8.453.2.2 srvStatus . . . . .	557
8.454ssdatasession_params Struct Reference . . . . .	557
8.454.1 Detailed Description . . . . .	558
8.454.2 Field Documentation . . . . .	559
8.454.2.1 action . . . . .	559
8.454.2.2 failureReason . . . . .	560
8.454.2.3 failureReasonv4 . . . . .	560
8.454.2.4 failureReasonv6 . . . . .	560
8.454.2.5 instanceId . . . . .	560
8.454.2.6 ipfamily . . . . .	560
8.454.2.7 pAuthentication . . . . .	560
8.454.2.8 pPassword . . . . .	560
8.454.2.9 pProfileId3GPP . . . . .	560
8.454.2.10pProfileId3GPP2 . . . . .	560
8.454.2.11pTechnology . . . . .	560
8.454.2.12pUsername . . . . .	560
8.454.2.13rcv4 . . . . .	560
8.454.2.14rcv6 . . . . .	560
8.454.2.15sessionId . . . . .	560
8.454.2.16v4sessionId . . . . .	560
8.454.2.17v6sessionId . . . . .	560



8.454.2.18verbFailReason	560
8.454.2.19verbFailReasonType	560
8.455SupportedMsgList Struct Reference	560
8.455.1 Detailed Description	560
8.455.2 Field Documentation	561
8.455.2.1 supportedMsgLen	561
8.455.2.2 supportedMsgs	561
8.456SUPSInfo Struct Reference	561
8.456.1 Detailed Description	561
8.456.2 Field Documentation	562
8.456.2.1 isModByCC	562
8.456.2.2 svcType	562
8.457SV Struct Reference	562
8.457.1 Detailed Description	562
8.457.2 Field Documentation	563
8.457.2.1 id	563
8.457.2.2 mask	563
8.457.2.3 system	563
8.458SVInfo Struct Reference	563
8.458.1 Detailed Description	563
8.458.2 Field Documentation	564
8.458.2.1 len	564
8.458.2.2 pSV	564
8.459svUsedforFix_s Struct Reference	564
8.459.1 Detailed Description	564
8.459.2 Field Documentation	564
8.459.2.1 gnssSvUsedList	564
8.459.2.2 gnssSvUsedList_len	565
8.460SWI_STRUCT_CarrierImage Struct Reference	565
8.460.1 Detailed Description	565
8.460.2 Field Documentation	565
8.460.2.1 m_FwBuildId	565
8.460.2.2 m_FwImageld	566
8.460.2.3 m_nCarrierId	566
8.460.2.4 m_nFolderId	566
8.460.2.5 m_nStorage	566
8.460.2.6 m_PriBuildId	566
8.460.2.7 m_PrImageld	566
8.461swiModemStatusResp Struct Reference	566
8.461.1 Detailed Description	566

8.461.2 Field Documentation	566
8.461.2.1 commonInfo	566
8.461.2.2 pLTEInfo	566
8.462SwiOTAMsg_s Struct Reference	566
8.462.1 Detailed Description	566
8.462.2 Field Documentation	567
8.462.2.1 data	567
8.462.2.2 data_len	567
8.462.2.3 pLteNasRelInfo	567
8.462.2.4 pTime	567
8.462.2.5 type	567
8.463swiPDPRuntimeSettingsReq Struct Reference	567
8.463.1 Detailed Description	567
8.463.2 Field Documentation	568
8.463.2.1 contextId	568
8.463.2.2 contextType	568
8.464swiPDPRuntimeSettingsResp Struct Reference	568
8.464.1 Detailed Description	568
8.464.2 Field Documentation	570
8.464.2.1 pAPNName	570
8.464.2.2 pBearerId	570
8.464.2.3 pContextId	570
8.464.2.4 pIPv4Address	570
8.464.2.5 pIPv4GWAddress	570
8.464.2.6 pIPv6Address	570
8.464.2.7 pIPv6GWAddress	570
8.464.2.8 pPrDNSIPv4Address	570
8.464.2.9 pPrDNSIPv6Address	571
8.464.2.10pPrPCSCFIPv4Address	571
8.464.2.11pPrPCSCFIPv6Address	571
8.464.2.12pSeDNSIPv4Address	571
8.464.2.13pSeDNSIPv6Address	571
8.464.2.14pSePCSCFIPv4Address	571
8.464.2.15pSePCSCFIPv6Address	571
8.465swiQosFilter Struct Reference	571
8.465.1 Detailed Description	571
8.465.2 Field Documentation	573
8.465.2.1 index	573
8.465.2.2 pEspSpi	573
8.465.2.3 pId	573

8.465.2.4 pIPv4DstAddr . . . . .	573
8.465.2.5 pIPv4SrcAddr . . . . .	573
8.465.2.6 pIPv6DstAddr . . . . .	573
8.465.2.7 pIPv6Label . . . . .	573
8.465.2.8 pIPv6SrcAddr . . . . .	573
8.465.2.9 pIPv6TrafCls . . . . .	573
8.465.2.10pNxtHdrProto . . . . .	573
8.465.2.11pPrecedence . . . . .	573
8.465.2.12pTCPDstPort . . . . .	573
8.465.2.13pTCPSrcPort . . . . .	574
8.465.2.14pTos . . . . .	574
8.465.2.15pTranDstPort . . . . .	574
8.465.2.16pTranSrcPort . . . . .	574
8.465.2.17pUDPDstPort . . . . .	574
8.465.2.18pUDPSrcPort . . . . .	574
8.465.2.19version . . . . .	574
8.466swiQosFlow Struct Reference . . . . .	574
8.466.1 Detailed Description . . . . .	574
8.466.2 Field Documentation . . . . .	577
8.466.2.1 index . . . . .	577
8.466.2.2 p3GPP2Pri . . . . .	577
8.466.2.3 p3GPPImCn . . . . .	577
8.466.2.4 p3GPPResResidualBER . . . . .	577
8.466.2.5 p3GPPSigInd . . . . .	577
8.466.2.6 p3GPPTraHdlPri . . . . .	577
8.466.2.7 pDataRate . . . . .	577
8.466.2.8 pJitter . . . . .	577
8.466.2.9 pLatency . . . . .	577
8.466.2.10pLteQci . . . . .	577
8.466.2.11pMaxAllowedPktSz . . . . .	577
8.466.2.12pMinPolicedPktSz . . . . .	577
8.466.2.13pPktErrRate . . . . .	577
8.466.2.14pProfileId3GPP2 . . . . .	577
8.466.2.15pTokenBucket . . . . .	577
8.466.2.16pTrafficClass . . . . .	577
8.467swiQosGranted Struct Reference . . . . .	578
8.467.1 Detailed Description . . . . .	578
8.467.2 Field Documentation . . . . .	578
8.467.2.1 pRxFlow . . . . .	578
8.467.2.2 pTxFlow . . . . .	578

8.468swiQosIds Struct Reference . . . . .	578
8.468.1 Detailed Description . . . . .	578
8.468.2 Field Documentation . . . . .	578
8.468.2.1 plds . . . . .	578
8.468.2.2 sz . . . . .	578
8.469swiQosModifyReq Struct Reference . . . . .	578
8.469.1 Detailed Description . . . . .	579
8.469.2 Field Documentation . . . . .	579
8.469.2.1 id . . . . .	579
8.469.2.2 pRxFilter . . . . .	579
8.469.2.3 pRxFlow . . . . .	579
8.469.2.4 pTxFilter . . . . .	579
8.469.2.5 pTxFlow . . . . .	579
8.470swiQosReq Struct Reference . . . . .	579
8.470.1 Detailed Description . . . . .	579
8.470.2 Field Documentation . . . . .	580
8.470.2.1 index . . . . .	580
8.470.2.2 pRxFilter . . . . .	580
8.470.2.3 pRxFlow . . . . .	580
8.470.2.4 pTxFilter . . . . .	580
8.470.2.5 pTxFlow . . . . .	580
8.471swiRMTrasnferStaticsReq Struct Reference . . . . .	580
8.471.1 Detailed Description . . . . .	580
8.471.2 Field Documentation . . . . .	581
8.471.2.1 bResetStatistics . . . . .	581
8.471.2.2 ulMask . . . . .	581
8.472sysInfoCommon Struct Reference . . . . .	581
8.472.1 Detailed Description . . . . .	581
8.472.2 Field Documentation . . . . .	583
8.472.2.1 isSysForbidden . . . . .	583
8.472.2.2 isSysForbiddenValid . . . . .	583
8.472.2.3 roamStatus . . . . .	583
8.472.2.4 roamStatusValid . . . . .	583
8.472.2.5 srvCapability . . . . .	583
8.472.2.6 srvCapabilityValid . . . . .	583
8.472.2.7 srvDomain . . . . .	583
8.472.2.8 srvDomainValid . . . . .	583
8.473TDSCDMAECIOThresh Struct Reference . . . . .	583
8.473.1 Detailed Description . . . . .	583
8.473.2 Field Documentation . . . . .	584

8.473.2.1 pTDSCDMAECIOThreshList . . . . .	584
8.473.2.2 TDSCDMAECIOThreshListLen . . . . .	584
8.474TDSCDMARSCPThresh Struct Reference . . . . .	584
8.474.1 Detailed Description . . . . .	584
8.474.2 Field Documentation . . . . .	584
8.474.2.1 pTDSCDMARSCPThreshList . . . . .	584
8.474.2.2 TDSCDMARSCPThreshListLen . . . . .	584
8.475TDSCDMARSSIThresh Struct Reference . . . . .	584
8.475.1 Detailed Description . . . . .	584
8.475.2 Field Documentation . . . . .	585
8.475.2.1 pTDSCDMARSSIThreshList . . . . .	585
8.475.2.2 TDSCDMARSSIThreshListLen . . . . .	585
8.476TDSCDMASigInfoExt Struct Reference . . . . .	585
8.476.1 Detailed Description . . . . .	585
8.476.2 Field Documentation . . . . .	585
8.476.2.1 ecio . . . . .	585
8.476.2.2 rscp . . . . .	585
8.476.2.3 rssi . . . . .	585
8.476.2.4 sinr . . . . .	586
8.477TDSCDMASINRThresh Struct Reference . . . . .	586
8.477.1 Detailed Description . . . . .	586
8.477.2 Field Documentation . . . . .	586
8.477.2.1 pTDSCDMASINRThreshList . . . . .	586
8.477.2.2 TDSCDMASINRThreshListLen . . . . .	586
8.478TFTIDParams Struct Reference . . . . .	586
8.478.1 Detailed Description . . . . .	586
8.478.2 Field Documentation . . . . .	587
8.478.2.1 destPortRangeEnd . . . . .	588
8.478.2.2 destPortRangeStart . . . . .	588
8.478.2.3 eValid . . . . .	588
8.478.2.4 filterId . . . . .	588
8.478.2.5 flowLabel . . . . .	588
8.478.2.6 IPSECSPi . . . . .	588
8.478.2.7 ipVersion . . . . .	588
8.478.2.8 nextHeader . . . . .	588
8.478.2.9 pSourceIP . . . . .	588
8.478.2.10sourceIPMask . . . . .	588
8.478.2.11srcPortRangeEnd . . . . .	588
8.478.2.12srcPortRangeStart . . . . .	588
8.478.2.13osMask . . . . .	588

8.479tokenBucket Struct Reference . . . . .	588
8.479.1 Detailed Description . . . . .	588
8.479.2 Field Documentation . . . . .	588
8.479.2.1 bucketSz . . . . .	588
8.479.2.2 peakRate . . . . .	588
8.479.2.3 tokenRate . . . . .	588
8.480Tos Struct Reference . . . . .	589
8.480.1 Detailed Description . . . . .	589
8.480.2 Field Documentation . . . . .	589
8.480.2.1 mask . . . . .	589
8.480.2.2 val . . . . .	589
8.481TransferStatInd Struct Reference . . . . .	589
8.481.1 Detailed Description . . . . .	589
8.481.2 Field Documentation . . . . .	590
8.481.2.1 StatsMask . . . . .	590
8.481.2.2 StatsPeriod . . . . .	590
8.482TransferStatsDataType Struct Reference . . . . .	590
8.482.1 Field Documentation . . . . .	590
8.482.1.1 interval . . . . .	590
8.483TrStatInd Struct Reference . . . . .	590
8.483.1 Detailed Description . . . . .	590
8.483.2 Field Documentation . . . . .	591
8.483.2.1 statsMask . . . . .	591
8.483.2.2 statsPeriod . . . . .	591
8.484trueIMSI Struct Reference . . . . .	591
8.484.1 Detailed Description . . . . .	591
8.484.2 Field Documentation . . . . .	592
8.484.2.1 imsiT1112 . . . . .	592
8.484.2.2 imsiTaddrNum . . . . .	592
8.484.2.3 imsiTS1 . . . . .	592
8.484.2.4 imsiTS2 . . . . .	592
8.484.2.5 mccT . . . . .	592
8.485TXAGCList Struct Reference . . . . .	592
8.485.1 Detailed Description . . . . .	592
8.485.2 Field Documentation . . . . .	593
8.485.2.1 pTXAIG . . . . .	593
8.485.2.2 pTXComprSlope . . . . .	593
8.485.2.3 pTXComprThres . . . . .	593
8.485.2.4 pTXExpSlope . . . . .	593
8.485.2.5 pTXExpThres . . . . .	593

8.485.2.6 pTXStaticGain . . . . .	593
8.486txInfo Struct Reference . . . . .	593
8.486.1 Detailed Description . . . . .	593
8.486.2 Field Documentation . . . . .	594
8.486.2.1 isInTraffic . . . . .	594
8.486.2.2 txPower . . . . .	594
8.487TXPCMIIRFiltr Struct Reference . . . . .	594
8.487.1 Detailed Description . . . . .	594
8.487.2 Field Documentation . . . . .	596
8.487.2.1 pFlag . . . . .	596
8.487.2.2 pStage0Val . . . . .	596
8.487.2.3 pStage1Val . . . . .	596
8.487.2.4 pStage2Val . . . . .	596
8.487.2.5 pStage3Val . . . . .	596
8.487.2.6 pStage4Val . . . . .	596
8.487.2.7 pStageCnt . . . . .	596
8.488UIMAuthenticateReq Struct Reference . . . . .	596
8.488.1 Detailed Description . . . . .	596
8.488.2 Field Documentation . . . . .	597
8.488.2.1 authData . . . . .	597
8.488.2.2 pIndicationToken . . . . .	597
8.488.2.3 sessionInfo . . . . .	597
8.489UIMAuthenticateResp Struct Reference . . . . .	597
8.489.1 Detailed Description . . . . .	597
8.489.2 Field Documentation . . . . .	598
8.489.2.1 pAuthenticateResult . . . . .	598
8.489.2.2 pCardResult . . . . .	598
8.489.2.3 pIndicationToken . . . . .	598
8.490UIMChangePinReq Struct Reference . . . . .	598
8.490.1 Detailed Description . . . . .	598
8.490.2 Field Documentation . . . . .	598
8.490.2.1 changePIN . . . . .	598
8.490.2.2 pIndicationToken . . . . .	599
8.490.2.3 pKeyReferenceID . . . . .	599
8.490.2.4 sessionInfo . . . . .	599
8.491UIMDepersonalizationReq Struct Reference . . . . .	599
8.491.1 Detailed Description . . . . .	599
8.491.2 Field Documentation . . . . .	599
8.491.2.1 depersonilisationInfo . . . . .	599
8.492UIMDepersonalizationResp Struct Reference . . . . .	599

8.492.1 Detailed Description . . . . .	599
8.492.2 Field Documentation . . . . .	599
8.492.2.1 pRemainingRetries . . . . .	599
8.493UIMEventRegisterReqResp Struct Reference . . . . .	599
8.493.1 Detailed Description . . . . .	600
8.493.2 Field Documentation . . . . .	600
8.493.2.1 eventMask . . . . .	600
8.494UIMGetCardStatusResp Struct Reference . . . . .	600
8.494.1 Detailed Description . . . . .	600
8.494.2 Field Documentation . . . . .	600
8.494.2.1 pCardStatus . . . . .	600
8.494.2.2 pHotSwapStatus . . . . .	600
8.495UIMGetFileAttributesReq Struct Reference . . . . .	601
8.495.1 Detailed Description . . . . .	601
8.495.2 Field Documentation . . . . .	601
8.495.2.1 fileIndex . . . . .	601
8.495.2.2 pIndicationToken . . . . .	601
8.495.2.3 sessionInfo . . . . .	601
8.496UIMGetFileAttributesResp Struct Reference . . . . .	601
8.496.1 Detailed Description . . . . .	601
8.496.2 Field Documentation . . . . .	602
8.496.2.1 pCardResult . . . . .	602
8.496.2.2 pFileAttributes . . . . .	602
8.496.2.3 pIndicationToken . . . . .	602
8.497UIMPinResp Struct Reference . . . . .	602
8.497.1 Detailed Description . . . . .	602
8.497.2 Field Documentation . . . . .	603
8.497.2.1 pEncryptedPIN1 . . . . .	603
8.497.2.2 pIndicationToken . . . . .	603
8.497.2.3 pRemainingRetries . . . . .	603
8.498UIMPowerDownReq Struct Reference . . . . .	603
8.498.1 Detailed Description . . . . .	603
8.498.2 Field Documentation . . . . .	603
8.498.2.1 slot . . . . .	603
8.499UIMPowerUpReq Struct Reference . . . . .	603
8.499.1 Detailed Description . . . . .	603
8.499.2 Field Documentation . . . . .	604
8.499.2.1 pIgnoreHotSwapSwitch . . . . .	604
8.499.2.2 slot . . . . .	604
8.500UIMReadTransparentReq Struct Reference . . . . .	604



8.500.1 Detailed Description . . . . .	604
8.500.2 Field Documentation . . . . .	605
8.500.2.1 fileIndex . . . . .	605
8.500.2.2 pEncryptData . . . . .	605
8.500.2.3 pIndicationToken . . . . .	605
8.500.2.4 readTransparent . . . . .	605
8.500.2.5 sessionInfo . . . . .	605
8.501 UIMReadTransparentResp Struct Reference . . . . .	605
8.501.1 Detailed Description . . . . .	605
8.501.2 Field Documentation . . . . .	606
8.501.2.1 pCardResult . . . . .	606
8.501.2.2 pEncryptedData . . . . .	606
8.501.2.3 pIndicationToken . . . . .	606
8.501.2.4 pReadResult . . . . .	606
8.502 UIMRefreshCompleteReq Struct Reference . . . . .	606
8.502.1 Detailed Description . . . . .	606
8.502.2 Field Documentation . . . . .	607
8.502.2.1 refreshComplete . . . . .	607
8.502.2.2 sessionInfo . . . . .	607
8.503 UIMRefreshEvent Struct Reference . . . . .	607
8.503.1 Detailed Description . . . . .	607
8.503.2 Field Documentation . . . . .	609
8.503.2.1 aid . . . . .	609
8.503.2.2 aidLength . . . . .	609
8.503.2.3 arrfileInfo . . . . .	609
8.503.2.4 mode . . . . .	609
8.503.2.5 numOfFiles . . . . .	609
8.503.2.6 sessionType . . . . .	609
8.503.2.7 stage . . . . .	609
8.504 UIMRefreshGetLastEventReq Struct Reference . . . . .	609
8.504.1 Detailed Description . . . . .	609
8.504.2 Field Documentation . . . . .	609
8.504.2.1 sessionInfo . . . . .	609
8.505 UIMRefreshGetLastEventResp Struct Reference . . . . .	609
8.505.1 Detailed Description . . . . .	609
8.505.2 Field Documentation . . . . .	610
8.505.2.1 pRefreshEvent . . . . .	610
8.506 UIMRefreshOKReq Struct Reference . . . . .	610
8.506.1 Detailed Description . . . . .	610
8.506.2 Field Documentation . . . . .	610

8.506.2.1 OKtoRefresh . . . . .	610
8.506.2.2 sessionInfo . . . . .	610
8.507UIMRefreshRegisterReq Struct Reference . . . . .	610
8.507.1 Detailed Description . . . . .	610
8.507.2 Field Documentation . . . . .	611
8.507.2.1 regRefresh . . . . .	611
8.507.2.2 sessionInfo . . . . .	611
8.508UIMSessionInformation Struct Reference . . . . .	611
8.508.1 Detailed Description . . . . .	611
8.508.2 Field Documentation . . . . .	612
8.508.2.1 aid . . . . .	612
8.508.2.2 aidLength . . . . .	612
8.508.2.3 sessionType . . . . .	612
8.509UIMSetPinProtectionReq Struct Reference . . . . .	612
8.509.1 Detailed Description . . . . .	612
8.509.2 Field Documentation . . . . .	613
8.509.2.1 pIndicationToken . . . . .	613
8.509.2.2 pinProtection . . . . .	613
8.509.2.3 pKeyReferenceID . . . . .	613
8.509.2.4 sessionInfo . . . . .	613
8.510UIMStatusChangeInfo Struct Reference . . . . .	613
8.510.1 Detailed Description . . . . .	613
8.510.2 Field Documentation . . . . .	613
8.510.2.1 statusChange . . . . .	613
8.511UIMUnblockPinReq Struct Reference . . . . .	613
8.511.1 Detailed Description . . . . .	613
8.511.2 Field Documentation . . . . .	614
8.511.2.1 pIndicationToken . . . . .	614
8.511.2.2 pKeyReferenceID . . . . .	614
8.511.2.3 sessionInfo . . . . .	614
8.511.2.4 unblockPIN . . . . .	614
8.512UIMVerifyPinReq Struct Reference . . . . .	614
8.512.1 Detailed Description . . . . .	614
8.512.2 Field Documentation . . . . .	615
8.512.2.1 pEncryptedPIN1 . . . . .	615
8.512.2.2 pIndicationToken . . . . .	615
8.512.2.3 pKeyReferenceID . . . . .	615
8.512.2.4 sessionInfo . . . . .	615
8.512.2.5 verifyPIN . . . . .	615
8.513UMTSInfo Struct Reference . . . . .	615

8.513.1 Detailed Description	616
8.513.2 Field Documentation	617
8.513.2.1 cellID	617
8.513.2.2 ecio	617
8.513.2.3 geranInst	617
8.513.2.4 GeranInstInfo	617
8.513.2.5 lac	617
8.513.2.6 plmn	617
8.513.2.7 psc	617
8.513.2.8 rscp	617
8.513.2.9 uarfcn	617
8.513.2.10 umtsInst	617
8.513.2.11 UMTSInstInfo	617
8.514 UMTSinstInfo Struct Reference	617
8.514.1 Detailed Description	617
8.514.2 Field Documentation	618
8.514.2.1 umtsEcio	618
8.514.2.2 umtsPsc	618
8.514.2.3 umtsRscp	618
8.514.2.4 umtsUarfcn	618
8.515 umtsLTENbrCell Struct Reference	618
8.515.1 Detailed Description	618
8.515.2 Field Documentation	619
8.515.2.1 cellsTDD	619
8.515.2.2 earfcn	619
8.515.2.3 pci	619
8.515.2.4 rsrp	619
8.515.2.5 rsrq	619
8.515.2.6 srxlev	619
8.516 UMTSMinQoS Struct Reference	619
8.516.1 Detailed Description	619
8.516.2 Field Documentation	622
8.516.2.1 deliveryErrSDU	622
8.516.2.2 grntDownlinkBitrate	622
8.516.2.3 grntUplinkBitrate	622
8.516.2.4 maxDownlinkBitrate	622
8.516.2.5 maxSDUSize	622
8.516.2.6 maxUplinkBitrate	622
8.516.2.7 qosDeliveryOrder	622
8.516.2.8 resBerRatio	622

8.516.2.9 sduErrorRatio . . . . .	622
8.516.2.10 trafficClass . . . . .	622
8.516.2.11 trafficPriority . . . . .	622
8.516.2.12 transferDelay . . . . .	622
8.517 UMTSQoS Struct Reference . . . . .	622
8.517.1 Detailed Description . . . . .	623
8.517.2 Field Documentation . . . . .	625
8.517.2.1 deliveryErrSDU . . . . .	625
8.517.2.2 grntDownlinkBitrate . . . . .	625
8.517.2.3 grntUplinkBitrate . . . . .	625
8.517.2.4 maxDownlinkBitrate . . . . .	625
8.517.2.5 maxSDUSize . . . . .	625
8.517.2.6 maxUplinkBitrate . . . . .	625
8.517.2.7 qosDeliveryOrder . . . . .	625
8.517.2.8 resBerRatio . . . . .	625
8.517.2.9 sduErrorRatio . . . . .	625
8.517.2.10 trafficClass . . . . .	625
8.517.2.11 trafficPriority . . . . .	625
8.517.2.12 transferDelay . . . . .	625
8.518 UMTSReqQoS SigInd Struct Reference . . . . .	626
8.518.1 Detailed Description . . . . .	626
8.518.2 Field Documentation . . . . .	626
8.518.2.1 SigInd . . . . .	626
8.518.2.2 UMTSReqQoS . . . . .	626
8.519 unblockUIMPIN Struct Reference . . . . .	626
8.519.1 Detailed Description . . . . .	626
8.519.2 Field Documentation . . . . .	627
8.519.2.1 newPINLen . . . . .	627
8.519.2.2 newPINVal . . . . .	627
8.519.2.3 pinID . . . . .	627
8.519.2.4 pukLen . . . . .	627
8.519.2.5 pukVal . . . . .	627
8.520 UniversalTime Struct Reference . . . . .	627
8.520.1 Detailed Description . . . . .	627
8.520.2 Field Documentation . . . . .	628
8.520.2.1 day . . . . .	628
8.520.2.2 dayOfWeek . . . . .	628
8.520.2.3 hour . . . . .	628
8.520.2.4 minute . . . . .	628
8.520.2.5 month . . . . .	628

8.520.2.6 second	628
8.520.2.7 year	628
8.521 USBCompConfig Struct Reference	629
8.521.1 Detailed Description	629
8.521.2 Field Documentation	629
8.521.2.1 pUSBComp	629
8.522 USBCompParams Struct Reference	629
8.522.1 Detailed Description	630
8.522.2 Field Documentation	632
8.522.2.1 pNumSupUSBComps	632
8.522.2.2 pSupUSBComps	632
8.522.2.3 pUSBComp	632
8.523 USSDNoWaitIndicationInfo Struct Reference	632
8.523.1 Detailed Description	632
8.523.2 Field Documentation	632
8.523.2.1 pAlphaIdentifier	632
8.523.2.2 pError	632
8.523.2.3 pFailureCause	632
8.523.2.4 pUSSDData	632
8.524 USSDRespFNetwork Struct Reference	632
8.524.1 Detailed Description	633
8.524.2 Field Documentation	634
8.524.2.1 pRespData	634
8.524.2.2 pTypeCode	634
8.525 USSInfo Struct Reference	634
8.525.1 Detailed Description	634
8.525.2 Field Documentation	634
8.525.2.1 ussData	634
8.525.2.2 ussDCS	634
8.525.2.3 ussLen	634
8.526 USSResp Struct Reference	634
8.526.1 Field Documentation	635
8.526.1.1 pAlphaIDInfo	635
8.526.1.2 pCallId	635
8.526.1.3 pCcResultType	635
8.526.1.4 pCCSuppsType	635
8.526.1.5 pfailureCause	635
8.526.1.6 pUSSDInfo	635
8.527 UUSInfo Struct Reference	635
8.527.1 Detailed Description	635

8.527.2 Field Documentation . . . . .	636
8.527.2.1 UUSData . . . . .	636
8.527.2.2 UUSDatalen . . . . .	636
8.527.2.3 UUSDcs . . . . .	636
8.527.2.4 UUSType . . . . .	636
8.528verifyUIMPIN Struct Reference . . . . .	636
8.528.1 Detailed Description . . . . .	637
8.528.2 Field Documentation . . . . .	637
8.528.2.1 pinID . . . . .	637
8.528.2.2 pinLen . . . . .	637
8.528.2.3 pinVal . . . . .	637
8.529voiceALSSelectLineInfo Struct Reference . . . . .	637
8.529.1 Detailed Description . . . . .	637
8.529.2 Field Documentation . . . . .	638
8.529.2.1 lineValue . . . . .	638
8.530voiceALSSetLineSwitchInfo Struct Reference . . . . .	638
8.530.1 Detailed Description . . . . .	638
8.530.2 Field Documentation . . . . .	638
8.530.2.1 switchOption . . . . .	638
8.531voiceAnswerCall Struct Reference . . . . .	638
8.531.1 Detailed Description . . . . .	638
8.531.2 Field Documentation . . . . .	639
8.531.2.1 pCallId . . . . .	639
8.532voiceBindSubscriptionInfo Struct Reference . . . . .	639
8.532.1 Detailed Description . . . . .	639
8.532.2 Field Documentation . . . . .	639
8.532.2.1 subType . . . . .	639
8.533voiceBurstDTMFInfo Struct Reference . . . . .	639
8.533.1 Detailed Description . . . . .	639
8.533.2 Field Documentation . . . . .	640
8.533.2.1 BurstDTMFInfo . . . . .	640
8.533.2.2 pBurstDTMFLengths . . . . .	640
8.534voiceCallInfoReq Struct Reference . . . . .	640
8.534.1 Detailed Description . . . . .	640
8.534.2 Field Documentation . . . . .	640
8.534.2.1 callID . . . . .	640
8.535voiceCallInfoResp Struct Reference . . . . .	640
8.535.1 Detailed Description . . . . .	641
8.535.2 Field Documentation . . . . .	643
8.535.2.1 pAlertingPattern . . . . .	643

8.535.2.2 pAlertType . . . . .	643
8.535.2.3 pAlphaIDInfo . . . . .	643
8.535.2.4 pCallInfo . . . . .	643
8.535.2.5 pConnectNumInfo . . . . .	643
8.535.2.6 pDiagInfo . . . . .	644
8.535.2.7 pOTASPStatus . . . . .	644
8.535.2.8 pRemotePartyName . . . . .	644
8.535.2.9 pRemotePartyNum . . . . .	644
8.535.2.10 pSrvOpt . . . . .	644
8.535.2.11 pUUSInfo . . . . .	644
8.535.2.12 pVoicePrivacy . . . . .	644
8.536 voiceCallRequestParams Struct Reference . . . . .	644
8.536.1 Detailed Description . . . . .	644
8.536.2 Field Documentation . . . . .	646
8.536.2.1 callNumber . . . . .	646
8.536.2.2 pCallPartySubAdd . . . . .	646
8.536.2.3 pCallType . . . . .	646
8.536.2.4 pCLIRType . . . . .	646
8.536.2.5 pCUGInfo . . . . .	646
8.536.2.6 pEmergencyCategory . . . . .	646
8.536.2.7 pSvcType . . . . .	646
8.536.2.8 pUUSInfo . . . . .	646
8.537 voiceCallResponseParams Struct Reference . . . . .	646
8.537.1 Detailed Description . . . . .	646
8.537.2 Field Documentation . . . . .	647
8.537.2.1 pAlphaIDInfo . . . . .	647
8.537.2.2 pCallID . . . . .	647
8.537.2.3 pCCResultType . . . . .	647
8.537.2.4 pCCSUPSType . . . . .	647
8.538 voiceContDTMFInfo Struct Reference . . . . .	647
8.538.1 Detailed Description . . . . .	647
8.538.2 Field Documentation . . . . .	648
8.538.2.1 DTMFdigit . . . . .	648
8.538.2.2 pCallID . . . . .	648
8.539 voiceDTMFEventInfo Struct Reference . . . . .	648
8.539.1 Detailed Description . . . . .	648
8.539.2 Field Documentation . . . . .	649
8.539.2.1 DTMFInformation . . . . .	649
8.539.2.2 pOffLength . . . . .	649
8.539.2.3 pOnLength . . . . .	649

8.540voiceFlashInfo Struct Reference . . . . .	649
8.540.1 Detailed Description . . . . .	649
8.540.2 Field Documentation . . . . .	650
8.540.2.1 pCallID . . . . .	650
8.540.2.2 pFlashPayLd . . . . .	650
8.540.2.3 pFlashType . . . . .	650
8.541voiceGetAllCallInfo Struct Reference . . . . .	650
8.541.1 Detailed Description . . . . .	650
8.541.2 Field Documentation . . . . .	652
8.541.2.1 pArrAlertingPattern . . . . .	652
8.541.2.2 pArrAlertingType . . . . .	652
8.541.2.3 pArrAlphaID . . . . .	652
8.541.2.4 pArrCalledPartyNum . . . . .	652
8.541.2.5 pArrCallEndReason . . . . .	652
8.541.2.6 pArrCallInfo . . . . .	652
8.541.2.7 pArrConnectPartyNum . . . . .	652
8.541.2.8 pArrDiagInfo . . . . .	652
8.541.2.9 pArrRedirPartyNum . . . . .	652
8.541.2.10pArrRemotePartyName . . . . .	652
8.541.2.11pArrRemotePartyNum . . . . .	653
8.541.2.12pArrSvcOption . . . . .	653
8.541.2.13pArrUUSInfo . . . . .	653
8.541.2.14pOTASPStatus . . . . .	653
8.541.2.15pVoicePrivacy . . . . .	653
8.542voiceGetCallBarringReq Struct Reference . . . . .	653
8.542.1 Detailed Description . . . . .	653
8.542.2 Field Documentation . . . . .	654
8.542.2.1 pSvcClass . . . . .	654
8.542.2.2 reason . . . . .	654
8.543voiceGetCallBarringResp Struct Reference . . . . .	654
8.543.1 Detailed Description . . . . .	654
8.543.2 Field Documentation . . . . .	655
8.543.2.1 pAlphaIDInfo . . . . .	655
8.543.2.2 pCallID . . . . .	655
8.543.2.3 pCCRResType . . . . .	655
8.543.2.4 pCCSUPSType . . . . .	655
8.543.2.5 pFailCause . . . . .	655
8.543.2.6 pSvcClass . . . . .	655
8.544voiceGetCallFWReq Struct Reference . . . . .	655
8.544.1 Detailed Description . . . . .	656



8.544.2 Field Documentation . . . . .	657
8.544.2.1 pSvcClass . . . . .	657
8.544.2.2 Reason . . . . .	657
8.545voiceGetCallFWResp Struct Reference . . . . .	657
8.545.1 Detailed Description . . . . .	657
8.545.2 Field Documentation . . . . .	658
8.545.2.1 pAlphaIDInfo . . . . .	658
8.545.2.2 pCallID . . . . .	659
8.545.2.3 pCCResType . . . . .	659
8.545.2.4 pCCSUPSType . . . . .	659
8.545.2.5 pFailCause . . . . .	659
8.545.2.6 pGetCallFWExtInfo . . . . .	659
8.545.2.7 pGetCallFWInfo . . . . .	659
8.546voiceGetCallWaitInfo Struct Reference . . . . .	659
8.546.1 Detailed Description . . . . .	659
8.546.2 Field Documentation . . . . .	660
8.546.2.1 pAlphaIDInfo . . . . .	660
8.546.2.2 pCallID . . . . .	660
8.546.2.3 pCCResType . . . . .	660
8.546.2.4 pCCSUPSType . . . . .	660
8.546.2.5 pFailCause . . . . .	660
8.546.2.6 pSvcClass . . . . .	660
8.547voiceGetCLIPResp Struct Reference . . . . .	660
8.547.1 Detailed Description . . . . .	661
8.547.2 Field Documentation . . . . .	662
8.547.2.1 pAlphaIDInfo . . . . .	662
8.547.2.2 pCallID . . . . .	662
8.547.2.3 pCCResType . . . . .	662
8.547.2.4 pCCSUPSType . . . . .	663
8.547.2.5 pCLIPResp . . . . .	663
8.547.2.6 pFailCause . . . . .	663
8.548voiceGetCLIRResp Struct Reference . . . . .	663
8.548.1 Detailed Description . . . . .	663
8.548.2 Field Documentation . . . . .	664
8.548.2.1 pAlphaIDInfo . . . . .	664
8.548.2.2 pCallID . . . . .	664
8.548.2.3 pCCResType . . . . .	664
8.548.2.4 pCCSUPSType . . . . .	664
8.548.2.5 pCLIRResp . . . . .	664
8.548.2.6 pFailCause . . . . .	664

8.549voiceGetCNAPResp Struct Reference . . . . .	664
8.549.1 Detailed Description . . . . .	664
8.549.2 Field Documentation . . . . .	665
8.549.2.1 pAlphaIDInfo . . . . .	665
8.549.2.2 pCallID . . . . .	665
8.549.2.3 pCCResType . . . . .	665
8.549.2.4 pCCSUPSType . . . . .	666
8.549.2.5 pCNAPResp . . . . .	666
8.549.2.6 pFailCause . . . . .	666
8.550voiceGetCOLPResp Struct Reference . . . . .	666
8.550.1 Detailed Description . . . . .	666
8.550.2 Field Documentation . . . . .	667
8.550.2.1 pAlphaIDInfo . . . . .	667
8.550.2.2 pCallID . . . . .	667
8.550.2.3 pCCResType . . . . .	667
8.550.2.4 pCCSUPSType . . . . .	667
8.550.2.5 pCOLPResp . . . . .	667
8.550.2.6 pFailCause . . . . .	667
8.551voiceGetCOLRResp Struct Reference . . . . .	667
8.551.1 Detailed Description . . . . .	667
8.551.2 Field Documentation . . . . .	668
8.551.2.1 pAlphaIDInfo . . . . .	668
8.551.2.2 pCallID . . . . .	668
8.551.2.3 pCCResType . . . . .	668
8.551.2.4 pCCSUPSType . . . . .	669
8.551.2.5 pCOLRResp . . . . .	669
8.551.2.6 pFailCause . . . . .	669
8.552voiceGetConfigReq Struct Reference . . . . .	669
8.552.1 Detailed Description . . . . .	669
8.552.2 Field Documentation . . . . .	670
8.552.2.1 pAirTimer . . . . .	670
8.552.2.2 pAMRStatus . . . . .	670
8.552.2.3 pAutoAnswer . . . . .	670
8.552.2.4 pNamID . . . . .	670
8.552.2.5 pPrefVoicePrivacy . . . . .	670
8.552.2.6 pPrefVoiceSO . . . . .	670
8.552.2.7 pRoamTimer . . . . .	670
8.552.2.8 pTTYMode . . . . .	671
8.552.2.9 pVoiceDomainPref . . . . .	671
8.553voiceGetConfigResp Struct Reference . . . . .	671

8.553.1 Detailed Description . . . . .	671
8.553.2 Field Documentation . . . . .	672
8.553.2.1 pAirTimerCnt . . . . .	672
8.553.2.2 pAutoAnswerStat . . . . .	673
8.553.2.3 pCurAMRConfig . . . . .	673
8.553.2.4 pCurPrefVoiceSO . . . . .	673
8.553.2.5 pCurrTTYMode . . . . .	673
8.553.2.6 pCurVoiceDomainPref . . . . .	673
8.553.2.7 pCurVoicePrivacyPref . . . . .	673
8.553.2.8 pRoamTimerCnt . . . . .	673
8.554voiceIndicationRegisterInfo Struct Reference . . . . .	673
8.554.1 Detailed Description . . . . .	673
8.554.2 Field Documentation . . . . .	674
8.554.2.1 pRegDTMFEvents . . . . .	674
8.554.2.2 pRegVoicePrivacyEvents . . . . .	674
8.554.2.3 pSuppsNotifEvents . . . . .	674
8.555voiceInfoRec Struct Reference . . . . .	674
8.555.1 Detailed Description . . . . .	674
8.555.2 Field Documentation . . . . .	676
8.555.2.1 callID . . . . .	676
8.555.2.2 pCalledPartyInfo . . . . .	676
8.555.2.3 pCallerIDInfo . . . . .	676
8.555.2.4 pCallerNameInfo . . . . .	676
8.555.2.5 pCallingPartyInfo . . . . .	676
8.555.2.6 pCallWaitInd . . . . .	676
8.555.2.7 pCLIRCause . . . . .	676
8.555.2.8 pConnectNumInfo . . . . .	676
8.555.2.9 pDispInfo . . . . .	676
8.555.2.10pExtDispInfo . . . . .	676
8.555.2.11pExtDispRecInfo . . . . .	676
8.555.2.12pLineCtrlInfo . . . . .	676
8.555.2.13pNSSAudioCtrl . . . . .	676
8.555.2.14pNSSRelease . . . . .	676
8.555.2.15pRedirNumInfo . . . . .	676
8.555.2.16pSignalInfo . . . . .	676
8.556voiceManageCallsReq Struct Reference . . . . .	676
8.556.1 Detailed Description . . . . .	677
8.556.2 Field Documentation . . . . .	678
8.556.2.1 pCallID . . . . .	678
8.556.2.2 SUPSType . . . . .	678

8.557voiceManageCallsResp Struct Reference . . . . .	678
8.557.1 Detailed Description . . . . .	678
8.557.2 Field Documentation . . . . .	678
8.557.2.1 pFailCause . . . . .	678
8.558voiceOrigUSSDNoWaitInfo Struct Reference . . . . .	678
8.558.1 Detailed Description . . . . .	678
8.558.2 Field Documentation . . . . .	679
8.558.2.1 USSInformation . . . . .	679
8.559voiceOTASPStatusInfo Struct Reference . . . . .	679
8.559.1 Detailed Description . . . . .	679
8.559.2 Field Documentation . . . . .	680
8.559.2.1 callID . . . . .	680
8.559.2.2 OTASPStatus . . . . .	680
8.560voicePrivacyInfo Struct Reference . . . . .	680
8.560.1 Detailed Description . . . . .	680
8.560.2 Field Documentation . . . . .	680
8.560.2.1 callID . . . . .	680
8.560.2.2 voicePrivacy . . . . .	680
8.561voiceSetAllCallStatusCbkJInfo Struct Reference . . . . .	680
8.561.1 Detailed Description . . . . .	681
8.561.2 Field Documentation . . . . .	682
8.561.2.1 arrCallInformation . . . . .	682
8.561.2.2 pArrAlertingPattern . . . . .	682
8.561.2.3 pArrAlertingType . . . . .	682
8.561.2.4 pArrAlphaID . . . . .	682
8.561.2.5 pArrCalledPartyNum . . . . .	682
8.561.2.6 pArrCallEndReason . . . . .	683
8.561.2.7 pArrConnectPartyNum . . . . .	683
8.561.2.8 pArrDiagInfo . . . . .	683
8.561.2.9 pArrRedirPartyNum . . . . .	683
8.561.2.10pArrRemotePartyName . . . . .	683
8.561.2.11pArrRemotePartyNum . . . . .	683
8.561.2.12pArrSvcOption . . . . .	683
8.562voiceSetCallBarringPwdInfo Struct Reference . . . . .	683
8.562.1 Detailed Description . . . . .	683
8.562.2 Field Documentation . . . . .	684
8.562.2.1 newPasswd . . . . .	684
8.562.2.2 newPasswdAgain . . . . .	684
8.562.2.3 oldPasswd . . . . .	684
8.562.2.4 Reason . . . . .	684

8.563voiceSetCallBarringPwdResp Struct Reference	684
8.563.1 Detailed Description	684
8.563.2 Field Documentation	685
8.563.2.1 pAlphaIDInfo	685
8.563.2.2 pCallID	685
8.563.2.3 pCCResType	685
8.563.2.4 pCCSUPSType	685
8.563.2.5 pFailCause	685
8.564voiceSetConfigReq Struct Reference	685
8.564.1 Detailed Description	686
8.564.2 Field Documentation	687
8.564.2.1 pAirTimerConfig	687
8.564.2.2 pAutoAnswer	687
8.564.2.3 pPrefVoiceDomain	687
8.564.2.4 pPrefVoiceSO	687
8.564.2.5 pRoamTimerConfig	687
8.564.2.6 pTTYMode	687
8.565voiceSetConfigResp Struct Reference	687
8.565.1 Detailed Description	687
8.565.2 Field Documentation	689
8.565.2.1 pAirTimerStatus	689
8.565.2.2 pAutoAnsStatus	689
8.565.2.3 pPrefVoiceSOStatus	689
8.565.2.4 pRoamTimerStatus	689
8.565.2.5 pTTYConfigStatus	689
8.565.2.6 pVoiceDomainPrefStatus	689
8.566voiceSetPrefPrivacy Struct Reference	689
8.566.1 Detailed Description	689
8.566.2 Field Documentation	690
8.566.2.1 privacyPref	690
8.567voiceSetSUPSServiceReq Struct Reference	690
8.567.1 Detailed Description	690
8.567.2 Field Documentation	692
8.567.2.1 pCallBarringPasswd	692
8.567.2.2 pCallForwardingNumber	692
8.567.2.3 pCallFwdTypeAndPlan	692
8.567.2.4 pServiceClass	692
8.567.2.5 pTimerVal	692
8.567.2.6 reason	692
8.567.2.7 voiceSvc	692

8.568voiceSetSUPSServiceResp Struct Reference	692
8.568.1 Detailed Description	693
8.568.2 Field Documentation	693
8.568.2.1 pAlphaIDInfo	693
8.568.2.2 pCallID	693
8.568.2.3 pCCResultType	693
8.568.2.4 pCCSUPSType	693
8.568.2.5 pFailCause	693
8.569voiceStopContDTMFInfo Struct Reference	694
8.569.1 Detailed Description	694
8.569.2 Field Documentation	694
8.569.2.1 callID	694
8.570voiceSUPSInfo Struct Reference	694
8.570.1 Detailed Description	694
8.570.2 Field Documentation	696
8.570.2.1 pAlphaIDInfo	696
8.570.2.2 pCallBarPasswd	696
8.570.2.3 pCallFwdInfo	696
8.570.2.4 pCallFWNum	696
8.570.2.5 pCallFWTimerVal	696
8.570.2.6 pCallID	696
8.570.2.7 pCLIPstatus	696
8.570.2.8 pCLIRstatus	696
8.570.2.9 pCNAPstatus	696
8.570.2.10pCOLPstatus	696
8.570.2.11pCOLRstatus	696
8.570.2.12pDataSrc	696
8.570.2.13pFailCause	697
8.570.2.14pNewPwdData	697
8.570.2.15pReason	697
8.570.2.16pSvcClass	697
8.570.2.17pUSSInfo	697
8.570.2.18SUPSInformation	697
8.571voiceSUPSNotification Struct Reference	697
8.571.1 Detailed Description	697
8.571.2 Field Documentation	699
8.571.2.1 callID	699
8.571.2.2 notifType	699
8.571.2.3 pCUGIndex	699
8.571.2.4 pECTNum	699

8.572wcdmaCellInfo Struct Reference	699
8.572.1 Detailed Description	699
8.572.2 Field Documentation	700
8.572.2.1 cpich_ecno	700
8.572.2.2 cpich_rscp	700
8.572.2.3 psc	700
8.572.2.4 srxlev	700
8.573WCDMAECIOThresh Struct Reference	700
8.573.1 Detailed Description	700
8.573.2 Field Documentation	700
8.573.2.1 pWCDMAECIOThreshList	700
8.573.2.2 WCDMAECIOThreshListLen	700
8.574WCDMAInfoLTENNeighborCell Struct Reference	700
8.574.1 Detailed Description	701
8.574.2 Field Documentation	701
8.574.2.1 UMTSLTENbrCell	701
8.574.2.2 umtsLTENbrCellLen	701
8.574.2.3 wcdmaRRCState	701
8.575wcdmaLongMsgDecodingParams Struct Reference	701
8.575.1 Detailed Description	702
8.575.2 Field Documentation	703
8.575.2.1 Date	703
8.575.2.2 plsUDHPresent	703
8.575.2.3 pMessage	703
8.575.2.4 pPartNum	703
8.575.2.5 pReferenceNum	703
8.575.2.6 pScAddr	703
8.575.2.7 pScAddrLength	703
8.575.2.8 pSenderAddr	703
8.575.2.9 pSenderAddrLength	703
8.575.2.10pTextMsg	703
8.575.2.11pTextMsgLength	703
8.575.2.12pTotalNum	703
8.575.2.13Time	703
8.576wcdmaMsgDecodingParams Struct Reference	703
8.576.1 Detailed Description	703
8.576.2 Field Documentation	704
8.576.2.1 Date	704
8.576.2.2 pMessage	704
8.576.2.3 pScAddr	704

8.576.2.4 pScAddrLength . . . . .	704
8.576.2.5 pSenderAddr . . . . .	704
8.576.2.6 pSenderAddrLength . . . . .	704
8.576.2.7 pTextMsg . . . . .	705
8.576.2.8 pTextMsgLength . . . . .	705
8.576.2.9 Time . . . . .	705
8.577wcdmaMsgEncodingParams Struct Reference . . . . .	705
8.577.1 Detailed Description . . . . .	705
8.577.2 Field Documentation . . . . .	705
8.577.2.1 alphabet . . . . .	705
8.577.2.2 messageSize . . . . .	705
8.577.2.3 pDestAddr . . . . .	705
8.577.2.4 pPDUMessage . . . . .	705
8.577.2.5 pTextMsg . . . . .	706
8.578WCDMARSSIThresh Struct Reference . . . . .	706
8.578.1 Detailed Description . . . . .	706
8.578.2 Field Documentation . . . . .	706
8.578.2.1 pWCDMARSSIThreshList . . . . .	706
8.578.2.2 WCDMARSSIThreshListLen . . . . .	706
8.579WCDMASysInfo Struct Reference . . . . .	706
8.579.1 Detailed Description . . . . .	707
8.579.2 Field Documentation . . . . .	710
8.579.2.1 cellId . . . . .	710
8.579.2.2 cellIdValid . . . . .	710
8.579.2.3 hsCallStatus . . . . .	710
8.579.2.4 hsCallStatusValid . . . . .	710
8.579.2.5 hsInd . . . . .	710
8.579.2.6 hsIndValid . . . . .	710
8.579.2.7 lac . . . . .	711
8.579.2.8 lacValid . . . . .	711
8.579.2.9 MCC . . . . .	711
8.579.2.10MNC . . . . .	711
8.579.2.11networkIdValid . . . . .	711
8.579.2.12psc . . . . .	711
8.579.2.13pscValid . . . . .	711
8.579.2.14regRejectInfoValid . . . . .	711
8.579.2.15rejCause . . . . .	711
8.579.2.16rejectSrvDomain . . . . .	711
8.579.2.17sysInfoWCDMA . . . . .	711
8.580WdsByteTotals Struct Reference . . . . .	711



8.580.1 Detailed Description . . . . .	711
8.580.2 Field Documentation . . . . .	712
8.580.2.1 ByteTotalsElmntsV4 . . . . .	712
8.580.2.2 ByteTotalsElmntsV6 . . . . .	712
8.580.2.3 pV4sessionId . . . . .	712
8.580.2.4 pV6sessionId . . . . .	712
8.581WdsByteTotalsElmnts Struct Reference . . . . .	712
8.581.1 Detailed Description . . . . .	712
8.581.2 Field Documentation . . . . .	712
8.581.2.1 pRXTotalBytes . . . . .	712
8.581.2.2 pTXTotalBytes . . . . .	712
8.582WdsConnectionRate Struct Reference . . . . .	712
8.582.1 Detailed Description . . . . .	712
8.582.2 Field Documentation . . . . .	713
8.582.2.1 ConnRateElmntsV4 . . . . .	713
8.582.2.2 ConnRateElmntsV6 . . . . .	713
8.582.2.3 pV4sessionId . . . . .	713
8.582.2.4 pV6sessionId . . . . .	713
8.583WdsConnectionRateElmnts Struct Reference . . . . .	713
8.583.1 Detailed Description . . . . .	713
8.583.2 Field Documentation . . . . .	714
8.583.2.1 pCurrentChannelRXRate . . . . .	714
8.583.2.2 pCurrentChannelTXRate . . . . .	714
8.583.2.3 pMaxChannelRXRate . . . . .	714
8.583.2.4 pMaxChannelTXRate . . . . .	714
8.584WDSGetLoopbackData Struct Reference . . . . .	714
8.584.1 Detailed Description . . . . .	714
8.584.2 Field Documentation . . . . .	715
8.584.2.1 ByteLoopbackMode . . . . .	715
8.584.2.2 ByteLoopbackMultiplier . . . . .	715
8.585WdsIpAddressInfoReq Struct Reference . . . . .	715
8.585.1 Field Documentation . . . . .	715
8.585.1.1 ip . . . . .	715
8.585.1.2 pv4sessionId . . . . .	715
8.585.1.3 pv6sessionId . . . . .	715
8.586WdsPktStatisticsElmnts Struct Reference . . . . .	715
8.586.1 Detailed Description . . . . .	716
8.586.2 Field Documentation . . . . .	717
8.586.2.1 pRXDroppedCount . . . . .	717
8.586.2.2 pRXOkBytesCount . . . . .	717

8.586.2.3 pRXOKBytesLastCall . . . . .	717
8.586.2.4 pRXPacketErrors . . . . .	717
8.586.2.5 pRXPacketOverflows . . . . .	718
8.586.2.6 pRXPacketSuccesses . . . . .	718
8.586.2.7 pTXDroppedCount . . . . .	718
8.586.2.8 pTXOkBytesCount . . . . .	718
8.586.2.9 pTXOKBytesLastCall . . . . .	718
8.586.2.10pTXPacketErrors . . . . .	718
8.586.2.11pTXPacketOverflows . . . . .	718
8.586.2.12pTXPacketSuccesses . . . . .	718
8.587WdsPktStatisticsReq Struct Reference . . . . .	718
8.587.1 Detailed Description . . . . .	718
8.587.2 Field Documentation . . . . .	718
8.587.2.1 pStatMask . . . . .	718
8.588WdsPktStatisticsResp Struct Reference . . . . .	718
8.588.1 Detailed Description . . . . .	718
8.588.2 Field Documentation . . . . .	719
8.588.2.1 PktStatElmntsV4 . . . . .	719
8.588.2.2 PktStatElmntsV6 . . . . .	719
8.588.2.3 pV4sessionId . . . . .	719
8.588.2.4 pV6sessionId . . . . .	719
8.589WdsProfileParam Union Reference . . . . .	719
8.589.1 Detailed Description . . . . .	719
8.589.2 Field Documentation . . . . .	719
8.589.2.1 SlqsProfile3GPP . . . . .	719
8.589.2.2 SlqsProfile3GPP2 . . . . .	719
8.590WdsRunTimeSettings Struct Reference . . . . .	719
8.590.1 Detailed Description . . . . .	720
8.590.2 Field Documentation . . . . .	720
8.590.2.1 rts . . . . .	720
8.590.2.2 v4sessionId . . . . .	720
8.590.2.3 v6sessionId . . . . .	720
8.591wdsSetEventReportReq Struct Reference . . . . .	720
8.591.1 Detailed Description . . . . .	721
8.591.2 Field Documentation . . . . .	723
8.591.2.1 pCurrChannelRateInd . . . . .	723
8.591.2.2 pCurrDataBearerTechInd . . . . .	723
8.591.2.3 pCurrPrefDataSysInd . . . . .	723
8.591.2.4 pDataBearerTechInd . . . . .	723
8.591.2.5 pDataCallStatusChangeInd . . . . .	723

8.591.2.6 pDataSystemStatusChangeInd . . . . .	723
8.591.2.7 pDormancyStatusInd . . . . .	723
8.591.2.8 pEVDOPageMonPerChangeInd . . . . .	723
8.591.2.9 pMIPStatusInd . . . . .	723
8.591.2.10 pTransferStatInd . . . . .	723
8.592 WDS SetLoopbackData Struct Reference . . . . .	723
8.592.1 Detailed Description . . . . .	724
8.592.2 Field Documentation . . . . .	725
8.592.2.1 pLoopbackMode . . . . .	725
8.592.2.2 pLoopbackMultiplier . . . . .	725
8.593 WDSSWICurrentChannelRates Struct Reference . . . . .	725
8.593.1 Detailed Description . . . . .	725
8.593.2 Field Documentation . . . . .	725
8.593.2.1 current_channel_rx_rate . . . . .	725
8.593.2.2 current_channel_tx_rate . . . . .	725
8.593.2.3 max_channel_rx_rate . . . . .	726
8.593.2.4 max_channel_tx_rate . . . . .	726
<b>9 File Documentation</b>	<b>727</b>
9.1 apdxy pages.c File Reference . . . . .	727
9.1.1 Detailed Description . . . . .	727
9.2 qaCbkCatEventReportInd.h File Reference . . . . .	727
9.2.1 Macro Definition Documentation . . . . .	728
9.2.1.1 QMI_CAN_COMMON_EVENT_TLV_NUMBER . . . . .	728
9.2.1.2 QMI_MAX_CAT_EVENT_DATA_LENGTH . . . . .	728
9.2.2 Enumeration Type Documentation . . . . .	728
9.2.2.1 eQMI_CAT_EVENT_REPORT_IND_TLV . . . . .	728
9.2.2.2 eQMI_CAT_EVENT_REPORT_IND_TLV_LENGTH . . . . .	728
9.2.3 Function Documentation . . . . .	729
9.2.3.1 UpkQmiCbkCatEventReportInd . . . . .	729
9.3 qaCbkSwiOmaDmEventReportInd.h File Reference . . . . .	729
9.3.1 Macro Definition Documentation . . . . .	729
9.3.1.1 QMI_SWIOMA_DM_CONFIG . . . . .	729
9.3.1.2 QMI_SWIOMA_DM_FOTA . . . . .	729
9.3.1.3 QMI_SWIOMA_DM_NOT . . . . .	729
9.3.2 Enumeration Type Documentation . . . . .	729
9.3.2.1 eQMI_SWIOMA_DM_EVENT_REPORT_IND . . . . .	729
9.3.3 Function Documentation . . . . .	730
9.3.3.1 UpkQmiCbkSwiOmaDmEventReportInd . . . . .	730
9.3.3.2 UpkQmiCbkSwiOmaDmEventReportIndExt . . . . .	730

9.4	qaGobiApiAudio.h File Reference	730
9.4.1	Detailed Description	730
9.4.2	Function Documentation	730
9.4.2.1	SLQSGetAudioPathConfig	730
9.4.2.2	SLQSGetAudioProfile	731
9.4.2.3	SLQSGetAudioVoTLBConfig	731
9.4.2.4	SLQSSetAudioPathConfig	732
9.4.2.5	SLQSSetAudioProfile	732
9.4.2.6	SLQSSetAudioVoTLBConfig	733
9.5	qaGobiApiCat.h File Reference	733
9.5.1	Detailed Description	734
9.5.2	Function Documentation	734
9.5.2.1	CATSendEnvelopeCommand	734
9.5.2.2	CATSendTerminalResponse	734
9.6	qaGobiApiCbK.h File Reference	735
9.6.1	Detailed Description	742
9.6.2	Macro Definition Documentation	742
9.6.2.1	CBK_DISABLE_EVENT	742
9.6.2.2	CBK_ENABLE_EVENT	742
9.6.2.3	CBK_NOCHANGE	742
9.6.2.4	DEREGISTER_EVENT	742
9.6.2.5	DEREGISTER_SRV	742
9.6.2.6	EVENT_MASK_CARD	742
9.6.2.7	EVENT_MASK_DEREGISTER_ALL	742
9.6.2.8	FIRST_INSTANCE	742
9.6.2.9	INVALID_INSTACNE	742
9.6.2.10	IPV4	742
9.6.2.11	IPV4V6	742
9.6.2.12	IPV6	742
9.6.2.13	LOC_EVENT_MASK_ENG_STATE	742
9.6.2.14	LOC_EVENT_MASK_GNSS_SV_INFO	742
9.6.2.15	LOC_EVENT_MASK_INJECT_TIME	742
9.6.2.16	LOC_EVENT_MASK_SENSOR_STREAM	742
9.6.2.17	LOC_EVENT_MASK_TIME_SYNC	742
9.6.2.18	LOC_EVENT_POSITION_REPORT	742
9.6.2.19	MAX_NO_OF_APPLICATIONS	742
9.6.2.20	MAX_NO_OF_CALLS	742
9.6.2.21	MAX_NO_OF_FILES	742
9.6.2.22	MAX_NO_OF_SLOTS	742
9.6.2.23	MAX_NO_OF_UUSINFO	743

9.6.2.24	MAX_PATH_LENGTH . . . . .	743
9.6.2.25	MAX_RADIO_INTERFACE_LIST . . . . .	743
9.6.2.26	MAXUSSDLENGTH . . . . .	743
9.6.2.27	NAS_SRV . . . . .	743
9.6.2.28	NUM_OF_SET . . . . .	743
9.6.2.29	PDS_SRV . . . . .	743
9.6.2.30	QMI_ETWS_MAX_PAYLOAD_LENGTH . . . . .	743
9.6.2.31	QMI_MAX_VOICE_NUMBER_LENGTH . . . . .	743
9.6.2.32	QMI_WMS_MAX_PAYLOAD_LENGTH . . . . .	743
9.6.2.33	REGISTER_EVENT . . . . .	743
9.6.2.34	REGISTER_SRV . . . . .	743
9.6.2.35	SECOND_INSTANCE . . . . .	743
9.6.2.36	SIGSTRENGTH_THRESHOLD_ARR_SZ . . . . .	743
9.6.2.37	THIRD_INSTANCE . . . . .	743
9.6.2.38	USSD_DCS_8BIT . . . . .	743
9.6.2.39	USSD_DCS_ASCII . . . . .	743
9.6.2.40	USSD_DCS_UCS2 . . . . .	743
9.6.2.41	VOICE_SRV . . . . .	743
9.6.2.42	WDS_SRV . . . . .	743
9.6.3	Typedef Documentation . . . . .	743
9.6.3.1	accelAcceptReady . . . . .	743
9.6.3.2	accelTempAcceptReady . . . . .	744
9.6.3.3	eDevState . . . . .	744
9.6.3.4	eSMSEventType . . . . .	745
9.6.3.5	gpsTime . . . . .	745
9.6.3.6	gyroAcceptReady . . . . .	745
9.6.3.7	gyroTempAcceptReady . . . . .	746
9.6.3.8	LteNasReleaseInfo . . . . .	746
9.6.3.9	modemTempNotification . . . . .	746
9.6.3.10	packetSrvStatus . . . . .	747
9.6.3.11	precisionDilution . . . . .	749
9.6.3.12	sensorDataUsage . . . . .	749
9.6.3.13	sessionInformation . . . . .	750
9.6.3.14	sessionInformationExt . . . . .	750
9.6.3.15	SMSAsyncRawSend . . . . .	750
9.6.3.16	SMSCAddressInfo . . . . .	751
9.6.3.17	SMSEtwsMessageInfo . . . . .	751
9.6.3.18	SMSEtwsPlmnInfo . . . . .	752
9.6.3.19	SMSEventInfo . . . . .	752
9.6.3.20	SMSMessageModelInfo . . . . .	753

9.6.3.21	SMSMTMessageInfo	753
9.6.3.22	SMSONIMSInfo	753
9.6.3.23	SMSTransferRouteMTMessageInfo	754
9.6.3.24	svUsedforFix	754
9.6.3.25	SwiOTAMsg	755
9.6.3.26	tFNActivationStatus	755
9.6.3.27	tFNAllCallStatus	756
9.6.3.28	tFNASwiLTECphyCallInfo	756
9.6.3.29	tFNASwiOTAMsg	756
9.6.3.30	tFNAsyncRawSend	757
9.6.3.31	tFNBandPreference	758
9.6.3.32	tFNCATEvent	760
9.6.3.33	tFNDataCapabilities	760
9.6.3.34	tFNDataSysStatus	761
9.6.3.35	tFNDelAssistData	761
9.6.3.36	tFNDeviceStateChange	761
9.6.3.37	tFNDTMFEvent	762
9.6.3.38	tFNDUNCallInfo	762
9.6.3.39	tFNEventPosition	762
9.6.3.40	tFNFWdIdCompletion	762
9.6.3.41	tFNGnssSvInfo	762
9.6.3.42	tFNHDRPersonality	763
9.6.3.43	tFNImsaPdpStatus	763
9.6.3.44	tFNImsaRatStatus	763
9.6.3.45	tFNImsaRegStatus	763
9.6.3.46	tFNImsaSvcStatus	763
9.6.3.47	tFNImRegMgrConfig	764
9.6.3.48	tFNImSIPConfig	764
9.6.3.49	tFNImSMSConfig	764
9.6.3.50	tFNImUserConfig	764
9.6.3.51	tFNImVoIPConfig	764
9.6.3.52	tFNInfoRec	765
9.6.3.53	tFNInjectSensorData	765
9.6.3.54	tFNInjectTimeStatus	765
9.6.3.55	tFNLURreject	765
9.6.3.56	tFNMemoryFull	765
9.6.3.57	tFNMessageWaiting	766
9.6.3.58	tFNMobileIPStatus	766
9.6.3.59	tFNModemTempInfo	766
9.6.3.60	tFNNet	766

9.6.3.61	tFNNetworkTime	767
9.6.3.62	tFNNewGPS	767
9.6.3.63	tFNNewNMEA	767
9.6.3.64	tFNNewRMTransferStatistics	767
9.6.3.65	tFNNewSMS	768
9.6.3.66	tFNOMADMState	768
9.6.3.67	tFNOpMode	769
9.6.3.68	tFNOTASPStatus	769
9.6.3.69	tFNPacketSrvState	770
9.6.3.70	tFNPDSState	771
9.6.3.71	tFNPower	771
9.6.3.72	tFNPrivacyChange	771
9.6.3.73	tFNQosNWStatus	772
9.6.3.74	tFNQosPriEvent	772
9.6.3.75	tFNQosStatus	772
9.6.3.76	tFNRInfo	773
9.6.3.77	tFNRoamingIndicator	774
9.6.3.78	tFNSDKTerminated	774
9.6.3.79	tFNSensorStreaming	774
9.6.3.80	tFNServingSystem	774
9.6.3.81	tFNSetCradleMount	775
9.6.3.82	tFNSetEventTimeSync	775
9.6.3.83	tFNSigInfo	775
9.6.3.84	tFNSignalStrength	775
9.6.3.85	tFNSLQSOMADMAAlert	775
9.6.3.86	tFNSLQSQOSEvent	776
9.6.3.87	tFNSLQSSessionState	777
9.6.3.88	tFNSLQSSignalStrengths	777
9.6.3.89	tFNSLQSWDSEvent	777
9.6.3.90	tFNSMSEvents	777
9.6.3.91	tFNSUPSInfo	777
9.6.3.92	tFNSUPSNotification	778
9.6.3.93	tFNSysInfo	778
9.6.3.94	tFNSysSelectionPref	778
9.6.3.95	tFNtransLayerInfo	778
9.6.3.96	tFNtransNWRegInfo	779
9.6.3.97	tFNUIMRefresh	780
9.6.3.98	tFNUIMStatusChangeInfo	780
9.6.3.99	tFNUSSDNotification	780
9.6.3.100	tFNUSSDNoWaitIndication	780

9.6.3.101	tFNUSSDRelease	781
9.6.3.102	transLayerNotification	781
9.6.3.103	transNWRegInfoNotification	781
9.6.4	Enumeration Type Documentation	782
9.6.4.1	device_state_enum	782
9.6.4.2	eQaQMIService	782
9.6.4.3	SMSEventType	782
9.6.5	Function Documentation	782
9.6.5.1	iSetCATEventCallback	782
9.6.5.2	iSetSignalStrengthCallback	782
9.6.5.3	iSLQSSetDUNCallInfoCallback	782
9.6.5.4	iSLQSSetSignalStrengthsCallback	782
9.6.5.5	iSLQSSetWdsFirstInstEventCallback	782
9.6.5.6	iSLQSSetWdsSecondInstEventCallback	782
9.6.5.7	iSLQSSetWdsThirdInstEventCallback	782
9.6.5.8	iSLQSSetWdsXferStatsFirstInstCallback	783
9.6.5.9	iSLQSSetWdsXferStatsSecondInstCallback	783
9.6.5.10	SetActivationStatusCallback	783
9.6.5.11	SetCATEventCallback	783
9.6.5.12	SetDataCapabilitiesCallback	785
9.6.5.13	SetDeviceStateChangeCbK	786
9.6.5.14	SetFwDldCompletionCbK	786
9.6.5.15	SetGPSCallback	787
9.6.5.16	SetLocCradleMountCallback	787
9.6.5.17	SetLocDeleteAssistDataCallback	787
9.6.5.18	SetLocEventPositionCallback	787
9.6.5.19	SetLocEventTimeSyncCallback	788
9.6.5.20	SetLocGnssSvInfoCallback	788
9.6.5.21	SetLocInjectSensorDataCallback	788
9.6.5.22	SetLocInjectTimeCallback	789
9.6.5.23	SetLocOpModeCallback	789
9.6.5.24	SetLocSensorStreamingCallback	789
9.6.5.25	SetLURejectCallback	789
9.6.5.26	SetMobileIPStatusCallback	790
9.6.5.27	SetNasLTECphyCalIndCallback	790
9.6.5.28	SetNetChangeCbK	791
9.6.5.29	SetNewSMSCallback	791
9.6.5.30	SetNMEACallback	792
9.6.5.31	SetOMADMStateCallback	792
9.6.5.32	SetPDSSStateCallback	792



9.6.5.33	SetPowerCallback	793
9.6.5.34	SetRFInfoCallback	793
9.6.5.35	SetRMTransferStatisticsCallback	794
9.6.5.36	SetRoamingIndicatorCallback	795
9.6.5.37	SetSignalStrengthCallback	795
9.6.5.38	SetSLQSOMADMAAlertCallback	796
9.6.5.39	SetSLQSOMADMAAlertCallbackExt	796
9.6.5.40	SetUSSDNotificationCallback	797
9.6.5.41	SetUSSDNoWaitIndicationCallback	797
9.6.5.42	SetUSSDReleaseCallback	797
9.6.5.43	SLQSNasNetworkTimeCallBack	798
9.6.5.44	SLQSNasSigInfo2CallBack	798
9.6.5.45	SLQSNasSigInfoCallBack	799
9.6.5.46	SLQSNasSwiOTAMessageCallback	800
9.6.5.47	SLQSNasSysInfoCallBack	800
9.6.5.48	SLQSSetBandPreferenceCbk	801
9.6.5.49	SLQSSetDataSystemStatusCallback	801
9.6.5.50	SLQSSetDUNCallInfoCallback	802
9.6.5.51	SLQSSetIMSAPdpStatusCallback	802
9.6.5.52	SLQSSetIMSARegStatusCallback	803
9.6.5.53	SLQSSetIMSARatStatusCallback	803
9.6.5.54	SLQSSetIMSASvcStatusCallback	803
9.6.5.55	SLQSSetIMSSMSConfigCallback	804
9.6.5.56	SLQSSetIMSUserConfigCallback	804
9.6.5.57	SLQSSetIMSVoIPConfigCallback	805
9.6.5.58	SLQSSetModemTempCallback	805
9.6.5.59	SLQSSetPacketSrvStatusCallback	805
9.6.5.60	SLQSSetQosEventCallback	806
9.6.5.61	SLQSSetQosNWStatusCallback	806
9.6.5.62	SLQSSetQosPriEventCallback	807
9.6.5.63	SLQSSetQosStatusCallback	807
9.6.5.64	SLQSSetRegMgrConfigCallback	808
9.6.5.65	SLQSSetSDKTerminatedCallback	808
9.6.5.66	SLQSSetServingSystemCallback	809
9.6.5.67	SLQSSetSessionStateCallback	809
9.6.5.68	SLQSSetSignalStrengthsCallback	809
9.6.5.69	SLQSSetSIPConfigCallback	810
9.6.5.70	SLQSSetSMSEventCallback	810
9.6.5.71	SLQSSetSwiHDRPersCallback	811
9.6.5.72	SLQSSetSysSelectionPrefCallBack	811

9.6.5.73	SLQSSetTransLayerInfoCallback	812
9.6.5.74	SLQSSetTransNWRegInfoCallback	813
9.6.5.75	SLQSSetWdsEventCallback	813
9.6.5.76	SLQSSetWdsTransferStatisticCallback	814
9.6.5.77	SLQSUIMSetRefreshCallBack	815
9.6.5.78	SLQSUIMSetStatusChangeCallBack	816
9.6.5.79	SLQSVoiceInfoRecCallback	816
9.6.5.80	SLQSVoiceSetAllCallStatusCallBack	816
9.6.5.81	SLQSVoiceSetDTMFEventCallBack	817
9.6.5.82	SLQSVoiceSetOTASPStatusCallBack	817
9.6.5.83	SLQSVoiceSetPrivacyChangeCallBack	818
9.6.5.84	SLQSVoiceSetSUPSCallBack	818
9.6.5.85	SLQSVoiceSetSUPSNotificationCallback	819
9.6.5.86	SLQSWmsAsyncRawSendCallBack	820
9.6.5.87	SLQSWmsMemoryFullCallBack	820
9.6.5.88	SLQSWmsMessageWaitingCallBack	821
9.7	qaGobiApiDcs.h File Reference	822
9.7.1	Detailed Description	823
9.7.2	Macro Definition Documentation	823
9.7.2.1	LEN	823
9.7.2.2	PORTNAM_LEN	823
9.7.3	Function Documentation	823
9.7.3.1	QCWWAN2kConnect	823
9.7.3.2	QCWWAN2kEnumerateDevices	824
9.7.3.3	QCWWAN2kGetConnectedDeviceID	825
9.7.3.4	QCWWANConnect	825
9.7.3.5	QCWWANDisconnect	826
9.7.3.6	QCWWANEnumerateDevices	826
9.7.3.7	SetSDKImagePath	827
9.7.3.8	SLQSGetDeviceMode	827
9.7.3.9	SLQSGetNetStatistic	827
9.7.3.10	SLQSGetUsbPortNames	828
9.7.3.11	SLQSKillSDKProcess	828
9.7.3.12	SLQSQosClearMap	829
9.7.3.13	SLQSQosDumpMap	829
9.7.3.14	SLQSQosEditMap	830
9.7.3.15	SLQSQosMap	831
9.7.3.16	SLQSQosReadMap	831
9.7.3.17	SLQSQosUnmap	832
9.7.3.18	SLQSSetLoggingMask	833

9.7.3.19	SLQSSStart	833
9.7.3.20	SLQSSStart_AVAgent	834
9.7.3.21	SLQSSStartSrv	834
9.8	qaGobiApiDms.h File Reference	835
9.8.1	Detailed Description	837
9.8.2	Macro Definition Documentation	837
9.8.2.1	IMGDETAILS_LEN	837
9.8.2.2	MAX_BUILD_ID_LEN	837
9.8.2.3	MAX_CUST_ID_LEN	837
9.8.2.4	MAX_CUST_VALUE_LEN	837
9.8.2.5	MAX_FSN_LENGTH	837
9.8.2.6	UNIQUE_ID_LEN	837
9.8.3	Typedef Documentation	837
9.8.3.1	custFeaturesInfo	837
9.8.3.2	custFeaturesSetting	840
9.8.3.3	dmsCurrentPRLInfo	841
9.8.3.4	ERIFileparams	842
9.8.3.5	serialNumbersInfo	842
9.8.3.6	SLQSSwiGetHostDevInfoParams	843
9.8.3.7	SLQSSwiGetOSInfoParams	843
9.8.3.8	SLQSSwiGetSerialNoExtParams	843
9.8.3.9	SLQSSwiSetHostDevInfoParams	844
9.8.3.10	SLQSSwiSetOSInfoParams	844
9.8.4	Function Documentation	844
9.8.4.1	ActivateAutomatic	844
9.8.4.2	GetActivationState	844
9.8.4.3	GetDeviceCapabilities	845
9.8.4.4	GetFirmwareRevision	846
9.8.4.5	GetFirmwareRevisions	847
9.8.4.6	GetHardwareRevision	848
9.8.4.7	GetIMSI	849
9.8.4.8	GetManufacturer	849
9.8.4.9	GetModelID	850
9.8.4.10	GetNetworkTime	850
9.8.4.11	GetOfflineReason	851
9.8.4.12	GetPower	851
9.8.4.13	GetPRLVersion	852
9.8.4.14	GetSerialNumbers	852
9.8.4.15	GetVoiceNumber	853
9.8.4.16	ResetToFactoryDefaults	854

9.8.4.17	SetPower	854
9.8.4.18	SLQSGetBandCapability	855
9.8.4.19	SLQSGetCurrentPRLInfo	857
9.8.4.20	SLQSGetCustFeatures	857
9.8.4.21	SLQSGetCustFeaturesV2	858
9.8.4.22	SLQSGetERIFile	858
9.8.4.23	SLQSGetSerialNumbers	858
9.8.4.24	SLQSSetCustFeatures	859
9.8.4.25	SLQSSetCustFeaturesV2	859
9.8.4.26	SLQSSwiGetCrashAction	859
9.8.4.27	SLQSSwiGetCrashInfo	860
9.8.4.28	SLQSSwiGetFirmwareCurr	861
9.8.4.29	SLQSSwiGetFSN	861
9.8.4.30	SLQSSwiGetFwUpdateStatus	861
9.8.4.31	SLQSSwiGetHostDevInfo	862
9.8.4.32	SLQSSwiGetOSInfo	862
9.8.4.33	SLQSSwiGetSerialNoExt	863
9.8.4.34	SLQSSwiGetUSBComp	863
9.8.4.35	SLQSSwiSetCrashAction	864
9.8.4.36	SLQSSwiSetHostDevInfo	864
9.8.4.37	SLQSSwiSetOSInfo	865
9.8.4.38	SLQSSwiSetUSBComp	866
9.8.4.39	SLQSUIMGetState	866
9.8.4.40	UIMChangePIN	867
9.8.4.41	UIMGetControlKeyStatus	867
9.8.4.42	UIMGetIccID	868
9.8.4.43	UIMGetPINStatus	869
9.8.4.44	UIMSetControlKeyProtection	870
9.8.4.45	UIMSetPINProtection	871
9.8.4.46	UIMUnblockControlKey	872
9.8.4.47	UIMUnblockPIN	873
9.8.4.48	UIMVerifyPIN	874
9.8.4.49	ValidateSPC	875
9.9	qaGobiApiFms.h File Reference	875
9.9.1	Detailed Description	878
9.9.2	Macro Definition Documentation	878
9.9.2.1	DEVICE_RESET	878
9.9.2.2	DEVICE_SHUTDOWN	878
9.9.2.3	FIRMWARE_UPDATE_FAIL	878
9.9.2.4	FIRMWARE_UPDATE_SUCCESS	878

9.9.2.5	FIRMWARE_UPGRADE_SUCCESS	878
9.9.2.6	GOBI_LISTENTRIES_MAX	878
9.9.2.7	GOBI_MBN_BUILD_ID_STR_LEN	879
9.9.2.8	GOBI_MBN_IMG_ID_STR_LEN	879
9.9.2.9	GOBI_SET_IMG_PREF_RSPLEN	879
9.9.2.10	PRI_UPDATE_FAIL	879
9.9.2.11	SLQSFWINFO_APPVERSION_SZ	879
9.9.2.12	SLQSFWINFO_BOOTVERSION_SZ	879
9.9.2.13	SLQSFWINFO_CARRIER_SZ	879
9.9.2.14	SLQSFWINFO_CUR_CARR_NAME	879
9.9.2.15	SLQSFWINFO_CUR_CARR_REV	879
9.9.2.16	SLQSFWINFO_MODELID_SZ	879
9.9.2.17	SLQSFWINFO_PACKAGEID_SZ	879
9.9.2.18	SLQSFWINFO_PRIVERSION_SZ	879
9.9.2.19	SLQSFWINFO_SKU_SZ	879
9.9.3	Enumeration Type Documentation	879
9.9.3.1	eGobiDeviceSeries	879
9.9.3.2	eGobiImageCarrier	879
9.9.3.3	eGobiImageGPS	881
9.9.3.4	eGobiImageRegion	881
9.9.3.5	eGobiImageTech	881
9.9.4	Function Documentation	881
9.9.4.1	DeleteStoredImage	881
9.9.4.2	eGetDeviceSeries	882
9.9.4.3	GetImagesPreference	882
9.9.4.4	GetImageStore	882
9.9.4.5	GetStoredImages	883
9.9.4.6	SetImagesPreference	883
9.9.4.7	SLQSDownloadFirmwareToSlot	884
9.9.4.8	SLQSGetBootVersionNumber	885
9.9.4.9	SLQSGetFirmwareInfo	885
9.9.4.10	SLQSGetImageInfo	886
9.9.4.11	SLQSGetImageInfo_9x15	886
9.9.4.12	SLQSGetImageInfoMC77xx	887
9.9.4.13	SLQSGetImageInfoMC83xx	888
9.9.4.14	SLQSIIsSpkgFormatRequired	888
9.9.4.15	SLQSSwiGetAllCarrierImages	888
9.9.4.16	SLQSUgradeFirmware9x15	889
9.9.4.17	upgrade_mc77xx_fw	890
9.9.4.18	UpgradeFirmware2k	890

9.10	qaGobiApilms.h File Reference	891
9.10.1	Detailed Description	891
9.10.2	Function Documentation	892
9.10.2.1	SLQSGetIMSSMSConfig	892
9.10.2.2	SLQSGetIMSUserConfig	892
9.10.2.3	SLQSGetIMSVoIPConfig	892
9.10.2.4	SLQSGetRegMgrConfig	893
9.10.2.5	SLQSGetSIPConfig	893
9.10.2.6	SLQSImsConfigIndicationRegister	894
9.10.2.7	SLQSSetIMSSMSConfig	894
9.10.2.8	SLQSSetIMSUserConfig	895
9.10.2.9	SLQSSetIMSVoIPConfig	895
9.10.2.10	SLQSSetRegMgrConfig	896
9.10.2.11	SLQSSetSIPConfig	896
9.11	qaGobiApilmsa.h File Reference	897
9.11.1	Detailed Description	897
9.11.2	Function Documentation	898
9.11.2.1	SLQSGetIMSAREgStatus	898
9.11.2.2	SLQSGetIMSAServiceStatus	899
9.11.2.3	SLQSGetIMSASupportedFields	899
9.11.2.4	SLQSGetIMSASupportedMsg	900
9.11.2.5	SLQSRegisterIMSASIndication	900
9.12	qaGobiApiLoc.h File Reference	901
9.12.1	Detailed Description	902
9.12.2	Function Documentation	902
9.12.2.1	SLQSLOCDeIAssData	902
9.12.2.2	SLQSLOCEventRegister	902
9.12.2.3	SLQSLOCSetExtPowerState	902
9.12.2.4	SLQSLOCSetOpMode	903
9.12.2.5	SLQSLOCStart	903
9.12.2.6	SLQSLOCStop	904
9.13	qaGobiApiNas.h File Reference	904
9.13.1	Detailed Description	909
9.13.2	Macro Definition Documentation	909
9.13.2.1	IMSI_M_S1_LENGTH	909
9.13.2.2	IMSI_M_S2_LENGTH	909
9.13.2.3	MAX_DATA_SRV_CAPABILITIES	909
9.13.2.4	MAX_DESCRIPTION_LENGTH	909
9.13.2.5	MAX_PILOT_SETS	909
9.13.2.6	MAX_SERV_SYSTEM_RADIO_INTERFACES	909

9.13.2.7	NAM_NAME_LENGTH	909
9.13.2.8	PLMN_LENGTH	909
9.13.2.9	SLQS_SS_INFO_LIST_MAX_ELEMENTS	909
9.13.2.10	SLQS_SYSTEM_ID_SIZE	909
9.13.2.11	UATISIZE	909
9.13.3	Typedef Documentation	909
9.13.3.1	slqsNetworkScanInfo	910
9.13.3.2	sysSelectPrefInfo	911
9.13.3.3	sysSelectPrefParams	914
9.13.4	Enumeration Type Documentation	918
9.13.4.1	_NAMS_RADIO_IF_TECHNOLOGY_	918
9.13.4.2	NAS_LTE_CPHY_CA_BW_NRB	918
9.13.4.3	NAS_LTE_CPHY_SCELL_STATE	918
9.13.5	Function Documentation	918
9.13.5.1	GetACCOLC	918
9.13.5.2	GetANAAAAAuthenticationStatus	919
9.13.5.3	GetCDMANetworkParameters	919
9.13.5.4	GetHomeNetwork	922
9.13.5.5	GetHomeNetwork3GPP2	924
9.13.5.6	GetNetworkPreference	926
9.13.5.7	GetRFInfo	927
9.13.5.8	GetServingNetwork	928
9.13.5.9	GetServingNetworkCapabilities	930
9.13.5.10	GetSignalStrengths	931
9.13.5.11	InitiateDomainAttach	932
9.13.5.12	InitiateNetworkRegistration	933
9.13.5.13	PerformNetworkScan	934
9.13.5.14	SetACCOLC	935
9.13.5.15	SetCDMANetworkParameters	935
9.13.5.16	SetNetworkPreference	937
9.13.5.17	SLQSConfigSigInfo	938
9.13.5.18	SLQSGetErrorRate	938
9.13.5.19	SLQSGetOperatorNameData	939
9.13.5.20	SLQSGetPLMNName	939
9.13.5.21	SLQSGetServingSystem	939
9.13.5.22	SLQSGetSignalStrength	940
9.13.5.23	SLQSGetSysSelectionPref	940
9.13.5.24	SLQSInitiateNetworkRegistration	941
9.13.5.25	SLQSNasConfigSigInfo2	941
9.13.5.26	SLQSNasGet3GPP2Subscription	942

9.13.5.27 SLQSNasGetCellLocationInfo . . . . .	943
9.13.5.28 SLQSNasGetHDRColorCode . . . . .	943
9.13.5.29 SLQSNASGetLTECPHYCaInfo . . . . .	944
9.13.5.30 SLQSNasGetSigInfo . . . . .	944
9.13.5.31 SLQSNasGetSysInfo . . . . .	945
9.13.5.32 SLQSNasGetTxRxInfo . . . . .	945
9.13.5.33 SLQSNasIndicationRegister . . . . .	946
9.13.5.34 SLQSNasIndicationRegisterExt . . . . .	947
9.13.5.35 SLQSNasIndicationRegisterLTECphyCa . . . . .	948
9.13.5.36 SLQSNasSwiIndicationRegister . . . . .	948
9.13.5.37 SLQSNasSwiModemStatus . . . . .	949
9.13.5.38 SLQSPerformNetworkScan . . . . .	949
9.13.5.39 SLQSSetBandPreference . . . . .	950
9.13.5.40 SLQSSetSysSelectionPref . . . . .	952
9.13.5.41 SLQSSwiGetHDRPersonality . . . . .	952
9.13.5.42 SLQSSwiGetHDRProtSubtype . . . . .	952
9.13.5.43 SLQSSwiGetHRPDStats . . . . .	953
9.13.5.44 SLQSSwiGetLteCQI . . . . .	953
9.13.5.45 SLQSSwiNetworkDebug . . . . .	954
9.13.5.46 SLQSSwiPSDetach . . . . .	954
9.14 qaGobiApiOadm.h File Reference . . . . .	954
9.14.1 Detailed Description . . . . .	955
9.14.2 Function Documentation . . . . .	955
9.14.2.1 OMADMCancelSession . . . . .	955
9.14.2.2 OMADMGetPendingNIA . . . . .	955
9.14.2.3 OMADMGetSessionInfo . . . . .	956
9.14.2.4 OMADMStartSession . . . . .	957
9.15 qaGobiApiPds.h File Reference . . . . .	958
9.15.1 Detailed Description . . . . .	959
9.15.2 Macro Definition Documentation . . . . .	959
9.15.2.1 DEFAULTBYTEVALUE . . . . .	959
9.15.2.2 DEFAULTLONGVALUE . . . . .	959
9.15.2.3 DEFAULTWORDVALUE . . . . .	959
9.15.3 Enumeration Type Documentation . . . . .	959
9.15.3.1 anonymous enum . . . . .	959
9.15.4 Function Documentation . . . . .	959
9.15.4.1 ForceXTRADownload . . . . .	959
9.15.4.2 GetPDSDDefaults . . . . .	960
9.15.4.3 GetPDSSState . . . . .	960
9.15.4.4 GetPortAutomaticTracking . . . . .	961



9.15.4.5	GetServiceAutomaticTracking	962
9.15.4.6	GetXTRAAutomaticDownload	963
9.15.4.7	GetXTRANetwork	963
9.15.4.8	GetXTRAValidity	964
9.15.4.9	PDSInjectTimeReference	964
9.15.4.10	ResetPDSDData	965
9.15.4.11	SetPDSDDefaults	966
9.15.4.12	SetPDSSState	967
9.15.4.13	SetPortAutomaticTracking	967
9.15.4.14	SetServiceAutomaticTracking	968
9.15.4.15	SetXTRAAutomaticDownload	968
9.15.4.16	SetXTRANetwork	969
9.15.4.17	SLQSGetAGPSConfig	969
9.15.4.18	SLQSGetGPSStateInfo	970
9.15.4.19	SLQSPDSDeterminePosition	971
9.15.4.20	SLQSPDSInjectAbsoluteTimeReference	971
9.15.4.21	SLQSPDSInjectPositionData	971
9.15.4.22	SLQSSetAGPSConfig	972
9.15.4.23	SLQSSetPositionMethodState	973
9.15.4.24	StartPDSTrackingSessionExt	974
9.15.4.25	StopPDSTrackingSession	975
9.16	qaGobiApiQos.h File Reference	975
9.16.1	Detailed Description	976
9.16.2	Macro Definition Documentation	976
9.16.2.1	MAX_QOS_FILTER_TLV	976
9.16.2.2	MAX_QOS_SPEC_PER_APN	976
9.16.3	Function Documentation	976
9.16.3.1	SLQSQosGetFlowStatus	976
9.16.3.2	SLQSQosGetGranted	977
9.16.3.3	SLQSQosGetNetworkStatus	977
9.16.3.4	SLQSQosGetNWProf	978
9.16.3.5	SLQSQosModify	978
9.16.3.6	SLQSQosRel	979
9.16.3.7	SLQSQosReq	979
9.16.3.8	SLQSQosReset	980
9.16.3.9	SLQSQosResume	980
9.16.3.10	SLQSQosSuspend	981
9.16.3.11	SLQSQosSwiReadApnExtraParams	981
9.16.3.12	SLQSQosSwiReadDataStats	982
9.17	qaGobiApiRms.h File Reference	982

9.17.1	Detailed Description	982
9.17.2	Function Documentation	982
9.17.2.1	GetSMSWake	982
9.17.2.2	SetSMSWake	983
9.18	qaGobiApiSar.h File Reference	984
9.18.1	Detailed Description	984
9.18.2	Enumeration Type Documentation	984
9.18.2.1	eQMISARRFState	984
9.18.3	Function Documentation	985
9.18.3.1	SLQSGetRfSarState	985
9.18.3.2	SLQSSetRfSarState	985
9.19	qaGobiApiSms.h File Reference	986
9.19.1	Detailed Description	988
9.19.2	Macro Definition Documentation	988
9.19.2.1	ABSOLUTE_VALIDITY	988
9.19.2.2	CONFIG_LEN	988
9.19.2.3	MAX_SMS_ROUTES	988
9.19.2.4	NUM_OF_SET	988
9.19.2.5	TIME_DATE_BUF	988
9.19.2.6	TIME_STAMP_BUF	988
9.19.3	Typedef Documentation	988
9.19.3.1	getIndicationRegResp	988
9.19.3.2	getTransLayerInfoResp	989
9.19.3.3	getTransNWRegInfoResp	990
9.19.3.4	qaQmi3GPP2BroadcastCfgInfo	990
9.19.3.5	qaQmi3GPPBroadcastCfgInfo	991
9.19.3.6	setIndicationRegReq	991
9.19.3.7	transLayerInfo	992
9.19.4	Function Documentation	993
9.19.4.1	GetSMSCAddress	993
9.19.4.2	SaveSMS	993
9.19.4.3	SendSMS	994
9.19.4.4	SetSMSCAddress	995
9.19.4.5	SLQSCDMADecodeMTTextMsg	995
9.19.4.6	SLQSCDMAEncodeMOTextMsg	996
9.19.4.7	SLQSDeleteSMS	996
9.19.4.8	SLQSGetIndicationRegister	997
9.19.4.9	SLQSGetMessageWaiting	998
9.19.4.10	SLQSGetSMS	998
9.19.4.11	SLQSGetSmsBroadcastConfig	999

9.19.4.12 SLQSGetSMSList . . . . .	1000
9.19.4.13 SLQSGetTransLayerInfo . . . . .	1001
9.19.4.14 SLQSGetTransNWRegInfo . . . . .	1001
9.19.4.15 SLQSModifySMSStatus . . . . .	1002
9.19.4.16 SLQSSendAsyncSMS . . . . .	1003
9.19.4.17 SLQSSendLongSMS . . . . .	1003
9.19.4.18 SLQSSendSMS . . . . .	1004
9.19.4.19 SLQSSetIndicationRegister . . . . .	1005
9.19.4.20 SLQSSetSmsBroadcastActivation . . . . .	1005
9.19.4.21 SLQSSetSmsBroadcastConfig . . . . .	1006
9.19.4.22 SLQSSetSmsStorage . . . . .	1006
9.19.4.23 SLQSSmsGetMaxStorageSize . . . . .	1007
9.19.4.24 SLQSSmsGetMessageProtocol . . . . .	1007
9.19.4.25 SLQSSmsSetRoutes . . . . .	1008
9.19.4.26 SLQSSwiGetSMSStorage . . . . .	1008
9.19.4.27 SLQSWCDMADecodeLongTextMsg . . . . .	1009
9.19.4.28 SLQSWCDMADecodeMTTextMsg . . . . .	1009
9.19.4.29 SLQSWCDMAEncodeMOTextMsg . . . . .	1010
9.20 qaGobiApiSwi.h File Reference . . . . .	1010
9.20.1 Detailed Description . . . . .	1011
9.20.2 Function Documentation . . . . .	1011
9.20.2.1 SLQSGetPidof . . . . .	1011
9.20.2.2 SLQSGetSdkVersion . . . . .	1011
9.20.2.3 SLQSSendRawQMI . . . . .	1011
9.21 qaGobiApiSwiAudio.h File Reference . . . . .	1011
9.21.1 Detailed Description . . . . .	1012
9.21.2 Macro Definition Documentation . . . . .	1012
9.21.2.1 MAX_LEN_IFACE_TABLE . . . . .	1012
9.21.3 Function Documentation . . . . .	1012
9.21.3.1 SLQSGetM2MAudioProfile . . . . .	1012
9.21.3.2 SLQSGetM2MAudioVolume . . . . .	1013
9.21.3.3 SLQSGetM2MAVMute . . . . .	1013
9.21.3.4 SLQSGetM2MSpkrGain . . . . .	1014
9.21.3.5 SLQSSetM2MAudioAVCFG . . . . .	1014
9.21.3.6 SLQSSetM2MAudioLPBK . . . . .	1015
9.21.3.7 SLQSSetM2MAudioNVDef . . . . .	1015
9.21.3.8 SLQSSetM2MAudioProfile . . . . .	1015
9.21.3.9 SLQSSetM2MAudioVolume . . . . .	1016
9.21.3.10 SLQSSetM2MAVMute . . . . .	1016
9.21.3.11 SLQSSetM2MSpkrGain . . . . .	1017

9.22	qaGobiApiSwiOmadms.h File Reference	1017
9.22.1	Detailed Description	1018
9.22.2	Typedef Documentation	1018
9.22.2.1	SLQSOMADMSessionInfo	1018
9.22.2.2	SLQSOMADMSettings	1020
9.22.2.3	SLQSOMADMSettingsReqParams	1021
9.22.2.4	SLQSOMADMSettingsReqParams3	1022
9.22.3	Function Documentation	1023
9.22.3.1	SLQSOMADMCancelSession	1023
9.22.3.2	SLQSOMADMGetSessionInfo	1023
9.22.3.3	SLQSOMADMGetSettings	1024
9.22.3.4	SLQSOMADMGetSettings2	1025
9.22.3.5	SLQSOMADMSendSelection	1025
9.22.3.6	SLQSOMADMSendSelection2	1026
9.22.3.7	SLQSOMADMSetSettings	1026
9.22.3.8	SLQSOMADMSetSettings2	1027
9.22.3.9	SLQSOMADMSetSettings3	1027
9.22.3.10	SLQSOMADMStartSession	1028
9.22.3.11	SLQSOMADMStartSession2	1028
9.23	qaGobiApiTableBandClasses.h File Reference	1029
9.23.1	Detailed Description	1029
9.23.2	Band Classes (Value - Description)	1029
9.23.2.1	LTE Bands	1031
9.24	qaGobiApiTableCallControlReturnReasons.h File Reference	1032
9.24.1	Detailed Description	1032
9.24.2	Call Control Result Reasons (Value - Name - Description)	1032
9.25	qaGobiApiTableCallEndReasons.h File Reference	1033
9.25.1	Detailed Description	1033
9.25.2	Call end reason codes (Code - Reason)	1033
9.25.2.1	Technology-agnostic call end reasons	1033
9.25.2.2	EVDO CDMA 1xEV-DO	1033
9.25.2.3	WCDMA/GSM call end reasons	1034
9.25.2.4	EVDO CDMA 1xEV-DO	1036
9.25.2.5	call end reason type	1036
9.25.2.6	Mobile IP call end reasons (Type=1)	1037
9.25.2.7	Internal call end reasons (Type=2)	1038
9.25.2.8	Call Manager defined call end reasons (Type=3)	1040
9.25.2.9	3GPP specification defined call end reasons (Type=6)	1044
9.25.2.10	PPP call end reasons (Type=7)	1046
9.25.2.11	EHRPD call end reasons (Type=8)	1047

9.25.2.12 IPv6 call end reasons (Type=9)	1048
9.26 qaGobiApiTableCarrierCodes.h File Reference	1048
9.26.1 Detailed Description	1048
9.26.2 Carrier Codes (Number - Carrier)	1048
9.27 qaGobiApiTableCodingScheme.h File Reference	1050
9.27.1 Detailed Description	1050
9.27.2 Call Control Result Reasons (Value - Name - Description)	1050
9.27.2.1 Use of bits 3..0	1050
9.27.3 Coding Group Bits 7..4(0001)	1050
9.27.3.1 use of bits 3..0	1050
9.27.4 Coding Group Bits 7..4(0010)	1051
9.27.4.1 use of bits 3..0	1051
9.27.5 Coding Group Bits 7..4(0011)	1051
9.27.5.1 use of bits 3..0	1051
9.27.6 Coding Group Bits 7..4(01xx)	1051
9.27.6.1 use of bits 3..0	1051
9.27.7 Coding Group Bits 7..4(1001)	1052
9.27.7.1 Reserved coding groups	1052
9.27.8 Coding Group Bits 7..4(1010..1101)	1052
9.27.8.1 Reserved coding groups	1052
9.27.9 Coding Group Bits 7..4(1110)	1052
9.27.9.1 Defined by the WAP Forum	1052
9.27.10 Coding Group Bits 7..4 (1111)	1052
9.27.10.1 Data coding / message handling	1052
9.27.11 Macro Definition Documentation	1052
9.27.11.1 __GOBI_API_CODING_SCHEME_H__	1052
9.28 qaGobiApiTableGpsCapabilityCodes.h File Reference	1052
9.28.1 Detailed Description	1053
9.28.2 GPS capability (Value - Capability)	1053
9.29 qaGobiApiTablePowerModes.h File Reference	1053
9.29.1 Detailed Description	1053
9.29.2 Power Modes (Value - Description)	1053
9.30 qaGobiApiTableRadioInterfaces.h File Reference	1053
9.30.1 Detailed Description	1053
9.30.2 Radio interface	1054
9.30.2.1 Technology (Value - Radio Interface Technology)	1054
9.31 qaGobiApiTableRegionCodes.h File Reference	1054
9.31.1 Detailed Description	1054
9.31.2 Region Codes (Code - Region)	1054
9.32 qaGobiApiTableServiceOptions.h File Reference	1054

9.32.1	Detailed Description	1054
9.32.2	Service Option codes (Code - Reason)	1055
9.32.2.1	Description	1055
9.33	qaGobiApiTableSupServiceInfoClasses.h File Reference	1057
9.33.1	Detailed Description	1057
9.33.2	Supplementary Service Information Classes (Value - Service Class)	1057
9.34	qaGobiApiTableSwiAudio.h File Reference	1057
9.34.1	Detailed Description	1057
9.34.2	ACDB Device (Device ID - description)	1057
9.34.3	Physical Interface (Device ID - description - Interface parameters)	1058
9.35	qaGobiApiTableSwiOMADMUpdateCompleteStatus.h File Reference	1058
9.35.1	Detailed Description	1058
9.35.2	OMA DM Update Complete Status (Update Complete Status - Meaning - Usage)	1058
9.36	qaGobiApiTableVoiceCallEndReasons.h File Reference	1059
9.36.1	Detailed Description	1059
9.36.2	Voice Call and supplementary services end reason codes (Code - Reason)	1059
9.36.2.1	General	1059
9.36.2.2	service Errors	1061
9.36.2.3	control cause values	1062
9.36.2.4	reject causes	1064
9.36.2.5	reject causes	1064
9.36.2.6	reject causes	1064
9.36.2.7	stratum reject causes	1065
9.36.2.8	reject causes	1065
9.36.2.9	IP end reasons	1065
9.37	qaGobiApiUim.h File Reference	1066
9.37.1	Detailed Description	1067
9.37.2	Macro Definition Documentation	1067
9.37.2.1	MAX_CONTENT_LENGTH	1067
9.37.2.2	MAX_DESCRIPTION_LENGTH	1067
9.37.2.3	MAX_NO_OF_APPLICATIONS	1067
9.37.2.4	MAX_NO_OF_SLOTS	1067
9.37.2.5	MAX_PATH_LENGTH	1067
9.37.2.6	MAX_PUK_LENGTH	1068
9.37.3	Function Documentation	1068
9.37.3.1	SLQSUIMAuthenticate	1068
9.37.3.2	SLQSUIMChangePin	1068
9.37.3.3	SLQSUIMDepersonalization	1069
9.37.3.4	SLQSUIEventRegister	1069
9.37.3.5	SLQSUIMGetCardStatus	1070

9.37.3.6	SLQSUIMGetFileAttributes	1070
9.37.3.7	SLQSUIMPowerDown	1071
9.37.3.8	SLQSUIMPowerUp	1071
9.37.3.9	SLQSUIMReadTransparent	1072
9.37.3.10	SLQSUIMRefreshComplete	1072
9.37.3.11	SLQSUIMRefreshGetLastEvent	1073
9.37.3.12	SLQSUIMRefreshOK	1073
9.37.3.13	SLQSUIMRefreshRegister	1074
9.37.3.14	SLQSUIMReset	1074
9.37.3.15	SLQSUIMSetPinProtection	1075
9.37.3.16	SLQSUIMUnblockPin	1075
9.37.3.17	SLQSUIMVerifyPin	1076
9.38	qaGobiApiVoice.h File Reference	1076
9.38.1	Detailed Description	1079
9.38.2	Macro Definition Documentation	1079
9.38.2.1	MAX_CALL_NO_LEN	1079
9.38.2.2	MAX_DESCRIPTION_LENGTH	1079
9.38.2.3	MAX_NO_OF_CALLS	1079
9.38.2.4	MAXUSSDLENGTH	1080
9.38.2.5	PASSWORD_LENGTH	1080
9.38.3	Enumeration Type Documentation	1080
9.38.3.1	serviceClassInformation	1080
9.38.4	Function Documentation	1080
9.38.4.1	AnswerUSSD	1080
9.38.4.2	CancelUSSD	1080
9.38.4.3	OriginateUSSD	1081
9.38.4.4	SLQSOriginateUSSD	1081
9.38.4.5	SLQSVoiceALSSelectLine	1082
9.38.4.6	SLQSVoiceALSSetLineSwitching	1082
9.38.4.7	SLQSVoiceAnswerCall	1083
9.38.4.8	SLQSVoiceBindSubscription	1083
9.38.4.9	SLQSVoiceBurstDTMF	1084
9.38.4.10	SLQSVoiceDialCall	1084
9.38.4.11	SLQSVoiceEndCall	1085
9.38.4.12	SLQSVoiceGetAllCallInfo	1085
9.38.4.13	SLQSVoiceGetCallBarring	1086
9.38.4.14	SLQSVoiceGetCallForwardingStatus	1087
9.38.4.15	SLQSVoiceGetCallInfo	1088
9.38.4.16	SLQSVoiceGetCallWaiting	1089
9.38.4.17	SLQSVoiceGetCLIP	1089

9.38.4.18 SLQSVoiceGetCLIR . . . . .	1090
9.38.4.19 SLQSVoiceGetCNAP . . . . .	1090
9.38.4.20 SLQSVoiceGetCOLP . . . . .	1091
9.38.4.21 SLQSVoiceGetCOLR . . . . .	1091
9.38.4.22 SLQSVoiceGetConfig . . . . .	1092
9.38.4.23 SLQSVoiceIndicationRegister . . . . .	1092
9.38.4.24 SLQSVoiceManageCalls . . . . .	1093
9.38.4.25 SLQSVoiceOrigUSSDNoWait . . . . .	1094
9.38.4.26 SLQSVoiceSendFlash . . . . .	1095
9.38.4.27 SLQSVoiceSetCallBarringPassword . . . . .	1095
9.38.4.28 SLQSVoiceSetConfig . . . . .	1096
9.38.4.29 SLQSVoiceSetPreferredPrivacy . . . . .	1097
9.38.4.30 SLQSVoiceSetSUPSService . . . . .	1097
9.38.4.31 SLQSVoiceStartContDTMF . . . . .	1098
9.38.4.32 SLQSVoiceStopContDTMF . . . . .	1098
9.39 qaGobiApiWds.h File Reference . . . . .	1099
9.39.1 Detailed Description . . . . .	1103
9.39.2 Macro Definition Documentation . . . . .	1103
9.39.2.1 IPV6_ADDRESS_ARRAY_SIZE . . . . .	1103
9.39.3 Typedef Documentation . . . . .	1103
9.39.3.1 GetProfileSettingIn . . . . .	1103
9.39.3.2 GetProfileSettingOut . . . . .	1103
9.39.3.3 QmiProfileInfo . . . . .	1103
9.39.3.4 QmiWDSDataBearers . . . . .	1104
9.39.3.5 QmiWDSDataBearerTechnology . . . . .	1105
9.39.3.6 slqs3GPPConfigItem . . . . .	1107
9.39.4 Enumeration Type Documentation . . . . .	1108
9.39.4.1 qmiDataBearerMasks . . . . .	1108
9.39.5 Function Documentation . . . . .	1108
9.39.5.1 GetAutoconnect . . . . .	1108
9.39.5.2 GetByteTotals . . . . .	1109
9.39.5.3 GetConnectionRate . . . . .	1109
9.39.5.4 GetDataBearerTechnology . . . . .	1110
9.39.5.5 GetDefaultProfile . . . . .	1111
9.39.5.6 GetDefaultProfileLTE . . . . .	1113
9.39.5.7 GetDormancyState . . . . .	1115
9.39.5.8 GetIPAddressLTE . . . . .	1116
9.39.5.9 GetLastMobileIPError . . . . .	1116
9.39.5.10 GetMobileIP . . . . .	1117
9.39.5.11 GetMobileIPProfile . . . . .	1117



9.39.5.12 GetPacketStatistics . . . . .	1119
9.39.5.13 GetPacketStatus . . . . .	1120
9.39.5.14 GetSessionDuration . . . . .	1120
9.39.5.15 GetSessionState . . . . .	1121
9.39.5.16 iGetByteTotals . . . . .	1122
9.39.5.17 iGetConnectionRate . . . . .	1122
9.39.5.18 iGetPacketStatistics . . . . .	1122
9.39.5.19 iLQSMISetIPFamilyPreference . . . . .	1122
9.39.5.20 RMSetTransferStatistics . . . . .	1122
9.39.5.21 SetActiveMobileIPProfile . . . . .	1122
9.39.5.22 SetAutoconnect . . . . .	1123
9.39.5.23 SetDefaultProfile . . . . .	1124
9.39.5.24 SetDefaultProfileLTE . . . . .	1125
9.39.5.25 SetDefaultProfileLTEV2 . . . . .	1127
9.39.5.26 SetMobileIP . . . . .	1129
9.39.5.27 SetMobileIPParameters . . . . .	1130
9.39.5.28 SetMobileIPProfile . . . . .	1131
9.39.5.29 SLQSAutoConnect . . . . .	1132
9.39.5.30 SLQSCreateProfile . . . . .	1132
9.39.5.31 SLQSDeleteProfile . . . . .	1133
9.39.5.32 SLQSGet3GPPConfigItem . . . . .	1134
9.39.5.33 SLQSGetByteTotals . . . . .	1134
9.39.5.34 SLQSGetConnectionRate . . . . .	1134
9.39.5.35 SLQSGetCurrDataSystemStat . . . . .	1135
9.39.5.36 SLQSGetCurrentChannelRate . . . . .	1135
9.39.5.37 SLQSGetDataBearerTechnology . . . . .	1136
9.39.5.38 SLQSGetDataBearerTechnologyExt . . . . .	1136
9.39.5.39 SLQSGetDUNCallInfo . . . . .	1137
9.39.5.40 SLQSGetPacketStatistics . . . . .	1137
9.39.5.41 SLQSGetProfile . . . . .	1138
9.39.5.42 SLQSGetProfileSettings . . . . .	1140
9.39.5.43 SLQSGetRuntimeSettings . . . . .	1141
9.39.5.44 SLQSGetSessionState . . . . .	1141
9.39.5.45 SLQSMModifyProfile . . . . .	1142
9.39.5.46 SLQSResetPacketStatics . . . . .	1143
9.39.5.47 SLQSSet3GPPConfigItem . . . . .	1143
9.39.5.48 SLQSSetHostMTU . . . . .	1143
9.39.5.49 SLQSSetProfile . . . . .	1144
9.39.5.50 SLQSSGetLoopback . . . . .	1146
9.39.5.51 SLQSSSetLoopback . . . . .	1146

9.39.5.52 SLQSSStartStopDataSession . . . . .	1147
9.39.5.53 SLQSWdsGoActive . . . . .	1148
9.39.5.54 SLQSWdsGoDormant . . . . .	1148
9.39.5.55 SLQSWdsSetEventReport . . . . .	1149
9.39.5.56 SLQSWdsSwiPDPRuntimeSettings . . . . .	1149
9.39.5.57 WDS_IsGobiDevice . . . . .	1150
9.40 qaNasGetRFBandInfo.h File Reference . . . . .	1150
9.40.1 Enumeration Type Documentation . . . . .	1150
9.40.1.1 eQMI_NAS_GET_RF_INFO_RESP . . . . .	1150
9.40.2 Function Documentation . . . . .	1150
9.40.2.1 PkQmiNasGetRFBandInfo . . . . .	1150
9.40.2.2 UpkQmiNasGetRFBandInfo . . . . .	1150
9.41 qaNasPerformNetworkScan.h File Reference . . . . .	1150
9.41.1 Macro Definition Documentation . . . . .	1151
9.41.1.1 FORBIDDEN_INDEX . . . . .	1151
9.41.1.2 INDEX_ZERO . . . . .	1151
9.41.1.3 MAX_DESCRIPTION_LENGTH . . . . .	1151
9.41.1.4 PREFERRED_INDEX . . . . .	1151
9.41.1.5 QMI_NAS_MAX_INSTANCES . . . . .	1151
9.41.1.6 QMI_NAS_NETSTATUS_MASK . . . . .	1151
9.41.1.7 ROAMING_INDEX . . . . .	1151
9.41.2 Enumeration Type Documentation . . . . .	1151
9.41.2.1 eQMI_NAS_PERFORM_NETWORK_SCAN_RESP . . . . .	1151
9.41.3 Function Documentation . . . . .	1151
9.41.3.1 PkQmiNasPerformNetworkScan . . . . .	1151
9.41.3.2 UpkQmiNasPerformNetworkScan . . . . .	1151
9.42 qmerrno.h File Reference . . . . .	1151
9.42.1 Enumeration Type Documentation . . . . .	1153
9.42.1.1 eQCWWANError . . . . .	1153
9.42.1.2 qm_wds_ds_profile_extended_err_codes . . . . .	1157
9.43 SwiDataTypes.h File Reference . . . . .	1158
9.43.1 Detailed Description . . . . .	1159
9.43.2 Macro Definition Documentation . . . . .	1159
9.43.2.1 QMI_NO_LTE_FW_SUPPORT . . . . .	1159
9.43.2.2 QMI_TLV_PLACEHOLDER . . . . .	1159
9.43.2.3 SWI_API . . . . .	1159
9.43.2.4 UNUSEDPARAM . . . . .	1159
9.43.3 Typedef Documentation . . . . .	1159
9.43.3.1 BOOL . . . . .	1159
9.43.3.2 BYTE . . . . .	1159

9.43.3.3	CHAR	1159
9.43.3.4	FLOAT	1159
9.43.3.5	INT32	1159
9.43.3.6	INT8	1159
9.43.3.7	LPCSTR	1159
9.43.3.8	SHORT	1159
9.43.3.9	ULONG	1159
9.43.3.10	ULONGLONG	1159
9.43.3.11	USHORT	1160
9.43.3.12	WORD	1160
9.44	SWIWWANCMAPI.h File Reference	1160
<b>Index</b>		<b>1161</b>



# Chapter 1

## Welcome to the Sierra Wireless Linux QMI SDK API Reference Guide

This API reference guide contains information about all the modules, in the Sierra Wireless Linux QMI SDK (SLQS). Use the tabs at the top of the page to navigate the reference guide.

- Modules tab – lists all the service modules and provides a link to the API header file in each module.
- References tab – links to reference material.

### 1.1 Important Notice

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

### 1.2 Limitation of Liability

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

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

### 1.3 Patents

**Patents** This product may contain technology developed by or for Sierra Wireless Inc. This product includes technology licensed from QUALCOMM® 3G. This product is manufactured or sold by Sierra Wireless Inc. or its affiliates

under one or more patents licensed from InterDigital Group.

## 1.4 Copyright

© 2011-2015 Sierra Wireless. All rights reserved.

## 1.5 Trademarks

AirCard® and Heart of the Wireless Machine® are registered trademarks of Sierra Wireless. Watcher® is a trademark of Sierra Wireless, registered in the European Community. Sierra Wireless, the Sierra Wireless logo, the red wave design, and the red-tipped antenna are trademarks of Sierra Wireless. Windows® is a registered trademark of Microsoft Corporation. QUALCOMM® is a registered trademark of QUALCOMM Incorporated. Used under license. Linux is a registered trademark of Linus Torvalds. Other trademarks are the property of the respective owners.

## 1.6 Contact Information

If you have any questions about the Sierra Wireless Linux SDK, contact your Sierra Wireless account manager.

Consult our website for up-to-date product descriptions, documentation, application notes, firmware upgrades, troubleshooting tips, and press releases: <http://www.sierrawireless.com>.

## Chapter 2

# Module Index

### 2.1 Modules

Here is a list of all modules:

Device Connectivity Service (DCS) . . . . .	21
Wireless Data Service (WDS) . . . . .	22
Device Management Service (DMS) . . . . .	23
Network Access Service (NAS) . . . . .	24
CallBack registration (CBK) . . . . .	25
Short Message Service (SMS) . . . . .	26
Position Determination Service (PDS) . . . . .	27
Card Application Toolkit (CAT) . . . . .	28
Remote Management Service (RMS) . . . . .	29
Firmware Management Service (FMS) . . . . .	30
Open Mobile Alliance Service (OMA) . . . . .	31
Specific Absorption Rate (SAR) . . . . .	32
SWI Open Mobile Alliance Service (SWIOMA) . . . . .	33
Voice Service (VOICE) . . . . .	34
Non-service specific APIs (SWI) . . . . .	35
User Identity Module Service (UIM) . . . . .	36
Audio Service (AUDIO) . . . . .	37
Quality of Service (QOS) . . . . .	38
IMS Service (IMS) . . . . .	39
SWI Audio Service(SWIAUDIO) . . . . .	40
Location Service(LOC) . . . . .	41





## Chapter 3

# Namespace Index

### 3.1 Namespace List

Here is a list of all namespaces with brief descriptions:

<a href="#">Tables</a> . . . . .	43
----------------------------------	----



## Chapter 4

# Data Structure Index

### 4.1 Data Structures

Here are the data structures with brief descriptions:

<a href="#">_getIndicationRegResp</a>	45
<a href="#">_GetProfileSettingIn</a>	46
<a href="#">_GetProfileSettingOut</a>	47
<a href="#">_getTransLayerInfoResp</a>	47
<a href="#">_getTransNWRegInfoResp</a>	49
<a href="#">_modemTempNotification</a>	50
<a href="#">_packetSrvStatus</a>	50
<a href="#">_qaQmi3GPP2BroadcastCfgInfo</a>	53
<a href="#">_qaQmi3GPPBroadcastCfgInfo</a>	53
<a href="#">_setIndicationRegReq</a>	55
<a href="#">_slqs3GPPConfigItem</a>	56
<a href="#">_slqsNetworkScanInfo</a>	59
<a href="#">_SLQSOMADMSessionInfo</a>	61
<a href="#">_SLQSOMADMSettings</a>	64
<a href="#">_SLQSOMADMSettingsReqParams</a>	65
<a href="#">_SLQSOMADMSettingsReqParams3</a>	66
<a href="#">_SLQSSwiGetHostDevInfoParams</a>	67
<a href="#">_SLQSSwiGetOSInfoParams</a>	68
<a href="#">_SLQSSwiGetSerialNoExtParams</a>	68
<a href="#">_SLQSSwiSetHostDevInfoParams</a>	69
<a href="#">_SLQSSwiSetOSInfoParams</a>	69
<a href="#">_sysSelectPrefInfo</a>	70
<a href="#">_sysSelectPrefParams</a>	74
<a href="#">_transLayerinfo</a>	78
<a href="#">_transLayerInfoNotification</a>	80
<a href="#">_transNWRegInfoNotification</a>	81
<a href="#">accelAcceptReady_s</a>	81
<a href="#">accelTempAcceptReady_s</a>	82
<a href="#">acqOrderPref</a>	83
<a href="#">ActPilotPNElement</a>	84
<a href="#">AddCDMASysInfo</a>	84
<a href="#">AddSysInfo</a>	85
<a href="#">airTimer</a>	86
<a href="#">allCallsAlphaIDInfo</a>	87
<a href="#">allCallsDiagInfo</a>	87
<a href="#">allCallsUUSInfo</a>	88
<a href="#">alphaIDInfo</a>	88
<a href="#">appStatus</a>	89

arrAlertingPattern	93
arrAlertingType	93
arrAlphaID	94
arrCalledPartyNum	95
arrCallEndReason	95
arrCallInfo	96
arrConnectPartyNum	97
arrDiagInfo	97
arrRedirPartyNum	98
arrRemotePartyName	98
arrRemotePartyNum	99
arrSvcOption	99
arrUUSInfo	100
authenticateResult	101
authenticationData	101
BdsSV	102
BdsSVInfo	103
BroadcastConfig	104
burstDTMFInfo	104
CallBarringSysInfo	105
callBarStatus	106
calledPartyInfo	107
calledPartySubAdd	109
callerIDInfo	110
callFwdTypeAndPlan	111
callFWExtInfo	112
callFWInfo	115
callInfo	116
callingPartyInfo	118
cardResult	120
cardStatus	121
CatAlPhalIdentifierTlv	122
CatCommonEventTlv	123
CatEndProactiveSessionTlv	123
CATEventDataType	124
CatEventIDDDataTlv	124
CatEventListTlv	124
CatRefreshTlv	125
ccSUPSType	125
CDMABroadcastConfig	126
CDMAChannel	127
CDMAECIOThresh	128
CDMAInfo	128
cdmaMsgDecodingParams	130
cdmaMsgEncodingParams	133
CDMARSSIThresh	135
CDMASSInfo	135
CDMASysInfo	136
CDMASysInfoExt	140
CellIDb	141
cellParams	141
changeUIMPIN	142
channelRate	143
ChannelRate	144
CLIPResp	145
CLIRResp	146
CikInfo	147
CNAPResp	149

COLPResp	149
COLRResp	150
CommInfo	151
ConnectionStatus	153
connectNumInfo	154
CrashInfo	156
CrashInfoParams	158
CreateProfileIn	159
CreateProfileOut	160
CSGID	160
CUGInfo	161
curAMRConfig	162
CurrDataSysStat	163
currentCatEvent	164
CurrentImgList	165
currentPLMN	166
CurrImageInfo	167
CurrNetworkInfo	168
custFeaturesInfo	170
custFeaturesSetting	172
custSettingInfo	174
custSettingList	175
dataBearers	176
DataBearerTech	177
DataBearerTechExt	179
dataBearerTechnology	179
dataRate	181
dataSrvCapabilities	181
DataStatusDetail	182
DataULongLongTlv	184
DataULongTlv	184
DcsUsbPortNames	184
delAssistDataStatus	184
depersonalizationInformation	185
detailSvcInfo	187
DeviceConfigDetail	189
diagInfo	190
dirNum	191
dmsCurrentPRLInfo	191
Domain	192
DomainNameList	193
DRCParams	193
DTMFInfo	194
DTMFLengths	194
DUNCallInfoInd	195
ecioListElement	196
ECIOThresh	196
ECTNum	198
encryptedPIN1	199
ERIFileparams	200
errorRateListElement	200
extDispRecInfo	201
FactorySequenceNumber	203
fileAttributes	203
fileInfo	208
FirmwareUpdatStat	210
finfo_s	212
GERANInfo	212

geranInstInfo	214
getAllCallInformation	216
getAllCallRmtPtyName	217
getAllCallRmtPtyNum	217
GetAudioPathConfigReq	219
GetAudioPathConfigResp	220
GetAudioProfileReq	222
GetAudioProfileResp	222
GetAudioVolTLBConfigReq	224
GetAudioVolTLBConfigResp	224
getCallFWExtInfo	225
getCallFWInfo	225
getCustomFeatureV2	226
getCustomInput	227
getDUNCallInfoReq	227
getDUNCallInfoResp	229
GetErrRateResp	231
GetHRPDStatsResp	232
GetIMSSMSConfigParams	233
GetIMSUserConfigParams	234
GetIMSVoIPConfigResp	235
GetInstIDResp	238
GetM2MAudioProfileReq	238
GetM2MAudioProfileResp	238
GetM2MAudioVolumeReq	241
GetM2MAudioVolumeResp	241
GetM2MAVMuteReq	242
GetM2MAVMuteResp	242
GetM2MSpkrGainReq	243
GetM2MSpkrGainResp	243
getMsgWaitingInfo	244
GetRegMgrConfigParams	244
GetSessionIDResp	246
GetSIPConfigResp	247
GnssData	248
gnssSvInfoNotification	250
GPRSQoS	250
GPRSRequestedQoS	251
GPSSStateInfo	252
gpsTime_s	256
gsmCellInfo	257
GSMRSSIThresh	258
GSMSrvStatusInfo	259
GSMSysInfo	260
gyroAcceptReady_s	263
gyroTempAcceptReady_s	264
HDRECIOThresh	265
HDRIOThresh	266
HDRPersonalityInd	266
HDRPersonalityResp	267
HDRProtSubtypResp	267
HDRRSSIThresh	268
HDRSINRThresh	269
HDRSINRThreshold	269
HDRSSInfo	270
HDRSysInfo	271
homeSIDNID	275
hotSwapStatus	275

ImageElement	276
ImageIdElement	277
ImageIdEntries	278
ImageList	278
IMSIndRegisterInfo	279
imsaPdpStatusInfo	280
imsaRatStatusInfo	281
IMSARegistrationStatus	282
imsaRegStatusInfo	283
IMSAServiceStatus	284
IMSASupportedFieldsResp	287
IMSASupportedMsgInfo	287
imsaSvcStatusInfo	288
imsCfgIndRegisterInfo	289
imsRegMgrConfigInfo	290
imsSIPConfigInfo	292
imsSMSCConfigInfo	293
imsUserConfigInfo	294
imsVoIPConfigInfo	294
IndFieldsList	297
infoInterFreq	298
IOTresh	299
IPv4Addr	300
IPv6Addr	300
IPV6AddressInfo	302
IPV6GWAddressInfo	302
IPv6TrafCls	303
lineCtrlInfo	303
LocApplicationInfo	304
LocDelAssDataReq	305
LOCEventRegisterReqResp	306
LOCExtPowerStateReqResp	308
LOCStartReq	308
LOCStopReq	310
LteCQIParm	310
lteGsmCellInfo	311
LTEInfo	312
LTEInfoInterfreq	315
LTEInfoIntrafreq	316
LTEInfoNeighboringGSM	319
LTEInfoNeighboringWCDMA	319
LteNasReleaseInfo_s	320
lteRsrpInformation	321
LTERSRPThresh	321
LTERSQRThresh	322
LTERSSIThresh	322
LTESigRptCfg	323
LTESigRptConfig	324
lteSnrinformation	325
LTESNRThresh	325
LTESNRThreshold	326
LTESSTInfo	326
LTESysInfo	327
lteWcdmaCellInfo	331
messageWaitingInfoContent	332
minBasedIMSI	333
MNRInfo	333
ModifyProfileIn	335

ModifyProfileOut	336
msgWaitingInfo	336
namName	337
nasCellLocationInfoResp	337
nasGet3GPP2SubscriptionInfoReq	339
nasGet3GPP2SubscriptionInfoResp	339
nasGetHDRColorCodeResp	340
nasGetLTECphyCa	341
nasGetLTECphyCaResp	341
nasGetSigInfoResp	342
nasGetSysInfoResp	343
nasGetTxRxInfoReq	345
nasGetTxRxInfoResp	346
nasIndicationRegisterReq	347
nasInitNetworkReg	350
nasNetworkTime	351
nasOperatorNameResp	352
nasPLMNNameReq	353
nasPLMNNameResp	354
nasSigInfo	357
NasSwIndReg	358
nasSysInfo	359
netSelectionPref	362
NetStats	363
NetworkDebugResp	364
NetworkStat1x	366
NetworkStatEVDO	369
newPwdData	371
nmrCellInfo	372
NSSAudioCtrl	373
NWProfile	374
omaDmConfigTlv	374
omaDmConfigTlvExt	375
omaDmFotaTlv	378
omaDmFotaTlvExt	380
omaDmNotificationsTlv	383
operatorNameString	383
OperatorPLMNData	383
operatorPLMNList	384
PCMparams	385
PCSCFFQDNAddress	385
PCSCFFQDNAddressList	387
PCSCFIPv4ServerAddressList	387
PDSPositionData	388
PDSPosMethodStateReq	390
peerNumberInfo	391
PhyCaAggPcellInfo	393
PhyCaAggScellIDBw	394
PhyCaAggScellIndex	395
PhyCaAggScellIndType	395
PhyCaAggScellInfo	396
PilotSetData	397
PilotSetParams	398
pktErrRate	399
PLMNNetworkName	399
PLMNNetworkNameData	400
Port	402
precisionDilution_s	403



PrefImageList	403
prefVoiceSO	404
Profile3GPP	406
Profile3GPP2	413
ProfileIdentifier	419
protocolSubtypeElement	419
PSDetachReq	421
qaQmi3Gpp2TimeZone	422
qaQmiInterfaceInfo	422
qaQmiServingSystemParam	423
QmiCbkCatEventStatusReportInd	427
QmiCbkLocCradleMountInd	428
QmiCbkLocEventTimeSyncInd	428
QmiCbkLocInjectSensorDataInd	429
QmiCbkLocInjectTimeInd	430
QmiCbkLocPositionReportInd	431
QmiCbkLocSensorStreamingInd	437
QmiCbkNasLTECphyCalInfo	438
QmiCbkSwiOmaDmEventStatusReportInd	439
QmiCbkSwiOmaDmEventStatusReportIndExt	439
QmiCbkWdsStatisticsIndState	439
qmifwinfo_s	440
QmiNas3GppNetworkInfo	441
QmiNasGetRFBandInfoResp	443
QmiNasPerformNetworkScanResp	443
QmiWdsIpAddressInfo	443
qmiWdsRunTimeSettings	444
QosClassID	448
QosEventInfo	449
QosFlowInfo	451
QosFlowInfoState	452
QosMap	452
readResult	453
readTransparentInfo	454
redirNumInfo	454
registerRefresh	456
remainingRetries	458
remotePartyName	459
remotePartyNum	460
ReqFieldsList	461
RespFieldsList	462
RFBandInfoElements	462
roamIndList	463
RoamingInfo	464
roamTimer	464
RSRPThresh	466
rsrqInformation	467
RSRQThresh	467
RSSIThresh	468
RXAGCList	469
RXAVCList	470
rxInfo	470
RXPCMIIRFitr	472
rxSignalStrengthListElement	474
sApnExtraParams	474
satelliteInfo	476
sensorDataUsage_s	478
serialNumbersInfo	479

serviceProviderName	480
ServingSystemInfo	481
servSystem	483
sessionInfo	485
sessionInfoExt	485
sessionInfoTlv	485
sessionInfoTlvExt	486
SetAudioPathConfigReq	486
SetAudioProfileReq	489
SetAudioVolTLBConfigReq	491
SetAudioVolTLBConfigResp	492
setCustomSettingV2	492
SetIMSSMSConfigReq	493
SetIMSSMSConfigResp	494
SetIMSUserConfigReq	494
SetIMSUserConfigResp	495
SetIMSVoIPConfigReq	495
SetIMSVoIPConfigResp	499
SetM2MAudioAVCFGReq	499
SetM2MAudioLPBKReq	500
SetM2MAudioProfileReq	501
SetM2MAudioVolumeReq	502
SetM2MAVMuteReq	503
SetM2MSpkrGainReq	504
setPINProtection	504
SetRegMgrConfigReq	505
SetRegMgrConfigResp	506
setSignalStrengthInfo	507
SetSIPConfigReq	512
SetSIPConfigResp	513
sGetDeviceSeriesResult	513
sidNid	514
sigInfo	514
signalInfo	516
SignalStrengthDataType	516
slotInfo	517
slqsautoconnect	519
SLQSSDeleteProfileParams	519
slqsfwinfo_s	520
SlqsNas3GppNetworkInfo	521
SlqsNas3GppNetworkRAT	524
SlqsNasPcsDigit	524
slqssendasyncsmsparams_s	526
slqssendsmsparams_s	529
slqsSessionStateInfo	530
slqsSignalStrengthInfo	531
SLQSSignalStrengthsIndReq	534
SLQSSignalStrengthsInformation	536
slqsWdsEventInfo	537
SMSAsyncRawSend_s	539
SMSCAddress	542
SMSEtwMessage	543
SMSEtwPImn	543
SMSEventInfo_s	545
smsMaxStorageSizeReq	546
smsMaxStorageSizeResp	547
SMSMemoryInfo	548
SMSMessageMode	548

smsMsgprotocolResp	548
SMSMTMessage	549
SMSOnIMS	549
smsRouteEntry	551
smsSetRoutesReq	553
SMSTransferRouteMTMessage	553
sQosFlowStat	554
sQosStat	555
SrvStatusInfo	557
ssdatasession_params	557
SupportedMsgList	560
SUPInfo	561
SV	562
SVInfo	563
svUsedforFix_s	564
SWI_STRUCT_CarrierImage	565
swiModemStatusResp	566
SwiOTAMsg_s	566
swiPDPRuntimeSettingsReq	567
swiPDPRuntimeSettingsResp	568
swiQosFilter	571
swiQosFlow	574
swiQosGranted	578
swiQosIds	578
swiQosModifyReq	578
swiQosReq	579
swiRMTrasnferStaticsReq	580
sysInfoCommon	581
TDSCDMAECIOThresh	583
TDSCDMARSCPTThresh	584
TDSCDMARSSIThresh	584
TDSCDMASigInfoExt	585
TDSCDMASINRThresh	586
TFTIDParams	586
tokenBucket	588
Tos	589
TransferStatInd	589
TransferStatsDataType	590
TrStatInd	590
trueIMSI	591
TXAGCList	592
txInfo	593
TXPCMIIRFtr	594
UIMAuthenticateReq	596
UIMAuthenticateResp	597
UIMChangePinReq	598
UIMDepersonalizationReq	599
UIMDepersonalizationResp	599
UIMEventRegisterReqResp	599
UIMGetCardStatusResp	600
UIMGetFileAttributesReq	601
UIMGetFileAttributesResp	601
UIMPinResp	602
UIMPowerDownReq	603
UIMPowerUpReq	603
UIMReadTransparentReq	604
UIMReadTransparentResp	605
UIMRefreshCompleteReq	606

UIMRefreshEvent	607
UIMRefreshGetLastEventReq	609
UIMRefreshGetLastEventResp	609
UIMRefreshOKReq	610
UIMRefreshRegisterReq	610
UIMSessionInformation	611
UIMSetPinProtectionReq	612
UIMStatusChangeInfo	613
UIMUnblockPinReq	613
UIMVerifyPinReq	614
UMTSInfo	615
UMTSInstInfo	617
umtsLTENbrCell	618
UMTSMinQoS	619
UMTSQoS	622
UMTSReqQoSSigInd	626
unblockUIMPIN	626
UniversalTime	627
USBCompConfig	629
USBCompParams	629
USSDNoWaitIndicationInfo	632
USSDRespFNetwork	632
USSInfo	634
USSResp	634
UUSInfo	635
verifyUIMPIN	636
voiceALSSelectLineInfo	637
voiceALSSetLineSwitchInfo	638
voiceAnswerCall	638
voiceBindSubscriptionInfo	639
voiceBurstDTMFInfo	639
voiceCallInfoReq	640
voiceCallInfoResp	640
voiceCallRequestParams	644
voiceCallResponseParams	646
voiceContDTMFInfo	647
voiceDTMFEventInfo	648
voiceFlashInfo	649
voiceGetAllCallInfo	650
voiceGetCallBarringReq	653
voiceGetCallBarringResp	654
voiceGetCallFWReq	655
voiceGetCallFWResp	657
voiceGetCallWaitInfo	659
voiceGetCLIPResp	660
voiceGetCLIRResp	663
voiceGetCNAPResp	664
voiceGetCOLPResp	666
voiceGetCOLRResp	667
voiceGetConfigReq	669
voiceGetConfigResp	671
voiceIndicationRegisterInfo	673
voiceInfoRec	674
voiceManageCallsReq	676
voiceManageCallsResp	678
voiceOrigUSSDNoWaitInfo	678
voiceOTASPStatusInfo	679
voicePrivacyInfo	680

voiceSetAllCallStatusCbkInfo . . . . .	680
voiceSetCallBarringPwdInfo . . . . .	683
voiceSetCallBarringPwdResp . . . . .	684
voiceSetConfigReq . . . . .	685
voiceSetConfigResp . . . . .	687
voiceSetPrefPrivacy . . . . .	689
voiceSetSUPSServiceReq . . . . .	690
voiceSetSUPSServiceResp . . . . .	692
voiceStopContDTMFInfo . . . . .	694
voiceSUPSInfo . . . . .	694
voiceSUPSNotification . . . . .	697
wcdmaCellInfo . . . . .	699
WCDMAECIOTresh . . . . .	700
WCDMAInfoLTENeighborCell . . . . .	700
wcdmaLongMsgDecodingParams . . . . .	701
wcdmaMsgDecodingParams . . . . .	703
wcdmaMsgEncodingParams . . . . .	705
WCDMARSSITresh . . . . .	706
WCDMASysInfo . . . . .	706
WdsByteTotals . . . . .	711
WdsByteTotalsElmnts . . . . .	712
WdsConnectionRate . . . . .	712
WdsConnectionRateElmnts . . . . .	713
WDSGetLoopbackData . . . . .	714
WdsIpAddressInfoReq . . . . .	715
WdsPktStatisticsElmnts . . . . .	715
WdsPktStatisticsReq . . . . .	718
WdsPktStatisticsResp . . . . .	718
WdsProfileParam . . . . .	719
WdsRunTimeSettings . . . . .	719
wdsSetEventReportReq . . . . .	720
WDSSetLoopbackData . . . . .	723
WDSSWICurrentChannelRates . . . . .	725



## Chapter 5

# File Index

### 5.1 File List

Here is a list of all files with brief descriptions:

<a href="#">apdoxypages.c</a>	Contains the module declaration for the Doxygen output. Also contains the content of the main page and related pages . . . . .	727
<a href="#">qaCbkCatEventReportInd.h</a>		727
<a href="#">qaCbkSwiOmaDmEventReportInd.h</a>		729
<a href="#">qaGobiApiAudio.h</a>	Audio Service API function prototypes . . . . .	730
<a href="#">qaGobiApiCat.h</a>	Card Application Toolkit API function headers . . . . .	733
<a href="#">qaGobiApiCbk.h</a>	Callback Service API function prototypes . . . . .	735
<a href="#">qaGobiApiDcs.h</a>	Device Connectivity Service API function prototypes . . . . .	822
<a href="#">qaGobiApiDms.h</a>	Device Management Service API function prototypes . . . . .	835
<a href="#">qaGobiApiFms.h</a>	Firmware Management Service API function prototypes . . . . .	875
<a href="#">qaGobiApiIms.h</a>	IMS Service API function prototypes . . . . .	891
<a href="#">qaGobiApiImsa.h</a>	IMSA Service API function prototypes . . . . .	897
<a href="#">qaGobiApiLoc.h</a>	Location API function prototypes . . . . .	901
<a href="#">qaGobiApiNas.h</a>	Network Access Service API function prototypes . . . . .	904
<a href="#">qaGobiApiOmadm.h</a>	Open Mobile Alliance Device Management Service API function prototypes . . . . .	954
<a href="#">qaGobiApiPds.h</a>	Position Determination Service API function prototypes . . . . .	958
<a href="#">qaGobiApiQos.h</a>	Quality of Service API function prototypes . . . . .	975
<a href="#">qaGobiApiRms.h</a>	Remote Management Service API function prototypes . . . . .	982
<a href="#">qaGobiApiSar.h</a>	Specific Absorption Rate API function prototypes . . . . .	984
<a href="#">qaGobiApiSms.h</a>	Short Message Service API function prototypes . . . . .	986

<a href="#">qaGobiApiSwi.h</a>	
SWI API function prototypes	1010
<a href="#">qaGobiApiSwiAudio.h</a>	
M2M Audio Service API function prototypes	1011
<a href="#">qaGobiApiSwiOmadms.h</a>	
SWI Open Mobile Alliance Device Management Service API function prototypes	SWI OMA-DM
QMI Service revision 1.6	1017
<a href="#">qaGobiApiTableBandClasses.h</a>	
Network Access Service API Band Classes table	1029
<a href="#">qaGobiApiTableCallControlReturnReasons.h</a>	
Call Control Return Reasons table	1032
<a href="#">qaGobiApiTableCallEndReasons.h</a>	
Wireless Data Service Call End Reasons	1033
<a href="#">qaGobiApiTableCarrierCodes.h</a>	
Carrier Codes table	1048
<a href="#">qaGobiApiTableCodingScheme.h</a>	
Data Coding Scheme	1050
<a href="#">qaGobiApiTableGpsCapabilityCodes.h</a>	
Position Determination Service API GPS Capability Codes	1052
<a href="#">qaGobiApiTablePowerModes.h</a>	
Device Management Service API Power Modes table	1053
<a href="#">qaGobiApiTableRadioInterfaces.h</a>	
Network Access Service API Radio Interfaces table	1053
<a href="#">qaGobiApiTableRegionCodes.h</a>	
Region Codes table	1054
<a href="#">qaGobiApiTableServiceOptions.h</a>	
Voice Service Options	1054
<a href="#">qaGobiApiTableSupServiceInfoClasses.h</a>	
Voice Supplementary Service Information Classes	1057
<a href="#">qaGobiApiTableSwiAudio.h</a>	
Swi Audio related tables	1057
<a href="#">qaGobiApiTableSwiOMADMUpdateCompleteStatus.h</a>	
Update Complete Status table	1058
<a href="#">qaGobiApiTableVoiceCallEndReasons.h</a>	
Voice Service Call and supplementary services end reasons	1059
<a href="#">qaGobiApiUim.h</a>	
Uim Service API function prototypes	1066
<a href="#">qaGobiApiVoice.h</a>	
Voice Service API function prototypes	1076
<a href="#">qaGobiApiWds.h</a>	
Wireless Data Service API function prototypes	1099
<a href="#">qaNasGetRFBandInfo.h</a>	1150
<a href="#">qaNasPerformNetworkScan.h</a>	1150
<a href="#">qmerrno.h</a>	1151
<a href="#">SwiDataTypes.h</a>	
SWI data types	1158
<a href="#">SWIWWANCMAPI.h</a>	1160



## Chapter 6

# Module Documentation

### 6.1 Device Connectivity Service (DCS)

#### Files

- file [qaGobiApiDcs.h](#)  
*Device Connectivity Service API function prototypes.*

#### 6.1.1 Detailed Description

## 6.2 Wireless Data Service (WDS)

### Files

- file [qaGobiApiTableCallEndReasons.h](#)  
*Wireless Data Service Call End Reasons.*
- file [qaGobiApiTableCarrierCodes.h](#)  
*Carrier Codes table.*
- file [qaGobiApiTableRegionCodes.h](#)  
*Region Codes table.*
- file [qaGobiApiWds.h](#)  
*Wireless Data Service API function prototypes.*

### 6.2.1 Detailed Description

## 6.3 Device Management Service (DMS)

### Files

- file [qaGobiApiTablePowerModes.h](#)  
*Device Management Service API Power Modes table.*
- file [qaGobiApiDms.h](#)  
*Device Management Service API function prototypes.*

### 6.3.1 Detailed Description

## 6.4 Network Access Service (NAS)

### Files

- file [qaGobiApiTableRadioInterfaces.h](#)  
*Network Access Service API Radio Interfaces table.*
- file [qaGobiApiTableBandClasses.h](#)  
*Network Access Service API Band Classes table.*
- file [qaGobiApiNas.h](#)  
*Network Access Service API function prototypes.*

### 6.4.1 Detailed Description

## 6.5 CallBack registration (CBK)

### Files

- file [qaGobiApiCbK.h](#)  
*Callback Service API function prototypes.*

### 6.5.1 Detailed Description

## 6.6 Short Message Service (SMS)

### Files

- file [qaGobiApiSms.h](#)  
*Short Message Service API function prototypes.*

### 6.6.1 Detailed Description

## 6.7 Position Determination Service (PDS)

### Files

- file [qaGobiApiTableGpsCapabilityCodes.h](#)  
*Position Determination Service API GPS Capability Codes.*
- file [qaGobiApiPds.h](#)  
*Position Determination Service API function prototypes.*

### 6.7.1 Detailed Description

## 6.8 Card Application Toolkit (CAT)

### Files

- file [qaGobiApiCat.h](#)  
*Card Application Toolkit API function headers.*

### 6.8.1 Detailed Description



## 6.9 Remote Management Service (RMS)

### Files

- file [qaGobiApiRms.h](#)  
*Remote Management Service API function prototypes.*

### 6.9.1 Detailed Description

## 6.10 Firmware Management Service (FMS)

### Files

- file [qaGobiApiFms.h](#)  
*Firmware Management Service API function prototypes.*

### 6.10.1 Detailed Description

## 6.11 Open Mobile Alliance Service (OMA)

### Files

- file [qaGobiApiOmadm.h](#)

*Open Mobile Alliance Device Management Service API function prototypes.*

### 6.11.1 Detailed Description

## 6.12 Specific Absorption Rate (SAR)

### Files

- file [qaGobiApiSar.h](#)  
*Specific Absorption Rate API function prototypes.*

### 6.12.1 Detailed Description

## 6.13 SWI Open Mobile Alliance Service (SWIOMA)

### Files

- file [qaGobiApiTableSwiOMADMUpdateCompleteStatus.h](#)  
*Update Complete Status table.*
- file [qaGobiApiSwiOmadms.h](#)  
*SWI Open Mobile Alliance Device Management Service API function prototypes SWI OMA-DM QMI Service revision 1.6.*

### 6.13.1 Detailed Description

## 6.14 Voice Service (VOICE)

### Files

- file [qaGobiApiTableServiceOptions.h](#)  
*Voice Service Options.*
- file [qaGobiApiTableVoiceCallEndReasons.h](#)  
*Voice Service Call and supplementary services end reasons.*
- file [qaGobiApiTableCodingScheme.h](#)  
*Data Coding Scheme.*
- file [qaGobiApiTableCallControlReturnReasons.h](#)  
*Call Control Return Reasons table.*
- file [qaGobiApiTableSupServiceInfoClasses.h](#)  
*Voice Supplementary Service Information Classes.*
- file [qaGobiApiVoice.h](#)  
*Voice Service API function prototypes.*

### 6.14.1 Detailed Description

## 6.15 Non-service specific APIs (SWI)

### Files

- file [qaGobiApiSwi.h](#)  
*SWI API function prototypes.*

### 6.15.1 Detailed Description

## 6.16 User Identity Module Service (UIM)

### Files

- file [qaGobiApiUim.h](#)  
*Uim Service API function prototypes.*

### 6.16.1 Detailed Description



## 6.17 Audio Service (AUDIO)

### Files

- file [qaGobiApiAudio.h](#)  
*Audio Service API function prototypes.*

### 6.17.1 Detailed Description

## 6.18 Quality of Service (QOS)

### Files

- file [qaGobiApiQos.h](#)  
*Quality of Service API function prototypes.*

### 6.18.1 Detailed Description

## 6.19 IMS Service (IMS)

### Files

- file [qaGobiApilms.h](#)  
*IMS Service API function prototypes.*
- file [qaGobiApilmsa.h](#)  
*IMSA Service API function prototypes.*

### 6.19.1 Detailed Description

## 6.20 SWI Audio Service(SWIAUDIO)

### Files

- file [qaGobiApiTableSwiAudio.h](#)  
*Swi Audio related tables.*
- file [qaGobiApiSwiAudio.h](#)  
*M2M Audio Service API function prototypes.*

### 6.20.1 Detailed Description

## 6.21 Location Service(LOC)

### Files

- file [qaGobiApiLoc.h](#)  
*Location API function prototypes.*

### 6.21.1 Detailed Description



## Chapter 7

# Namespace Documentation

### 7.1 Tables Namespace Reference

#### 7.1.1 Detailed Description

[Tables](#) referenced in the API function headers:

- Table 1 - Call End Reason Codes The reason a call (either in process or connected) was ended. [qaGobiApiTableCallEndReasons.h](#)
- Table 2 - Carrier codes List of carrier identification codes. [qaGobiApiTableCarrierCodes.h](#)
- Table 3 - Region codes List of region identification codes. [qaGobiApiTableRegionCodes.h](#)
- Table 4 - GPS capability codes List of GPS capability codes. [qaGobiApiTableGpsCapabilityCodes.h](#)
- Table 5 - Radio Interfaces List of radio interface technologies. [qaGobiApiTableRadioInterfaces.h](#)
- Table 6 - Band classes List of band classes. [qaGobiApiTableBandClasses.h](#)
- Table 7 - Power modes List of operating modes. [qaGobiApiTablePowerModes.h](#)
- Table 8 - Service Options List of Service Options. [qaGobiApiTableServiceOptions.h](#)
- Table 9 - Voice Call End Reason List of Voice Call End Reason. [qaGobiApiTableVoiceCallEndReasons.h](#)
- Table 10 - Data Coding Scheme List of Data Coding Scheme. [qaGobiApiTableCodingScheme.h](#)
- Table 11 - Call Control Return Reasons List of Voice Call Control Return Reasons. [qaGobiApiTableCallControlReturnReasons.h](#)
- Table 12 - Supplementary Service Information Classes List of Voice Supplementary Service Information Classes. [qaGobiApiTableSupServiceInfoClasses.h](#)
- Table 13 - Audio Calibration Data Base(ACDB) Device List of ACDB devices. [qaGobiApiTableSwiAudio.h](#)
- Table 14 - PIFACE List of physical interfaces. [qaGobiApiTableSwiAudio.h](#)





## Chapter 8

# Data Structure Documentation

### 8.1 `_getIndicationRegResp` Struct Reference

#### Data Fields

- [BYTE](#) \* [pRegTransLayerInfoEvt](#)
- [BYTE](#) \* [pRegTransNWRegInfoEvt](#)
- [BYTE](#) \* [pRegCallStatInfoEvt](#)

#### 8.1.1 Detailed Description

This structure contains Get Indication Register Response parameters

#### Parameters

<i>pRegTrans-LayerInfoEvt</i>	- <ul style="list-style-type: none"><li>• Optional 1 BYTE parameter indicating registration status of transport layer information events</li><li>• Values:<ul style="list-style-type: none"><li>– 0x00 - Disabled</li><li>– 0x01 - Enabled</li></ul></li><li>• function <a href="#">SLQSGetIndicationRegister()</a> returns a default value 0xFF if this parameter is allocated memory in the structure and no response is received from the device.</li></ul>
-------------------------------	--

<i>pRegTransNW-RegInfoEvt</i>	- <ul style="list-style-type: none"> <li>Optional 1 BYTE parameter indicating registration status of transport network registration information events</li> <li>Values: <ul style="list-style-type: none"> <li>0x00 - Disabled</li> <li>0x01 - Enabled</li> </ul> </li> <li>function <a href="#">SLQSGetIndicationRegister()</a> returns a default value 0xFF if this parameter is allocated memory in the structure and no response is received from the device.</li> </ul>
<i>pRegCallStat-InfoEvt</i>	- <ul style="list-style-type: none"> <li>Optional 1 BYTE parameter indicating registration status of call status information events</li> <li>Values: <ul style="list-style-type: none"> <li>0x00 - Disabled</li> <li>0x01 - Enabled</li> </ul> </li> <li>function <a href="#">SLQSGetIndicationRegister()</a> returns a default value 0xFF if this parameter is allocated memory in the structure and no response is received from the device.</li> </ul>

## 8.1.2 Field Documentation

8.1.2.1 BYTE\* \_getIndicationRegResp::pRegCallStatInfoEvt

8.1.2.2 BYTE\* \_getIndicationRegResp::pRegTransLayerInfoEvt

8.1.2.3 BYTE\* \_getIndicationRegResp::pRegTransNWRegInfoEvt

## 8.2 \_GetProfileSettingIn Struct Reference

### Data Fields

- [BYTE ProfileType](#)
- [BYTE ProfileID](#)

### 8.2.1 Detailed Description

This structure contains the input parameters for SLQSGetProfileSettings

#### Parameters

<i>ProfileType</i>	<ul style="list-style-type: none"> <li>Identifies the technology type of the profile <ul style="list-style-type: none"> <li>0x00 - 3GPP</li> <li>0x01 - 3GPP2</li> </ul> </li> </ul>
--------------------	--

<i>ProfileID</i>	<ul style="list-style-type: none"> <li>index identifying the profile</li> </ul>
------------------	---

## 8.2.2 Field Documentation

8.2.2.1 **BYTE** \_GetProfileSettingIn::ProfileID

8.2.2.2 **BYTE** \_GetProfileSettingIn::ProfileType

## 8.3 \_GetProfileSettingOut Struct Reference

### Data Fields

- [QmiProfileInfo curProfile](#)
- WORD** \* [pExtErrCode](#)

### 8.3.1 Detailed Description

This structure contains the profile settings retrieved by the API SLQSGetProfileSettings

#### Parameters

<i>curProfile</i>	<ul style="list-style-type: none"> <li>Structure containing details of the profile</li> <li>See <a href="#">QmiProfileInfo</a> for more details</li> </ul>
<i>pExtErrCode</i>	<ul style="list-style-type: none"> <li>pointer to a 2 byte extended error code</li> <li>Error code will only will be present if error code eQCWWAN_ERR_QMI_EXTENDED-INTERNAL is returned by device.</li> <li>See <a href="#">qm_wds_ds_profile_extended_err_codes</a> enum in <a href="#">qmerrno.h</a> for received error description.</li> </ul>

## 8.3.2 Field Documentation

8.3.2.1 **QmiProfileInfo** \_GetProfileSettingOut::curProfile

8.3.2.2 **WORD**\* \_GetProfileSettingOut::pExtErrCode

## 8.4 \_getTransLayerInfoResp Struct Reference

### Data Fields

- BYTE** \* [pRegInd](#)
- [transLayerInfo](#) \* [pTransLayerInfo](#)

### 8.4.1 Detailed Description

This structure contains Get Transport Layer Info Response parameters

## Parameters

<i>pRegInd</i>	- <ul style="list-style-type: none"> <li>Optional parameter indicating if transport layer is registered</li> <li>Values: <ul style="list-style-type: none"> <li>0x00 - Transport layer is not registered</li> <li>0x01 - Transport layer is registered</li> </ul> </li> <li>function <a href="#">SLQSGetTransLayerInfo()</a> returns a default value 0xFF if no response is received from the device.</li> </ul>
<i>pTransLayerInfo</i>	<ul style="list-style-type: none"> <li>Pointer to structure of transLayerInfo. <ul style="list-style-type: none"> <li>Optional parameter</li> <li>See <a href="#">transLayerInfo</a> for more information</li> </ul> </li> <li>function <a href="#">SLQSGetTransLayerInfo()</a> returns a default value 0xFF for parameter values if no response is received from the device.</li> </ul>

## 8.4.2 Field Documentation

8.4.2.1 `BYTE* _getTransLayerInfoResp::pRegInd`8.4.2.2 `transLayerInfo* _getTransLayerInfoResp::pTransLayerInfo`

## 8.5 \_getTransNWRegInfoResp Struct Reference

## Data Fields

- `BYTE * pRegStatus`

## 8.5.1 Detailed Description

This structure contains transport network registration info parameter

## Parameters

<i>pRegStatus</i>	- <ul style="list-style-type: none"> <li>Optional 1 BYTE parameter indicating transport layer network registration status</li> <li>Values: <ul style="list-style-type: none"> <li>0x00 - No service</li> <li>0x01 - In progress</li> <li>0x02 - Failed</li> <li>0x03 - Limited Service</li> <li>0x04 - Full Service</li> </ul> </li> <li>function <a href="#">SLQSGetTransNWRegInfo()</a> returns a default value 0xFF if no response is received from the device.</li> </ul>
-------------------	---

## 8.5.2 Field Documentation

### 8.5.2.1 BYTE\* \_getTransNWRegInfoResp::pRegStatus

## 8.6 \_modemTempNotification Struct Reference

### Data Fields

- [BYTE ModemTempState](#)
- [WORD ModemTemperature](#)

### 8.6.1 Detailed Description

Contains the parameters passed for SLQSSetModemTempCallback by the device.

#### Parameters

<i>ModemTemp- State</i>	<ul style="list-style-type: none"> <li>• provides the temperature state of the modem</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0 - unknown</li> <li>– 1 - normal</li> <li>– 2 - high(warning)</li> <li>– 3 - high(critical)</li> <li>– 4 - low(critical)</li> </ul> </li> </ul>
<i>Modem- Temperature</i>	<ul style="list-style-type: none"> <li>• provides the temperature of the modem</li> </ul>

#### Note

None

## 8.6.2 Field Documentation

### 8.6.2.1 WORD \_modemTempNotification::ModemTemperature

### 8.6.2.2 BYTE \_modemTempNotification::ModemTempState

## 8.7 \_packetSrvStatus Struct Reference

### Data Fields

- [qaQmiInterfaceInfo \\* pQmiInterfaceInfo](#)
- [BYTE connStatus](#)
- [BYTE reconfigReqd](#)
- [WORD sessionEndReason](#)
- [WORD verboseSessnEndReasonType](#)
- [WORD verboseSessnEndReason](#)
- [BYTE ipFamily](#)

- [WORD techName](#)
- [BYTE bearerID](#)

### 8.7.1 Detailed Description

Contains the parameters passed for SLQSSetPacketSrvStatusCallback by the device.

#### Parameters

<i>pQmiInterface-Info</i>	<ul style="list-style-type: none"> <li>• See <a href="#">qaQmiInterfaceInfo</a> for more information</li> </ul>
<i>connStatus</i>	<ul style="list-style-type: none"> <li>• Current Link Status <ul style="list-style-type: none"> <li>– 1 - Disconnected</li> <li>– 2 - Connected</li> <li>– 3 - Suspended</li> <li>– 4 - Authenticating</li> </ul> </li> </ul>
<i>reconfigReqd</i>	<ul style="list-style-type: none"> <li>• Indicates if the network interface on the host needs to be reconfigured <ul style="list-style-type: none"> <li>– 0 - No need to reconfigure</li> <li>– 1 - Reconfiguration required</li> </ul> </li> </ul>
<i>sessionEnd-Reason</i>	<ul style="list-style-type: none"> <li>• See <a href="#">qaGobiApiTableCallEndReasons.h</a> for Call End Reason, 0xFFFF means invalid value</li> </ul>
<i>verboseSessn-EndReasonType</i>	<ul style="list-style-type: none"> <li>• Call End Reason Type <ul style="list-style-type: none"> <li>– 0 - Unspecified</li> <li>– 1 - Mobile IP</li> <li>– 2 - Internal</li> <li>– 3 - Call Manager defined</li> <li>– 6 - 3GPP Specification defined</li> <li>– 7 - PPP</li> <li>– 8 - EHRPD</li> <li>– 9 - IPv6</li> <li>– 0xFFFF - invalid value</li> </ul> </li> </ul>

<i>verboseSessn-EndReason</i>	<ul style="list-style-type: none"> <li>See <a href="#">qaGobiApiTableCallEndReasons.h</a> for verbose Call End Reason. The values depend on verboseSessnEndReasonType parameter 0xFFFF means invalid value</li> </ul>
<i>ipFamily</i>	<ul style="list-style-type: none"> <li>IP Family of the packet data connection <ul style="list-style-type: none"> <li>4 - IPv4</li> <li>6 - IPv6</li> <li>0xFF - invalid value</li> </ul> </li> </ul>
<i>techName</i>	<ul style="list-style-type: none"> <li>Technology name of the packet data connection. <ul style="list-style-type: none"> <li>32767 - CDMA</li> <li>32764 - UMTS</li> <li>30592 - EPC</li> <li>30590 - EMBMS</li> <li>30584 - Modem Link Local</li> <li>0xFFFF - invalid value EPC is a logical interface to support LTE/eHRPD handoff. Modem Link is an interface for transferring data between entities on the AP and modem.</li> </ul> </li> </ul>
<i>bearerID</i>	<ul style="list-style-type: none"> <li>Bearer ID (3GPP) or RLP ID (3GPP2) of the packet data connection 0xFF means invalid value</li> </ul>

**Note**

Any parameter not returned by the device is returned as its maximum unsigned value by the callback.

**8.7.2 Field Documentation**

**8.7.2.1** `BYTE _packetSrvStatus::bearerID`

**8.7.2.2** `BYTE _packetSrvStatus::connStatus`

**8.7.2.3** `BYTE _packetSrvStatus::ipFamily`

**8.7.2.4** `qaQmiInterfaceInfo* _packetSrvStatus::pQmiInterfaceInfo`

**8.7.2.5** `BYTE _packetSrvStatus::reconfigReqd`

**8.7.2.6** `WORD _packetSrvStatus::sessionEndReason`

**8.7.2.7** `WORD _packetSrvStatus::techName`

**8.7.2.8** `WORD _packetSrvStatus::verboseSessnEndReason`

**8.7.2.9** `WORD _packetSrvStatus::verboseSessnEndReasonType`



## 8.8 \_qaQmi3GPP2BroadcastCfgInfo Struct Reference

### Data Fields

- [BYTE](#) `activated_ind`
- [WORD](#) `num_instances`
- struct [CDMABroadcastConfig](#) `CDMABroadcastConfig` [0x05]

### 8.8.1 Detailed Description

This structure contains the 3GPP2 Broadcast Configuration Information parameters

#### Parameters

<i>activated_ind</i>	<ul style="list-style-type: none"> <li>• Broadcast SMS <ul style="list-style-type: none"> <li>– 0x00 - Deactivated</li> <li>– 0x01 - Activated</li> </ul> </li> </ul>
<i>num_instances</i>	<ul style="list-style-type: none"> <li>• Number of sets (N) of parameters Following each set describes one entry in the broadcast configuration table. <ul style="list-style-type: none"> <li>– serviceCategory</li> <li>– language</li> <li>– selected</li> </ul> </li> </ul>
<i>broadcastConfig</i>	<ul style="list-style-type: none"> <li>• A <a href="#">CDMABroadcastConfig</a> structure array.</li> <li>• Further defined by the structure <a href="#">CDMABroadcastConfig</a></li> </ul>

### 8.8.2 Field Documentation

8.8.2.1 [BYTE](#) `_qaQmi3GPP2BroadcastCfgInfo::activated_ind`

8.8.2.2 struct [CDMABroadcastConfig](#) `_qaQmi3GPP2BroadcastCfgInfo::CDMABroadcastConfig`[0x05]

8.8.2.3 [WORD](#) `_qaQmi3GPP2BroadcastCfgInfo::num_instances`

## 8.9 \_qaQmi3GPPBroadcastCfgInfo Struct Reference

### Data Fields

- [BYTE](#) `activated_ind`
- [WORD](#) `num_instances`
- struct [BroadcastConfig](#) `broadcastConfig` [0x05]

### 8.9.1 Detailed Description

This structure contains the 3GPP Broadcast Configuration Information parameters

## Parameters

<i>activated_ind</i>	<ul style="list-style-type: none"> <li>• Broadcast SMS <ul style="list-style-type: none"> <li>– 0x00 - Deactivated</li> <li>– 0x01 - Activated</li> </ul> </li> </ul>
<i>num_instances</i>	<ul style="list-style-type: none"> <li>• Number of sets (N) of parameters Following each set describes one entry in the broadcast configuration table. <ul style="list-style-type: none"> <li>– fromServiceId</li> <li>– toServiceId</li> <li>– selected</li> </ul> </li> </ul>
<i>broadcastConfig</i>	<ul style="list-style-type: none"> <li>• A <a href="#">BroadcastConfig</a> structure array.</li> <li>• Further defined by the structure <a href="#">BroadcastConfig</a></li> </ul>

## 8.9.2 Field Documentation

8.9.2.1 BYTE \_qaQmi3GPPBroadcastCfgInfo::activated\_ind

8.9.2.2 struct BroadcastConfig \_qaQmi3GPPBroadcastCfgInfo::broadcastConfig[0x05]

8.9.2.3 WORD \_qaQmi3GPPBroadcastCfgInfo::num\_instances

## 8.10 \_setIndicationRegReq Struct Reference

## Data Fields

- BYTE \* [pRegTransLayerInfoEvt](#)
- BYTE \* [pRegTransNWRegInfoEvt](#)
- BYTE \* [pRegCallStatInfoEvt](#)

## 8.10.1 Detailed Description

This structure contains Indication Register request parameters

## Parameters

<i>pRegTransLayerInfoEvt</i>	- <ul style="list-style-type: none"> <li>Optional 1 BYTE parameter indicating registration status of transport layer information events</li> <li>Values: <ul style="list-style-type: none"> <li>0x00 - Disabled</li> <li>0x01 - Enabled</li> <li>NULL - No change - specifying NULL indicates that the device will continue to use the existing setting (disable/enable) which has been previously set for the device</li> </ul> </li> </ul>
<i>pRegTransNWRegInfoEvt</i>	- <ul style="list-style-type: none"> <li>Optional 1 BYTE parameter indicating registration status of transport network registration information events</li> <li>Values: <ul style="list-style-type: none"> <li>0x00 - Disabled</li> <li>0x01 - Enabled</li> <li>NULL - No change - specifying NULL indicates that the device will continue to use the existing setting (disable/enable) which has been previously set for the device</li> </ul> </li> </ul>
<i>pRegCallStatInfoEvt</i>	- <ul style="list-style-type: none"> <li>Optional 1 BYTE parameter indicating registration status of call status information events</li> <li>Values: <ul style="list-style-type: none"> <li>0x00 - Disabled</li> <li>0x01 - Enabled</li> <li>NULL - No change - specifying NULL indicates that the device will continue to use the existing setting (disable/enable) which has been previously set for the device</li> </ul> </li> </ul>

## 8.10.2 Field Documentation

8.10.2.1 **BYTE\*** `_setIndicationRegReq::pRegCallStatInfoEvt`8.10.2.2 **BYTE\*** `_setIndicationRegReq::pRegTransLayerInfoEvt`8.10.2.3 **BYTE\*** `_setIndicationRegReq::pRegTransNWRegInfoEvt`8.11 `_slqs3GPPConfigItem` Struct Reference

## Data Fields

- WORD** \* `pLTEAttachProfile`
- WORD** \* `pProfileList`
- BYTE** \* `pDefaultPDNEnabled`
- BYTE** \* `p3gppRelease`
- WORD** `LTEAttachProfileListLen`
- WORD** \* `pLTEAttachProfileList`

### 8.11.1 Detailed Description

This structure contains the 3gpp Configuration Item information.

## Parameters

<i>pLTEAttach-Profile</i>	<ul style="list-style-type: none"> <li>• LTE Attach Profile <ul style="list-style-type: none"> <li>– points to a single WORD Value indicating the attached LTE Profile</li> <li>– Optional parameter with possible values 1-16</li> <li>– function <a href="#">SLQSGet3GPPConfigItem()</a> returns a default value 255 if no LTE Attach Profile is configured</li> </ul> </li> <li>• This setting is deprecated on MC/EM74xx</li> </ul>
<i>pProfileList</i>	<ul style="list-style-type: none"> <li>• Profile List <ul style="list-style-type: none"> <li>– an array of 4 profile configurations</li> <li>– Each element points to a single WORD value indicating profile</li> <li>– Optional parameter with possible values <ul style="list-style-type: none"> <li>* 1 - 16 (MC/EM73xx and before)</li> <li>* 1 - 24 (MC/EM74xx and onwards)</li> </ul> </li> <li>– function <a href="#">SLQSGet3GPPConfigItem()</a> returns a default value 255 if no 3gpp configuration is present</li> </ul> </li> </ul>
<i>pDefaultPDN-Enabled</i>	<ul style="list-style-type: none"> <li>• Always Connect Default PDN <ul style="list-style-type: none"> <li>– A single BYTE value indicating the status of Always connect default PDN <ul style="list-style-type: none"> <li>* 0 - disabled</li> <li>* 1 - enabled</li> </ul> </li> <li>– Optional parameter</li> <li>– function <a href="#">SLQSGet3GPPConfigItem()</a> returns a default value 255 if no 3gpp configuration is present</li> </ul> </li> </ul>
<i>p3gppRelease</i>	<ul style="list-style-type: none"> <li>• 3gpp release <ul style="list-style-type: none"> <li>– A single BYTE value indicating the 3gpp release <ul style="list-style-type: none"> <li>* 0 - Release 99</li> <li>* 1 - Release 5</li> <li>* 2 - Release 6</li> <li>* 3 - Release 7</li> <li>* 4 - Release 8</li> </ul> </li> <li>– Optional parameter</li> <li>– function <a href="#">SLQSGet3GPPConfigItem()</a> returns a default value 255 if no 3gpp configuration is present</li> </ul> </li> </ul>
<i>pLTEAttach-ProfileList</i>	<ul style="list-style-type: none"> <li>• pointer to WORD array indicating LTE Attach Profile List <ul style="list-style-type: none"> <li>– Optional parameter</li> <li>– possible values: 1-24</li> <li>– This setting is only supported for MC/EM74xx onwards</li> </ul> </li> </ul>
	<ul style="list-style-type: none"> <li>– The new equivalent option for "pLTEAttachProfile" on 74xx modems is "pLTE-AttachProfileList". Please provide attach profiles in order of decreasing priority in this list.</li> </ul>

<i>LTEAttachProfileListLen</i>	<ul style="list-style-type: none"> <li>Number of element in pLTEAttachProfileList <ul style="list-style-type: none"> <li>valid range: 0-2</li> <li>This setting is only supported for MC/EM74xx onwards</li> </ul> </li> </ul>
--------------------------------	--

## 8.11.2 Field Documentation

8.11.2.1 **WORD** \_slqs3GPPConfigItem::LTEAttachProfileListLen

8.11.2.2 **BYTE\*** \_slqs3GPPConfigItem::p3gppRelease

8.11.2.3 **BYTE\*** \_slqs3GPPConfigItem::pDefaultPDNEnabled

8.11.2.4 **WORD\*** \_slqs3GPPConfigItem::pLTEAttachProfile

8.11.2.5 **WORD\*** \_slqs3GPPConfigItem::pLTEAttachProfileList

8.11.2.6 **WORD\*** \_slqs3GPPConfigItem::pProfileList

## 8.12 \_slqsNetworkScanInfo Struct Reference

### Data Fields

- BYTE** \* pNetworkInfoInstances
- struct SlqsNas3GppNetworkInfo \* pNetworkInfo
- BYTE** \* pRATInstances
- struct SlqsNas3GppNetworkRAT \* pRATInfo
- BYTE** \* pPCSDigitInstances
- struct SlqsNasPcsDigit \* pPCSDigitInfo
- ULONG** \* pScanResult

### 8.12.1 Detailed Description

Contain the network scan information.

#### Parameters

<i>pNetworkInfoInstances</i> [IN/OUT]	<ul style="list-style-type: none"> <li>Upon input, maximum number of elements that the network info instance array can contain.</li> <li>Upon successful output, the actual number of elements in the network info instance array.</li> </ul>
---------------------------------------	---

<i>pNetworkInfo</i> [O-UT]	<ul style="list-style-type: none"> <li>• Network info instance array <ul style="list-style-type: none"> <li>– See <a href="#">SlqsNas3GppNetworkInfo</a> for more information</li> </ul> </li> </ul>
<i>pRATInstances</i> [-IN/OUT]	<ul style="list-style-type: none"> <li>• Upon input, maximum number of elements that the RAT info instance array can contain.</li> <li>• Upon successful output, the actual number of elements in the RAT info instance array.</li> </ul>
<i>pRATInfo</i> [OUT]	<ul style="list-style-type: none"> <li>• RAT info instance array <ul style="list-style-type: none"> <li>– See <a href="#">SlqsNas3GppNetworkRAT</a> for more information</li> </ul> </li> </ul>
<i>pPCSDigitInstances</i> [IN/O-UT]	<ul style="list-style-type: none"> <li>• Upon input, maximum number of elements that the PCS Digit info instance array can contain.</li> <li>• Upon successful output, the actual number of elements in the PCS Digit info instance array.</li> </ul>
<i>pPCSDigitInfo</i> [-OUT]	<ul style="list-style-type: none"> <li>• PCS Digit info instance array <ul style="list-style-type: none"> <li>– See <a href="#">SlqsNasPcsDigit</a> for more information</li> </ul> </li> </ul>
<i>pScanResult</i> [O-UT]	<ul style="list-style-type: none"> <li>• status of network scan</li> <li>• 0x00 - scan successful</li> <li>• 0x01 - scan was aborted</li> <li>• 0x02 - scan did not complete due to a radio link failure recovery in progress</li> </ul>

## 8.12.2 Field Documentation

8.12.2.1 **struct** `SlqsNas3GppNetworkInfo* _slqsNetworkScanInfo::pNetworkInfo`

8.12.2.2 **BYTE\*** `_slqsNetworkScanInfo::pNetworkInfoInstances`

8.12.2.3 **struct** `SlqsNasPcsDigit* _slqsNetworkScanInfo::pPCSDigitInfo`

8.12.2.4 **BYTE\*** `_slqsNetworkScanInfo::pPCSDigitInstances`

8.12.2.5 **struct** `SlqsNas3GppNetworkRAT* _slqsNetworkScanInfo::pRATInfo`

8.12.2.6 **BYTE\*** `_slqsNetworkScanInfo::pRATInstances`

8.12.2.7 **ULONG\*** `_slqsNetworkScanInfo::pScanResult`



## 8.13 \_SLQSOMADMSessionInfo Struct Reference

### Data Fields

- [BYTE](#) \* [pStatus](#)
- [WORD](#) \* [pUpdateCompleteStatus](#)
- [BYTE](#) \* [pSeverity](#)
- [WORD](#) \* [pSourceLength](#)
- [BYTE](#) \* [pSource](#)
- [WORD](#) \* [pPkgNameLength](#)
- [BYTE](#) \* [pPkgName](#)
- [WORD](#) \* [pPkgDescLength](#)
- [BYTE](#) \* [pPkgDescription](#)
- [WORD](#) \* [pDateLength](#)
- [BYTE](#) \* [pDate](#)
- [WORD](#) \* [pTimeLength](#)
- [BYTE](#) \* [pTime](#)
- [BYTE](#) \* [pSessionType](#)
- [BYTE](#) \* [pSessionState](#)
- [BYTE](#) \* [pRetryCount](#)

### 8.13.1 Detailed Description

Structure containing the OMA DM Session Info returned by the device. Also used as input parameter to specify the size of variable parameters. (ref. notes)

#### Parameters

<i>pStatus</i>	<ul style="list-style-type: none"><li>• 1 Byte parameter indicating status<ul style="list-style-type: none"><li>– 0x01 - No Firmware available</li><li>– 0x02 - Query Firmware Download</li><li>– 0x03 - Firmware Downloading</li><li>– 0x04 - Firmware Downloaded</li><li>– 0x05 - Query Firmware Update</li><li>– 0x06 - Firmware Updating</li><li>– 0x07 - Firmware Updated</li></ul></li></ul>
----------------	--

<i>pUpdate-CompleteStatus</i>	<ul style="list-style-type: none"> <li>• 2 byte parameter indicating Update Complete Status <ul style="list-style-type: none"> <li>– See <a href="#">qaGobiApiTableSwiOMADMUpdateCompleteStatus.h</a> Update Complete Status</li> </ul> </li> </ul>
<i>pSeverity</i>	<ul style="list-style-type: none"> <li>• 1 byte parameter indicating severity <ul style="list-style-type: none"> <li>– 0x01 - Mandatory</li> <li>– 0x02 - Optional</li> </ul> </li> </ul>
<i>pSourceLength</i>	<ul style="list-style-type: none"> <li>• 2 byte parameter indicating Length of Vendor Name String in Bytes.</li> </ul>
<i>pSource</i>	<ul style="list-style-type: none"> <li>• Variable length parameter indicating Vendor Name in ASCII</li> </ul>
<i>pPkgName-Length</i>	<ul style="list-style-type: none"> <li>• 2 byte parameter indicating Length of Package Name String in Bytes.</li> </ul>
<i>pPkgName</i>	<ul style="list-style-type: none"> <li>• Variable length parameter indicating Package Name in ASCII</li> </ul>
<i>pPkgDesc-Length</i>	<ul style="list-style-type: none"> <li>• 2 byte parameter indicating Length of Package Description String in Bytes.</li> </ul>
<i>pPkgDescription</i>	<ul style="list-style-type: none"> <li>• Variable length parameter indicating Package Description in ASCII</li> </ul>
<i>pDateLength</i>	<ul style="list-style-type: none"> <li>• 2 byte parameter indicating Length of Package Description String in Bytes.</li> </ul>
<i>pDate</i>	<ul style="list-style-type: none"> <li>• Variable length parameter indicating Package Description in ASCII</li> </ul>
<i>pTimeLength</i>	<ul style="list-style-type: none"> <li>• 2 byte parameter indicating Length of Time String in Bytes.</li> </ul>
<i>pTime</i>	<ul style="list-style-type: none"> <li>• Variable length parameter indicating Time String in ASCII</li> </ul>
<i>pSessionType</i>	<ul style="list-style-type: none"> <li>• 1 byte parameter reflects the last session started for Sprint <ul style="list-style-type: none"> <li>– 0x00 - No session since boot</li> <li>– 0x01 - Sprint CI-DC Session</li> <li>– 0x02 - Sprint CI-PRL Session</li> <li>– 0x03 - Sprint CI-FUMO Session</li> <li>– 0x04 - Sprint HFA-DC Session</li> <li>– 0x05 - Sprint HFA-PRL Session</li> <li>– 0x06 - Sprint HFA-FUMO Session</li> <li>– 0x07 - Sprint NI Session</li> </ul> </li> </ul>
	Generated on Wed Oct 28 2015 00:40:12 for LinuxQMISDK by Doxygen

<i>pSessionState</i>	<ul style="list-style-type: none"> <li>• 1 byte parameter indicating session state <ul style="list-style-type: none"> <li>– 0x01 - idle</li> <li>– 0x02 - active</li> <li>– 0x03 - pending</li> </ul> </li> </ul>
<i>pRetryCount</i>	<ul style="list-style-type: none"> <li>• 1 byte parameter indicating retries left count <ul style="list-style-type: none"> <li>– valid values 0 to 6</li> </ul> </li> </ul>

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

As input parameter the members pSourceLength, pPkgNameLength, pPkgDescLength, pDateLength, pTimeLength have to be specified. These should contain the initialized size of pSource, pPkgName, pPkgDescription, pDate, pTime respectively.

**8.13.2 Field Documentation**

8.13.2.1 **BYTE\*** \_SLQSOMADMSessionInfo::pDate

8.13.2.2 **WORD\*** \_SLQSOMADMSessionInfo::pDateLength

8.13.2.3 **WORD\*** \_SLQSOMADMSessionInfo::pPkgDescLength

8.13.2.4 **BYTE\*** \_SLQSOMADMSessionInfo::pPkgDescription

8.13.2.5 **BYTE\*** \_SLQSOMADMSessionInfo::pPkgName

8.13.2.6 **WORD\*** \_SLQSOMADMSessionInfo::pPkgNameLength

8.13.2.7 **BYTE\*** \_SLQSOMADMSessionInfo::pRetryCount

8.13.2.8 **BYTE\*** \_SLQSOMADMSessionInfo::pSessionState

8.13.2.9 **BYTE\*** \_SLQSOMADMSessionInfo::pSessionType

8.13.2.10 **BYTE\*** \_SLQSOMADMSessionInfo::pSeverity

8.13.2.11 **BYTE\*** \_SLQSOMADMSessionInfo::pSource

8.13.2.12 **WORD\*** \_SLQSOMADMSessionInfo::pSourceLength

8.13.2.13 **BYTE\*** \_SLQSOMADMSessionInfo::pStatus

8.13.2.14 **BYTE\*** `_SLQSOMADMSessionInfo::pTime`

8.13.2.15 **WORD\*** `_SLQSOMADMSessionInfo::pTimeLength`

8.13.2.16 **WORD\*** `_SLQSOMADMSessionInfo::pUpdateCompleteStatus`

## 8.14 `_SLQSOMADMSettings` Struct Reference

### Data Fields

- [ULONG](#) \* `pOMADMEEnabled`
- [BYTE](#) \* `pFOTAdownload`
- [BYTE](#) \* `pFOTAUpdate`
- [BYTE](#) \* `pAutosdm`
- [BYTE](#) \* `pFwAutoCheck`

### 8.14.1 Detailed Description

Structure containing the OMA DM settings retrieved from the device

#### Parameters

<i><code>pOMADM-Enabled[OUT]</code></i>	<ul style="list-style-type: none"> <li>• Optional 4 byte parameter indicating OMADM service enabled               <ul style="list-style-type: none"> <li>– 0x00000001 - Client-initiated device configuration</li> <li>– 0x00000002 - Network-initiated device configuration</li> <li>– 0x00000010 - Client-initiated FUMO</li> <li>– 0x00000020 - Network-initiated FUMO</li> </ul> </li> <li>• function <a href="#">SLQSOMADMGetSettings2()</a> returns a default value 0xFFFFFFFF in case this parameter is not returned by the modem.</li> </ul>
<i><code>pFOTAdownload[OUT]</code></i>	<ul style="list-style-type: none"> <li>• Optional 1 Byte parameter indicating support for FOTA Automatic download               <ul style="list-style-type: none"> <li>– 0x00 - Host permission required before downloading</li> <li>– 0x01 - Automatically start downloading, no host permission required</li> <li>– 0x02 - Automatically start downloading, while not roaming</li> <li>– 0x03 - Automatically reject download</li> <li>– 0x04 - Automatically reject download with “Enterprise Reject Policy”</li> </ul> </li> <li>• function <a href="#">SLQSOMADMGetSettings2()</a> returns a default value 0xFF in case this parameter is not returned by the modem.</li> </ul>

<i>pFOTAUpdate[OUT]</i>	<ul style="list-style-type: none"> <li>Optional 1 byte parameter indicating FOTA Automatic update <ul style="list-style-type: none"> <li>0x00 - User permission required before updating firmware</li> <li>0x01 - No user permission required before updating firmware</li> <li>0x02 - User permission required, auto update on power up</li> </ul> </li> <li>function <a href="#">SLQSOMADMGetSettings2()</a> returns a default value 0xFF in case this parameter is not returned by the modem.</li> </ul>
<i>pAutosdm[OUT]</i>	<ul style="list-style-type: none"> <li>Optional 1 byte parameter indicating OMA Automatic UI Alert Response <ul style="list-style-type: none"> <li>0x00 - Disabled</li> <li>0x01 - Enabled Accept</li> <li>0x02 - Enabled Reject</li> </ul> </li> <li>function <a href="#">SLQSOMADMGetSettings2()</a> returns a default value 0xFF in case this parameter is not returned by the modem.</li> </ul>
<i>pFwAutoCheck[OUT]</i>	<ul style="list-style-type: none"> <li>Optional 1 byte parameter indicating OMA Automatic Check for Firmware Update on Power-Up Response <ul style="list-style-type: none"> <li>0x00 - Disabled</li> <li>0x01 - Enabled</li> </ul> </li> <li>function <a href="#">SLQSOMADMGetSettings2()</a> returns a default value 0xFF in case this parameter is not returned by the modem.</li> </ul>

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**8.14.2 Field Documentation**

8.14.2.1 **BYTE\*** \_SLQSOMADMSettings::pAutosdm

8.14.2.2 **BYTE\*** \_SLQSOMADMSettings::pFOTAdownload

8.14.2.3 **BYTE\*** \_SLQSOMADMSettings::pFOTAUpdate

8.14.2.4 **BYTE\*** \_SLQSOMADMSettings::pFwAutoCheck

8.14.2.5 **ULONG\*** \_SLQSOMADMSettings::pOMADMAEnabled

**8.15 \_SLQSOMADMSettingsReqParams Struct Reference****Data Fields**

- [BYTE](#) FOTAdownload

- [BYTE FOTAUpdate](#)
- [BYTE \\* pAutosdm](#)

### 8.15.1 Detailed Description

Structure containing the OMA DM settings to be set on the device

#### Parameters

<i>FOTAdownload</i>	<ul style="list-style-type: none"> <li>• 1 Byte parameter indicating support for FOTA Automatic download <ul style="list-style-type: none"> <li>– 0x00 - Firmware auto download FALSE</li> <li>– 0x01 - Firmware auto download TRUE</li> </ul> </li> </ul>
<i>FOTAUpdate</i>	<ul style="list-style-type: none"> <li>• 1 byte parameter indicating FOTA Automatic update <ul style="list-style-type: none"> <li>– 0x00 - Firmware auto update FALSE</li> <li>– 0x01 - Firmware auto update TRUE</li> </ul> </li> </ul>
<i>pAutosdm[IN]</i>	<ul style="list-style-type: none"> <li>• Optional 1 byte parameter indicating OMA Automatic UI Alert Response <ul style="list-style-type: none"> <li>– 0x00 - Disabled</li> <li>– 0x01 - Enabled Accept</li> <li>– 0x02 - Enabled Reject</li> </ul> </li> </ul>

#### Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

#### See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

### 8.15.2 Field Documentation

8.15.2.1 [BYTE \\_SLQSOMADMSettingsReqParams::FOTAdownload](#)

8.15.2.2 [BYTE \\_SLQSOMADMSettingsReqParams::FOTAUpdate](#)

8.15.2.3 [BYTE\\* \\_SLQSOMADMSettingsReqParams::pAutosdm](#)

## 8.16 \_SLQSOMADMSettingsReqParams3 Struct Reference

#### Data Fields

- [BYTE FOTAdownload](#)
- [BYTE FOTAUpdate](#)
- [BYTE \\* pAutosdm](#)
- [BYTE \\* pFwAutoCheck](#)

### 8.16.1 Detailed Description

Structure containing the OMA DM settings to be set on the device

#### Parameters

<i>FOTAdownload</i>	<ul style="list-style-type: none"> <li>• 1 Byte parameter indicating support for FOTA Automatic download <ul style="list-style-type: none"> <li>– 0x00 - Firmware auto download FALSE</li> <li>– 0x01 - Firmware auto download TRUE</li> </ul> </li> </ul>
<i>FOTAUpdate</i>	<ul style="list-style-type: none"> <li>• 1 byte parameter indicating FOTA Automatic update <ul style="list-style-type: none"> <li>– 0x00 - Firmware auto update FALSE</li> <li>– 0x01 - Firmware auto update TRUE</li> </ul> </li> </ul>
<i>pAutosdm[IN]</i>	<ul style="list-style-type: none"> <li>• Optional 1 byte parameter indicating OMA Automatic UI Alert Response <ul style="list-style-type: none"> <li>– 0x00 - Disabled</li> <li>– 0x01 - Enabled Accept</li> <li>– 0x02 - Enabled Reject</li> </ul> </li> </ul>
<i>pFwAutoCheck[IN]</i>	<ul style="list-style-type: none"> <li>• Optional 1 byte parameter indicating OMA Automatic Check for Firmware Update on Power-Up Response <ul style="list-style-type: none"> <li>– 0x00 - Disabled</li> <li>– 0x01 - Enabled</li> </ul> </li> </ul>

#### Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

#### See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

### 8.16.2 Field Documentation

8.16.2.1 **BYTE** \_SLQSOMADMSettingsReqParams3::FOTAdownload

8.16.2.2 **BYTE** \_SLQSOMADMSettingsReqParams3::FOTAUpdate

8.16.2.3 **BYTE\*** \_SLQSOMADMSettingsReqParams3::pAutosdm

8.16.2.4 **BYTE\*** \_SLQSOMADMSettingsReqParams3::pFwAutoCheck

## 8.17 \_SLQSSwiGetHostDevInfoParams Struct Reference

## Data Fields

- [BYTE bManSize](#)
- [CHAR \\* pManString](#)
- [BYTE bModelSize](#)
- [CHAR \\* pModelString](#)
- [BYTE bSWVerSize](#)
- [CHAR \\* pSWVerString](#)
- [BYTE bPlasmaIDSize](#)
- [CHAR \\* pPlasmaIDString](#)

### 8.17.1 Field Documentation

8.17.1.1 **BYTE** \_SLQSSwiGetHostDevInfoParams::bManSize

8.17.1.2 **BYTE** \_SLQSSwiGetHostDevInfoParams::bModelSize

8.17.1.3 **BYTE** \_SLQSSwiGetHostDevInfoParams::bPlasmaIDSize

8.17.1.4 **BYTE** \_SLQSSwiGetHostDevInfoParams::bSWVerSize

8.17.1.5 **CHAR\*** \_SLQSSwiGetHostDevInfoParams::pManString

8.17.1.6 **CHAR\*** \_SLQSSwiGetHostDevInfoParams::pModelString

8.17.1.7 **CHAR\*** \_SLQSSwiGetHostDevInfoParams::pPlasmaIDString

8.17.1.8 **CHAR\*** \_SLQSSwiGetHostDevInfoParams::pSWVerString

## 8.18 \_SLQSSwiGetOSInfoParams Struct Reference

### Data Fields

- [BYTE bNameSize](#)
- [CHAR \\* pNameString](#)
- [BYTE bVersionSize](#)
- [CHAR \\* pVersionString](#)

### 8.18.1 Field Documentation

8.18.1.1 **BYTE** \_SLQSSwiGetOSInfoParams::bNameSize

8.18.1.2 **BYTE** \_SLQSSwiGetOSInfoParams::bVersionSize

8.18.1.3 **CHAR\*** \_SLQSSwiGetOSInfoParams::pNameString

8.18.1.4 **CHAR\*** \_SLQSSwiGetOSInfoParams::pVersionString

## 8.19 \_SLQSSwiGetSerialNoExtParams Struct Reference

### Data Fields

- [BYTE meidLength](#)
- [CHAR \\* pMeidString](#)



### 8.19.1 Detailed Description

This structure is used to store MEID Information

Parameters

<i>meidLength</i> [OUT]	<ul style="list-style-type: none"> <li>String length of the of MEID received</li> </ul>
<i>pMeidString</i> [OUT]	<ul style="list-style-type: none"> <li>Pointer to receive String containing the Mobile Equipment Identifier(MEID) of the device.</li> </ul>

### 8.19.2 Field Documentation

8.19.2.1 **BYTE** \_SLQSSwiGetSerialNoExtParams::meidLength

8.19.2.2 **CHAR\*** \_SLQSSwiGetSerialNoExtParams::pMeidString

## 8.20 \_SLQSSwiSetHostDevInfoParams Struct Reference

Data Fields

- [BYTE](#) bManSize
- [CHAR](#) \* pManString
- [BYTE](#) bModelSize
- [CHAR](#) \* pModelString
- [BYTE](#) bSWVerSize
- [CHAR](#) \* pSWVerString
- [BYTE](#) bPlasmaIDSize
- [CHAR](#) \* pPlasmaIDString

### 8.20.1 Field Documentation

8.20.1.1 **BYTE** \_SLQSSwiSetHostDevInfoParams::bManSize

8.20.1.2 **BYTE** \_SLQSSwiSetHostDevInfoParams::bModelSize

8.20.1.3 **BYTE** \_SLQSSwiSetHostDevInfoParams::bPlasmaIDSize

8.20.1.4 **BYTE** \_SLQSSwiSetHostDevInfoParams::bSWVerSize

8.20.1.5 **CHAR\*** \_SLQSSwiSetHostDevInfoParams::pManString

8.20.1.6 **CHAR\*** \_SLQSSwiSetHostDevInfoParams::pModelString

8.20.1.7 **CHAR\*** \_SLQSSwiSetHostDevInfoParams::pPlasmaIDString

8.20.1.8 **CHAR\*** \_SLQSSwiSetHostDevInfoParams::pSWVerString

## 8.21 \_SLQSSwiSetOSInfoParams Struct Reference

## Data Fields

- [BYTE](#) [bNameSize](#)
- [CHAR](#) \* [pNameString](#)
- [BYTE](#) [bVersionSize](#)
- [CHAR](#) \* [pVersionString](#)

### 8.21.1 Field Documentation

8.21.1.1 [BYTE](#) [\\_SLQSSwiSetOSInfoParams::bNameSize](#)

8.21.1.2 [BYTE](#) [\\_SLQSSwiSetOSInfoParams::bVersionSize](#)

8.21.1.3 [CHAR](#)\* [\\_SLQSSwiSetOSInfoParams::pNameString](#)

8.21.1.4 [CHAR](#)\* [\\_SLQSSwiSetOSInfoParams::pVersionString](#)

## 8.22 [\\_sysSelectPrefInfo](#) Struct Reference

### Data Fields

- [BYTE](#) \* [pEmerMode](#)
- [WORD](#) \* [pModePref](#)
- [ULONGLONG](#) \* [pBandPref](#)
- [WORD](#) \* [pPRLPref](#)
- [WORD](#) \* [pRoamPref](#)
- [ULONGLONG](#) \* [pLTEBandPref](#)
- [BYTE](#) \* [pNetSelPref](#)
- [ULONG](#) \* [pSrvDomainPref](#)
- [ULONG](#) \* [pGWAcqOrderPref](#)

### 8.22.1 Detailed Description

Structure for storing the current preferred system selection settings for the device.

#### Parameters

<i><a href="#">pEmerMode</a></i>	<ul style="list-style-type: none"> <li>• Optional parameter specifying the emergency Mode</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0 - OFF (normal)</li> <li>– 1 - ON (Emergency)</li> </ul> </li> <li>• function <a href="#">SLQSGetSysSelectionPref()</a> returns a default value FF if no value is returned by the device.</li> </ul>
----------------------------------	---

<i>pModePref</i>	<ul style="list-style-type: none"> <li>• Optional parameter</li> <li>• Bit Mask indicating the radio technology mode preference</li> <li>• Bit values: <ul style="list-style-type: none"> <li>– Bit 0 - cdma2000 1x</li> <li>– Bit 1 - cdma2000 HRPD(1xEV-DO)</li> <li>– Bit 2 - GSM</li> <li>– Bit 3 - UMTS</li> <li>– Bit 4 - LTE</li> </ul> </li> <li>• function <a href="#">SLQSGetSysSelectionPref()</a> returns a default value FF if no value is returned by the device.</li> </ul>
<i>pBandPref</i>	<ul style="list-style-type: none"> <li>• Optional parameter</li> <li>• Bit mask representing the band preference</li> <li>• Bit values: <ul style="list-style-type: none"> <li>– Bit 0 - Band Class 0, A-System</li> <li>– Bit 1 - Band Class 0, B-System, Band Class 0 AB, GSM 850 Band</li> <li>– Bit 2 - Band Class 1, all blocks</li> <li>– Bit 3 - Band Class 2 place holder</li> <li>– Bit 4 - Band Class 3, A-System</li> <li>– Bit 5 - Band Class 4, all blocks</li> <li>– Bit 6 - Band Class 5, all blocks</li> <li>– Bit 7 - GSM_DCS_1800 band</li> <li>– Bit 8 - GSM Extended GSM (E-GSM) 900 band</li> <li>– Bit 9 - GSM Primary GSM (P-GSM) 900 band</li> <li>– Bit 10 - Band Class 6</li> <li>– Bit 11 - Band Class 7</li> <li>– Bit 12 - Band Class 8</li> <li>– Bit 13 - Band Class 9</li> <li>– Bit 14 - Band Class 10</li> <li>– Bit 15 - Band Class 11</li> <li>– Bit 16 - GSM 450 band</li> <li>– Bit 17 - GSM 480 band</li> <li>– Bit 18 - GSM 750 band</li> <li>– Bit 19 - GSM 850 band</li> <li>– Bit 20 - GSM Railways GSM 900 Band</li> <li>– Bit 21 - GSM PCS 1900 band</li> <li>– Bit 22 - WCDMA Europe, Japan, and China IMT 2100 band</li> <li>– Bit 23 - WCDMA U.S. PCS 1900 band</li> <li>– Bit 24 - WCDMA Europe and China DCS 1800 band</li> <li>– Bit 25 - WCDMA U.S. 1700 band</li> <li>– Bit 26 - WCDMA U.S. 850 band</li> <li>– Bit 27 - WCDMA Japan 800 band</li> </ul> </li> </ul>
Generated on Wed Oct 28 2015 00:40:12 for Linux by Doxygen 1.8.12	<ul style="list-style-type: none"> <li>– Bit 29 - Band Class 14</li> <li>– Bit 30 - Reserved</li> <li>– Bit 31 - Band Class 15</li> </ul>

<i>pPRLPref</i>	<ul style="list-style-type: none"> <li>Optional parameter indicating the CDMA PRL Preference</li> <li>Values: <ul style="list-style-type: none"> <li>0x0001 - Acquire available system only on the A side</li> <li>0x0002 - Acquire available system only on the B side</li> <li>0x3FFF - Acquire any available systems</li> </ul> </li> <li>function <a href="#">SLQSGetSysSelectionPref()</a> returns a default value FFFF if no value is returned by the device.</li> </ul>
<i>pRoamPref</i>	<ul style="list-style-type: none"> <li>Optional parameter indicating the roaming Preference</li> <li>Values: <ul style="list-style-type: none"> <li>0x01 - Acquire only systems for which the roaming indicator is off</li> <li>0x02 - Acquire a system as long as its roaming indicator is not off</li> <li>0x03 - Acquire only systems for which the roaming indicator is off or solid on, i.e. not flashing; CDMA only</li> <li>0xFF - Acquire systems, regardless of their roaming indicator</li> </ul> </li> <li>function <a href="#">SLQSGetSysSelectionPref()</a> returns a default value FFFF if no value is returned by the device.</li> </ul>
<i>pLTEBandPref</i>	<ul style="list-style-type: none"> <li>Optional parameter</li> <li>Bit mask representing the LTE band preference</li> <li>Bit Values <ul style="list-style-type: none"> <li>Bit 0 - E-UTRA Operating Band 1</li> <li>Bit 1 - E-UTRA Operating Band 2</li> <li>Bit 2 - E-UTRA Operating Band 3</li> <li>Bit 3 - E-UTRA Operating Band 4</li> <li>Bit 4 - E-UTRA Operating Band 5</li> <li>Bit 5 - E-UTRA Operating Band 6</li> <li>Bit 6 - E-UTRA Operating Band 7</li> <li>Bit 7 - E-UTRA Operating Band 8</li> <li>Bit 8 - E-UTRA Operating Band 9</li> <li>Bit 9 - E-UTRA Operating Band 10</li> <li>Bit 10 - E-UTRA Operating Band 11</li> <li>Bit 11 - E-UTRA Operating Band 12</li> <li>Bit 12 - E-UTRA Operating Band 13</li> <li>Bit 13 - E-UTRA Operating Band 14</li> <li>Bit 16 - E-UTRA Operating Band 17</li> <li>Bit 17 - E-UTRA Operating Band 18</li> <li>Bit 18 - E-UTRA Operating Band 19</li> <li>Bit 19 - E-UTRA Operating Band 20</li> <li>Bit 20 - E-UTRA Operating Band 21</li> <li>Bit 32 - E-UTRA Operating Band 33</li> </ul> </li> </ul>
	<ul style="list-style-type: none"> <li>Bit 33 - E-UTRA Operating Band 34</li> <li>Bit 34 - E-UTRA Operating Band 35</li> <li>Bit 35 - E-UTRA Operating Band 36</li> <li>Bit 36 - E-UTRA Operating Band 37</li> </ul>

<i>pNetSelPref</i>	<ul style="list-style-type: none"> <li>Optional parameter indicating network selection preference</li> <li>Values: <ul style="list-style-type: none"> <li>0x00 - Automatic network selection</li> <li>0x01 - Manual network selection</li> </ul> </li> <li>function <a href="#">SLQSGetSysSelectionPref()</a> returns a default value FF if no value is returned by the device.</li> </ul>
<i>pSrvDomainPref</i>	<ul style="list-style-type: none"> <li>Optional parameter indicating Service domain preference</li> <li>Values: <ul style="list-style-type: none"> <li>0x00 - Circuit switched only</li> <li>0x01 - Packet switched only</li> <li>0x02 - Circuit switched and packet switched</li> <li>0x03 - Packet switched attach</li> <li>0x04 - Packet switched detach</li> </ul> </li> <li>function <a href="#">SLQSGetSysSelectionPref()</a> returns a default value FFFFFFFF if no value is returned by the device.</li> </ul>
<i>pGWAcqOrder-Pref</i>	<ul style="list-style-type: none"> <li>Optional parameter indicating GSM/WCDMA Acquisition order Preference</li> <li>Values: <ul style="list-style-type: none"> <li>0x00 - Automatic</li> <li>0x01 - GSM then WCDMA</li> <li>0x02 - WCDMA then GSM</li> </ul> </li> <li>function <a href="#">SLQSGetSysSelectionPref()</a> returns a default value FFFFFFFF if no value is returned by the device.</li> </ul>

## Note

None

## 8.22.2 Field Documentation

8.22.2.1 ULONGLONG\* \_sysSelectPrefInfo::pBandPref

8.22.2.2 BYTE\* \_sysSelectPrefInfo::pEmerMode

8.22.2.3 ULONG\* \_sysSelectPrefInfo::pGWAcqOrderPref

8.22.2.4 ULONGLONG\* \_sysSelectPrefInfo::pLTEBandPref

8.22.2.5 WORD\* \_sysSelectPrefInfo::pModePref

8.22.2.6 BYTE\* \_sysSelectPrefInfo::pNetSelPref

8.22.2.7 **WORD**\* \_sysSelectPrefInfo::pPRLPref

8.22.2.8 **WORD**\* \_sysSelectPrefInfo::pRoamPref

8.22.2.9 **ULONG**\* \_sysSelectPrefInfo::pSrvDomainPref

## 8.23 \_sysSelectPrefParams Struct Reference

### Data Fields

- **BYTE**\* pEmerMode
- **WORD**\* pModePref
- **ULONGLONG**\* pBandPref
- **WORD**\* pPRLPref
- **WORD**\* pRoamPref
- **ULONGLONG**\* pLTEBandPref
- **struct netSelectionPref**\* pNetSelPref
- **BYTE**\* pChgDuration
- **BYTE**\* pMNCIncPCSDigStat
- **ULONG**\* pSrvDomainPref
- **ULONG**\* pGWAcqOrderPref
- **ULONGLONG**\* pTdsdmaBandPref
- **struct acqOrderPref**\* pAcqOrderPref
- **ULONG**\* pSrvRegRestriction
- **struct CSGID**\* pCSGID
- **BYTE**\* pRAT

### 8.23.1 Detailed Description

Contain the system selection preferences.

## Parameters

<i>pEmerMode</i>	<ul style="list-style-type: none"> <li>Optional parameter specifying the emergency Mode</li> <li>Values: <ul style="list-style-type: none"> <li>0 - OFF (normal)</li> <li>1 - ON (Emergency)</li> </ul> </li> </ul>
<i>pModePref</i>	<ul style="list-style-type: none"> <li>Optional parameter</li> <li>Bit Mask indicating the radio technology mode preference</li> <li>Bit values: <ul style="list-style-type: none"> <li>Bit 0 - cdma2000 1x</li> <li>Bit 1 - cdma2000 HRPD(1xEV-DO)</li> <li>Bit 2 - GSM</li> <li>Bit 3 - UMTS</li> <li>Bit 4 - LTE</li> </ul> </li> </ul>
<i>pBandPref</i>	<ul style="list-style-type: none"> <li>Optional parameter</li> <li>Bit mask representing the band preference</li> <li>Bit values: <ul style="list-style-type: none"> <li>Bit 0 - Band Class 0, A-System</li> <li>Bit 1 - Band Class 0, B-System, Band Class 0 AB, GSM 850 Band</li> <li>Bit 2 - Band Class 1, all blocks</li> <li>Bit 3 - Band Class 2 place holder</li> <li>Bit 4 - Band Class 3, A-System</li> <li>Bit 5 - Band Class 4, all blocks</li> <li>Bit 6 - Band Class 5, all blocks</li> <li>Bit 7 - GSM_DCS_1800 band</li> <li>Bit 8 - GSM Extended GSM (E-GSM) 900 band</li> <li>Bit 9 - GSM Primary GSM (P-GSM) 900 band</li> <li>Bit 10 - Band Class 6</li> <li>Bit 11 - Band Class 7</li> <li>Bit 12 - Band Class 8</li> <li>Bit 13 - Band Class 9</li> <li>Bit 14 - Band Class 10</li> <li>Bit 15 - Band Class 11</li> <li>Bit 16 - GSM 450 band</li> <li>Bit 17 - GSM 480 band</li> <li>Bit 18 - GSM 750 band</li> <li>Bit 19 - GSM 850 band</li> <li>Bit 20 - GSM Railways GSM 900 Band</li> <li>Bit 21 - GSM PCS 1900 band</li> <li>Bit 22 - WCDMA Europe, Japan, and China IMT 2100 band</li> <li>Bit 23 - WCDMA U.S. PCS 1900 band</li> <li>Bit 24 - WCDMA Europe and China DCS 1800 band</li> </ul> </li> </ul>

<i>pPRLPref</i>	<ul style="list-style-type: none"> <li>• Optional parameter indicating the CDMA PRL Preference</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0x0001 - Acquire available system only on the A side</li> <li>– 0x0002 - Acquire available system only on the B side</li> <li>– 0x3FFF - Acquire any available systems</li> </ul> </li> </ul>
<i>pRoamPref</i>	<ul style="list-style-type: none"> <li>• Optional parameter indicating the roaming Preference</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0x01 - Acquire only systems for which the roaming indicator is off</li> <li>– 0x02 - Acquire a system as long as its roaming indicator is not off</li> <li>– 0x03 - Acquire only systems for which the roaming indicator is off or solid on, i.e. not flashing; CDMA only</li> <li>– 0xFF - Acquire systems, regardless of their roaming indicator</li> </ul> </li> </ul>
<i>pLTEBandPref</i>	<ul style="list-style-type: none"> <li>• Optional parameter</li> <li>• Bit mask representing the LTE band preference</li> <li>• Bit Values <ul style="list-style-type: none"> <li>– Bit 0 - E-UTRA Operating Band 1</li> <li>– Bit 1 - E-UTRA Operating Band 2</li> <li>– Bit 2 - E-UTRA Operating Band 3</li> <li>– Bit 3 - E-UTRA Operating Band 4</li> <li>– Bit 4 - E-UTRA Operating Band 5</li> <li>– Bit 5 - E-UTRA Operating Band 6</li> <li>– Bit 6 - E-UTRA Operating Band 7</li> <li>– Bit 7 - E-UTRA Operating Band 8</li> <li>– Bit 8 - E-UTRA Operating Band 9</li> <li>– Bit 9 - E-UTRA Operating Band 10</li> <li>– Bit 10 - E-UTRA Operating Band 11</li> <li>– Bit 11 - E-UTRA Operating Band 12</li> <li>– Bit 12 - E-UTRA Operating Band 13</li> <li>– Bit 13 - E-UTRA Operating Band 14</li> <li>– Bit 16 - E-UTRA Operating Band 17</li> <li>– Bit 17 - E-UTRA Operating Band 18</li> <li>– Bit 18 - E-UTRA Operating Band 19</li> <li>– Bit 19 - E-UTRA Operating Band 20</li> <li>– Bit 20 - E-UTRA Operating Band 21</li> <li>– Bit 32 - E-UTRA Operating Band 33</li> <li>– Bit 33 - E-UTRA Operating Band 34</li> <li>– Bit 34 - E-UTRA Operating Band 35</li> <li>– Bit 35 - E-UTRA Operating Band 36</li> </ul> </li> </ul>
	<ul style="list-style-type: none"> <li>– Bit 36 - E-UTRA Operating Band 37</li> <li>– Bit 37 - E-UTRA Operating Band 38</li> <li>– Bit 38 - E-UTRA Operating Band 39</li> <li>– Bit 39 - E-UTRA Operating Band 40</li> </ul>



<i>pNetSelPref</i>	<ul style="list-style-type: none"> <li>- <a href="#">netSelectionPref</a></li> <li>• Optional parameter for specifying Network Selection Preference</li> <li>• Modem selects networks based on this parameter(if present).</li> <li>• see <a href="#">netSelectionPref</a> for more information</li> </ul>
<i>pChgDuration</i>	<ul style="list-style-type: none"> <li>• Optional parameter specifying the duration of the change</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0x00 - Power cycle - Remains active until the next device power cycle</li> <li>– 0x01 - Permanent - Remains active through power cycles until changed by client</li> <li>– Device will use "0x01 - permanent" as default if this parameter is omitted</li> </ul> </li> </ul>
<i>pMNCIncPCS-DigStat</i>	<ul style="list-style-type: none"> <li>• Optional parameter indicating if MNC includes PCS digit</li> <li>• Values: <ul style="list-style-type: none"> <li>– TRUE - MNC is a 3 digit value; e.g., a reported value of 90 corresponds to an MNC value of 090</li> <li>– FALSE - MNC is a 2-digit value; e.g., a reported value of 90 corresponds to an MNC value of 90</li> </ul> </li> </ul>
<i>pSrvDomainPref</i>	<ul style="list-style-type: none"> <li>• Optional parameter indicating Service domain preference</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0x00 - Circuit switched only</li> <li>– 0x01 - Packet switched only</li> <li>– 0x02 - Circuit switched and packet switched</li> <li>– 0x03 - Packet switched attach</li> <li>– 0x04 - Packet switched detach</li> </ul> </li> </ul>
<i>pGWAcqOrder-Pref</i>	<ul style="list-style-type: none"> <li>• Optional parameter indicating GSM/WCDMA Acquisition order Preference</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0x00 - Automatic</li> <li>– 0x01 - GSM then WCDMA</li> <li>– 0x02 - WCDMA then GSM</li> </ul> </li> </ul>

<i>pRAT</i>	<ul style="list-style-type: none"> <li>• Optional parameter Radio Access Technology order Preference</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0x04 - GSM</li> <li>– 0x05 - UMTS</li> <li>– 0x08 - LTE</li> <li>– 0x09 - TDSCDMA</li> </ul> </li> </ul>
-------------	---

### 8.23.2 Field Documentation

8.23.2.1 `struct acqOrderPref* _sysSelectPrefParams::pAcqOrderPref`

8.23.2.2 `ULONGLONG* _sysSelectPrefParams::pBandPref`

8.23.2.3 `BYTE* _sysSelectPrefParams::pChgDuration`

8.23.2.4 `struct CSGID* _sysSelectPrefParams::pCSGID`

8.23.2.5 `BYTE* _sysSelectPrefParams::pEmerMode`

8.23.2.6 `ULONG* _sysSelectPrefParams::pGWAcqOrderPref`

8.23.2.7 `ULONGLONG* _sysSelectPrefParams::pLTEBandPref`

8.23.2.8 `BYTE* _sysSelectPrefParams::pMNCIncPCSDigStat`

8.23.2.9 `WORD* _sysSelectPrefParams::pModePref`

8.23.2.10 `struct netSelectionPref* _sysSelectPrefParams::pNetSelPref`

8.23.2.11 `WORD* _sysSelectPrefParams::pPRLPref`

8.23.2.12 `BYTE* _sysSelectPrefParams::pRAT`

8.23.2.13 `WORD* _sysSelectPrefParams::pRoamPref`

8.23.2.14 `ULONG* _sysSelectPrefParams::pSrvDomainPref`

8.23.2.15 `ULONG* _sysSelectPrefParams::pSrvRegRestriction`

8.23.2.16 `ULONGLONG* _sysSelectPrefParams::pTdsdmaBandPref`

## 8.24 \_transLayerinfo Struct Reference

### Data Fields

- [BYTE TransType](#)
- [BYTE TransCap](#)

### 8.24.1 Detailed Description

This structure contains Transport Layer Information

## Parameters

<i>TransType</i>	<ul style="list-style-type: none"> <li>• Transport Type <ul style="list-style-type: none"> <li>– 0x00 - IMS</li> </ul> </li> </ul>
<i>TransCap</i>	<ul style="list-style-type: none"> <li>• Transport Capability</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0x00 - CDMA</li> <li>– 0x01 - GW</li> </ul> </li> </ul>

## 8.24.2 Field Documentation

## 8.24.2.1 BYTE \_transLayerInfo::TransCap

## 8.24.2.2 BYTE \_transLayerInfo::TransType

## 8.25 \_transLayerInfoNotification Struct Reference

## Data Fields

- [BYTE regInd](#)
- [transLayerInfo](#) \* [pTransLayerInfo](#)

## 8.25.1 Detailed Description

Contains the parameters passed for SLQSSetTransLayerInfoCallback by the device.

## Parameters

<i>regInd</i>	<ul style="list-style-type: none"> <li>• Indicates whether the transport layer is registered or not</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0x00 - Transport layer is not registered</li> <li>– 0x01 - Transport layer is registered</li> </ul> </li> </ul>
---------------	---

<i>pTransLayerInfo</i>	<ul style="list-style-type: none"> <li>Optional parameter</li> <li>See <a href="#">transLayerInfo</a> for more information</li> </ul>
------------------------	---

**Note**

None

**8.25.2 Field Documentation**8.25.2.1 `transLayerInfo* _transLayerInfoNotification::pTransLayerInfo`8.25.2.2 `BYTE _transLayerInfoNotification::regInd`**8.26 \_transNWRegInfoNotification Struct Reference****Data Fields**

- [BYTE NWRegStat](#)

**8.26.1 Detailed Description**

Contains the parameters passed for SLQSSetTransNWRegInfoCallback by the device.

**Parameters**

<i>NWRegStat</i>	<ul style="list-style-type: none"> <li>provides the transport network registration information</li> <li>Values: <ul style="list-style-type: none"> <li>0x00 - No Service</li> <li>0x01 - In Progress</li> <li>0x02 - Failed</li> <li>0x03 - Limited Service</li> <li>0x04 - Full Service</li> </ul> </li> </ul>
------------------	---

**Note**

None

**8.26.2 Field Documentation**8.26.2.1 `BYTE _transNWRegInfoNotification::NWRegStat`**8.27 accelAcceptReady\_s Struct Reference****Data Fields**

- [BYTE injectEnable](#)

- [WORD samplesPerBatch](#)
- [WORD batchPerSec](#)

### 8.27.1 Detailed Description

This structure contains Accelerometer Accept Ready Info

#### Parameters

<i>injectEnable</i>	<ul style="list-style-type: none"> <li>• GNSS location engine is ready to accept data from sensor.</li> <li>• Values</li> <li>• 0x01 - Ready to accept sensor data</li> <li>• 0x00 - Not ready to accept sensor data</li> </ul>
<i>samplesPerBatch</i>	<ul style="list-style-type: none"> <li>• number of samples per batch the GNSS location engine is to receive.</li> <li>• <math>\text{samplingFrequency} = \text{samplesPerBatch} * \text{batchesPerSecond}</math></li> <li>• samplesPerBatch must be a nonzero positive value.</li> </ul>
<i>batchPerSec</i>	<ul style="list-style-type: none"> <li>• LTE NAS version minor</li> <li>• Number of sensor-data batches the GNSS location engine is to receive per second.</li> <li>• BatchesPerSecond must be a nonzero positive value.</li> </ul>

### 8.27.2 Field Documentation

8.27.2.1 **WORD** accelAcceptReady\_s::batchPerSec

8.27.2.2 **BYTE** accelAcceptReady\_s::injectEnable

8.27.2.3 **WORD** accelAcceptReady\_s::samplesPerBatch

## 8.28 accelTempAcceptReady\_s Struct Reference

#### Data Fields

- [BYTE injectEnable](#)
- [WORD samplesPerBatch](#)
- [WORD batchPerSec](#)

### 8.28.1 Detailed Description

This structure contains Accelerometer Temperature Accept Ready Info

## Parameters

<i>injectEnable</i>	<ul style="list-style-type: none"> <li>GNSS location engine is ready to accept data from sensor.</li> <li>Values</li> <li>0x01 - Ready to accept sensor data</li> <li>0x00 - Not ready to accept sensor data</li> </ul>
<i>samplesPerBatch</i>	<ul style="list-style-type: none"> <li>number of samples per batch the GNSS location engine is to receive.</li> <li>samplingFrequency = samplesPerBatch * batchesPerSecond</li> <li>samplesPerBatch must be a nonzero positive value.</li> </ul>
<i>batchPerSec</i>	<ul style="list-style-type: none"> <li>LTE NAS version minor</li> <li>Number of sensor-data batches the GNSS location engine is to receive per second.</li> <li>BatchesPerSecond must be a nonzero positive value.</li> </ul>

## 8.28.2 Field Documentation

8.28.2.1 WORD accelTempAcceptReady\_s::batchPerSec

8.28.2.2 BYTE accelTempAcceptReady\_s::injectEnable

8.28.2.3 WORD accelTempAcceptReady\_s::samplesPerBatch

## 8.29 acqOrderPref Struct Reference

## Data Fields

- [BYTE acqOrdeLen](#)
- [BYTE \\* pAcqOrder](#)

## 8.29.1 Detailed Description

Contain the Acquisition Order Preference.

## Parameters

<i>acqOrdeLen</i>	<ul style="list-style-type: none"> <li>• Number of sets of the following elements.</li> </ul>
<i>pAcqOrder</i>	<ul style="list-style-type: none"> <li>• Acquisition order preference to be set. Values: <ul style="list-style-type: none"> <li>– 0x01 - NAS_RADIO_IF_CDMA_1X - cdma2000 1X</li> <li>– 0x02 - NAS_RADIO_IF_CDMA_1XEVD0 - cdma2000 HRPD (1xEV-DO)</li> <li>– 0x04 - NAS_RADIO_IF_GSM - GSM</li> <li>– 0x05 - NAS_RADIO_IF_UMTS - UMTS</li> <li>– 0x08 - NAS_RADIO_IF_LTE - LTE</li> <li>– 0x09 - NAS_RADIO_IF_TDSCDMA - TD-SCDMA</li> </ul> </li> </ul>

## 8.29.2 Field Documentation

8.29.2.1 BYTE acqOrderPref::acqOrdeLen

8.29.2.2 BYTE\* acqOrderPref::pAcqOrder

## 8.30 ActPilotPNElement Struct Reference

## Data Fields

- [WORD ActSetPilotPN](#)
- [BYTE ActSetPilotPNStrength](#)

## 8.30.1 Detailed Description

This structure describes Active Pilot PN elements

## Parameters

<i>ActSetPilotPN</i>	<ul style="list-style-type: none"> <li>• Active Pilot PN</li> </ul>
<i>ActSetPilotPN- Strength</i>	<ul style="list-style-type: none"> <li>• Active Pilot PN strength</li> </ul>

## 8.30.2 Field Documentation

8.30.2.1 WORD ActPilotPNElement::ActSetPilotPN

8.30.2.2 BYTE ActPilotPNElement::ActSetPilotPNStrength

## 8.31 AddCDMASysInfo Struct Reference



## Data Fields

- [WORD geoSysIdx](#)
- [WORD regPrd](#)

### 8.31.1 Detailed Description

Structure for storing the Additional CDMA System Information.

#### Parameters

<i>geoSysIdx</i>	<ul style="list-style-type: none"><li>• System table index referencing the beginning of the geo in which the current serving system is present.</li><li>• When the system index is not known, 0xFFFF is used.</li></ul>
<i>regPrd</i>	<ul style="list-style-type: none"><li>• Registration period after the CDMA system is acquired.</li><li>• When the CDMA registration period is not valid, 0xFFFF is used.</li></ul>

### 8.31.2 Field Documentation

8.31.2.1 [WORD AddCDMASysInfo::geoSysIdx](#)

8.31.2.2 [WORD AddCDMASysInfo::regPrd](#)

## 8.32 AddSysInfo Struct Reference

## Data Fields

- [WORD geoSysIdx](#)
- [ULONG cellBroadcastCap](#)

### 8.32.1 Detailed Description

Structure for storing the Additional GSM and WCDMA System Information.

#### Parameters

<i>geoSysIdx</i>	<ul style="list-style-type: none"><li>• System table index referencing the beginning of the geo in which the current serving system is present.</li><li>• When the system index is not known, 0xFFFF is used.</li></ul>
------------------	---

<i>cellBroadcast-Cap</i>	<ul style="list-style-type: none"> <li>• Cell broadcast capability of the serving system.</li> <li>• When the CDMA registration period is not valid, 0xFFFF is used. <ul style="list-style-type: none"> <li>– NAS_CELL_BROADCAST_CAP_UNKNOWN - Cell broadcast support is unknown</li> <li>– NAS_CELL_BROADCAST_CAP_OFF - Cell broadcast is not supported</li> <li>– NAS_CELL_BROADCAST_CAP_ON - Cell broadcast is supported</li> </ul> </li> </ul>
--------------------------	--

### 8.32.2 Field Documentation

8.32.2.1 **ULONG** AddSysInfo::cellBroadcastCap

8.32.2.2 **WORD** AddSysInfo::geoSysIdx

## 8.33 airTimer Struct Reference

### Data Fields

- [BYTE](#) *namID*
- [ULONG](#) *airTimerValue*

### 8.33.1 Detailed Description

This structure contains information about the Air Timer.

#### Parameters

<i>namID</i>	<ul style="list-style-type: none"> <li>• Index of the NAM(Number Assignment Module) to be configured.</li> <li>• Range 0 to 3.</li> <li>• Some modems support only 1 or 2 NAMs.</li> <li>• 0xFF,if not available.</li> </ul>
<i>airTimerValue</i>	<ul style="list-style-type: none"> <li>• Time in minutes.</li> <li>• Cumulative air time is slammed.</li> <li>• 0xFFFFFFFF,if not available.</li> </ul>

### 8.33.2 Field Documentation

8.33.2.1 **ULONG** airTimer::airTimerValue

8.33.2.2 **BYTE** airTimer::namID

## 8.34 allCallsAlphaIDInfo Struct Reference

### Data Fields

- [BYTE callID](#)
- [alphaIDInfo AlphaIDInfo](#)

#### 8.34.1 Detailed Description

This structure contains information for Alpha Identifier for All Calls

##### Parameters

<i>callID</i>	<ul style="list-style-type: none"><li>• Unique call identifier for the call.</li></ul>
<i>AlphaIDInfo</i>	<ul style="list-style-type: none"><li>• See <a href="#">alphaIDInfo</a> for more information.</li></ul>

#### 8.34.2 Field Documentation

8.34.2.1 [alphaIDInfo](#) allCallsAlphaIDInfo::AlphaIDInfo

8.34.2.2 [BYTE](#) allCallsAlphaIDInfo::callID

## 8.35 allCallsDiagInfo Struct Reference

### Data Fields

- [BYTE callID](#)
- [diagInfo DiagInfo](#)

#### 8.35.1 Detailed Description

This structure contains Diagnostic Information for All Calls

##### Parameters

<i>callID</i>	<ul style="list-style-type: none"><li>• Unique call identifier for the call.</li></ul>
<i>DiagInfo</i>	<ul style="list-style-type: none"><li>• See <a href="#">diagInfo</a> for more information.</li></ul>

#### 8.35.2 Field Documentation

8.35.2.1 [BYTE](#) allCallsDiagInfo::callID

8.35.2.2 [diagInfo](#) allCallsDiagInfo::DiagInfo

## 8.36 allCallsUUSInfo Struct Reference

### Data Fields

- [BYTE callID](#)
- [UUSInfo uusInfo](#)

### 8.36.1 Detailed Description

This structure contains information for User to User Signaling Service for All Calls.

#### Parameters

<i>callID</i>	<ul style="list-style-type: none"><li>• Unique call identifier for the call.</li></ul>
<i>uusInfo</i>	<ul style="list-style-type: none"><li>• See <a href="#">UUSInfo</a> for more information.</li></ul>

### 8.36.2 Field Documentation

#### 8.36.2.1 BYTE allCallsUUSInfo::callID

#### 8.36.2.2 UUSInfo allCallsUUSInfo::uusInfo

## 8.37 alphaIDInfo Struct Reference

### Data Fields

- [BYTE alphaDcs](#)
- [BYTE alphaLen](#)
- [BYTE alphaText](#) [255]

### 8.37.1 Detailed Description

This structure contains information about the Alpha Identifier.

#### Parameters

<i>alphaDcs</i>	<ul style="list-style-type: none"><li>• Alpha coding scheme<ul style="list-style-type: none"><li>– 0x01 - GSM Default_Char</li><li>– 0x02 - UCS2</li><li>– 0xFF - Not Available</li></ul></li></ul>
-----------------	---

<i>alphaLen</i>	<ul style="list-style-type: none"> <li>• Number of sets of the following elements: <ul style="list-style-type: none"> <li>– pAlpha_text</li> </ul> </li> <li>• If zero(0) then no further information exists.</li> </ul>
<i>alphaText</i> [MAX_DESCRIPTION_LENGTH]	<ul style="list-style-type: none"> <li>• Data encoded as per the alpha_dcs</li> </ul>

### 8.37.2 Field Documentation

8.37.2.1 **BYTE** alphaIDInfo::alphaDcs

8.37.2.2 **BYTE** alphaIDInfo::alphaLen

8.37.2.3 **BYTE** alphaIDInfo::alphaText[255]

## 8.38 appStatus Struct Reference

### Data Fields

- [BYTE appType](#)
- [BYTE appState](#)
- [BYTE persoState](#)
- [BYTE persoFeature](#)
- [BYTE persoRetries](#)
- [BYTE persoUnblockRetries](#)
- [BYTE aidLength](#)
- [BYTE aidVal](#) [255]
- [BYTE univPin](#)
- [BYTE pin1State](#)
- [BYTE pin1Retries](#)
- [BYTE puk1Retries](#)
- [BYTE pin2State](#)
- [BYTE pin2Retries](#)
- [BYTE puk2Retries](#)

### 8.38.1 Detailed Description

This structure contains Application Status Information loaded on the card.

## Parameters

<i>appType</i>	<ul style="list-style-type: none"><li>• Indicates the type of the application.<ul style="list-style-type: none"><li>– 0 - Unknown</li><li>– 1 - SIM card</li><li>– 2 - USIM application</li><li>– 3 - RUIM card</li><li>– 4 - CSIM application</li><li>– 5 - ISIM application</li></ul></li><li>• Other values are reserved for the future and are to be handled as "Unknown".</li></ul>
<i>appState</i>	<ul style="list-style-type: none"><li>• Indicates the state of the application.<ul style="list-style-type: none"><li>– 0 - Unknown</li><li>– 1 - Detected</li><li>– 2 - PIN1 or UPIN is required</li><li>– 3 - PUK1 or PUK for UPIN is required</li><li>– 4 - Personalization state must be checked</li><li>– 5 - PIN1 is blocked</li><li>– 6 - Illegal</li><li>– 7 - Ready</li></ul></li></ul>
<i>persoState</i>	<ul style="list-style-type: none"><li>• Indicates the state of the personalization for the application.<ul style="list-style-type: none"><li>– 0 - Unknown</li><li>– 1 - Personalization operation is in progress</li><li>– 2 - Ready</li><li>– 3 - Personalization code is required</li><li>– 4 - PUK for personalization code is required</li><li>– 5 - Permanently blocked</li></ul></li></ul>

<i>persoFeature</i>	<ul style="list-style-type: none"> <li>Indicates the personalization feature.</li> <li>This applies only when a personalization code is required to deactivate or unblock personalization. <ul style="list-style-type: none"> <li>0 - GW network personalization</li> <li>1 - GW network subset personalization</li> <li>2 - GW service provider personalization</li> <li>3 - GW corporate personalization</li> <li>4 - GW UIM personalization</li> <li>5 - 1X network type 1 personalization</li> <li>6 - 1X network type 2 personalization</li> <li>7 - 1X HRPD personalization</li> <li>8 - 1X service provider personalization</li> <li>9 - 1X corporate personalization</li> <li>10 - 1X RUIM personalization</li> <li>11 - Unknown</li> </ul> </li> </ul>
<i>persoRetries</i>	<ul style="list-style-type: none"> <li>Indicates the number of retries remaining to disable the personalization.</li> </ul>
<i>persoUnblock-Retries</i>	<ul style="list-style-type: none"> <li>Indicates the number of retries remaining to unblock the personalization.</li> </ul>
<i>aidLength</i>	<ul style="list-style-type: none"> <li>Number of sets of the following elements. i.e. aidVal</li> <li>If zero(0) then no aidVal information exists.</li> </ul>
<i>aidVal</i> [MAX_DESCRIPTION_LENGTH]	<ul style="list-style-type: none"> <li>Application identifier value.</li> </ul>
<i>univPin</i>	<ul style="list-style-type: none"> <li>Indicates whether UPIN replaces PIN1. <ul style="list-style-type: none"> <li>0 - PIN1 is used</li> <li>1 - UPIN replaces PIN1</li> </ul> </li> </ul>
<i>pin1State</i>	<ul style="list-style-type: none"> <li>Indicates the state of PIN1. <ul style="list-style-type: none"> <li>0 - Unknown</li> <li>1 - Enabled and not verified</li> <li>2 - Enabled and verified</li> <li>3 - Disabled</li> <li>4 - Blocked</li> <li>5 - Permanently blocked</li> </ul> </li> </ul>

<i>pin1Retries</i>	<ul style="list-style-type: none"> <li>Indicates the number of retries remaining to verify PIN1.</li> </ul>
<i>puk1Retries</i>	<ul style="list-style-type: none"> <li>Indicates the number of retries remaining to unblock PIN1.</li> </ul>
<i>pin2State</i>	<ul style="list-style-type: none"> <li>Indicates the state of PIN2. <ul style="list-style-type: none"> <li>0 - Unknown</li> <li>1 - Enabled and not verified</li> <li>2 - Enabled and verified</li> <li>3 - Disabled</li> <li>4 - Blocked</li> <li>5 - Permanently blocked</li> </ul> </li> </ul>
<i>pin2Retries</i>	<ul style="list-style-type: none"> <li>Indicates the number of retries remaining to verify PIN2.</li> </ul>
<i>puk2Retries</i>	<ul style="list-style-type: none"> <li>Indicates the number of retries remaining to unblock PIN2.</li> </ul>

## 8.38.2 Field Documentation

8.38.2.1 **BYTE** appStatus::aidLength

8.38.2.2 **BYTE** appStatus::aidVal[255]

8.38.2.3 **BYTE** appStatus::appState

8.38.2.4 **BYTE** appStatus::appType

8.38.2.5 **BYTE** appStatus::persoFeature

8.38.2.6 **BYTE** appStatus::persoRetries

8.38.2.7 **BYTE** appStatus::persoState

8.38.2.8 **BYTE** appStatus::persoUnblockRetries

8.38.2.9 **BYTE** appStatus::pin1Retries

8.38.2.10 **BYTE** appStatus::pin1State

8.38.2.11 **BYTE** appStatus::pin2Retries

8.38.2.12 **BYTE** appStatus::pin2State

8.38.2.13 **BYTE** appStatus::puk1Retries



8.38.2.14 **BYTE** appStatus::puk2Retries8.38.2.15 **BYTE** appStatus::univPin

## 8.39 arrAlertingPattern Struct Reference

### Data Fields

- [BYTE numInstances](#)
- [BYTE callID \[20\]](#)
- [ULONG alertingPattern \[20\]](#)

### 8.39.1 Detailed Description

This structure contains an array of Alerting Pattern.

#### Parameters

<i>numInstances</i>	<ul style="list-style-type: none"> <li>• Number of callID, alertingPattern that follow.</li> <li>• If zero(0) then no further information exists.</li> </ul>
<i>callID[MAX_NO_OF_CALLS]</i>	<ul style="list-style-type: none"> <li>• Array of Unique call identifier for the call.</li> </ul>
<i>alertingPattern[MAX_NO_OF_CALLS]</i>	<ul style="list-style-type: none"> <li>• Array of Alerting pattern. <ul style="list-style-type: none"> <li>– 0x00 - QMI_VOICE_ALERTING_PATTERN_1 - Pattern 1</li> <li>– 0x01 - QMI_VOICE_ALERTING_PATTERN_2 - Pattern 2</li> <li>– 0x02 - QMI_VOICE_ALERTING_PATTERN_3 - Pattern 3</li> <li>– 0x04 - QMI_VOICE_ALERTING_PATTERN_5 - Pattern 5</li> <li>– 0x05 - QMI_VOICE_ALERTING_PATTERN_6 - Pattern 6</li> <li>– 0x06 - QMI_VOICE_ALERTING_PATTERN_7 - Pattern 7</li> <li>– 0x07 - QMI_VOICE_ALERTING_PATTERN_8 - Pattern 8</li> <li>– 0x08 - QMI_VOICE_ALERTING_PATTERN_9 - Pattern 9</li> </ul> </li> </ul>

### 8.39.2 Field Documentation

8.39.2.1 **ULONG** arrAlertingPattern::alertingPattern[20]8.39.2.2 **BYTE** arrAlertingPattern::callID[20]8.39.2.3 **BYTE** arrAlertingPattern::numInstances

## 8.40 arrAlertingType Struct Reference

### Data Fields

- [BYTE numInstances](#)

- [BYTE callID](#) [20]
- [BYTE AlertingType](#) [20]

### 8.40.1 Detailed Description

This structure contains an array of Alerting Type.

#### Parameters

<i>numInstances</i>	<ul style="list-style-type: none"> <li>• Number of callID, AlertingType that follow.</li> <li>• If zero(0) then no further information exists.</li> </ul>
<i>callID</i> [MAX_NO_OF_CALLS]	<ul style="list-style-type: none"> <li>• Array of Unique call identifier for the call.</li> </ul>
<i>AlertingType</i> [MAX_NO_OF_CALLS]	<ul style="list-style-type: none"> <li>• Array of Alerting type. <ul style="list-style-type: none"> <li>– 0x00 - ALERTING_LOCAL - Local</li> <li>– 0x01 - ALERTING_REMOTE - Remote</li> </ul> </li> </ul>

### 8.40.2 Field Documentation

8.40.2.1 [BYTE arrAlertingType::AlertingType](#)[20]

8.40.2.2 [BYTE arrAlertingType::callID](#)[20]

8.40.2.3 [BYTE arrAlertingType::numInstances](#)

## 8.41 arrAlphaID Struct Reference

#### Data Fields

- [BYTE numInstances](#)
- [allCallsAlphaIDInfo allCallsAlphaIDInfoArr](#) [20]

### 8.41.1 Detailed Description

This structure contains an array of Alpha ID Info

#### Parameters

<i>numInstances</i>	<ul style="list-style-type: none"> <li>• Number of <a href="#">allCallsAlphaIDInfo</a> that follow.</li> <li>• If zero(0) then no further information exists.</li> </ul>
---------------------	--

<i>allCallsAlphaID-Info</i> [MAX_NO_OF_CALLS]	<ul style="list-style-type: none"> <li>• Array of <a href="#">allCallsAlphaIDInfo</a>.</li> <li>• See <a href="#">allCallsAlphaIDInfo</a> for more information.</li> </ul>
---	--

### 8.41.2 Field Documentation

8.41.2.1 **allCallsAlphaIDInfo** arrAlphaID::allCallsAlphaIDInfoArr[20]

8.41.2.2 **BYTE** arrAlphaID::numInstances

## 8.42 arrCalledPartyNum Struct Reference

### Data Fields

- [BYTE](#) numInstances
- [peerNumberInfo](#) [CalledPartyNum](#) [20]

### 8.42.1 Detailed Description

This structure contains an array of Called Party Numbers consisting of information of all the numbers which have been called from the device.

#### Parameters

<i>numInstances</i>	<ul style="list-style-type: none"> <li>• Number of calledPartyNum that follow.</li> <li>• If zero(0) then no further information exists.</li> </ul>
<i>CalledParty-Num</i> [MAX_NO_OF_CALLS]	<ul style="list-style-type: none"> <li>• Array of CalledPartyNum.</li> <li>• See <a href="#">peerNumberInfo</a> for more information.</li> </ul>

### 8.42.2 Field Documentation

8.42.2.1 **peerNumberInfo** arrCalledPartyNum::CalledPartyNum[20]

8.42.2.2 **BYTE** arrCalledPartyNum::numInstances

## 8.43 arrCallEndReason Struct Reference

### Data Fields

- [BYTE](#) numInstances
- [BYTE](#) callID [20]
- [WORD](#) callEndReason [20]

### 8.43.1 Detailed Description

This structure contains an array of Call End Reasons.

## Parameters

<i>numInstances</i>	<ul style="list-style-type: none"> <li>• Number of callID, callEndReason that follow.</li> <li>• If zero(0) then no further information exists.</li> </ul>
<i>callID</i> [MAX_NO_OF_CALLS]	<ul style="list-style-type: none"> <li>• Array of Unique call identifier for the call.</li> </ul>
<i>callEndReason</i> [MAX_NO_OF_CALLS]	<ul style="list-style-type: none"> <li>• Array of Call End Reason .</li> <li>• See Table9 <a href="#">qaGobiApiTableVoiceCallEndReasons.h</a> for a list of valid voice-related call end reasons</li> </ul>

## 8.43.2 Field Documentation

8.43.2.1 WORD arrCallEndReason::callEndReason[20]

8.43.2.2 BYTE arrCallEndReason::callID[20]

8.43.2.3 BYTE arrCallEndReason::numInstances

## 8.44 arrCallInfo Struct Reference

## Data Fields

- [BYTE numInstances](#)
- [getAllCallInformation getAllCallInfo](#) [20]

## 8.44.1 Detailed Description

This structure contains an array of Call Info

## Parameters

<i>numInstances</i>	<ul style="list-style-type: none"> <li>• Number of getAllCallInfo that follow.</li> <li>• If zero(0) then no further information exists.</li> </ul>
<i>getAllCallInfo</i> [MAX_NO_OF_CALLS]	<ul style="list-style-type: none"> <li>• Array of CallInfo.</li> <li>• See <a href="#">getAllCallInfo</a> for more information.</li> </ul>

## 8.44.2 Field Documentation

8.44.2.1 getAllCallInformation arrCallInfo::getAllCallInfo[20]

8.44.2.2 BYTE arrCallInfo::numInstances

## 8.45 arrConnectPartyNum Struct Reference

### Data Fields

- [BYTE numInstances](#)
- [peerNumberInfo ConnectedPartyNum](#) [20]

### 8.45.1 Detailed Description

This structure contains an array of Connected Party Numbers consisting of information regarding all the devices connected.

#### Parameters

<i>numInstances</i>	<ul style="list-style-type: none"> <li>• Number of ConnectedPartyNum that follow.</li> <li>• If zero(0) then no further information exists.</li> </ul>
<i>ConnectedParty-Num[MAX_NO_OF_CALLS]</i>	<ul style="list-style-type: none"> <li>• Array of ConnectedPartyNum.</li> <li>• See <a href="#">peerNumberInfo</a> for more information.</li> </ul>

### 8.45.2 Field Documentation

8.45.2.1 [peerNumberInfo](#) arrConnectPartyNum::ConnectedPartyNum[20]

8.45.2.2 [BYTE](#) arrConnectPartyNum::numInstances

## 8.46 arrDiagInfo Struct Reference

### Data Fields

- [BYTE numInstances](#)
- [allCallsDiagInfo DiagInfo](#) [20]

### 8.46.1 Detailed Description

This structure contains an array of Diagnostic Information.

#### Parameters

<i>numInstances</i>	<ul style="list-style-type: none"> <li>• Number of DiagInfo that follow.</li> <li>• If zero(0) then no further information exists.</li> </ul>
---------------------	---

<i>DiagInfo</i> [MAX_ NO_OF_CALLS]	<ul style="list-style-type: none"> <li>• Array of DiagInfo.</li> <li>• See <a href="#">allCallsDiagInfo</a> for more information.</li> </ul>
---------------------------------------	--

## 8.46.2 Field Documentation

8.46.2.1 [allCallsDiagInfo](#) arrDiagInfo::DiagInfo[20]

8.46.2.2 [BYTE](#) arrDiagInfo::numInstances

## 8.47 arrRedirPartyNum Struct Reference

### Data Fields

- [BYTE](#) numInstances
- [peerNumberInfo](#) RedirPartyNum [20]

### 8.47.1 Detailed Description

This structure contains an array of Redirecting Party Numbers consisting of information of all the numbers which have been redirected from the device.

#### Parameters

<i>numInstances</i>	<ul style="list-style-type: none"> <li>• Number of redirPartyNum that follow.</li> <li>• If zero(0) then no further information exists.</li> </ul>
<i>RedirPartyNum</i> [- MAX_NO_OF_ CALLS]	<ul style="list-style-type: none"> <li>• Array of RedirPartyNum.</li> <li>• See <a href="#">peerNumberInfo</a> for more information.</li> </ul>

## 8.47.2 Field Documentation

8.47.2.1 [BYTE](#) arrRedirPartyNum::numInstances

8.47.2.2 [peerNumberInfo](#) arrRedirPartyNum::RedirPartyNum[20]

## 8.48 arrRemotePartyName Struct Reference

### Data Fields

- [BYTE](#) numInstances
- [getAllCallRmtPtyName](#) GetAllCallRmtPtyName [20]

### 8.48.1 Detailed Description

This structure contains an array of Remote Party Names

## Parameters

<i>numInstances</i>	<ul style="list-style-type: none"> <li>• Number of <a href="#">remotePartyName</a> that follow.</li> <li>• If zero(0) then no further information exists.</li> </ul>
<i>GetAllCallRmtPtyName[MAX_NO_OF_CALLS]</i>	<ul style="list-style-type: none"> <li>• Array of <a href="#">remotePartyName</a>.</li> <li>• See <a href="#">getAllCallRmtPtyName</a> for more information.</li> </ul>

## 8.48.2 Field Documentation

8.48.2.1 [getAllCallRmtPtyName](#) arrRemotePartyName::GetAllCallRmtPtyName[20]8.48.2.2 [BYTE](#) arrRemotePartyName::numInstances

## 8.49 arrRemotePartyNum Struct Reference

## Data Fields

- [BYTE](#) numInstances
- [getAllCallRmtPtyNum](#) RmtPtyNum [20]

## 8.49.1 Detailed Description

This structure contains an array of Remote Party Numbers

## Parameters

<i>numInstances</i>	<ul style="list-style-type: none"> <li>• Number of <a href="#">remotePartyNum</a> that follow.</li> <li>• If zero(0) then no further information exists.</li> </ul>
<i>RmtPtyNum[MAX_NO_OF_CALLS]</i>	<ul style="list-style-type: none"> <li>• Array of <a href="#">remotePartyNum</a>.</li> <li>• See <a href="#">getAllCallRmtPtyNum</a> for more information.</li> </ul>

## 8.49.2 Field Documentation

8.49.2.1 [BYTE](#) arrRemotePartyNum::numInstances8.49.2.2 [getAllCallRmtPtyNum](#) arrRemotePartyNum::RmtPtyNum[20]

## 8.50 arrSvcOption Struct Reference

## Data Fields

- [BYTE](#) numInstances



- [BYTE callID](#) [20]
- [WORD srvOption](#) [20]

### 8.50.1 Detailed Description

This structure contains array an of Servicing option.

#### Parameters

<i>numInstances</i>	<ul style="list-style-type: none"> <li>• Number of callID, srvOption that follow.</li> <li>• If zero(0) then no further information exists.</li> </ul>
<i>callID</i> [MAX_NO_OF_CALLS]	<ul style="list-style-type: none"> <li>• Array of Unique call identifier for the call.</li> </ul>
<i>srvOption</i> [MAX_NO_OF_CALLS]	<ul style="list-style-type: none"> <li>• Array of Service option.</li> <li>• See Table9 <a href="#">qaGobiApiTableServiceOptions.h</a> for standard service option number assignments.</li> </ul>

### 8.50.2 Field Documentation

8.50.2.1 **BYTE** arrSvcOption::callID[20]

8.50.2.2 **BYTE** arrSvcOption::numInstances

8.50.2.3 **WORD** arrSvcOption::srvOption[20]

## 8.51 arrUUSInfo Struct Reference

### Data Fields

- [BYTE numInstances](#)
- [allCallsUUSInfo AllCallsUUSInfo](#) [20]

### 8.51.1 Detailed Description

This structure contains an array of User to User Signaling Service Information

#### Parameters

<i>numInstances</i>	<ul style="list-style-type: none"> <li>• Number of <a href="#">allCallsUUSInfo</a> that follow.</li> <li>• If zero(0) then no further information exists.</li> </ul>
---------------------	--

<i>AllCallsUUSInfo</i> [MAX_NO_OF_CALLS]	<ul style="list-style-type: none"> <li>• Array of <a href="#">allCallsUUSInfo</a>.</li> <li>• See <a href="#">allCallsUUSInfo</a> for more information.</li> </ul>
--	--

## 8.51.2 Field Documentation

8.51.2.1 **allCallsUUSInfo** arrUUSInfo::AllCallsUUSInfo[20]

8.51.2.2 **BYTE** arrUUSInfo::numInstances

## 8.52 authenticateResult Struct Reference

### Data Fields

- [WORD](#) *contentLen*
- [BYTE](#) *content* [1024]

### 8.52.1 Detailed Description

This structure contains the information about the authenticate result.

#### Parameters

<i>contentLen</i>	<ul style="list-style-type: none"> <li>• Length of the following elements i.e. <i>content</i>.</li> </ul>
<i>content</i> [MAX_CONTENT_LENGTH]	<ul style="list-style-type: none"> <li>• Authenticate data.</li> <li>• This value is a sequence of bytes returned from the card.</li> </ul>

## 8.52.2 Field Documentation

8.52.2.1 **BYTE** authenticateResult::content[1024]

8.52.2.2 **WORD** authenticateResult::contentLen

## 8.53 authenticationData Struct Reference

### Data Fields

- [BYTE](#) *context*
- [WORD](#) *dataLen*
- [BYTE](#) *data* [1024]

### 8.53.1 Detailed Description

This structure contains the Session Information.

## Parameters

<i>context</i>	<ul style="list-style-type: none"> <li>• Authenticate context. <ul style="list-style-type: none"> <li>– 0 - Runs the GSM algorithm (valid only on a 2GSIM card)</li> <li>– 1 - Runs the CAVE algorithm (valid only on a RUIM card)</li> <li>– 2 - GSM security context (valid only on a USIM application)</li> <li>– 3 - 3G security context (valid only on a USIM application)</li> <li>– 4 - VGCS/VBS security context (valid only on a USIM application)</li> <li>– 5 - GBA security context, Bootstrapping mode (valid only on a USIM or ISIM application)</li> <li>– 6 - GBA security context, NAF Derivation mode (valid only on a USIM or ISIM application)</li> <li>– 7 - MBMS security context, MSK Update mode (valid only on a USIM application)</li> <li>– 8 - MBMS security context, MTK Generation mode (valid only on a USIM application)</li> <li>– 9 - MBMS security context, MSK Deletion mode (valid only on a USIM application)</li> <li>– 10 - MBMS security context, MUK Deletion mode (valid only on a USIM application)</li> <li>– 11 - IMS AKA security context (valid only on a USIM application)</li> <li>– 12 - HTTP-digest security context (valid only on an ISIM application)</li> <li>– 13 - Compute IP authentication, CHAP (valid only on RUIM or CSIM)</li> <li>– 14 - Compute IP authentication, MN-HA authenticator (valid only on RUIM or CSIM)</li> <li>– 15 - Compute IP authentication, MIP-RRQ hash (valid only on RUIM or CSIM)</li> <li>– 16 - Compute IP authentication, MN-AAA authenticator (valid only on RUIM or CSIM)</li> <li>– 17 - Compute IP authentication, HRPD access authenticator (valid only on RUIM or CSIM)</li> </ul> </li> <li>• Other values are possible and reserved for future use.</li> </ul>
----------------	--

<i>dataLen</i>	<ul style="list-style-type: none"> <li>Length of the following elements i.e. data.</li> </ul>
<i>data[MAX_DESCRIPTION_LENGTH]</i>	<ul style="list-style-type: none"> <li>Authenticate Data.</li> </ul>

### 8.53.2 Field Documentation

8.53.2.1 **BYTE** authenticationData::context

8.53.2.2 **BYTE** authenticationData::data[1024]

8.53.2.3 **WORD** authenticationData::dataLen

## 8.54 BdsSV Struct Reference

### Data Fields

- [WORD](#) id
- [BYTE](#) mask

### 8.54.1 Detailed Description

This structure contains the BDS [SV](#) Info

#### Parameters

<i>id</i>	<ul style="list-style-type: none"> <li><a href="#">SV</a> ID of the satellite whose data is to be deleted. <ul style="list-style-type: none"> <li>Range for BDS: 201 to 237</li> </ul> </li> </ul>
<i>mask</i>	<ul style="list-style-type: none"> <li>Indicates if the ephemeris or almanac for a satellite is to be deleted</li> <li>Valid values: <ul style="list-style-type: none"> <li>QMI_LOC_MASK_DELETE_EPHEMERIS (0x01) - Delete ephemeris for the satellite</li> <li>QMI_LOC_MASK_DELETE_ALMANAC (0x02) - Delete almanac for the satellite</li> </ul> </li> </ul>

### 8.54.2 Field Documentation

8.54.2.1 **WORD** BdsSV::id

8.54.2.2 **BYTE** BdsSV::mask

## 8.55 BdsSVInfo Struct Reference

## Data Fields

- [BYTE len](#)
- [BdsSV \\* pSV](#)

### 8.55.1 Detailed Description

This structure contains the number of sets of the BDS SVN Info

#### Parameters

<i>len</i>	<ul style="list-style-type: none"><li>• Number of sets of the following elements:<ul style="list-style-type: none"><li>– gnssSvId</li><li>– deleteSvInfoMask</li></ul></li></ul>
<i>pSV</i>	<ul style="list-style-type: none"><li>• Pointer to struct <a href="#">BdsSV</a>. See <a href="#">BdsSV</a> for more information</li></ul>

### 8.55.2 Field Documentation

#### 8.55.2.1 BYTE BdsSVInfo::len

#### 8.55.2.2 BdsSV\* BdsSVInfo::pSV

## 8.56 BroadcastConfig Struct Reference

## Data Fields

- [WORD fromServiceId](#)
- [WORD toServiceId](#)
- [BYTE selected](#)

### 8.56.1 Detailed Description

This structure contains [BroadcastConfig](#) parameters

#### Parameters

<i>fromServiceId</i>	<ul style="list-style-type: none"><li>• Starting point of range of CBM message identifiers</li></ul>
<i>toServiceId</i>	<ul style="list-style-type: none"><li>• Ending point of range of CBM message identifiers</li></ul>
<i>selected</i>	<ul style="list-style-type: none"><li>• Range of CBM message identifiers indicated by from_service_id and to_service_id<ul style="list-style-type: none"><li>– 0x00 – Not selected</li><li>– 0x01 – Selected</li></ul></li></ul>

## 8.56.2 Field Documentation

8.56.2.1 WORD BroadcastConfig::fromServiceId

8.56.2.2 BYTE BroadcastConfig::selected

8.56.2.3 WORD BroadcastConfig::toServiceId

## 8.57 burstDTMFInfo Struct Reference

### Data Fields

- BYTE \* pCallID
- BYTE digitCnt
- BYTE pDigitBuff [255]

### 8.57.1 Detailed Description

This structure contains Voice Burst DTMF Information

#### Parameters

<i>pCallID</i> [IN/OUT]	<ul style="list-style-type: none"> <li>• Call ID associated with call on which the DTMF information has to be sent. A burst DTMF request is sent to the current active/alerting call when pCallId is set to 0xFF.</li> <li>• This is IN/OUT parameter, value passed by user will be packed in request and value received from the device would be returned to the user.</li> <li>• If the call ID value received is 0, no value has been returned by the device</li> </ul>
<i>digitCnt</i>	<ul style="list-style-type: none"> <li>• Length of DTMF digit buffer which follows</li> </ul>
<i>pDigitBuff</i> [MAX- _DESCRIPTIO- N_LENGTH]	<ul style="list-style-type: none"> <li>• DTMF digit buffer in ASCII, NULL terminated</li> </ul>

## 8.57.2 Field Documentation

8.57.2.1 BYTE burstDTMFInfo::digitCnt

8.57.2.2 BYTE\* burstDTMFInfo::pCallID

8.57.2.3 BYTE burstDTMFInfo::pDigitBuff[255]

## 8.58 CallBarringSysInfo Struct Reference

### Data Fields

- ULONG csBarStatus
- ULONG psBarStatus

### 8.58.1 Detailed Description

Structure for storing the GSM and WCDMA Call Barring System Information.

## Parameters

<i>csBarStatus</i>	<ul style="list-style-type: none"> <li>• Call barring status for circuit-switched calls. <ul style="list-style-type: none"> <li>– NAS_CELL_ACCESS_NORMAL_ONLY - Cell access is allowed for normal calls only</li> <li>– NAS_CELL_ACCESS_EMERGENCY_ONLY - Cell access is allowed for emergency calls only</li> <li>– NAS_CELL_ACCESS_NO_CALLS - Cell access is not allowed for any call type</li> <li>– NAS_CELL_ACCESS_ALL_CALLS - Cell access is allowed for all call types</li> <li>– NAS_CELL_ACCESS_UNKNOWN - Cell access type is unknown</li> </ul> </li> </ul>
<i>psBarStatus</i>	<ul style="list-style-type: none"> <li>• Call barring status for packet-switched calls. <ul style="list-style-type: none"> <li>– NAS_CELL_ACCESS_NORMAL_ONLY - Cell access is allowed for normal calls only</li> <li>– NAS_CELL_ACCESS_EMERGENCY_ONLY - Cell access is allowed for emergency calls only</li> <li>– NAS_CELL_ACCESS_NO_CALLS - Cell access is not allowed for any call type</li> <li>– NAS_CELL_ACCESS_ALL_CALLS - Cell access is allowed for all call types</li> <li>– NAS_CELL_ACCESS_UNKNOWN - Cell access type is unknown</li> </ul> </li> </ul>

## 8.58.2 Field Documentation

8.58.2.1 **ULONG** CallBarringSysInfo::csBarStatus8.58.2.2 **ULONG** CallBarringSysInfo::psBarStatus

## 8.59 callBarStatus Struct Reference

## Data Fields

- [ULONG csBarStatus](#)
- [ULONG psBarStatus](#)

## 8.59.1 Detailed Description

This structure contains Call Barring Status.

- Parameter values default to their data type's maximum unsigned value unless explicitly stated otherwise.



## Parameters

<i>csBarStatus</i>	<ul style="list-style-type: none"> <li>• Call Barring Status for circuit-switched calls.</li> <li>• Values: <ul style="list-style-type: none"> <li>• NAS_CELL_ACCESS_NORMAL_ONLY - Cell access is allowed for normal calls only</li> <li>• NAS_CELL_ACCESS_EMERGENCY_ONLY - Cell access is allowed for emergency calls only</li> <li>• NAS_CELL_ACCESS_NO_CALLS - Cell access is not allowed for any call type</li> <li>• NAS_CELL_ACCESS_ALL_CALLS - Cell access is allowed for all call types</li> <li>• NAS_CELL_ACCESS_UNKNOWN - Cell access type is unknown</li> </ul> </li> </ul>
<i>psBarStatus</i>	<ul style="list-style-type: none"> <li>• Call Barring Status for packet-switched calls.</li> <li>• Values: <ul style="list-style-type: none"> <li>– NAS_CELL_ACCESS_NORMAL_ONLY - Cell access is allowed for normal calls only</li> <li>– NAS_CELL_ACCESS_EMERGENCY_ONLY - Cell access is allowed for emergency calls only</li> <li>– NAS_CELL_ACCESS_NO_CALLS - Cell access is not allowed for any call type</li> <li>– NAS_CELL_ACCESS_ALL_CALLS - Cell access is allowed for all call types</li> <li>– NAS_CELL_ACCESS_UNKNOWN - Cell access type is unknown</li> </ul> </li> </ul>

## 8.59.2 Field Documentation

8.59.2.1 **ULONG** callBarStatus::csBarStatus8.59.2.2 **ULONG** callBarStatus::psBarStatus

## 8.60 calledPartyInfo Struct Reference

## Data Fields

- [BYTE](#) PI
- [BYTE](#) SI
- [BYTE](#) numType
- [BYTE](#) numPlan
- [BYTE](#) numLen
- [BYTE](#) number [255]

## 8.60.1 Detailed Description

This structure contains Called party Number Information

## Parameters

<i>PI</i>	<ul style="list-style-type: none"> <li>• Presentation indicator; refer to [S1, Table 2.7.4.4-1] for valid values.</li> </ul>
<i>SI</i>	<ul style="list-style-type: none"> <li>• Number of sets of following elements <ul style="list-style-type: none"> <li>– Caller Id</li> </ul> </li> </ul>
<i>SI</i>	<ul style="list-style-type: none"> <li>• Number screening indicator.</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0x00 - QMI_VOICE_SI_USER_PROVIDED_NOT_SCREENED - Provided user is not screened</li> <li>– 0x01 - QMI_VOICE_SI_USER_PROVIDED_VERIFIED_PASSED - Provided user passed verification</li> <li>– 0x02 - QMI_VOICE_SI_USER_PROVIDED_VERIFIED_FAILED - Provided user failed verification</li> <li>– 0x03 - QMI_VOICE_SI_NETWORK_PROVIDED - Provided network</li> </ul> </li> </ul>
<i>numType</i>	<ul style="list-style-type: none"> <li>• Number type.</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0x00 - QMI_VOICE_NUM_TYPE_UNKNOWN - Unknown</li> <li>– 0x01 - QMI_VOICE_NUM_TYPE_INTERNATIONAL - International</li> <li>– 0x02 - QMI_VOICE_NUM_TYPE_NATIONAL - National</li> <li>– 0x03 - QMI_VOICE_NUM_TYPE_NETWORK_SPECIFIC - Network-specific</li> <li>– 0x04 - QMI_VOICE_NUM_TYPE_SUBSCRIBER - Subscriber</li> <li>– 0x05 - QMI_VOICE_NUM_TYPE_RESERVED - Reserved</li> <li>– 0x06 - QMI_VOICE_NUM_TYPE_ABBREVIATED - Abbreviated</li> <li>– 0x07 - QMI_VOICE_NUM_TYPE_RESERVED_EXTENSION - Reserved extension</li> </ul> </li> </ul>

<i>numPlan</i>	<ul style="list-style-type: none"> <li>• Number plan.</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0x00 - QMI_VOICE_NUM_PLAN_UNKNOWN - Unknown</li> <li>– 0x01 - QMI_VOICE_NUM_PLAN_ISDN - ISDN</li> <li>– 0x03 - QMI_VOICE_NUM_PLAN_DATA - Data</li> <li>– 0x04 - QMI_VOICE_NUM_PLAN_TELEX - Telex</li> <li>– 0x08 - QMI_VOICE_NUM_PLAN_NATIONAL - National</li> <li>– 0x09 - QMI_VOICE_NUM_PLAN_PRIVATE - Private</li> <li>– 0x0B - QMI_VOICE_NUM_PLAN_RESERVED_CTS - Reserved cordless telephony system</li> <li>– 0x0F - QMI_VOICE_NUM_PLAN_RESERVED_EXTENSION - Reserved extension</li> </ul> </li> </ul>
<i>numLen</i>	<ul style="list-style-type: none"> <li>• Provides the length of number which follow.</li> </ul>
<i>number[255]</i>	<ul style="list-style-type: none"> <li>• number of numLen length, NULL terminated.</li> </ul>

## 8.60.2 Field Documentation

8.60.2.1 **BYTE** calledPartyInfo::number[255]

8.60.2.2 **BYTE** calledPartyInfo::numLen

8.60.2.3 **BYTE** calledPartyInfo::numPlan

8.60.2.4 **BYTE** calledPartyInfo::numType

8.60.2.5 **BYTE** calledPartyInfo::PI

8.60.2.6 **BYTE** calledPartyInfo::SI

## 8.61 calledPartySubAdd Struct Reference

### Data Fields

- [BYTE extBit](#)
- [BYTE subAddrType](#)
- [BYTE oddEvenInd](#)
- [BYTE subAddrLen](#)
- [BYTE subAddr](#) [255]

### 8.61.1 Detailed Description

This structure contains information about the Called Sub Party Addresses.

## Parameters

<i>extBit</i>	<ul style="list-style-type: none"> <li>• Extension bit.</li> </ul>
<i>subAddrType</i>	<ul style="list-style-type: none"> <li>• Subaddress type. <ul style="list-style-type: none"> <li>– 0x00 - NSAP</li> <li>– 0x01 - USER</li> </ul> </li> </ul>
<i>oddEvenInd</i>	<ul style="list-style-type: none"> <li>• Even/odd indicator. <ul style="list-style-type: none"> <li>– 0x00 - Even number of address signals</li> <li>– 0x01 - Odd number of address signals</li> </ul> </li> </ul>
<i>subAddrLen</i>	<ul style="list-style-type: none"> <li>• Number of sets of the following elements: <ul style="list-style-type: none"> <li>– SubAddress</li> </ul> </li> </ul>
<i>subAddr[<small>MAX_</small> - DESCRIPTION- _LENGTH]</i>	<ul style="list-style-type: none"> <li>• Array of the SubAddress in BCD number format.</li> </ul>

## 8.61.2 Field Documentation

8.61.2.1 BYTE calledPartySubAdd::extBit

8.61.2.2 BYTE calledPartySubAdd::oddEvenInd

8.61.2.3 BYTE calledPartySubAdd::subAddr[255]

8.61.2.4 BYTE calledPartySubAdd::subAddrLen

8.61.2.5 BYTE calledPartySubAdd::subAddrType

## 8.62 callerIDInfo Struct Reference

## Data Fields

- [BYTE PI](#)
- [BYTE callerIDLen](#)
- [BYTE callerID \[255\]](#)

## 8.62.1 Detailed Description

This structure contains Caller ID Information

## Parameters

<i>PI</i>	<ul style="list-style-type: none"><li>• Presentation indicator; refer to [S1, Table 2.7.4.4-1] for valid values.</li></ul>
<i>callerIDLen</i>	<ul style="list-style-type: none"><li>• Number of sets of following elements<ul style="list-style-type: none"><li>– Caller Id</li></ul></li></ul>
<i>pCallerID</i>	<ul style="list-style-type: none"><li>• Caller ID in ASCII string.</li></ul>

## 8.62.2 Field Documentation

8.62.2.1 BYTE callerIDInfo::callerID[255]

8.62.2.2 BYTE callerIDInfo::callerIDLen

8.62.2.3 BYTE callerIDInfo::PI

## 8.63 callFwdTypeAndPlan Struct Reference

## Data Fields

- [BYTE numberType](#)
- [BYTE numberPlan](#)

## 8.63.1 Detailed Description

This structure contains Supplementary Service request parameters related to different features and their activation, deactivation, registration and erasure (applicable only for 3GPP)

## Parameters

<i>numberType</i>	<ul style="list-style-type: none"> <li>• Call forwarding number type <ul style="list-style-type: none"> <li>– 0x00 - QMI_VOICE_NUM_TYPE_UNKNOWN Unknown</li> <li>– 0x01 - QMI_VOICE_NUM_TYPE_INTERNATIONAL International</li> <li>– 0x02 - QMI_VOICE_NUM_TYPE_NATIONAL National</li> <li>– 0x03 - QMI_VOICE_NUM_TYPE_NETWORK_SPECIFIC Network-specific</li> <li>– 0x04 - QMI_VOICE_NUM_TYPE_SUBSCRIBER Subscriber</li> <li>– 0x05 - QMI_VOICE_NUM_TYPE_RESERVED Reserved</li> <li>– 0x06 - QMI_VOICE_NUM_TYPE_ABBREVIATED Abbreviated</li> <li>– 0x07 - QMI_VOICE_NUM_TYPE_RESERVED_EXTENSION Reserved extension</li> </ul> </li> </ul>
<i>numberPlan</i>	<ul style="list-style-type: none"> <li>• Call forwarding number plan <ul style="list-style-type: none"> <li>– 0x00 - QMI_VOICE_NUM_PLAN_UNKNOWN Unknown</li> <li>– 0x01 - QMI_VOICE_NUM_PLAN_ISDN ISDN</li> <li>– 0x03 - QMI_VOICE_NUM_PLAN_DATA Data</li> <li>– 0x04 - QMI_VOICE_NUM_PLAN_TELEX Telex</li> <li>– 0x08 - QMI_VOICE_NUM_PLAN_NATIONAL National</li> <li>– 0x09 - QMI_VOICE_NUM_PLAN_PRIVATE Private</li> <li>– 0x0B - QMI_VOICE_NUM_PLAN_RESERVED_CTS Reserved cordless telephony system</li> <li>– 0x0F - QMI_VOICE_NUM_PLAN_RESERVED_EXTENSION Reserved extension</li> </ul> </li> </ul>

## 8.63.2 Field Documentation

8.63.2.1 BYTE callFwdTypeAndPlan::numberPlan

8.63.2.2 BYTE callFwdTypeAndPlan::numberType

## 8.64 callFWExtInfo Struct Reference

## Data Fields

- [BYTE SvcStatus](#)
- [BYTE SvcClass](#)
- [BYTE noReplyTimer](#)
- [BYTE PI](#)
- [BYTE SI](#)
- [BYTE numType](#)
- [BYTE numPlan](#)
- [BYTE numLen](#)
- [BYTE number](#) [255]

### 8.64.1 Detailed Description

This structure contains information for Get Call Forwarding Extended Information.

## Parameters

<i>SvcStatus</i>	<ul style="list-style-type: none"> <li>Service status. Values: <ul style="list-style-type: none"> <li>0x00 - SERVICE_STATUS_INACTIVE - Inactive</li> <li>0x01 - SERVICE_STATUS_ACTIVE - Active</li> </ul> </li> </ul>
<i>SvcClass</i>	<ul style="list-style-type: none"> <li>Service Class is a combination (sum) of information class constants</li> <li>See <a href="#">qaGobiApiTableSupServiceInfoClasses.h</a> for service classes.</li> </ul>
<i>noReplyTimer</i>	<ul style="list-style-type: none"> <li>No reply timer value in seconds</li> <li>A value of 0 indicates that no_reply_timer is ignored.</li> </ul>
<i>PI</i>	<ul style="list-style-type: none"> <li>Presentation indicator; refer to [S1, Table 2.7.4.4-1] for valid values.</li> </ul>
<i>SI</i>	<ul style="list-style-type: none"> <li>Number screening indicator.</li> <li>Values: <ul style="list-style-type: none"> <li>0x00 - QMI_VOICE_SI_USER_PROVIDED_NOT_SCREENED - Provided user is not screened</li> <li>0x01 - QMI_VOICE_SI_USER_PROVIDED_VERIFIED_PASSED - Provided user passed verification</li> <li>0x02 - QMI_VOICE_SI_USER_PROVIDED_VERIFIED_FAILED - Provided user failed verification</li> <li>0x03 - QMI_VOICE_SI_NETWORK_PROVIDED - Provided network</li> </ul> </li> </ul>
<i>numType</i>	<ul style="list-style-type: none"> <li>Number type.</li> <li>Values: <ul style="list-style-type: none"> <li>0x00 - QMI_VOICE_NUM_TYPE_UNKNOWN - Unknown</li> <li>0x01 - QMI_VOICE_NUM_TYPE_INTERNATIONAL - International</li> <li>0x02 - QMI_VOICE_NUM_TYPE_NATIONAL - National</li> <li>0x03 - QMI_VOICE_NUM_TYPE_NETWORK_SPECIFIC - Network-specific</li> <li>0x04 - QMI_VOICE_NUM_TYPE_SUBSCRIBER - Subscriber</li> <li>0x05 - QMI_VOICE_NUM_TYPE_RESERVED - Reserved</li> <li>0x06 - QMI_VOICE_NUM_TYPE_ABBREVIATED - Abbreviated</li> <li>0x07 - QMI_VOICE_NUM_TYPE_RESERVED_EXTENSION - Reserved extension</li> </ul> </li> </ul>



<i>numPlan</i>	<ul style="list-style-type: none"> <li>• Number plan.</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0x00 - QMI_VOICE_NUM_PLAN_UNKNOWN - Unknown</li> <li>– 0x01 - QMI_VOICE_NUM_PLAN_ISDN - ISDN</li> <li>– 0x03 - QMI_VOICE_NUM_PLAN_DATA - Data</li> <li>– 0x04 - QMI_VOICE_NUM_PLAN_TELEX - Telex</li> <li>– 0x08 - QMI_VOICE_NUM_PLAN_NATIONAL - National</li> <li>– 0x09 - QMI_VOICE_NUM_PLAN_PRIVATE - Private</li> <li>– 0x0B - QMI_VOICE_NUM_PLAN_RESERVED_CTS - Reserved cordless telephony system</li> <li>– 0x0F - QMI_VOICE_NUM_PLAN_RESERVED_EXTENSION - Reserved extension</li> </ul> </li> </ul>
<i>numLen</i>	<ul style="list-style-type: none"> <li>• Provides the length of number which follow.</li> </ul>
<i>number[255]</i>	<ul style="list-style-type: none"> <li>• number of numLen length, NULL terminated.</li> </ul>

## 8.64.2 Field Documentation

8.64.2.1 **BYTE** callFWExtInfo::noReplyTimer

8.64.2.2 **BYTE** callFWExtInfo::number[255]

8.64.2.3 **BYTE** callFWExtInfo::numLen

8.64.2.4 **BYTE** callFWExtInfo::numPlan

8.64.2.5 **BYTE** callFWExtInfo::numType

8.64.2.6 **BYTE** callFWExtInfo::PI

8.64.2.7 **BYTE** callFWExtInfo::SI

8.64.2.8 **BYTE** callFWExtInfo::SvcClass

8.64.2.9 **BYTE** callFWExtInfo::SvcStatus

## 8.65 callFWInfo Struct Reference

### Data Fields

- [BYTE SvcStatus](#)
- [BYTE SvcClass](#)
- [BYTE numLen](#)
- [BYTE number \[255\]](#)
- [BYTE noReplyTimer](#)

### 8.65.1 Detailed Description

This structure contains information for Get Call Forwarding Information.

#### Parameters

<i>SvcStatus</i>	<ul style="list-style-type: none"> <li>Service status. Values: <ul style="list-style-type: none"> <li>0x00 - SERVICE_STATUS_INACTIVE - Inactive</li> <li>0x01 - SERVICE_STATUS_ACTIVE - Active</li> </ul> </li> </ul>
<i>SvcClass</i>	<ul style="list-style-type: none"> <li>Service Class is a combination (sum) of information class constants</li> <li>See <a href="#">qaGobiApiTableSupServiceInfoClasses.h</a> for service classes.</li> </ul>
<i>numLen</i>	<ul style="list-style-type: none"> <li>Provides the length of number which follow.</li> </ul>
<i>number[255]</i>	<ul style="list-style-type: none"> <li>number of numLen length, NULL terminated.</li> </ul>
<i>noReplyTimer</i>	<ul style="list-style-type: none"> <li>No reply timer value in seconds</li> <li>A value of 0 indicates that noReplyTimer is ignored.</li> </ul>

### 8.65.2 Field Documentation

8.65.2.1 **BYTE** callFWInfo::noReplyTimer

8.65.2.2 **BYTE** callFWInfo::number[255]

8.65.2.3 **BYTE** callFWInfo::numLen

8.65.2.4 **BYTE** callFWInfo::SvcClass

8.65.2.5 **BYTE** callFWInfo::SvcStatus

### 8.66 callInfo Struct Reference

#### Data Fields

- [BYTE](#) callID
- [BYTE](#) callState
- [BYTE](#) callType
- [BYTE](#) direction
- [BYTE](#) mode

### 8.66.1 Detailed Description

This structure contains Information about call state changes. For example, when an incoming call is received, this structure is populated and indicate the incoming call information. When this incoming call is answered, the call status changes from INCOMING to CONVERSATION, which means a change in the call information and this structure is populated again with the changes and notified to/retrived by the user.

#### Parameters

<i>callID</i>	<ul style="list-style-type: none"><li>• Call identifier for the call queried for information.</li><li>• If zero(0) then invalid.</li></ul>
<i>callState</i>	<ul style="list-style-type: none"><li>• Call state.<ul style="list-style-type: none"><li>– 0x01 - CALL_STATE_ORIGINATION - Origination</li><li>– 0x02 - CALL_STATE_INCOMING - Incoming</li><li>– 0x03 - CALL_STATE_CONVERSATION - Conversation</li><li>– 0x04 - CALL_STATE_CC_IN_PROGRESS - Call is originating but waiting for call control to complete</li><li>– 0x05 - CALL_STATE_ALERTING - Alerting</li><li>– 0x06 - CALL_STATE_HOLD - Hold</li><li>– 0x07 - CALL_STATE_WAITING - Waiting</li><li>– 0x08 - CALL_STATE_DISCONNECTING - Disconnecting</li><li>– 0x09 - CALL_STATE_END - End</li><li>– 0x0A - CALL_STATE_SETUP - MT call is in Setup state in 3GPP</li><li>– 0xFF - Not Available</li></ul></li></ul>

<i>callType</i>	<ul style="list-style-type: none"> <li>• Call type. <ul style="list-style-type: none"> <li>– 0x00 - CALL_TYPE_VOICE - Voice</li> <li>– 0x02 - CALL_TYPE_VOICE_IP - Voice over IP</li> <li>– 0x06 - CALL_TYPE_OTAPA - OTAPA</li> <li>– 0x07 - CALL_TYPE_STD_OTASP - Standard OTASP</li> <li>– 0x08 - CALL_TYPE_NON_STD_OTASP - Nonstandard OTASP</li> <li>– 0x09 - CALL_TYPE_EMERGENCY - Emergency</li> <li>– 0xFF - Not Available</li> </ul> </li> </ul>
<i>direction</i>	<ul style="list-style-type: none"> <li>• Direction. <ul style="list-style-type: none"> <li>– 0x01 - CALL_DIRECTION_MO - MO call</li> <li>– 0x02 - CALL_DIRECTION_MT - MT call</li> <li>– 0xFF - Not Available</li> </ul> </li> </ul>
<i>mode</i>	<ul style="list-style-type: none"> <li>• Mode.</li> <li>• If the mode field is "0x01 - CDMA", the optional Service Option, Voice Privacy, and OTASP Status (only for OTASP calls) TLVs are included in the response. <ul style="list-style-type: none"> <li>– 0x01 - CALL_MODE_CDMA - CDMA</li> <li>– 0x02 - CALL_MODE_GSM - GSM</li> <li>– 0x03 - CALL_MODE_UMTS - UMTS</li> <li>– 0x04 - CALL_MODE_LTE - LTE</li> <li>– 0x05 - CALL_MODE_TDS - TD-SCDMA</li> <li>– 0xFF - Not Available</li> </ul> </li> </ul>

## 8.66.2 Field Documentation

### 8.66.2.1 BYTE callInfo::callID

### 8.66.2.2 BYTE callInfo::callState

### 8.66.2.3 BYTE callInfo::callType

### 8.66.2.4 BYTE callInfo::direction

### 8.66.2.5 BYTE callInfo::mode

## 8.67 callingPartyInfo Struct Reference

### Data Fields

- [BYTE PI](#)
- [BYTE SI](#)

- [BYTE numType](#)
- [BYTE numPlan](#)
- [BYTE numLen](#)
- [BYTE number](#) [255]

### 8.67.1 Detailed Description

This structure contains Calling party Number Information

#### Parameters

<i>PI</i>	<ul style="list-style-type: none"> <li>• Presentation indicator; refer to [S1, Table 2.7.4.4-1] for valid values.</li> </ul>
<i>SI</i>	<ul style="list-style-type: none"> <li>• Number of sets of following elements             <ul style="list-style-type: none"> <li>– Caller Id</li> </ul> </li> </ul>
<i>SI</i>	<ul style="list-style-type: none"> <li>• Number screening indicator.</li> <li>• Values:             <ul style="list-style-type: none"> <li>– 0x00 - QMI_VOICE_SI_USER_PROVIDED_NOT_SCREENED - Provided user is not screened</li> <li>– 0x01 - QMI_VOICE_SI_USER_PROVIDED_VERIFIED_PASSED - Provided user passed verification</li> <li>– 0x02 - QMI_VOICE_SI_USER_PROVIDED_VERIFIED_FAILED - Provided user failed verification</li> <li>– 0x03 - QMI_VOICE_SI_NETWORK_PROVIDED - Provided network</li> </ul> </li> </ul>
<i>numType</i>	<ul style="list-style-type: none"> <li>• Number type.</li> <li>• Values:             <ul style="list-style-type: none"> <li>– 0x00 - QMI_VOICE_NUM_TYPE_UNKNOWN - Unknown</li> <li>– 0x01 - QMI_VOICE_NUM_TYPE_INTERNATIONAL - International</li> <li>– 0x02 - QMI_VOICE_NUM_TYPE_NATIONAL - National</li> <li>– 0x03 - QMI_VOICE_NUM_TYPE_NETWORK_SPECIFIC - Network-specific</li> <li>– 0x04 - QMI_VOICE_NUM_TYPE_SUBSCRIBER - Subscriber</li> <li>– 0x05 - QMI_VOICE_NUM_TYPE_RESERVED - Reserved</li> <li>– 0x06 - QMI_VOICE_NUM_TYPE_ABBREVIATED - Abbreviated</li> <li>– 0x07 - QMI_VOICE_NUM_TYPE_RESERVED_EXTENSION - Reserved extension</li> </ul> </li> </ul>

<i>numPlan</i>	<ul style="list-style-type: none"> <li>• Number plan.</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0x00 - QMI_VOICE_NUM_PLAN_UNKNOWN - Unknown</li> <li>– 0x01 - QMI_VOICE_NUM_PLAN_ISDN - ISDN</li> <li>– 0x03 - QMI_VOICE_NUM_PLAN_DATA - Data</li> <li>– 0x04 - QMI_VOICE_NUM_PLAN_TELEX - Telex</li> <li>– 0x08 - QMI_VOICE_NUM_PLAN_NATIONAL - National</li> <li>– 0x09 - QMI_VOICE_NUM_PLAN_PRIVATE - Private</li> <li>– 0x0B - QMI_VOICE_NUM_PLAN_RESERVED_CTS - Reserved cordless telephony system</li> <li>– 0x0F - QMI_VOICE_NUM_PLAN_RESERVED_EXTENSION - Reserved extension</li> </ul> </li> </ul>
<i>numLen</i>	<ul style="list-style-type: none"> <li>• Provides the length of number which follow.</li> </ul>
<i>number[255]</i>	<ul style="list-style-type: none"> <li>• number of numLen length, NULL terminated.</li> </ul>

## 8.67.2 Field Documentation

8.67.2.1 **BYTE** callingPartyInfo::number[255]

8.67.2.2 **BYTE** callingPartyInfo::numLen

8.67.2.3 **BYTE** callingPartyInfo::numPlan

8.67.2.4 **BYTE** callingPartyInfo::numType

8.67.2.5 **BYTE** callingPartyInfo::PI

8.67.2.6 **BYTE** callingPartyInfo::SI

## 8.68 cardResult Struct Reference

### Data Fields

- [BYTE sw1](#)
- [BYTE sw2](#)

### 8.68.1 Detailed Description

This structure contains the information about the card result.

## Parameters

<i>sw1</i>	<ul style="list-style-type: none"><li>• SW1 received from the card.</li></ul>
<i>sw2</i>	<ul style="list-style-type: none"><li>• SW2 received from the card.</li></ul>

## 8.68.2 Field Documentation

## 8.68.2.1 BYTE cardResult::sw1

## 8.68.2.2 BYTE cardResult::sw2

## 8.69 cardStatus Struct Reference

## Data Fields

- WORD [indexGwPri](#)
- WORD [index1xPri](#)
- WORD [indexGwSec](#)
- WORD [index1xSec](#)
- BYTE [numSlot](#)
- [slotInfo SlotInfo](#) [5]

## 8.69.1 Detailed Description

This structure contains Card Status Information.

## Parameters

<i>indexGwPri</i>	<ul style="list-style-type: none"><li>• Index of the primary GW provisioning application.</li><li>• The most significant byte indicates the slot (starting from 0), while the least significant byte indicates the application for that slot (starting from 0).</li><li>• The value 0xFFFF identifies when the session does not exist.</li></ul>
<i>index1xPri</i>	<ul style="list-style-type: none"><li>• Index of the primary 1X provisioning application.</li><li>• The most significant byte indicates the slot (starting from 0), while the least significant byte indicates the application for that slot (starting from 0).</li><li>• The value 0xFFFF identifies when the session does not exist.</li></ul>

<i>indexGwSec</i>	<ul style="list-style-type: none"> <li>• Index of the secondary GW provisioning application.</li> <li>• The most significant byte indicates the slot (starting from 0), while the least significant byte indicates the application for that slot (starting from 0).</li> <li>• The value 0xFFFF identifies when the session does not exist.</li> </ul>
<i>index1xSec</i>	<ul style="list-style-type: none"> <li>• Index of the secondary GW provisioning application.</li> <li>• The most significant byte indicates the slot (starting from 0), while the least significant byte indicates the application for that slot (starting from 0).</li> <li>• The value 0xFFFF identifies when the session does not exist.</li> </ul>
<i>numSlot</i>	<ul style="list-style-type: none"> <li>• Indicates the number of slots available on the device.</li> <li>• The following block is repeated for each slot. i.e. cardState</li> <li>• If zero(0) then no cardState information exists.</li> </ul>
<i>SlotInfo[MAX_N-O_OF_SLOTS]</i>	<ul style="list-style-type: none"> <li>• See <a href="#">slotInfo</a> for more information.</li> </ul>

## 8.69.2 Field Documentation

8.69.2.1 WORD cardStatus::index1xPri

8.69.2.2 WORD cardStatus::index1xSec

8.69.2.3 WORD cardStatus::indexGwPri

8.69.2.4 WORD cardStatus::indexGwSec

8.69.2.5 BYTE cardStatus::numSlot

8.69.2.6 slotInfo cardStatus::SlotInfo[5]

## 8.70 CatAlPhalIdentifierTlv Struct Reference

### Data Fields

- [BYTE ReferenceID](#)
- [USHORT AlphaIDLength](#)
- [BYTE AlphaID](#) [255]

### 8.70.1 Detailed Description

structure used to store all Alpha Identifier parameters.



## Parameters

<i>ReferenceID</i>	- proactive command type that included the alpha identifier – 0x01; sends SMS proactive command
<i>AlphaDLength</i>	- length of AlphaID ( in bytes )
<i>AlphaID</i>	- alpha identifier, encoded as in ETSI TS 102 223 [Section 8.2]

## 8.70.2 Field Documentation

8.70.2.1 **BYTE** CatAlPhalIdentifierTlv::AlphaID[255]8.70.2.2 **USHORT** CatAlPhalIdentifierTlv::AlphaDLength8.70.2.3 **BYTE** CatAlPhalIdentifierTlv::ReferenceID

## 8.71 CatCommonEventTlv Struct Reference

## Data Fields

- [BYTE TlvPresent](#)
- [BYTE EventID](#)
- [WORD EventLength](#)
- union [currentCatEvent](#) [CatEvent](#)

## 8.71.1 Field Documentation

8.71.1.1 union [currentCatEvent](#) CatCommonEventTlv::CatEvent8.71.1.2 **BYTE** CatCommonEventTlv::EventID8.71.1.3 **WORD** CatCommonEventTlv::EventLength8.71.1.4 **BYTE** CatCommonEventTlv::TlvPresent

## 8.72 CatEndProactiveSessionTlv Struct Reference

## Data Fields

- [BYTE EndProactiveSession](#)

## 8.72.1 Detailed Description

structure used to store End Proactive Session event parameters.

## Parameters

<i>EndProactive-Session</i>	- The proactive session end type values are: <ul style="list-style-type: none"> <li>• 0x01 – End proactive session command type received from the card</li> <li>• 0x02 – End proactive session internal to ME</li> </ul>
-----------------------------	--

## 8.72.2 Field Documentation

### 8.72.2.1 BYTE CatEndProactiveSessionTlv::EndProactiveSession

## 8.73 CATEventDataType Struct Reference

### Data Fields

- [ULONG eventMask](#)
- [ULONG \\* pErrorMask](#)

## 8.73.1 Field Documentation

### 8.73.1.1 ULONG CATEventDataType::eventMask

### 8.73.1.2 ULONG\* CATEventDataType::pErrorMask

## 8.74 CatEventIDDataTlv Struct Reference

### Data Fields

- [ULONG ReferenceID](#)
- [USHORT DataLength](#)
- [BYTE Data](#) [255]

## 8.74.1 Detailed Description

structure used to store all Common CAT Event parameters.

### Parameters

<i>ReferenceID</i>	- proactive command reference ID.
<i>DataLength</i>	- length of pData ( in Bytes )
<i>Data</i>	- command specific to the CAT event ID, encoded as in ETSI TS 102 223 [Section 6.6.X]

## 8.74.2 Field Documentation

### 8.74.2.1 BYTE CatEventIDDataTlv::Data[255]

### 8.74.2.2 USHORT CatEventIDDataTlv::DataLength

### 8.74.2.3 ULONG CatEventIDDataTlv::ReferenceID

## 8.75 CatEventListTlv Struct Reference

### Data Fields

- [ULONG SetupEventList](#)

## 8.75.1 Detailed Description

structure used to store all Event List parameters.

## Parameters

<i>SetupEventList</i>	- Setup event list bit mask <ul style="list-style-type: none"> <li>• 0x00000001 – User Activity Notify</li> <li>• 0x00000002 – Idle Screen Available</li> <li>• 0x00000004 – Lang Selection Notify Each set bit indicates the availability of the corresponding event in Setup Event list proactive command; all unlisted bits are reserved for future use and will be ignored</li> </ul>
-----------------------	---

## 8.75.2 Field Documentation

## 8.75.2.1 ULONG CatEventListTlv::SetupEventList

## 8.76 CatRefreshTlv Struct Reference

## Data Fields

- [USHORT RefreshMode](#)
- [BYTE RefreshStage](#)

## 8.76.1 Detailed Description

structure used to store all Refresh Event parameters.

## Parameters

<i>RefreshMode</i>	- The Refresh Event as in ETSI TS 102 223 [Section 8.6]
<i>RefreshStage</i>	- Stage of a refresh procedure <ul style="list-style-type: none"> <li>• 0x01 – Refresh start</li> <li>• 0x02 – Refresh success</li> <li>• 0x03 – Refresh failed</li> </ul>

## 8.76.2 Field Documentation

## 8.76.2.1 USHORT CatRefreshTlv::RefreshMode

## 8.76.2.2 BYTE CatRefreshTlv::RefreshStage

## 8.77 ccSUPSType Struct Reference

## Data Fields

- [BYTE svcType](#)
- [BYTE reason](#)

## 8.77.1 Detailed Description

This structure contains information about the Call Control Supplementary Service Types

## Parameters

<i>svcType</i>	<ul style="list-style-type: none"> <li>• Service type. <ul style="list-style-type: none"> <li>– 0x01 - VOICE_CC_SUPS_RESULT_SERVICE_TYPE_ACTIVATE - Activate</li> <li>– 0x02 - VOICE_CC_SUPS_RESULT_SERVICE_TYPE_DEACTIVATE - Deactivate</li> <li>– 0x03 - VOICE_CC_SUPS_RESULT_SERVICE_TYPE_REGISTER - Register</li> <li>– 0x04 - VOICE_CC_SUPS_RESULT_SERVICE_TYPE_ERASE - Erase</li> <li>– 0x05 - VOICE_CC_SUPS_RESULT_SERVICE_TYPE_INTERROGATE - Interrogate</li> <li>– 0x06 - VOICE_CC_SUPS_RESULT_SERVICE_TYPE_REGISTER_PASSWORD - Register password</li> <li>– 0x07 - VOICE_CC_SUPS_RESULT_SERVICE_TYPE_USSD - USSD</li> <li>– 0xFF - Not Available</li> </ul> </li> </ul>
<i>reason</i>	<ul style="list-style-type: none"> <li>• Call control supplementary service result reason</li> <li>• Values: <ul style="list-style-type: none"> <li>– See <a href="#">qaGobiApiTableCallControlReturnReasons.h</a> for return reasons.</li> </ul> </li> </ul>

## 8.77.2 Field Documentation

## 8.77.2.1 BYTE ccSUPSType::reason

## 8.77.2.2 BYTE ccSUPSType::svcType

## 8.78 CDMABroadcastConfig Struct Reference

## Data Fields

- [WORD serviceCategory](#)
- [WORD language](#)
- [BYTE selected](#)

## 8.78.1 Detailed Description

This structure contains [CDMABroadcastConfig](#) parameters

## Parameters

<i>serviceCategory</i>	<ul style="list-style-type: none"> <li>• Service category</li> </ul>
------------------------	--

<i>language</i>	<ul style="list-style-type: none"> <li>• Language</li> </ul>
<i>selected</i>	<ul style="list-style-type: none"> <li>• Specified service_category and language <ul style="list-style-type: none"> <li>– 0x00 – Not selected</li> <li>– 0x01 – Selected</li> </ul> </li> </ul>

## 8.78.2 Field Documentation

8.78.2.1 WORD CDMABroadcastConfig::language

8.78.2.2 BYTE CDMABroadcastConfig::selected

8.78.2.3 WORD CDMABroadcastConfig::serviceCategory

## 8.79 CDMAChannel Struct Reference

### Data Fields

- [WORD priChA](#)
- [WORD priChB](#)
- [WORD secChA](#)
- [WORD secChB](#)

### 8.79.1 Detailed Description

This structure contains the parameters for CDMA Channel Information

#### Parameters

<i>priChA</i>	<ul style="list-style-type: none"> <li>• A Channel number for the primary carrier. <ul style="list-style-type: none"> <li>– 0xFFFF - Not Available</li> </ul> </li> </ul>
<i>priChB</i>	<ul style="list-style-type: none"> <li>• B Channel number for the primary carrier. <ul style="list-style-type: none"> <li>– 0xFFFF - Not Available</li> </ul> </li> </ul>
<i>secChA</i>	<ul style="list-style-type: none"> <li>• A Channel number for the secondary carrier. <ul style="list-style-type: none"> <li>– 0xFFFF - Not Available</li> </ul> </li> </ul>

<i>secChB</i>	<ul style="list-style-type: none"> <li>• B Channel number for the secondary carrier. <ul style="list-style-type: none"> <li>– 0xFFFF - Not Available</li> </ul> </li> </ul>
---------------	---

## 8.79.2 Field Documentation

8.79.2.1 WORD CDMAChannel::priChA

8.79.2.2 WORD CDMAChannel::priChB

8.79.2.3 WORD CDMAChannel::secChA

8.79.2.4 WORD CDMAChannel::secChB

## 8.80 CDMAECIOThresh Struct Reference

### Data Fields

- [BYTE CDMAECIOThreshListLen](#)
- [WORD \\* pCDMAECIOThreshList](#)

### 8.80.1 Detailed Description

This structure contains CDMA ECIO threshold related parameters.

#### Parameters

<i>CDMAECIO- ThreshListLen</i>	<ul style="list-style-type: none"> <li>• Length of the CDMA ECIO threshold list parameter to follow</li> </ul>
<i>pCDMAECIO- ThreshList</i>	<ul style="list-style-type: none"> <li>• Array of ECIO thresholds (in units of 0.1 dB)</li> <li>• Maximum of 32 values. Range for ECIO values: -31.5 to 0 (in dB)</li> </ul>

## 8.80.2 Field Documentation

8.80.2.1 BYTE CDMAECIOThresh::CDMAECIOThreshListLen

8.80.2.2 WORD\* CDMAECIOThresh::pCDMAECIOThreshList

## 8.81 CDMAInfo Struct Reference

### Data Fields

- [WORD sid](#)
- [WORD nid](#)
- [WORD baseld](#)
- [WORD refpn](#)

- [ULONG baseLat](#)
- [ULONG baseLong](#)

### 8.81.1 Detailed Description

This structure contains information about the CDMA Network.

Parameters

<i>sid</i>	<ul style="list-style-type: none"> <li>• System ID.</li> <li>– 0xFFFF - Not Available</li> </ul>
<i>nid</i>	<ul style="list-style-type: none"> <li>• Network ID.</li> <li>– 0xFFFF - Not Available</li> </ul>
<i>baseId</i>	<ul style="list-style-type: none"> <li>• Base station ID.</li> <li>– 0xFFFF - Not Available</li> </ul>
<i>refpn</i>	<ul style="list-style-type: none"> <li>• Reference PN.</li> <li>– 0xFFFF - Not Available</li> </ul>
<i>baseLat</i>	<ul style="list-style-type: none"> <li>• Latitude of the current base station in units of 0.25 sec.</li> <li>– 0xFFFFFFFF - Not Available</li> </ul>
<i>baseLong</i>	<ul style="list-style-type: none"> <li>• Longitude of the current base station in units of 0.25 sec.</li> <li>– 0xFFFFFFFF - Not Available</li> </ul>

### 8.81.2 Field Documentation

8.81.2.1 **WORD** CDMAInfo::baseId

8.81.2.2 **ULONG** CDMAInfo::baseLat

8.81.2.3 **ULONG** CDMAInfo::baseLong

8.81.2.4 **WORD** CDMAInfo::nid

8.81.2.5 **WORD** CDMAInfo::refpn

## 8.81.2.6 WORD CDMAInfo::sid

## 8.82 cdmaMsgDecodingParams Struct Reference

## Data Fields

- ULONG messageLength
- BYTE \* pMessage
- ULONG \* pMessageID
- BYTE \* pSenderAddrLength
- CHAR \* pSenderAddr
- BYTE \* pTextMsgLength
- WORD \* pTextMsg
- BYTE \* pPriority
- BYTE \* pPrivacy
- BYTE \* pLanguage
- BYTE mcTimeStamp [0x08]
- BYTE absoluteValidity [0x08]
- BYTE \* pRelativeValidity
- BYTE \* pDisplayMode
- BOOL \* pUserAcknowledgementReq
- BOOL \* pReadAcknowledgementReq
- BYTE \* pAlertPriority
- BYTE \* pCallbkAddrLength
- CHAR \* pCallbkAddr

## 8.82.1 Detailed Description

Structure contains parameters which need to be decoded from message

## Parameters

<i>messageLength</i> [IN]	<ul style="list-style-type: none"> <li>Length of the message to be decoded in bytes</li> </ul>
<i>pMessage</i> [IN]	<ul style="list-style-type: none"> <li>Message read off the device via GetSMS</li> </ul>
<i>pSenderAddrLength</i> [IN/OUT]	<ul style="list-style-type: none"> <li>Upon input, indicates the maximum number of ASCII characters (including NULL termination) that the pSenderAddr buffer can accommodate. Note that a length of 14 is reasonable. Upon successful output, returns the length of originating address string (including the NULL termination)</li> </ul>
<i>pSenderAddr</i> [OUT]	<ul style="list-style-type: none"> <li>Returns NULL-terminated ASCII String containing the originating address. International number will be prepended with a '+' character</li> </ul>



<i>pTextMsgLength</i> [IN/OUT]	<ul style="list-style-type: none"> <li>Upon input, specifies the number of UCS2 characters the given text message buffer can accommodate. Upon successful output, returns the number of UCS2 characters returns in the given text messagebuffer(including NULL-terminator)</li> </ul>
<i>pTextMsg</i> [OUT]	<ul style="list-style-type: none"> <li>Returns the text message as NULL-terminated UCS2 string</li> </ul>
<i>pPriority</i> [OUT]	(optional parameter) <ul style="list-style-type: none"> <li>Returns the priority setting of the message 0x00 - normal 0x01 - interactive 0x02 - urgent 0x03 - emergency 0xFF - unavailable setting</li> </ul>
<i>pPrivacy</i> [OUT-T](optional)	parameter) <ul style="list-style-type: none"> <li>Returns the privacy setting of the message 0x00 - not restricted 0x01 - restricted 0x02 - confidential 0x03 - secret 0xFF - unavailable setting</li> </ul>
<i>pLanguage</i> [OUT-T]	(optional parameter ) <ul style="list-style-type: none"> <li>Returns the language setting of the message 0x00 - unspecified 0x01 - english 0x02 - french 0x03 - spanish 0x04 - japanese 0x05 - korean 0x06 - chinese 0x07 - hebrew 0xFF - unavailable setting</li> </ul>
<i>mcTimeStamp</i> [8][OUT]	(optional parameter) <ul style="list-style-type: none"> <li>Returns the message center timestamp which takes the form: YYMMDDHHMMSS-TZ where YY - year MM - month DD - day HH - hour MM - minute SS - second TZ - timezone All values are in decimal. Timezone is in relation to GMT, one unit is equal to 15 minutes and MSB indicates a negative value. If this information is unavailable for message then this field will be filled with 0xFF</li> </ul>
<i>absoluteValidity</i> [8][OUT]	(optional parameter) <ul style="list-style-type: none"> <li>Returns the absolute validity period setting for this message. This field takes the same form as mcTimeStamp</li> </ul>
<i>pRelativeValidity</i> [OUT]	(optional parameter) <ul style="list-style-type: none"> <li>Returns the relative validity period. Values have the following meanings: 0 to 143: validity period = (value + 1) * 5 minutes 144 to 167: validity period = 12 hours + (value - 143) * 30 minutes 168 to 196: validity period = (value - 166) * 1 day 197 to 244: validity period = (value - 192) * 1 week 245: validity period = indefinite 246: validity period = immediate 247: validity period = valid until mobile becomes inactive 248: validity period = valid until registration area changes 249 to 254: reserved 255: unavailable information</li> </ul>

<i>pDisplayMode[OUT]</i>	(optional parameter) <ul style="list-style-type: none"><li>Returns the display mode parameter 0x00 - immediate display 0x01 - mobile default setting 0x02 - user invoked 0x03 - reserved 0xFF - unavailable parameter</li></ul>
<i>pUser-Acknowledgement-Req[OUT]</i>	(optional parameter) <ul style="list-style-type: none"><li>Returns the user (manual) acknowledgment request parameter TRUE - means the user is requested to manually acknowledge the delivery of the message. FALSE - means no such user acknowledgment is requested</li></ul>
<i>pRead-Acknowledgement-Req[OUT]</i>	(optional parameter) <ul style="list-style-type: none"><li>Returns the read acknowledgment request parameter TRUE - means acknowledgment of the message being viewed is requested. FALSE - means no such read acknowledgment is requested</li></ul>
<i>pAlertPriority[OUT]</i>	(optional parameter) <ul style="list-style-type: none"><li>Returns the alerting parameter setting 0x00 - use default alert 0x01 - use low priority alert 0x02 - use medium priority alert 0x03 - use high priority alert 0xFF - unavailable parameter</li></ul>
<i>pCallbkAddr-Length[OUT]</i>	(optional parameter) <ul style="list-style-type: none"><li>returns the length of Callback address string (including the NULL termination)</li></ul>
<i>pCallbkAddr[OUT]</i>	(optional parameter) <ul style="list-style-type: none"><li>returns NULL-terminated ASCII String containing callback address String containing the Call Back number with a 32 maximum characters.</li></ul>

## 8.82.2 Field Documentation

8.82.2.1 **BYTE** cdmaMsgDecodingParams::absoluteValidity[0x08]

8.82.2.2 **BYTE** cdmaMsgDecodingParams::mcTimeStamp[0x08]

8.82.2.3 **ULONG** cdmaMsgDecodingParams::messageLength

8.82.2.4 **BYTE\*** cdmaMsgDecodingParams::pAlertPriority

8.82.2.5 **CHAR\*** cdmaMsgDecodingParams::pCallbkAddr

8.82.2.6 **BYTE\*** cdmaMsgDecodingParams::pCallbkAddrLength

8.82.2.7 **BYTE\*** cdmaMsgDecodingParams::pDisplayMode

8.82.2.8 **BYTE\*** cdmaMsgDecodingParams::pLanguage

8.82.2.9 **BYTE\*** cdmaMsgDecodingParams::pMessage

8.82.2.10 **ULONG\*** cdmaMsgDecodingParams::pMessageID

8.82.2.11 **BYTE\*** cdmaMsgDecodingParams::pPriority

- 8.82.2.12 **BYTE\*** cdmaMsgDecodingParams::pPrivacy
- 8.82.2.13 **BOOL\*** cdmaMsgDecodingParams::pReadAcknowledgementReq
- 8.82.2.14 **BYTE\*** cdmaMsgDecodingParams::pRelativeValidity
- 8.82.2.15 **CHAR\*** cdmaMsgDecodingParams::pSenderAddr
- 8.82.2.16 **BYTE\*** cdmaMsgDecodingParams::pSenderAddrLength
- 8.82.2.17 **WORD\*** cdmaMsgDecodingParams::pTextMsg
- 8.82.2.18 **BYTE\*** cdmaMsgDecodingParams::pTextMsgLength
- 8.82.2.19 **BOOL\*** cdmaMsgDecodingParams::pUserAcknowledgementReq

## 8.83 cdmaMsgEncodingParams Struct Reference

### Data Fields

- **BYTE \*** pMessageSize
- **BYTE \*** pMessage
- **BYTE** msgaseld
- **CHAR \*** pDestAddr
- **CHAR \*** pCallbackAddr
- **ULONG** textMsgLength
- **WORD \*** pTextMsg
- **BYTE \*** pPriority
- **BYTE \*** pEncodingAlphabet
- **BYTE \*** pRelValidity

### 8.83.1 Detailed Description

Structure contains parameters for message to be encoded

#### Parameters

<i>pMessageSize</i> [I/-N/OUT]	<ul style="list-style-type: none"> <li>Upon input, specifies the total number of bytes that the given pMessage buffer can hold (a buffer of length 240 is recommended). Upon successful output, specifies the length of the constructed message placed in the pMessage buffer (in bytes)</li> </ul>
<i>pMessage</i> [OUT]	- The constructed raw message
<i>msgaseld</i> [IN]	<ul style="list-style-type: none"> <li>The message reference number for this message. This value should be incremented for every message the host application sends</li> </ul>
<i>pDestAddr</i> [IN]	<ul style="list-style-type: none"> <li>Gives NULL-terminated ASCII String containing a destination address. International number will be prepended with a '+' character</li> </ul>

<i>pCallbackAddr</i> [I-N]	<ul style="list-style-type: none"> <li>Gives NULL-terminated ASCII String containing a callback address. International number will be prepended with a '+' character</li> </ul>
<i>textMsgLength</i> [I-N]	<ul style="list-style-type: none"> <li>Number of UCS2 characters in the text message(excluding NULL)</li> </ul>
<i>pTextMsg</i> [IN]	<ul style="list-style-type: none"> <li>Text message to be encoded</li> </ul>
<i>pPriority</i> [I-N](optional)	<p>parameter)</p> <ul style="list-style-type: none"> <li>Gives the priority of the outgoing message: 0 - normal (default if NULL pointer is given) 1 - interactive 2 - urgent 3 - emergency 64 - 64 is decoded value for URGENT VZAM Support interactive. 128 - 128 is decoded value for URGENT VZAM Support urgent. 192 - 128 is decoded value for URGENT VZAM Support emergency.</li> </ul>
<i>pEncodingAlphabet</i> [IN/OUT](optional)	<p>parameter)</p> <ul style="list-style-type: none"> <li>Upon input, specifies the alphabet the text message should be encoded in 0 - 8bit ASCII (not supported at this time) 1 - IS91EP (not supported at this time) 2 - 7bit ASCII (default if NULL pointer is given) 3 - IA5 (not supported at this time) 4 - unicode (not supported at this time) 5 - shift JIS (not supported at this time) 6 - korean (not supported at this time) 7 - latin hebrew (not supported at this time) 8 - latin (not supported at this time) 9 - GSM 7 bit default Upon successful output, specifies the alphabet used to encode the message.</li> </ul>
<i>pRelValidity</i> [I-N](optional)	<p>parameter)</p> <ul style="list-style-type: none"> <li>Gives the relative validity period of the outgoing message 0 - Set Relative validity to 11 1 - Set Relative validity to 71 2 - Set Relative validity to 167 3 - Set Relative validity to 169 4 - Set Relative validity to 171 Values have the following meanings: 0 to 143: validity period = (value + 1)* 5 minutes 144 to 167: validity period = 12 hours + (value - 143)*30 minutes 168 to 196: validity period = (value - 166) * 1 day 197 to 244: validity period = (value - 192) * 1 week 245: validity period = indefinite</li> </ul>

Currently only encoding of 7bit ASCII messages is supported.

## 8.83.2 Field Documentation

8.83.2.1 **BYTE** cdmaMsgEncodingParams::messageld

8.83.2.2 **CHAR\*** cdmaMsgEncodingParams::pCallbackAddr

8.83.2.3 **CHAR\*** cdmaMsgEncodingParams::pDestAddr

8.83.2.4 **BYTE\*** cdmaMsgEncodingParams::pEncodingAlphabet

8.83.2.5 **BYTE\*** cdmaMsgEncodingParams::pMessage

8.83.2.6 **BYTE\*** cdmaMsgEncodingParams::pMessageSize

8.83.2.7 **BYTE\*** cdmaMsgEncodingParams::pPriority

8.83.2.8 **BYTE\*** cdmaMsgEncodingParams::pRelValidity

8.83.2.9 **WORD\*** cdmaMsgEncodingParams::pTextMsg

8.83.2.10 **ULONG** cdmaMsgEncodingParams::textMsgLength

## 8.84 CDMARSSIThresh Struct Reference

### Data Fields

- [BYTE](#) CDMARSSIThreshListLen
- [WORD](#) \* pCDMARSSIThreshList

### 8.84.1 Detailed Description

This structure contains CDMA RSSI threshold related parameters.

#### Parameters

<i>CDMARSSI- ThreshListLen</i>	<ul style="list-style-type: none"> <li>• Length of the CDMARSSI threshold list parameter to follow</li> </ul>
<i>pCDMARSSI- ThreshList</i>	<ul style="list-style-type: none"> <li>• Array of RSSI thresholds (in units of 0.1 dBm)</li> <li>• maximum of 32 values.</li> <li>• Range for RSSI values:-105 to -21 (in dBm).</li> </ul>

### 8.84.2 Field Documentation

8.84.2.1 **BYTE** CDMARSSIThresh::CDMARSSIThreshListLen

8.84.2.2 **WORD\*** CDMARSSIThresh::pCDMARSSIThreshList

## 8.85 CDMASSInfo Struct Reference

### Data Fields

- [INT8](#) rssi
- [SHORT](#) ecio

### 8.85.1 Detailed Description

This structure contains the parameters for CDMA/WCDMA Signal Strength Information

#### Parameters

<i>rssi</i>	<ul style="list-style-type: none"> <li>• RSSI in dBm (signed value).</li> <li>• A value of -125 dBm or lower is used to indicate No Signal.</li> </ul>
-------------	--

<i>ecio</i>	<ul style="list-style-type: none"> <li>• ECIO value representing negative 0.5 dBm increments, i.e., 2 means -1 dBm (14 means -7 dBm, 63 means -31.5 dBm).</li> </ul>
-------------	--

## 8.85.2 Field Documentation

### 8.85.2.1 SHORT CDMASysInfo::ecio

### 8.85.2.2 INT8 CDMASysInfo::rssi

## 8.86 CDMASysInfo Struct Reference

### Data Fields

- [sysInfoCommon sysInfoCDMA](#)
- [BYTE isSysPrIMatchValid](#)
- [BYTE isSysPrIMatch](#)
- [BYTE pRevInUseValid](#)
- [BYTE pRevInUse](#)
- [BYTE bsPRevValid](#)
- [BYTE bsPRev](#)
- [BYTE ccsSupportedValid](#)
- [BYTE ccsSupported](#)
- [BYTE cdmaSysIdValid](#)
- [WORD systemID](#)
- [WORD networkID](#)
- [BYTE bsInfoValid](#)
- [WORD baseId](#)
- [ULONG baseLat](#)
- [ULONG baseLong](#)
- [BYTE packetZoneValid](#)
- [WORD packetZone](#)
- [BYTE networkIdValid](#)
- [BYTE MCC \[3\]](#)
- [BYTE MNC \[3\]](#)

### 8.86.1 Detailed Description

Structure for storing the CDMA System Information.

## Parameters

<i>sysInfoCDMA</i>	<ul style="list-style-type: none"> <li>• See <a href="#">sysInfoCommon</a> for more information.</li> </ul>
<i>isSysPrlMatch-Valid</i>	<ul style="list-style-type: none"> <li>• Indicates whether the system PRL match is valid. <ul style="list-style-type: none"> <li>– 0x00 - Invalid</li> <li>– 0x01 - Valid</li> <li>– 0xFF - Not Available</li> </ul> </li> </ul>
<i>isSysPrlMatch</i>	<ul style="list-style-type: none"> <li>• Indicates whether the system is in a PRL.</li> <li>• Only applies to CDMA/HDR. <ul style="list-style-type: none"> <li>– 0x00 - System is not in a PRL</li> <li>– 0x01 - System is in a PRL</li> <li>– 0xFF - Not Available</li> </ul> </li> <li>• If the system is not in a PRL, roam_status carries the value from the default roaming indicator in the PRL.</li> <li>• If the system is in a PRL, roam_status is set to the value based on the standard specification.</li> </ul>
<i>pRevInUseValid</i>	<ul style="list-style-type: none"> <li>• Indicates whether the P_Rev in use is valid. <ul style="list-style-type: none"> <li>– 0x00 - Invalid</li> <li>– 0x01 - Valid</li> <li>– 0xFF - Not Available</li> </ul> </li> </ul>
<i>pRevInUse</i>	<ul style="list-style-type: none"> <li>• The lesser of the base station P_Rev and mobile P_Rev</li> <li>• Only applicable for CDMA. <ul style="list-style-type: none"> <li>– 0xFF - Not Available</li> </ul> </li> </ul>
<i>bsPRevValid</i>	<ul style="list-style-type: none"> <li>• Indicates whether the base station P_Rev is valid <ul style="list-style-type: none"> <li>– 0x00 - Invalid</li> <li>– 0x01 - Valid</li> <li>– 0xFF - Not Available</li> </ul> </li> </ul>

<i>bsPRev</i>	<ul style="list-style-type: none"> <li>• Base station P_Rev.</li> <li>• Only applicable for CDMA. <ul style="list-style-type: none"> <li>– 0xFF - Not Available</li> </ul> </li> </ul>
<i>ccsSupported-Valid</i>	<ul style="list-style-type: none"> <li>• Indicates whether the supported concurrent service is valid. <ul style="list-style-type: none"> <li>– 0x00 - Invalid</li> <li>– 0x01 - Valid</li> <li>– 0xFF - Not Available</li> </ul> </li> </ul>
<i>ccsSupported</i>	<ul style="list-style-type: none"> <li>• Whether concurrent service is supported.</li> <li>• Only applicable for CDMA. <ul style="list-style-type: none"> <li>– 0x00 - Not supported</li> <li>– 0x01 - Supported</li> <li>– 0xFF - Not Available</li> </ul> </li> </ul>
<i>cdmaSysIdValid</i>	<ul style="list-style-type: none"> <li>• Indicates whether the CDMA system ID is valid. <ul style="list-style-type: none"> <li>– 0x00 - Invalid</li> <li>– 0x01 - Valid</li> <li>– 0xFF - Not Available</li> </ul> </li> </ul>
<i>systemID</i>	<ul style="list-style-type: none"> <li>• System ID. <ul style="list-style-type: none"> <li>– 0xFFFF - Not Available</li> </ul> </li> </ul>
<i>networkID</i>	<ul style="list-style-type: none"> <li>• Network ID. <ul style="list-style-type: none"> <li>– 0xFFFF - Not Available</li> </ul> </li> </ul>
<i>bsInfoValid</i>	<ul style="list-style-type: none"> <li>• Indicates whether the base station information is valid. <ul style="list-style-type: none"> <li>– 0x00 - Invalid</li> <li>– 0x01 - Valid</li> <li>– 0xFF - Not Available</li> </ul> </li> </ul>



<i>baseLat</i>	<ul style="list-style-type: none"> <li>• Base station latitude in units of 0.25 sec.</li> <li>• Expressed as a two's complement signed number with positive numbers signifying North latitudes. <ul style="list-style-type: none"> <li>– 0xFFFFFFFF - Not Available</li> </ul> </li> </ul>
<i>baseLong</i>	<ul style="list-style-type: none"> <li>• Base station longitude in units of 0.25 sec.</li> <li>• Expressed as a two's complement signed number with positive numbers signifying East latitudes. <ul style="list-style-type: none"> <li>– 0xFFFFFFFF - Not Available</li> </ul> </li> </ul>
<i>packetZoneValid</i>	<ul style="list-style-type: none"> <li>• Indicates whether the packet zone is valid. <ul style="list-style-type: none"> <li>– 0x00 - Invalid</li> <li>– 0x01 - Valid</li> <li>– 0xFF - Not Available</li> </ul> </li> </ul>
<i>packetZone</i>	<ul style="list-style-type: none"> <li>• Packet zone (8-bit). <ul style="list-style-type: none"> <li>– 0xFFFF indicates no packet zone.</li> </ul> </li> <li>• Only applicable for CDMA.</li> </ul>
<i>networkIdValid</i>	<ul style="list-style-type: none"> <li>• Indicates whether the network ID is valid. <ul style="list-style-type: none"> <li>– 0x00 - Invalid</li> <li>– 0x01 - Valid</li> <li>– 0xFF - Not Available</li> </ul> </li> </ul>
<i>MCC[PLMN_LENGTH]</i>	<ul style="list-style-type: none"> <li>• Mobile Country Code.</li> <li>• MCC digits in ASCII characters</li> </ul>
<i>MNC[PLMN_LENGTH]</i>	<ul style="list-style-type: none"> <li>• Mobile Network Code.</li> <li>• MNC digits in ASCII characters</li> <li>• An unused byte is set to 0xFF.</li> </ul>

## 8.86.2 Field Documentation

### 8.86.2.1 WORD CDMA SysInfo::baseId

- 8.86.2.2 **ULONG** CDMA SysInfo::baseLat
- 8.86.2.3 **ULONG** CDMA SysInfo::baseLong
- 8.86.2.4 **BYTE** CDMA SysInfo::bsInfoValid
- 8.86.2.5 **BYTE** CDMA SysInfo::bsPRev
- 8.86.2.6 **BYTE** CDMA SysInfo::bsPRevValid
- 8.86.2.7 **BYTE** CDMA SysInfo::ccsSupported
- 8.86.2.8 **BYTE** CDMA SysInfo::ccsSupportedValid
- 8.86.2.9 **BYTE** CDMA SysInfo::cdmaSysIdValid
- 8.86.2.10 **BYTE** CDMA SysInfo::isSysPrIMatch
- 8.86.2.11 **BYTE** CDMA SysInfo::isSysPrIMatchValid
- 8.86.2.12 **BYTE** CDMA SysInfo::MCC[3]
- 8.86.2.13 **BYTE** CDMA SysInfo::MNC[3]
- 8.86.2.14 **WORD** CDMA SysInfo::networkID
- 8.86.2.15 **BYTE** CDMA SysInfo::networkIdValid
- 8.86.2.16 **WORD** CDMA SysInfo::packetZone
- 8.86.2.17 **BYTE** CDMA SysInfo::packetZoneValid
- 8.86.2.18 **BYTE** CDMA SysInfo::pRevInUse
- 8.86.2.19 **BYTE** CDMA SysInfo::pRevInUseValid
- 8.86.2.20 **sysInfoCommon** CDMA SysInfo::sysInfoCDMA
- 8.86.2.21 **WORD** CDMA SysInfo::systemID

## 8.87 CDMA SysInfoExt Struct Reference

### Data Fields

- [WORD MCC](#)
- [BYTE imsi\\_11\\_12](#)

### 8.87.1 Detailed Description

This structure contains CDMA system information extension

- Parameter values default to their data type's maximum unsigned value unless explicitly stated otherwise.

## Parameters

<i>MCC</i>	<ul style="list-style-type: none"><li>• Mobile Country Code</li></ul>
<i>imsi_11_12</i>	<ul style="list-style-type: none"><li>• IMSI_11_12</li></ul>

## 8.87.2 Field Documentation

8.87.2.1 BYTE CDMA SysInfoExt::imsi\_11\_12

8.87.2.2 WORD CDMA SysInfoExt::MCC

## 8.88 CellDb Struct Reference

## Data Fields

- [ULONG mask](#)

## 8.88.1 Detailed Description

This structure contains the cell database

## Parameters

<i>mask</i>	<ul style="list-style-type: none"><li>• Mask for the cell database assistance data that is to be deleted</li><li>• Valid values:<ul style="list-style-type: none"><li>– 0x00000001 - DELETE_CELLDB_POS</li><li>– 0x00000002 - DELETE_CELLDB_LATEST_GPS_POS</li><li>– 0x00000004 - DELETE_CELLDB_OTA_POS</li><li>– 0x00000008 - DELETE_CELLDB_EXT_REF_POS</li><li>– 0x00000010 - DELETE_CELLDB_TIMETAG</li><li>– 0x00000020 - DELETE_CELLDB_CELLID</li><li>– 0x00000040 - DELETE_CELLDB_CACHED_CELLID</li><li>– 0x00000080 - DELETE_CELLDB_LAST_SRV_CELL</li><li>– 0x00000100 - DELETE_CELLDB_CUR_SRV_CELL</li><li>– 0x00000200 - DELETE_CELLDB_NEIGHBOR_INFO</li></ul></li></ul>
-------------	--

## 8.88.2 Field Documentation

8.88.2.1 ULONG CellDb::mask

## 8.89 cellParams Struct Reference

## Data Fields

- [WORD pci](#)
- [SHORT rsrq](#)
- [SHORT rsrp](#)
- [SHORT rssi](#)
- [SHORT srxlev](#)

### 8.89.1 Detailed Description

This structure contains information about the Cell parameters.

#### Parameters

<i>pci</i>	<ul style="list-style-type: none"> <li>• Physical cell ID.</li> <li>• Range: 0 to 503.</li> </ul>
<i>rsrq</i>	<ul style="list-style-type: none"> <li>• Current RSRQ in 1/10 dB as measured by L1.</li> <li>• Range: -20.0 dB to -3.0 dB.</li> </ul>
<i>rsrp</i>	<ul style="list-style-type: none"> <li>• Current RSRP in 1/10 dBm as measured by L1.</li> <li>• Range: -140.0 dBm to -44.0 dBm.</li> </ul>
<i>rssi</i>	<ul style="list-style-type: none"> <li>• Current RSSI in 1/10 dBm as measured by L1.</li> <li>• Range: -120.0 dBm to 0.</li> </ul>
<i>srxlev</i>	<ul style="list-style-type: none"> <li>• Cell selection Rx level (Srxlev) value.</li> <li>• Range: -128 to 128.</li> <li>• This field is only valid when ue_in_idle is TRUE.</li> </ul>

### 8.89.2 Field Documentation

8.89.2.1 **WORD** cellParams::pci

8.89.2.2 **SHORT** cellParams::rsrp

8.89.2.3 **SHORT** cellParams::rsrq

8.89.2.4 **SHORT** cellParams::rssi

8.89.2.5 **SHORT** cellParams::srxlev

## 8.90 changeUIMPIN Struct Reference

## Data Fields

- [BYTE pinID](#)
- [BYTE oldPINLen](#)
- [BYTE oldPINVal \[255\]](#)
- [BYTE pinLen](#)
- [BYTE pinVal \[255\]](#)

## 8.90.1 Detailed Description

This structure contains the information about the pin parameters that need to be verified.

## Parameters

<i>pinID</i>	<ul style="list-style-type: none"> <li>• Indicates the PIN ID to be changed. <ul style="list-style-type: none"> <li>– 1 - PIN1 (also called PIN)</li> <li>– 2 - PIN2</li> <li>– 3 - Universal PIN</li> <li>– 4 - Hidden key</li> </ul> </li> </ul>
<i>oldPINLen</i>	<ul style="list-style-type: none"> <li>• Length of the following elements i.e. old pin value.</li> </ul>
<i>oldPINVal[<small>MAX_DESCRIPTION_LENGTH</small>]</i>	<ul style="list-style-type: none"> <li>• Old PIN value.</li> <li>• This value is a sequence of ASCII characters.</li> </ul>
<i>pinLen</i>	<ul style="list-style-type: none"> <li>• Length of the following elements i.e. new pin value.</li> </ul>
<i>pinVal[<small>MAX_DESCRIPTION_LENGTH</small>]</i>	<ul style="list-style-type: none"> <li>• New PIN value.</li> <li>• This value is a sequence of ASCII characters.</li> </ul>

## 8.90.2 Field Documentation

8.90.2.1 **BYTE** changeUIMPIN::oldPINLen

8.90.2.2 **BYTE** changeUIMPIN::oldPINVal[255]

8.90.2.3 **BYTE** changeUIMPIN::pinID

8.90.2.4 **BYTE** changeUIMPIN::pinLen

8.90.2.5 **BYTE** changeUIMPIN::pinVal[255]

## 8.91 channelRate Struct Reference

## Data Fields

- [ULONG CurrChanTxRate](#)
- [ULONG CurrChanRxRate](#)

### 8.91.1 Detailed Description

This structure contains Channel Rate

#### Parameters

<i>CurrChanTxRate</i>	<ul style="list-style-type: none"> <li>• Max channel Tx rate in bits per second</li> </ul>
<i>CurrChanRxRate</i>	<ul style="list-style-type: none"> <li>• Max channel Rx rate in bits per second</li> </ul>

### 8.91.2 Field Documentation

8.91.2.1 **ULONG** channelRate::CurrChanRxRate

8.91.2.2 **ULONG** channelRate::CurrChanTxRate

## 8.92 ChannelRate Struct Reference

## Data Fields

- [ULONG CurrChanTxRate](#)
- [ULONG CurrChanRxRate](#)
- [ULONG MaxChanTxRate](#)
- [ULONG MaxChanRxRate](#)

### 8.92.1 Detailed Description

This structure contains Channel Rate

#### Parameters

<i>CurrChanTxRate</i>	<ul style="list-style-type: none"> <li>• Instantaneous channel Tx rate in bits per second</li> </ul>
<i>CurrChanRxRate</i>	<ul style="list-style-type: none"> <li>• Instantaneous channel Rx rate in bits per second</li> </ul>
<i>MaxChanTxRate</i>	<ul style="list-style-type: none"> <li>• maximum Tx rate that can be assigned to the device by the serving system in bits per second</li> </ul>

<i>MaxChanRxRate</i>	<ul style="list-style-type: none"><li>• maximum Rx rate that can be assigned to the device by the serving system in bits per second</li></ul>
----------------------	---

## 8.92.2 Field Documentation

8.92.2.1 **ULONG** ChannelRate::CurrChanRxRate

8.92.2.2 **ULONG** ChannelRate::CurrChanTxRate

8.92.2.3 **ULONG** ChannelRate::MaxChanRxRate

8.92.2.4 **ULONG** ChannelRate::MaxChanTxRate

## 8.93 CLIPResp Struct Reference

### Data Fields

- [BYTE](#) *ActiveStatus*
- [BYTE](#) *ProvisionStatus*

### 8.93.1 Detailed Description

This structure contains information about the Calling Line Identification Presentation (CLIP) supplementary service responses.

#### Parameters

<i>ActiveStatus</i>	<ul style="list-style-type: none"><li>• Active status.</li><li>• Values:<ul style="list-style-type: none"><li>– 0x00 - ACTIVE_STATUS_INACTIVE - Inactive</li><li>– 0x01 - ACTIVE_STATUS_ACTIVE - Active</li><li>– 0xFF - Not Available</li></ul></li></ul>
<i>ProvisionStatus</i>	<ul style="list-style-type: none"><li>• Provisioned status.</li><li>• Values:<ul style="list-style-type: none"><li>– 0x00 - PROVISION_STATUS_NOT_PROVISIONED - Not provisioned</li><li>– 0x01 - PROVISION_STATUS_PROVISIONED - Provisioned</li><li>– 0xFF - Not Available</li></ul></li></ul>

## 8.93.2 Field Documentation

8.93.2.1 **BYTE** CLIPResp::ActiveStatus

## 8.93.2.2 BYTE CLIPResp::ProvisionStatus

## 8.94 CLIRResp Struct Reference

## Data Fields

- [BYTE ActiveStatus](#)
- [BYTE ProvisionStatus](#)

## 8.94.1 Detailed Description

This structure contains information about the Calling Line Identification Restriction (CLIR) supplementary service responses.

## Parameters

<i>ActiveStatus</i>	<ul style="list-style-type: none"> <li>• Active status.</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0x00 - ACTIVE_STATUS_INACTIVE - Inactive</li> <li>– 0x01 - ACTIVE_STATUS_ACTIVE - Active</li> <li>– 0xFF - Not Available</li> </ul> </li> </ul>
<i>ProvisionStatus</i>	<ul style="list-style-type: none"> <li>• Provisioned status.</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0x00 - PROVISION_STATUS_NOT_PROVISIONED - Not provisioned</li> <li>– 0x01 - PROVISION_STATUS_PROVISIONED_PERMANENT - Permanently provisioned</li> <li>– 0x02 - PROVISION_STATUS_PRESENTATION_RESTRICTED - Restricted presentation</li> <li>– 0x03 - PROVISION_STATUS_PRESENTATION_ALLOWED - Allowed presentation</li> <li>– 0xFF - Not Available</li> </ul> </li> </ul>

## 8.94.2 Field Documentation

## 8.94.2.1 BYTE CLIRResp::ActiveStatus

## 8.94.2.2 BYTE CLIRResp::ProvisionStatus

## 8.95 ClkInfo Struct Reference

## Data Fields

- [ULONG mask](#)



### 8.95.1 Detailed Description

This structure contains the clock Info

## Parameters

<i>mask</i>	<ul style="list-style-type: none"> <li>• Mask for the clock information assistance data that is to be deleted</li> <li>• Valid bitmasks: <ul style="list-style-type: none"> <li>– QMI_LOC_MASK_DELETE_CLOCK_INFO_TIME_EST (0x00000001) - Mask to delete time estimate from clock information</li> <li>– QMI_LOC_MASK_DELETE_CLOCK_INFO_FREQ_EST (0x00000002) - Mask to delete frequency estimate from clock information</li> <li>– QMI_LOC_MASK_DELETE_CLOCK_INFO_WEEK_NUMBER (0x00000004) - Mask to delete week number from clock information</li> <li>– QMI_LOC_MASK_DELETE_CLOCK_INFO_RTC_TIME (0x00000008) - Mask to delete RTC time from clock information</li> <li>– QMI_LOC_MASK_DELETE_CLOCK_INFO_TIME_TRANSFER (0x00000010) - Mask to delete time transfer from clock information</li> <li>– QMI_LOC_MASK_DELETE_CLOCK_INFO_GPSTIME_EST (0x00000020) - Mask to delete GPS time estimate from clock information</li> <li>– QMI_LOC_MASK_DELETE_CLOCK_INFO_GLOTIME_EST (0x00000040) - Mask to delete GLONASS time estimate from clock information</li> <li>– QMI_LOC_MASK_DELETE_CLOCK_INFO_GLODAY_NUMBER (0x00000080) - Mask to delete GLONASS day number from clock information</li> <li>– QMI_LOC_MASK_DELETE_CLOCK_INFO_GLO4YEAR_NUMBER (0x00000100) - Mask to delete GLONASS four year number from clock information</li> <li>– QMI_LOC_MASK_DELETE_CLOCK_INFO_GLO_RF_GRP_DELAY (0x00000200) - Mask to delete GLONASS RF GRP delay from clock information</li> <li>– QMI_LOC_MASK_DELETE_CLOCK_INFO_DISABLE_TT (0x00000400) - Mask to delete disable TT from clock information</li> <li>– QMI_LOC_MASK_DELETE_CLOCK_INFO_GG_LEAPSEC (0x00000800) - Mask to delete a BDS time estimate from the clock information</li> <li>– QMI_LOC_MASK_DELETE_CLOCK_INFO_GG_GGTB (0x00001000) - Mask to delete a BDS time estimate from the clock information</li> <li>– QMI_LOC_MASK_DELETE_CLOCK_INFO_BDSTIME_EST (0x00002000) - Mask to delete a BDS time estimate from the clock information</li> <li>– QMI_LOC_MASK_DELETE_CLOCK_INFO_GB_GBTB (0x00004000) - Mask to delete Glonass-to-BDS time bias-related information from the clock information</li> <li>– QMI_LOC_MASK_DELETE_CLOCK_INFO_BG_BGTB (0x00008000) - Mask to delete BDS-to-GLONASS time bias-related information from the clock information</li> <li>– QMI_LOC_MASK_DELETE_CLOCK_INFO_BDSWEEK_NUMBER (0x00010000) - Mask to delete the BDS week number from the clock information</li> <li>– QMI_LOC_MASK_DELETE_CLOCK_INFO_BDS_RF_GRP_DELAY (0x00020000) - Mask to delete the BDS RF GRP delay from the clock information</li> </ul> </li> </ul>
-------------	--

## 8.95.2 Field Documentation

## 8.95.2.1 ULONG CkInfo::mask

## 8.96 CNAPResp Struct Reference

### Data Fields

- [BYTE ActiveStatus](#)
- [BYTE ProvisionStatus](#)

### 8.96.1 Detailed Description

This structure contains information about the Calling Name Presentation (CNAP) supplementary service responses.

#### Parameters

<i>ActiveStatus</i>	<ul style="list-style-type: none"><li>• Active status.</li><li>• Values:<ul style="list-style-type: none"><li>– 0x00 - ACTIVE_STATUS_INACTIVE - Inactive</li><li>– 0x01 - ACTIVE_STATUS_ACTIVE - Active</li><li>– 0xFF - Not Available</li></ul></li></ul>
<i>ProvisionStatus</i>	<ul style="list-style-type: none"><li>• Provisioned status.</li><li>• Values:<ul style="list-style-type: none"><li>– 0x00 - PROVISION_STATUS_NOT_PROVISIONED - Not provisioned</li><li>– 0x01 - PROVISION_STATUS_PROVISIONED - Provisioned</li><li>– 0xFF - Not Available</li></ul></li></ul>

### 8.96.2 Field Documentation

## 8.96.2.1 BYTE CNAPResp::ActiveStatus

## 8.96.2.2 BYTE CNAPResp::ProvisionStatus

## 8.97 COLPResp Struct Reference

### Data Fields

- [BYTE ActiveStatus](#)
- [BYTE ProvisionStatus](#)

### 8.97.1 Detailed Description

This structure contains information about the Connected Line Identification Presentation (COLP) supplementary service responses.

## Parameters

<i>ActiveStatus</i>	<ul style="list-style-type: none"> <li>• Active status.</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0x00 - ACTIVE_STATUS_INACTIVE - Inactive</li> <li>– 0x01 - ACTIVE_STATUS_ACTIVE - Active</li> <li>– 0xFF - Not Available</li> </ul> </li> </ul>
<i>ProvisionStatus</i>	<ul style="list-style-type: none"> <li>• Provisioned status.</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0x00 - PROVISION_STATUS_NOT_PROVISIONED - Not provisioned</li> <li>– 0x01 - PROVISION_STATUS_PROVISIONED - Provisioned</li> <li>– 0xFF - Not Available</li> </ul> </li> </ul>

## 8.97.2 Field Documentation

## 8.97.2.1 BYTE COLPResp::ActiveStatus

## 8.97.2.2 BYTE COLPResp::ProvisionStatus

## 8.98 COLRResp Struct Reference

## Data Fields

- [BYTE ActiveStatus](#)
- [BYTE ProvisionStatus](#)

## 8.98.1 Detailed Description

This structure contains information about the Connected Line Identification Restriction (COLR) supplementary service responses.

## Parameters

<i>ActiveStatus</i>	<ul style="list-style-type: none"> <li>• Active status.</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0x00 - ACTIVE_STATUS_INACTIVE - Inactive</li> <li>– 0x01 - ACTIVE_STATUS_ACTIVE - Active</li> <li>– 0xFF - Not Available</li> </ul> </li> </ul>
---------------------	---

<i>ProvisionStatus</i>	<ul style="list-style-type: none"><li>• Provisioned status.</li><li>• Values:<ul style="list-style-type: none"><li>– 0x00 - PROVISION_STATUS_NOT_PROVISIONED - Not provisioned</li><li>– 0x01 - PROVISION_STATUS_PROVISIONED - Provisioned</li><li>– 0xFF - Not Available</li></ul></li></ul>
------------------------	---

## 8.98.2 Field Documentation

### 8.98.2.1 BYTE COLRResp::ActiveStatus

### 8.98.2.2 BYTE COLRResp::ProvisionStatus

## 8.99 CommInfo Struct Reference

### Data Fields

- [BYTE temperature](#)
- [BYTE modemMode](#)
- [BYTE systemMode](#)
- [BYTE imsRegState](#)
- [BYTE psState](#)

### 8.99.1 Detailed Description

Structure for storing the common information for the device.

## Parameters

<i>temperature</i>	<ul style="list-style-type: none"> <li>• Temperature. <ul style="list-style-type: none"> <li>– 8-bit signed integer</li> <li>– 0xFF - Not Available.</li> </ul> </li> </ul>
<i>modemMode</i>	<ul style="list-style-type: none"> <li>• Modem Operating Mode. <ul style="list-style-type: none"> <li>– 0x00 - POWERING OFF</li> <li>– 0x01 - FACTORY TEST</li> <li>– 0x02 - OFFLINE</li> <li>– 0x03 - OFFLINE_AMPS</li> <li>– 0x04 - OFFLINE_CDMA</li> <li>– 0x05 - ONLINE</li> <li>– 0x06 - LOW POWER MODE</li> <li>– 0x07 - RESETTING</li> <li>– 0x08 - NETWORK TEST</li> <li>– 0x09 - OFFLINE REQUEST</li> <li>– 0x0A - PSEUDO ONLINE</li> <li>– 0x0B - RESETTING MODEM</li> <li>– 0xFF - Unknown</li> </ul> </li> </ul>
<i>systemMode</i>	<ul style="list-style-type: none"> <li>• System Acquisition Mode. <ul style="list-style-type: none"> <li>– 0x00 - No service</li> <li>– 0x01 - AMPS</li> <li>– 0x02 - CDMA</li> <li>– 0x03 - GSM</li> <li>– 0x04 - HDR</li> <li>– 0x05 - WCDMA</li> <li>– 0x06 - GPS</li> <li>– 0x08 - WLAN</li> <li>– 0x09 - LTE</li> <li>– 0xFF - Unknown</li> </ul> </li> </ul>

<i>imsRegState</i>	<ul style="list-style-type: none"><li>• IMS Registration State.<ul style="list-style-type: none"><li>– 0x00 - NO SRV</li><li>– 0x01 - IN PROG</li><li>– 0x02 - FAILED</li><li>– 0x03 - LIMITED</li><li>– 0x04 - FULL SRV</li><li>– 0xFF - Unknown</li></ul></li></ul>
<i>psState</i>	<ul style="list-style-type: none"><li>• PS Attach State.<ul style="list-style-type: none"><li>– 0x00 - Attached</li><li>– 0x01 - Detached</li><li>– 0xFF - Unknown</li></ul></li></ul>

## 8.99.2 Field Documentation

8.99.2.1 **BYTE** CommInfo::imsRegState

8.99.2.2 **BYTE** CommInfo::modemMode

8.99.2.3 **BYTE** CommInfo::psState

8.99.2.4 **BYTE** CommInfo::systemMode

8.99.2.5 **BYTE** CommInfo::temperature

## 8.100 ConnectionStatus Struct Reference

### Data Fields

- [BYTE MDMConnStatus](#)
- [ULONGLONG MDMDuration](#)

### 8.100.1 Detailed Description

This structure contains modem connection status

## Parameters

<i>MDMConn-Status</i>	<ul style="list-style-type: none"> <li>Current link status             <ul style="list-style-type: none"> <li>0x01 - DISCONNECTED</li> <li>0x02 - CONNECTED</li> </ul> </li> </ul>
<i>MDMCall-Duration</i>	<ul style="list-style-type: none"> <li>Call duration in milliseconds.</li> <li>If the modem connection status is connected, this represent the duration of the current DUN call.</li> <li>If the modem connection status is disconnected, this represents the duration of the last DUN call since the device was powered up (zero, if no call has been made or if the last call was not DUN).</li> </ul>

## 8.100.2 Field Documentation

## 8.100.2.1 ULONGLONG ConnectionStatus::MDMCallDuration

## 8.100.2.2 BYTE ConnectionStatus::MDMConnStatus

## 8.101 connectNumInfo Struct Reference

## Data Fields

- [BYTE numPresInd](#)
- [BYTE screeningInd](#)
- [BYTE numType](#)
- [BYTE numPlan](#)
- [BYTE callerIDLen](#)
- [BYTE callerID](#) [81]

## 8.101.1 Detailed Description

This structure contains information about the numbers connected to a device. It contains information such as number type, eg International or Local.



## Parameters

<i>numPresInd</i>	<ul style="list-style-type: none"> <li>• Presentation indicator <ul style="list-style-type: none"> <li>– 0x00 - PRESENTATION_ALLOWED - Allowed presentation</li> <li>– 0x01 - PRESENTATION_RESTRICTED - Restricted presentation</li> <li>– 0x02 - PRESENTATION_NUM_UNAVAILABLE - Unavailable presentation</li> <li>– 0x04 - PRESENTATION_PAYPHONE - Payphone presentation (GSM/UMTS specific)</li> <li>– 0xFF - Not Available</li> </ul> </li> </ul>
<i>screeningInd</i>	<ul style="list-style-type: none"> <li>• Screening indicator. <ul style="list-style-type: none"> <li>– 0x00 - QMI_VOICE_SI_USER_PROVIDED_NOT_SCREENED - Provided user is not screened</li> <li>– 0x01 - QMI_VOICE_SI_USER_PROVIDED_VERIFIED_PASSED - Provided user passed verification</li> <li>– 0x02 - QMI_VOICE_SI_USER_PROVIDED_VERIFIED_FAILED - Provided user failed verification</li> <li>– 0x03 - QMI_VOICE_SI_NETWORK_PROVIDED - Provided network</li> <li>– 0xFF - Not Available</li> </ul> </li> </ul>
<i>numType</i>	<ul style="list-style-type: none"> <li>• Number type. <ul style="list-style-type: none"> <li>– 0x00 - QMI_VOICE_NUM_TYPE_UNKNOWN - Unknown</li> <li>– 0x01 - QMI_VOICE_NUM_TYPE_INTERNATIONAL - International</li> <li>– 0x02 - QMI_VOICE_NUM_TYPE_NATIONAL - National</li> <li>– 0x03 - QMI_VOICE_NUM_TYPE_NETWORK_SPECIFIC - Network-specific</li> <li>– 0x04 - QMI_VOICE_NUM_TYPE_SUBSCRIBER - Subscriber</li> <li>– 0x05 - QMI_VOICE_NUM_TYPE_RESERVED - Reserved</li> <li>– 0x06 - QMI_VOICE_NUM_TYPE_ABBREVIATED - Abbreviated</li> <li>– 0x07 - QMI_VOICE_NUM_TYPE_RESERVED_EXTENSION - Reserved extension</li> <li>– 0xFF - Not Available</li> </ul> </li> </ul>

<i>numPlan</i>	<ul style="list-style-type: none"> <li>• Number plan. <ul style="list-style-type: none"> <li>– 0x00 - QMI_VOICE_NUM_PLAN_UNKNOWN - Unknown</li> <li>– 0x01 - QMI_VOICE_NUM_PLAN_ISDN - ISDN</li> <li>– 0x03 - QMI_VOICE_NUM_PLAN_DATA - Data</li> <li>– 0x04 - QMI_VOICE_NUM_PLAN_TELEX - Telex</li> <li>– 0x08 - QMI_VOICE_NUM_PLAN_NATIONAL - National</li> <li>– 0x09 - QMI_VOICE_NUM_PLAN_PRIVATE - Private</li> <li>– 0x0B - QMI_VOICE_NUM_PLAN_RESERVED_CTS - Reserved cordless telephony system</li> <li>– 0x0F - QMI_VOICE_NUM_PLAN_RESERVED_EXTENSION - Reserved extension</li> <li>– 0xFF - Not Available</li> </ul> </li> </ul>
<i>callerIDLen</i>	<ul style="list-style-type: none"> <li>• Provides the length of caller ID which follow.</li> <li>• If zero(0) then no further information exists.</li> </ul>
<i>callerID[MAX_CALL_NO_LEN]</i>	<ul style="list-style-type: none"> <li>• callerID of numLen length, NULL terminated.</li> </ul>

## 8.101.2 Field Documentation

8.101.2.1 **BYTE** connectNumInfo::callerID[81]

8.101.2.2 **BYTE** connectNumInfo::callerIDLen

8.101.2.3 **BYTE** connectNumInfo::numPlan

8.101.2.4 **BYTE** connectNumInfo::numPresInd

8.101.2.5 **BYTE** connectNumInfo::numType

8.101.2.6 **BYTE** connectNumInfo::screeningInd

## 8.102 CrashInfo Struct Reference

### Data Fields

- [WORD](#) numCrashes
- [ULONG](#) crashId
- [ULONG](#) crashData
- [WORD](#) crashStrLen
- [CHAR](#) \* pCrashString
- [WORD](#) gcDumpStrLen
- [CHAR](#) \* pGCDumpString

### 8.102.1 Detailed Description

This structure is used to store Crash Information

## Parameters

<i>numCrashes</i> [O-UT]	<ul style="list-style-type: none"> <li>• Number of instances of the remaining fields</li> </ul>
<i>crashId</i> [OUT]	<ul style="list-style-type: none"> <li>• Random crash id assigned at crash</li> </ul>
<i>crashData</i> [OUT]	<ul style="list-style-type: none"> <li>• Crash Data</li> </ul>
<i>crashStrLen</i> [IN-OUT]	<ul style="list-style-type: none"> <li>• Length of the pCrashString field returned by API</li> </ul>
<i>pCrashString</i> [O-UT]	<ul style="list-style-type: none"> <li>• Pointer to store crash string</li> </ul>
<i>gcDumpStrLen</i> [I-N/OUT]	<ul style="list-style-type: none"> <li>• Length of the pGCDumpString field returned by API</li> </ul>
<i>pGCDumpString</i> [OUT]	<ul style="list-style-type: none"> <li>• gcdump string for the crash</li> </ul>

## 8.102.2 Field Documentation

8.102.2.1 **ULONG** CrashInfo::crashData8.102.2.2 **ULONG** CrashInfo::crashId8.102.2.3 **WORD** CrashInfo::crashStrLen8.102.2.4 **WORD** CrashInfo::gcDumpStrLen8.102.2.5 **WORD** CrashInfo::numCrashes8.102.2.6 **CHAR\*** CrashInfo::pCrashString8.102.2.7 **CHAR\*** CrashInfo::pGCDumpString

## 8.103 CrashInfoParams Struct Reference

## Data Fields

- [BYTE](#) \* [pDevCrashStatus](#)
- [CrashInfo](#) \* [pCrashInfo](#)

## 8.103.1 Detailed Description

This structure is used to store Crash Information

## Parameters

<i>pDevCrash-Status[OUT]</i>	<ul style="list-style-type: none"> <li>• Device Crash Status</li> <li>• 0 - no crash</li> <li>• 1 - crash has occurred</li> </ul>
<i>pCrashInfo[OUT]</i>	<ul style="list-style-type: none"> <li>• Pointer to structure <a href="#">CrashInfo</a></li> <li>• See <a href="#">CrashInfo</a> for more information</li> </ul>

## 8.103.2 Field Documentation

8.103.2.1 [CrashInfo](#)\* [CrashInfoParams::pCrashInfo](#)8.103.2.2 [BYTE](#)\* [CrashInfoParams::pDevCrashStatus](#)

## 8.104 CreateProfileIn Struct Reference

## Data Fields

- [BYTE](#) \* [pProfileID](#)
- [BYTE](#) \* [pProfileType](#)
- [QmiProfileInfo](#) [curProfile](#)

## 8.104.1 Detailed Description

This structure contains the input parameters for [SLQSCreateProfile](#)

## Parameters

<i>ProfileID</i>	<ul style="list-style-type: none"> <li>• 1 to 16 for 3GPP profile (EM/MC73xx or earlier)</li> <li>• 1 to 24 for 3GPP profile (EM/MC74xx onwards)</li> <li>• 101 to 106 for 3GPP2 profile</li> </ul>
<i>ProfileType</i>	<ul style="list-style-type: none"> <li>• Identifies the technology type of the profile <ul style="list-style-type: none"> <li>– 0x00 - 3GPP</li> <li>– 0x01 - 3GPP2</li> <li>– NULL is not allowed</li> </ul> </li> </ul>

<i>curProfile</i>	<ul style="list-style-type: none"> <li>• union of <a href="#">Profile3GPP</a> and <a href="#">Profile3GPP2</a></li> </ul>
-------------------	---

**Note**

- If profileID is NULL, 3GPP profile will be created and index will be assigned based on availability in device.
- If profileID is not NULL depending on pProfileType 3GPP/3GPP2 relevant profile will be created

**8.104.2 Field Documentation****8.104.2.1 QmiProfileInfo CreateProfileIn::curProfile****8.104.2.2 BYTE\* CreateProfileIn::pProfileID****8.104.2.3 BYTE\* CreateProfileIn::pProfileType****8.105 CreateProfileOut Struct Reference****Data Fields**

- [BYTE](#) \* [pProfileType](#)
- [BYTE](#) \* [pProfileIndex](#)
- [USHORT](#) \* [pExtErrorCode](#)

**8.105.1 Detailed Description**

structure contains out parameter Information

**Parameters**

<i>profileType</i>	<ul style="list-style-type: none"> <li>• Identifies the type of profile 0x00 = 3GPP 0x01 = 3GPP2</li> </ul>
<i>profileIndex</i>	<ul style="list-style-type: none"> <li>• Index identifying the profile that was created</li> </ul>
<i>pExtErrorCode</i>	<ul style="list-style-type: none"> <li>• The extended error code received from DS Profile subsystem</li> </ul>

**8.105.2 Field Documentation****8.105.2.1 USHORT\* CreateProfileOut::pExtErrorCode****8.105.2.2 BYTE\* CreateProfileOut::pProfileIndex****8.105.2.3 BYTE\* CreateProfileOut::pProfileType****8.106 CSGID Struct Reference**

## Data Fields

- [WORD mcc](#)
- [WORD mnc](#)
- [BYTE mncPcsDigits](#)
- [ULONG id](#)
- [BYTE rat](#)

### 8.106.1 Detailed Description

Contain the [CSGID](#).

Parameters

<i>mcc</i>	<ul style="list-style-type: none"><li>• MCC value. Range 0 to 999</li></ul>
<i>mnc</i>	<ul style="list-style-type: none"><li>• MNC value. Range 0 to 999</li></ul>
<i>mncPcsDigits</i>	<ul style="list-style-type: none"><li>• TRUE - MNC is a three-digit value; e.g., a reported value of 90 corresponds to an MNC value of 090</li><li>• FALSE - MNC is a two-digit value; e.g., a reported value of 90 corresponds to an MNC value of 90</li></ul>
<i>id</i>	<ul style="list-style-type: none"><li>• Closed subscriber group identifier.</li></ul>
<i>rat</i>	<ul style="list-style-type: none"><li>• Radio interface technology of the CSG network. Values:<ul style="list-style-type: none"><li>– 0x04 - RADIO_IF_GSM - GSM</li><li>– 0x05 - RADIO_IF_UMTS - UMTS</li><li>– 0x08 - RADIO_IF_LTE - LTE</li><li>– 0x09 - RADIO_IF_TDSCDMA - TDS</li></ul></li></ul>

### 8.106.2 Field Documentation

8.106.2.1 **ULONG** CSGID::id

8.106.2.2 **WORD** CSGID::mcc

8.106.2.3 **WORD** CSGID::mnc

8.106.2.4 **BYTE** CSGID::mncPcsDigits

8.106.2.5 **BYTE** CSGID::rat

## 8.107 CUGInfo Struct Reference

## Data Fields

- [WORD CUGIndex](#)
- [BYTE SuppPrefCUG](#)
- [BYTE SuppOA](#)

### 8.107.1 Detailed Description

This structure contains Closed User Group Information

#### Parameters

<i>CUGIndex</i>	<ul style="list-style-type: none"> <li>• Range 0x00... 0x7FFF</li> </ul>
<i>SuppPrefCUG</i>	<ul style="list-style-type: none"> <li>• Suppress preferential CUG <ul style="list-style-type: none"> <li>– 0x00 - FALSE</li> <li>– 0x01 - TRUE</li> </ul> </li> </ul>
<i>SuppOA</i>	<ul style="list-style-type: none"> <li>• Suppress OA subscription option <ul style="list-style-type: none"> <li>– 0x00 - FALSE</li> <li>– 0x01 - TRUE</li> </ul> </li> </ul>

### 8.107.2 Field Documentation

8.107.2.1 [WORD CUGInfo::CUGIndex](#)

8.107.2.2 [BYTE CUGInfo::SuppOA](#)

8.107.2.3 [BYTE CUGInfo::SuppPrefCUG](#)

## 8.108 curAMRConfig Struct Reference

## Data Fields

- [BYTE gsmAmrStat](#)
- [BYTE wcdmaAmrStat](#)

### 8.108.1 Detailed Description

This structure contains the Current Adaptive Multi Rate Configuration Information.

#### Parameters

---



<i>gsmAmrStat</i>	<ul style="list-style-type: none"> <li>• GSM AMR Status <ul style="list-style-type: none"> <li>– 0x00 - Disable</li> <li>– 0x01 - Enable</li> <li>– 0xFF - Not Available</li> </ul> </li> </ul>
<i>wcdmaAmrStat</i>	<ul style="list-style-type: none"> <li>• WCDMA AMR Status</li> <li>• One or a combination of the following bitmask values: <ul style="list-style-type: none"> <li>– Bit 0 - AMR codec advertised is not supported</li> <li>– Bit 1 - Controls WCDMA AMR wideband</li> <li>– Bit 2 - Controls GSM half rate AMR</li> <li>– Bit 3 - Controls GSM AMR wideband</li> <li>– Bit 4 - Controls GSM AMR narrowband</li> </ul> </li> <li>• 0xFF, if not available</li> </ul>

## 8.108.2 Field Documentation

8.108.2.1 BYTE curAMRConfig::gsmAmrStat

8.108.2.2 BYTE curAMRConfig::wcdmaAmrStat

## 8.109 CurrDataSysStat Struct Reference

### Data Fields

- [BYTE \\* pPrefNetwork](#)
- [BYTE \\* pNetworkInfoLen](#)
- [CurrNetworkInfo \\* pCurrNetworkInfo](#)

### 8.109.1 Detailed Description

Data System Status

Parameters

<i>pPrefNetwork</i>	[OUT] <ul style="list-style-type: none"> <li>• Preferred Network</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0 - 3GPP</li> <li>– 1 - 3GPP2</li> </ul> </li> </ul>
---------------------	---

<i>pNetworkInfoLen</i>	[IN/OUT] <ul style="list-style-type: none"> <li>• As input parameter size assigned to next parameter i.e. network information</li> <li>• As output the actual number of network information elements returned by the device</li> </ul>
<i>pCurrNetwork-Info</i>	[OUT] <ul style="list-style-type: none"> <li>• Network information</li> <li>• See <a href="#">CurrNetworkInfo</a> for more details</li> </ul>

## 8.109.2 Field Documentation

8.109.2.1 **CurrNetworkInfo\*** CurrDataSysStat::pCurrNetworkInfo

8.109.2.2 **BYTE\*** CurrDataSysStat::pNetworkInfoLen

8.109.2.3 **BYTE\*** CurrDataSysStat::pPrefNetwork

## 8.110 currentCatEvent Union Reference

### Data Fields

- struct [CatEventIDDataTlv](#) CatEvIDData
- struct [CatAlPhalIdentifierTlv](#) CatAlphaldtfr
- struct [CatEventListTlv](#) CatEventLst
- struct [CatRefreshTlv](#) CatRefresh
- struct [CatEndProactiveSessionTlv](#) CatEndPS

### 8.110.1 Detailed Description

Union used to represent the current CAT Event Data. Choose the structure based on the EventID received.

- Use [CatEventIDDataTlv](#) if the Event ID is any of the below.
  - 16
  - 17
  - 18
  - 19
  - 20
  - 23
  - 24
- Use [CatAlPhalIdentifierTlv](#) if the Event ID is 21
- Use [CatEventListTlv](#) if the Event ID is 22
- Use [CatRefreshTlv](#) if the Event ID is 25
- Use [CatEndProactiveSessionTlv](#) if the Event ID is 26

## 8.110.2 Field Documentation

8.110.2.1 struct CatAlPhalIdentifierTlv currentCatEvent::CatAlphaldtfr

8.110.2.2 struct CatEndProactiveSessionTlv currentCatEvent::CatEndPS

8.110.2.3 struct CatEventListTlv currentCatEvent::CatEventLst

8.110.2.4 struct CatEventIDDDataTlv currentCatEvent::CatEvIDDData

8.110.2.5 struct CatRefreshTlv currentCatEvent::CatRefresh

## 8.111 CurrentImgList Struct Reference

### Data Fields

- [BYTE numEntries](#)
- [CurrImgInfo](#) \* [pCurrImgInfo](#)
- [CHAR priver](#) [16]
- [CHAR pkgver](#) [16]
- [CHAR fwvers](#) [16]
- [CHAR carrier](#) [16]

### 8.111.1 Detailed Description

This structure is used to store image list

#### Parameters

<i>numEntries</i> [IN/-OUT]	<ul style="list-style-type: none"> <li>• Number of entries in the image list to follow</li> <li>• The size of the list <a href="#">pCurrImgInfo</a> must be specified when calling the API</li> </ul>
<i>pCurrImgInfo</i> [OUT]	<ul style="list-style-type: none"> <li>• Currently Active Image List</li> </ul>
<i>priver</i> [OUT]	<ul style="list-style-type: none"> <li>• PRI version of the currently running firmware</li> </ul>
<i>pkgver</i> [OUT]	<ul style="list-style-type: none"> <li>• Package version of the currently running firmware</li> </ul>
<i>fwvers</i> [OUT]	<ul style="list-style-type: none"> <li>• firmware version of the currently running firmware</li> </ul>
<i>carrier</i> [OUT]	<ul style="list-style-type: none"> <li>• Carrier string of the currently running firmware</li> </ul>

## 8.111.2 Field Documentation

8.111.2.1 [CHAR](#) [CurrentImgList::carrier](#)[16]

8.111.2.2 **CHAR** CurrentImgList::fwvers[16]

8.111.2.3 **BYTE** CurrentImgList::numEntries

8.111.2.4 **CurrImageInfo\*** CurrentImgList::pCurrImgInfo

8.111.2.5 **CHAR** CurrentImgList::pkgver[16]

8.111.2.6 **CHAR** CurrentImgList::priver[16]

## 8.112 currentPLMN Struct Reference

### Data Fields

- [WORD MCC](#)
- [WORD MNC](#)
- [BYTE netDescrLength](#)
- [BYTE netDescr](#) [255]

### 8.112.1 Detailed Description

This structure contains the current PLMN parameters

- Parameter values default to their data type's maximum unsigned value unless explicitly stated otherwise.

#### Parameters

<i>MCC</i>	<ul style="list-style-type: none"> <li>• mobile country code               <ul style="list-style-type: none"> <li>– A 16 bit representation of MCC</li> <li>– Range 0 to 999</li> </ul> </li> </ul>
<i>MNC</i>	<ul style="list-style-type: none"> <li>• mobile network code               <ul style="list-style-type: none"> <li>– A 16 bit representation of MNC</li> <li>– Range 0 to 999</li> </ul> </li> </ul>
<i>netDescrLength</i>	<ul style="list-style-type: none"> <li>• Length of Network description field</li> <li>• Defaults to zero</li> </ul>
<i>netDescr</i>	<ul style="list-style-type: none"> <li>• Network Description               <ul style="list-style-type: none"> <li>– optional string containing network name or description</li> </ul> </li> </ul>

### 8.112.2 Field Documentation

8.112.2.1 WORD currentPLMN::MCC

8.112.2.2 WORD currentPLMN::MNC

8.112.2.3 BYTE currentPLMN::netDescr[255]

8.112.2.4 BYTE currentPLMN::netDescrLength

## 8.113 CurrImageInfo Struct Reference

### Data Fields

- [BYTE imageType](#)
- [BYTE uniqueID](#) [16]
- [BYTE buildIDLen](#)
- [BYTE buildID](#) [255]

### 8.113.1 Detailed Description

This structure is used to store image information

#### Parameters

<i>imageType</i> [OUT]	<ul style="list-style-type: none"> <li>• Image Type</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0 - FW</li> <li>– 1 - configuration</li> </ul> </li> </ul>
<i>uniqueID</i> [OUT]	<ul style="list-style-type: none"> <li>• Image Unique Identifier ( ASCII characters )</li> </ul>
<i>buildIDLen</i> [OUT]	<ul style="list-style-type: none"> <li>• Length of build ID string to follow</li> <li>• If set to zero, build ID string will be blank</li> </ul>
<i>buildID</i> [OUT]	<ul style="list-style-type: none"> <li>• String containing image information( ASCII characters )</li> <li>• Maximum length of this string is 255 chars</li> </ul>

### 8.113.2 Field Documentation

8.113.2.1 BYTE CurrImageInfo::buildID[255]

8.113.2.2 BYTE CurrImageInfo::buildIDLen

8.113.2.3 BYTE CurrImageInfo::imageType

8.113.2.4 BYTE CurrImageInfo::uniqueID[16]

## 8.114 CurrNetworkInfo Struct Reference

### Data Fields

- [BYTE NetworkType](#)
- [ULONG RATMask](#)
- [ULONG SOMask](#)

### 8.114.1 Detailed Description

Network information structure

## Parameters

<i>NetworkType</i>	<ul style="list-style-type: none"> <li>• Values: <ul style="list-style-type: none"> <li>– 0 - 3GPP</li> <li>– 1 - 3GPP2</li> </ul> </li> </ul>
<i>RATMask</i>	<ul style="list-style-type: none"> <li>• Radio Access Technology (RAT) mask to indicate the type of technology.</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0 - Don't Care</li> <li>– 0x8000 - NULL Bearer</li> </ul> </li> <li>• CDMA RAT mask values: <ul style="list-style-type: none"> <li>– 0x01 - CDMA_1x</li> <li>– 0x02 - EVDO_REV0</li> <li>– 0x04 - EVDO_REVA</li> <li>– 0x08 - EVDO_REVB</li> <li>– 0x10 - EHRPD</li> <li>– 0x20 - FMC</li> </ul> </li> <li>• UMTS RAT mask values: <ul style="list-style-type: none"> <li>– 0x01 - WCDMA</li> <li>– 0x02 - GPRS</li> <li>– 0x04 - HSDPA</li> <li>– 0x08 - HSUPA</li> <li>– 0x10 - EDGE</li> <li>– 0x20 - LTE</li> <li>– 0x40 - HSDPA+</li> <li>– 0x80 - DC_HSDPA+</li> <li>– 0x100 - 64_QAM</li> <li>– 0x200 - TDSCDMA</li> </ul> </li> </ul>
<i>SOMask</i>	<ul style="list-style-type: none"> <li>• Service Option (SO) mask to indicate the service option or type of application.</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0 - Don't Care</li> </ul> </li> <li>• CDMA 1x SO mask values: <ul style="list-style-type: none"> <li>– 0x01 - CDMA_1X_IS95</li> <li>– 0x02 - CDMA_1X_IS2000</li> <li>– 0x04 - CDMA_1X_IS2000_REL_A</li> </ul> </li> <li>• CDMA EV-DO Rev 0 SO mask values: <ul style="list-style-type: none"> <li>– 0x01 - DPA</li> </ul> </li> <li>• CDMA EV-DO Rev A SO mask values: <ul style="list-style-type: none"> <li>– 0x01 - DPA</li> <li>– 0x02 - MFPA</li> <li>– 0x04 - EMPA</li> <li>– 0x08 - EMPA_EHRPD</li> </ul> </li> </ul>

### 8.114.2 Field Documentation

8.114.2.1 **BYTE** CurrNetworkInfo::NetworkType

8.114.2.2 **ULONG** CurrNetworkInfo::RATMask

8.114.2.3 **ULONG** CurrNetworkInfo::SOMask

## 8.115 custFeaturesInfo Struct Reference

### Data Fields

- [ULONG](#) GpsEnable
- [BYTE](#) \* pDisableIMSI
- [WORD](#) \* pIPFamSupport
- [BYTE](#) \* pRMAutoConnect
- [BYTE](#) \* pGPSSel
- [BYTE](#) \* pSMSSupport
- [BYTE](#) \* plsVoiceEnabled
- [BYTE](#) \* pDHCPRelayEnabled
- [BYTE](#) \* pGPSLPM

### 8.115.1 Detailed Description

This structure contains current settings of custom features

#### Parameters

<i>GpsEnable</i>	<ul style="list-style-type: none"><li>• describes if GPS is enabled or disabled</li><li>• values:<ul style="list-style-type: none"><li>– 0x00 - GPS is disabled</li><li>– 0x01 - GPS is enabled</li></ul></li><li>• function <a href="#">SLQSGetCustFeatures()</a> returns a default value FFFFFFFF if no value is returned by the modem</li></ul>
------------------	--



<i>pDisableIMSI</i>	<ul style="list-style-type: none"> <li>• optional 1 byte parameter</li> <li>• describes if IMSI display is enabled or disabled</li> <li>• values: <ul style="list-style-type: none"> <li>– 0x00 - Allow display of IMSI</li> <li>– 0x01 - Do not display IMSI</li> </ul> </li> <li>• function <a href="#">SLQSGetCustFeatures()</a> returns a default value FF if no value is returned by the modem</li> </ul>
<i>pIPFamSupport</i>	<ul style="list-style-type: none"> <li>• optional 2 byte BitMask</li> <li>• bitmask representing the IP families supported</li> <li>• values: <ul style="list-style-type: none"> <li>– 0x01 - IPv4</li> <li>– 0x02 - IPv6</li> <li>– 0x04 - IPv4v6</li> </ul> </li> <li>• function <a href="#">SLQSGetCustFeatures()</a> returns a default value FFFF if no value is returned by the modem</li> </ul>
<i>pRMAuto-Connect</i>	<ul style="list-style-type: none"> <li>• optional 1 byte parameter</li> <li>• QMI Mode RM Net Auto Connect Support</li> <li>• values: <ul style="list-style-type: none"> <li>– 0x00 - Not Supported</li> <li>– 0x01 - Supported</li> </ul> </li> <li>• function <a href="#">SLQSGetCustFeatures()</a> returns a default value FF if no value is returned by the modem</li> </ul>
<i>pGPSSel</i>	<ul style="list-style-type: none"> <li>• optional 1 byte parameter</li> <li>• GPS Antenna Select</li> <li>• values: <ul style="list-style-type: none"> <li>– 0x00 - Dedicated GPS <a href="#">Port</a></li> <li>– 0x01 - GPS Rx over AUX <a href="#">Port</a></li> <li>– 0x02 - GPS Rx over dedicated GPS port with no bias voltage applied</li> </ul> </li> <li>• function <a href="#">SLQSGetCustFeatures()</a> returns a default value FF if no value is returned by the modem</li> </ul>

<i>pSMSSupport</i>	<ul style="list-style-type: none"> <li>• optional 1 byte parameter</li> <li>• SMS support</li> <li>• values: <ul style="list-style-type: none"> <li>– 0x00 - Not supported</li> <li>– 0x01 - supported</li> </ul> </li> <li>• Used to determine whether or not to hide SMS from user</li> <li>• function <a href="#">SLQSGetCustFeatures()</a> returns a default value FF if no value is returned by the modem. In this case assume, SMS is supported.</li> </ul>
<i>plsVoiceEnabled</i>	<ul style="list-style-type: none"> <li>• optional 1 byte parameter</li> <li>• Voice support</li> <li>• values: <ul style="list-style-type: none"> <li>– 0x00 - Enable voice on both AT and QMI interface (default)</li> <li>– 0x01 - Reserved</li> <li>– 0x02 - Disable voice on both AT and QMI interface</li> </ul> </li> </ul>
<i>pDHCPRelay-Enabled</i>	<ul style="list-style-type: none"> <li>• optional 1 byte parameter</li> <li>• DHCP Relay support</li> <li>• values: <ul style="list-style-type: none"> <li>– 0x00 - Disable DHCP relay</li> <li>– 0x01 - Enable DHCP relay</li> </ul> </li> </ul>
<i>pGPSLPM</i>	<ul style="list-style-type: none"> <li>• optional 1 byte parameter</li> <li>• GPSLPM support</li> <li>• values: <ul style="list-style-type: none"> <li>– 0x00 - Enable GPS in Low Power Mode</li> <li>– 0x01 - Disable GPS in Low Power Mode</li> </ul> </li> </ul>

## 8.115.2 Field Documentation

### 8.115.2.1 ULONG custFeaturesInfo::GpsEnable

### 8.115.2.2 BYTE\* custFeaturesInfo::pDHCPRelayEnabled

### 8.115.2.3 BYTE\* custFeaturesInfo::pDisableMSI

8.115.2.4 **BYTE\*** custFeaturesInfo::pGPSLPM8.115.2.5 **BYTE\*** custFeaturesInfo::pGPSSel8.115.2.6 **WORD\*** custFeaturesInfo::pIPFamSupport8.115.2.7 **BYTE\*** custFeaturesInfo::plsVoiceEnabled8.115.2.8 **BYTE\*** custFeaturesInfo::pRMAutoConnect8.115.2.9 **BYTE\*** custFeaturesInfo::pSMSSupport

## 8.116 custFeaturesSetting Struct Reference

### Data Fields

- **BYTE \*** pGPSSel
- **ULONG \*** pGPSEnable
- **BYTE \*** plsVoiceEnabled
- **BYTE \*** pDHCPRelayEnabled
- **BYTE \*** pGPSLPM

### 8.116.1 Detailed Description

This structure contains settings to be used for custom features

#### Parameters

<i>pGPSSel</i>	<ul style="list-style-type: none"> <li>• optional 1 byte parameter</li> <li>• GPS Antenna Select</li> <li>• values: <ul style="list-style-type: none"> <li>– 0x00 - Dedicated GPS <a href="#">Port</a></li> <li>– 0x01 - GPS Rx over AUX <a href="#">Port</a></li> <li>– 0x02 - GPS Rx over dedicated GPS port with no bias voltage applied</li> </ul> </li> </ul>
<i>pGPSEnable</i>	<ul style="list-style-type: none"> <li>• optional 4 byte parameter</li> <li>• GPS Enable/Disable</li> <li>• values: The value of 7 least significant bits: <ul style="list-style-type: none"> <li>– 0 - Disabled</li> <li>– 1 - MT &amp; MO enabled</li> <li>– 2 - MO enabled</li> <li>– 3 - MT enabled</li> <li>– 4 - MT &amp; MO enabled if GPS_DISABLE pin is not asserted</li> <li>– 5 - MO GPS enabled if GPS_DISABLE pin is not asserted</li> <li>– 6 - MT GPS enabled if GPS_DISABLE pin is not asserted</li> </ul> </li> </ul>

**Note**

Only MC7750 3.5.x firmware supports above 0x04, 0x05 and 0x06 settings. To disable GLONASS, set the most significant bit - 0x80. This setting is only applicable if GPS is not Disabled.

**Parameters**

<i>plsVoiceEnabled</i>	<ul style="list-style-type: none"> <li>• optional 1 byte parameter</li> <li>• voice enabled/disabled</li> <li>• values: <ul style="list-style-type: none"> <li>– 0 - Enable voice on both AT and QMI interface (default)</li> <li>– 1 - Reserved</li> <li>– 2 - Disable voice on both AT and QMI interface</li> </ul> </li> </ul>
<i>pDHCPRelay-Enabled</i>	<ul style="list-style-type: none"> <li>• optional 1 byte parameter</li> <li>• DHCPRELAYENABLE support</li> <li>• values: <ul style="list-style-type: none"> <li>– 0 - Disable DHCP relay</li> <li>– 1 - Enable DHCP relay</li> </ul> </li> </ul>
<i>pGPSLPM</i>	<ul style="list-style-type: none"> <li>• optional 1 byte parameter</li> <li>• GPSLPM support</li> <li>• values: <ul style="list-style-type: none"> <li>– 0 - Enable GPS in Low Power Mode</li> <li>– 1 - Disable GPS in Low Power Mode</li> </ul> </li> </ul>

**8.116.2 Field Documentation**

8.116.2.1 **BYTE\*** `custFeaturesSetting::pDHCPRelayEnabled`

8.116.2.2 **ULONG\*** `custFeaturesSetting::pGPSEnable`

8.116.2.3 **BYTE\*** `custFeaturesSetting::pGPSLPM`

8.116.2.4 **BYTE\*** `custFeaturesSetting::pGPSSel`

8.116.2.5 **BYTE\*** `custFeaturesSetting::plsVoiceEnabled`

**8.117 custSettingInfo Struct Reference****Data Fields**

- [WORD id\\_length](#)

- [CHAR cust\\_id](#) [64+1]
- [WORD value\\_length](#)
- [BYTE cust\\_value](#) [8+1]
- [WORD cust\\_attr](#)

### 8.117.1 Detailed Description

This structure contains information about Customization Setting. This TLV is only applicable for 9x30 modules so far

#### Parameters

<i>id_length</i>	<ul style="list-style-type: none"> <li>• length of cust_id field</li> </ul>
<i>cust_id</i>	<ul style="list-style-type: none"> <li>• Customization ID (Maximum 64 bytes)</li> </ul>
<i>value_length</i>	<ul style="list-style-type: none"> <li>• length of cust_value field</li> </ul>
<i>cust_value</i>	<ul style="list-style-type: none"> <li>• Customization Setting Value (Maximum 8 bytes)</li> </ul>
<i>cust_attr</i>	<ul style="list-style-type: none"> <li>• Customization Setting attribute through QMI             <ul style="list-style-type: none"> <li>– bit 0: Values:                 <ul style="list-style-type: none"> <li>* 0 - read only</li> <li>* 1 - read/write</li> </ul> </li> </ul> </li> </ul>

### 8.117.2 Field Documentation

8.117.2.1 **WORD** custSettingInfo::cust\_attr

8.117.2.2 **CHAR** custSettingInfo::cust\_id[64+1]

8.117.2.3 **BYTE** custSettingInfo::cust\_value[8+1]

8.117.2.4 **WORD** custSettingInfo::id\_length

8.117.2.5 **WORD** custSettingInfo::value\_length

## 8.118 custSettingList Struct Reference

### Data Fields

- [BYTE list\\_type](#)
- [WORD num\\_instances](#)
- [custSettingInfo custSetting](#) [256]

### 8.118.1 Detailed Description

This structure contains the fields of TLV Customization Setting List. This TLV is only applicable for 9x30 modules so far

## Parameters

<i>list_type</i>	<ul style="list-style-type: none"> <li>list type requested</li> </ul>
<i>num_instances</i>	<ul style="list-style-type: none"> <li>number of instances of customization setting</li> </ul>
<i>custSetting</i>	<ul style="list-style-type: none"> <li>See <a href="#">custSettingInfo</a> for more information</li> </ul>

## 8.118.2 Field Documentation

8.118.2.1 `custSettingInfo custSettingList::custSetting[256]`8.118.2.2 `BYTE custSettingList::list_type`8.118.2.3 `WORD custSettingList::num_instances`

## 8.119 dataBearers Struct Reference

## Data Fields

- [BYTE dataBearerMask](#)
- [QmiWDSDataBearerTechnology](#) \* [pCurDataBearerTechnology](#)
- [QmiWDSDataBearerTechnology](#) \* [pLastCallDataBearerTechnology](#)

## 8.119.1 Detailed Description

Structure to hold the data bearer technology values

## Parameters

<i>dataBearerMask[OUT]</i>	<ul style="list-style-type: none"> <li>This bit mask indicates if data bearer information for the current and/or last call has been received from the device. If a bit is set, then the information is available in the corresponding structure i.e. the one provided by the caller. Refer to <a href="#">qmiDataBearerMasks</a> for bit-mask positions.</li> </ul>
<i>pCurDataBearerTechnology[OUT]</i>	<ul style="list-style-type: none"> <li>current data bearer technology value. <ul style="list-style-type: none"> <li>– NULL if the parameter is not required</li> </ul> </li> </ul>
<i>pLastCallDataBearerTechnology[OUT]</i>	<ul style="list-style-type: none"> <li>last call data bearer technology value. <ul style="list-style-type: none"> <li>– NULL if the parameter is not required</li> </ul> </li> </ul>

## 8.119.2 Field Documentation

8.119.2.1 **BYTE** `dataBearers::dataBearerMask`

8.119.2.2 **QmiWSDDataBearerTechnology\*** `dataBearers::pCurDataBearerTechnology`

8.119.2.3 **QmiWSDDataBearerTechnology\*** `dataBearers::pLastCallDataBearerTechnology`

## 8.120 DataBearerTech Struct Reference

### Data Fields

- [ULONG](#) `techType`
- [ULONG](#) `ratValue`
- [ULONGLONG](#) `soMask`

### 8.120.1 Detailed Description

Network information structure



## Parameters

<i>TechType</i>	<ul style="list-style-type: none"> <li>• Technology type</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0 - WDS_BEARER_TECH_NETWORK_3GPP - 3GPP</li> <li>– 1 - WDS_BEARER_TECH_NETWORK_3GPP2 - 3GPP2</li> </ul> </li> </ul>
<i>ratValue</i>	<ul style="list-style-type: none"> <li>• Radio Access Technology (RAT) value</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0x00 - WDS_BEARER_TECH_RAT_EX_NULL_BEARER - NULL bearer</li> <li>– 0x01 - WDS_BEARER_TECH_RAT_EX_3GPP_WCDMA - 3GPP WCDMA</li> <li>– 0x02 - WDS_BEARER_TECH_RAT_EX_3GPP_GERAN - 3GPP GERAN</li> <li>– 0x03 - WDS_BEARER_TECH_RAT_EX_3GPP_LTE - 3GPP LTE</li> <li>– 0x04 - WDS_BEARER_TECH_RAT_EX_3GPP_TDSCDMA - 3GPP TDSCDMA</li> <li>– 0x05 - WDS_BEARER_TECH_RAT_EX_3GPP_WLAN - 3GPP WLAN</li> <li>– 0x64 - WDS_BEARER_TECH_RAT_EX_3GPP_MAX - 3GPP maximum</li> <li>– 0x65 - WDS_BEARER_TECH_RAT_EX_3GPP2_1X - 3GPP2 1X</li> <li>– 0x66 - WDS_BEARER_TECH_RAT_EX_3GPP2_HRPD - 3GPP2 HRPD</li> <li>– 0x67 - WDS_BEARER_TECH_RAT_EX_3GPP2_EHRPD - 3GPP2 EHRPD</li> <li>– 0x68 - WDS_BEARER_TECH_RAT_EX_3GPP2_WLAN - 3GPP2 WLAN</li> <li>– 0xC8 - WDS_BEARER_TECH_RAT_EX_3GPP2_MAX - 3GPP2 maximum</li> </ul> </li> </ul>
<i>SOMask</i>	<ul style="list-style-type: none"> <li>• Service Option (SO) mask to indicate the service option or type of application. An SO mask value of zero indicates that this field is ignored.</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0x00 - SO mask unspecified</li> </ul> </li> <li>• 3GPP SO mask: <ul style="list-style-type: none"> <li>– 0x01 - WCDMA</li> <li>– 0x02 - HSDPA</li> <li>– 0x04 - HSUPA</li> <li>– 0x08 - HSDPAPLUS</li> <li>– 0x10 - DC HSDPAPLUS</li> <li>– 0x20 - 64 QAM</li> <li>– 0x40 - HSPA</li> <li>– 0x80 - GPRS</li> <li>– 0x100 - EDGE</li> <li>– 0x200 - GSM</li> <li>– 0x400 - S2B</li> <li>– 0x800 - LTE limited service</li> <li>– 0x1000 - LTE FDD</li> <li>– 0x2000 - LTE TDD</li> </ul> </li> </ul>
	<p>Generated on Wed Oct 28 2015 00:40:42 for Linux OMI SDK by Doxygen</p> <ul style="list-style-type: none"> <li>• 3GPP2 SO mask: <ul style="list-style-type: none"> <li>– 0x01000000 - 1X IS95</li> <li>– 0x02000000 - 1X IS2000</li> </ul> </li> </ul>

### 8.120.2 Field Documentation

8.120.2.1 **ULONG** DataBearerTech::ratValue

8.120.2.2 **ULONGLONG** DataBearerTech::soMask

8.120.2.3 **ULONG** DataBearerTech::techType

## 8.121 DataBearerTechExt Struct Reference

### Data Fields

- [DataBearerTech](#) \* [pBearerTech](#)
- [DataBearerTech](#) \* [pLastBearerTech](#)

### 8.121.1 Detailed Description

Data Bearer Technology Ext

#### Parameters

<i>pBearerTech</i>	[OUT] <ul style="list-style-type: none"><li>• See <a href="#">DataBearerTech</a> for more information</li></ul>
<i>pLastBearerTech</i>	[OUT] <ul style="list-style-type: none"><li>• See <a href="#">DataBearerTech</a> for more information</li></ul>

### 8.121.2 Field Documentation

8.121.2.1 **DataBearerTech\*** DataBearerTechExt::pBearerTech

8.121.2.2 **DataBearerTech\*** DataBearerTechExt::pLastBearerTech

## 8.122 dataBearerTechnology Struct Reference

### Data Fields

- **BYTE** currentNetwork
- **ULONG** ratMask
- **ULONG** soMask

### 8.122.1 Detailed Description

Structure to hold the current data bearer technology values

## Parameters

<i>pCurrent-Network[OUT]</i>	<ul style="list-style-type: none"><li>• current selected network<ul style="list-style-type: none"><li>– 0 - UNKNOWN</li><li>– 1 - 3GPP2</li><li>– 2 - 3GPP</li></ul></li></ul>
<i>pRatMask[OUT]</i>	<ul style="list-style-type: none"><li>• Radio Access Technology (RAT) mask to indicate the type of technology (RAT mask value of zero indicates that this field is ignored)<ul style="list-style-type: none"><li>– 0x8000 - NULL Bearer</li><li>– 0x0000 - DO_NOT_CARE CDMA RAT mask</li><li>– 0x01 - CDMA_1X</li><li>– 0x02 - EVDO_REV0</li><li>– 0x04 - EVDO_REVA UMTS RAT mask</li><li>– 0x01 - WCDMA</li><li>– 0x02 - GPRS</li><li>– 0x04 - HSDPA</li><li>– 0x08 - HSUPA</li><li>– 0x10 - EDGE</li><li>– 0x20 - LTE</li><li>– 0x40 - HSDPA+</li><li>– 0x80 - DC_HSDPA+</li></ul></li></ul>

<i>pSoMask[OUT]</i>	<ul style="list-style-type: none"> <li>• Service Option (SO) mask to indicate the SO or type of application (SO mask value of zero indicates that this field is ignored) <ul style="list-style-type: none"> <li>– 0x00 - DO_NOT_CARE CDMA 1X SO mask</li> <li>– 0x01 - CDMA_1X_IS95</li> <li>– 0x02 - CDMA_1X_IS2000</li> <li>– 0x04 - CDMA_1X_IS2000_REL_A CDMA EV-DO Rev A SO mask</li> <li>– 0x01 - EVDO_REVA_DPA</li> <li>– 0x02 - EVDO_REVA_MFPA</li> <li>– 0x04 - EVDO_REVA_EMPA</li> <li>– 0x08 - EVDO_REVA_EMPA_EHRPD</li> </ul> </li> </ul>
---------------------	--

### 8.122.2 Field Documentation

8.122.2.1 **BYTE** `dataBearerTechnology::currentNetwork`

8.122.2.2 **ULONG** `dataBearerTechnology::ratMask`

8.122.2.3 **ULONG** `dataBearerTechnology::soMask`

## 8.123 dataRate Struct Reference

### Data Fields

- [ULONG dataRateMax](#)
- [ULONG guaranteedRate](#)

### 8.123.1 Detailed Description

This structure contains the IP flow data rate min max

#### Parameters

<i>dataRateMax</i>	Maximum required data rate (bits per second)
<i>guaranteedRate</i>	Minimum guaranteed data rate (bits per second)

### 8.123.2 Field Documentation

8.123.2.1 **ULONG** `dataRate::dataRateMax`

8.123.2.2 **ULONG** `dataRate::guaranteedRate`

## 8.124 dataSrvCapabilities Struct Reference

### Data Fields

- [BYTE dataCapabilitiesLen](#)
- [BYTE dataCapabilities](#) [0x20]

### 8.124.1 Detailed Description

This structure contains the data services capability

- Parameter values default to their data type's maximum unsigned value unless explicitly stated otherwise.

#### Parameters

<i>dataCapabilitiesLen</i>	<ul style="list-style-type: none"><li>• Length of data capabilities list</li><li>• Defaults to zero</li></ul>
<i>dataCapabilities</i>	<ul style="list-style-type: none"><li>• List of data capabilities</li><li>• Values:<ul style="list-style-type: none"><li>– 0x01 - GPRS</li><li>– 0x02 - EDGE</li><li>– 0x03 - HSDPA</li><li>– 0x04 - HSUPA</li><li>– 0x05 - WCDMA</li><li>– 0x06 - CDMA</li><li>– 0x07 - EV-DO Rev0</li><li>– 0x08 - EV-DO RevA</li><li>– 0x09 - GSM</li><li>– 0x0A - EV-DO Rev B</li><li>– 0x0B - LTE</li><li>– 0x0C - HSDPA+</li><li>– 0x0D - DC-HSDPA+</li></ul></li></ul>

### 8.124.2 Field Documentation

8.124.2.1 **BYTE** dataSrvCapabilities::dataCapabilities[0x20]

8.124.2.2 **BYTE** dataSrvCapabilities::dataCapabilitiesLen

## 8.125 DataStatusDetail Struct Reference

### Data Fields

- [ULONG](#) IPAddress
- [BYTE](#) LastErrCode

### 8.125.1 Detailed Description

This structure contains Data Status Details

## Parameters

<i>IPAddress</i>	<ul style="list-style-type: none"> <li>• IP Address</li> <li>• 0xABCDEFGH - AB.CD.EF.GH</li> <li>• Example: <ul style="list-style-type: none"> <li>– 0x12345678 - 18.52.86.120</li> <li>0x12=18 0x34=52 0x56=86 0x78=120</li> </ul> </li> <li>• 0xFFFFFFFF - NA</li> </ul>
<i>LastErrCode</i>	<ul style="list-style-type: none"> <li>• MIP Error code <ul style="list-style-type: none"> <li>– 0x00 - MIP_RRP_CODE_SUCCESS</li> <li>– 0x01 - MIP_RRP_CODE_SUCCESS_NO_SIM_BINDINGS</li> <li>– 0x40 - MIP_RRP_CODE_FAILURE_FA_REASON_UNSPECIFIED</li> <li>– 0x41 - MIP_RRP_CODE_FAILURE_FA_ADMIN_PROHIBITED</li> <li>– 0x42 - MIP_RRP_CODE_FAILURE_FA_INSUFFICIENT_RESOURCES</li> <li>– 0x43 - MIP_RRP_CODE_FAILURE_FA_MOBILE_NODE_FAILED_AUTH</li> <li>– 0x44 - MIP_RRP_CODE_FAILURE_FA_HA_FAILED_AUTH</li> <li>– 0x45 - MIP_RRP_CODE_FAILURE_FA_REQUESTED_LIFETIME_TOO_LONG</li> <li>– 0x46 - MIP_RRP_CODE_FAILURE_FA_MALFORMED_REQUEST</li> <li>– 0x47 - MIP_RRP_CODE_FAILURE_FA_MALFORMED_REPLY</li> <li>– 0x48 - MIP_RRP_CODE_FAILURE_FA_ENCAPSULATION_UNAVAILABLE</li> <li>– 0x49 - MIP_RRP_CODE_FAILURE_FA_VJHC_UNAVAILABLE</li> <li>– 0x4A - MIP_RRP_CODE_FAILURE_FA_CANT_REV_TUN</li> <li>– 0x4B - MIP_RRP_CODE_FAILURE_FA_MUST_REV_TUN</li> <li>– 0x4C - MIP_RRP_CODE_FAILURE_FA_BAD_TTL</li> <li>– 0x4D - MIP_RRP_CODE_FAILURE_INVALID_COA</li> <li>– 0x4F - MIP_RRP_CODE_FAILURE_FA_DELIVERY_STYLE_NOT_SUPPORTED</li> <li>– 0x59 - MIP_RRP_CODE_FAILURE_FA_VS_REASON</li> <li>– 0x61 - MIP_RRP_CODE_FAILURE_MISSING_NAI</li> <li>– 0x62 - MIP_RRP_CODE_FAILURE_MISSING_HA_ADDR</li> <li>– 0x63 - MIP_RRP_CODE_FAILURE_MISSING_HOMEADDR</li> <li>– 0x68 - MIP_RRP_CODE_FAILURE_UNKNOWN_CHALLENGE</li> <li>– 0x69 - MIP_RRP_CODE_FAILURE_MISSING_CHALLENGE</li> <li>– 0x6A - MIP_RRP_CODE_FAILURE_STALE_CHALLENGE</li> <li>– 0x6B - MIP_RRP_CODE_FAILURE_MISSING_MN_FA</li> <li>– 0x80 - MIP_RRP_CODE_FAILURE_HA_REASON_UNSPECIFIED</li> <li>– 0x81 - MIP_RRP_CODE_FAILURE_HA_ADMIN_PROHIBITED</li> <li>– 0x82 - MIP_RRP_CODE_FAILURE_HA_INSUFFICIENT_RESOURCES</li> <li>– 0x83 - MIP_RRP_CODE_FAILURE_HA_MOBILE_NODE_FAILED_AUTH</li> <li>– 0x84 - MIP_RRP_CODE_FAILURE_HA_FA_FAILED_AUTH</li> <li>– 0x85 - MIP_RRP_CODE_FAILURE_HA_REG_ID_MISMATCH</li> <li>– 0x86 - MIP_RRP_CODE_FAILURE_HA_MALFORMED_REQUEST</li> <li>– 0x88 - MIP_RRP_CODE_FAILURE_UNKNOWN_HA</li> <li>– 0x89 - MIP_RRP_CODE_FAILURE_HA_CANT_REV_TUN</li> <li>– 0x8A - MIP_RRP_CODE_FAILURE_HA_MUST_REV_TUN</li> <li>– 0x8B - MIP_RRP_CODE_FAILURE_HA_ENCAPSULATION_UNAVAILABLE</li> </ul> </li> </ul>

### 8.125.2 Field Documentation

8.125.2.1 **ULONG** DataStatusDetail::IPAddress

8.125.2.2 **BYTE** DataStatusDetail::LastErrCode

## 8.126 DataUlongLongTlv Struct Reference

### Data Fields

- [BYTE](#) TlvPresent
- [ULONGLONG](#) ullData

### 8.126.1 Field Documentation

8.126.1.1 **BYTE** DataUlongLongTlv::TlvPresent

8.126.1.2 **ULONGLONG** DataUlongLongTlv::ullData

## 8.127 DataUlongTlv Struct Reference

### Data Fields

- [BYTE](#) TlvPresent
- [ULONG](#) ulData

### 8.127.1 Field Documentation

8.127.1.1 **BYTE** DataUlongTlv::TlvPresent

8.127.1.2 **ULONG** DataUlongTlv::ulData

## 8.128 DcsUsbPortNames Struct Reference

### Data Fields

- [CHAR](#) AtCmdPort [32]
- [CHAR](#) NmeaPort [32]
- [CHAR](#) DmPort [32]

### 8.128.1 Field Documentation

8.128.1.1 **CHAR** DcsUsbPortNames::AtCmdPort[32]

8.128.1.2 **CHAR** DcsUsbPortNames::DmPort[32]

8.128.1.3 **CHAR** DcsUsbPortNames::NmeaPort[32]

## 8.129 delAssistDataStatus Struct Reference

## Data Fields

- [ULONG status](#)

### 8.129.1 Detailed Description

Contain the parameters passed for SetLocDeleteAssistDataCallback by the device.

#### Parameters

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

#### Note

None

### 8.129.2 Field Documentation

#### 8.129.2.1 ULONG delAssistDataStatus::status

## 8.130 depersonalizationInformation Struct Reference

### Data Fields

- [BYTE feature](#)
- [BYTE operation](#)
- [BYTE ckLen](#)
- [BYTE ckVal](#) [255]



### 8.130.1 Detailed Description

This structure contains the Depersonalization Information.

## Parameters

<i>feature</i>	<ul style="list-style-type: none"> <li>Indicates the personalization feature to de-activate or unblock. <ul style="list-style-type: none"> <li>0 - GW network personalization</li> <li>1 - GW network subset personalization</li> <li>2 - GW service provider personalization</li> <li>3 - GW corporate personalization</li> <li>4 - GW UIM personalization</li> <li>5 - 1X network type 1 personalization</li> <li>6 - 1X network type 2 personalization</li> <li>7 - 1X HRPD personalization</li> <li>8 - 1X service provider personalization</li> <li>9 - 1X corporate personalization</li> <li>10 - 1X RUIM personalization</li> </ul> </li> </ul>
<i>operation</i>	<ul style="list-style-type: none"> <li>Indicates the operation to perform. <ul style="list-style-type: none"> <li>0 - Deactivate personalization.</li> <li>1 - Unblock personalization.</li> </ul> </li> </ul>
<i>ckLen</i>	<ul style="list-style-type: none"> <li>Length of the following elements i.e. control key value.</li> </ul>
<i>ckVal</i> [MAX_DESCRIPTION_LENGTH]	<ul style="list-style-type: none"> <li>Control key value.</li> <li>This value is a sequence of ASCII characters.</li> </ul>

## 8.130.2 Field Documentation

8.130.2.1 BYTE depersonalizationInformation::ckLen

8.130.2.2 BYTE depersonalizationInformation::ckVal[255]

8.130.2.3 BYTE depersonalizationInformation::feature

8.130.2.4 BYTE depersonalizationInformation::operation

## 8.131 detailSvcInfo Struct Reference

## Data Fields

- [BYTE srvStatus](#)
- [BYTE srvCapability](#)
- [BYTE hdrSrvStatus](#)
- [BYTE hdrHybrid](#)
- [BYTE isSysForbidden](#)

### 8.131.1 Detailed Description

This structure contains Detailed Service information

- Parameter values default to their data type's maximum unsigned value unless explicitly stated otherwise.

#### Parameters

<i>srvStatus</i>	<ul style="list-style-type: none"><li>• Service status</li><li>• Values:<ul style="list-style-type: none"><li>– 0x00 - No service</li><li>– 0x01 - Limited service</li><li>– 0x02 - Service available</li><li>– 0x03 - Limited regional service</li><li>– 0x04 - MS in power save or deep sleep</li></ul></li></ul>
<i>srvCapability</i>	<ul style="list-style-type: none"><li>• System's service capability</li><li>• Values:<ul style="list-style-type: none"><li>– 0x00 - No Service</li><li>– 0x01 - Circuit-switched only</li><li>– 0x02 - Packet-switched only</li><li>– 0x03 - Circuit-switched and packet-switched</li><li>– 0x04 - MS found the right system but not yet registered/attached</li></ul></li></ul>
<i>hdrSrvStatus</i>	<ul style="list-style-type: none"><li>• HDR service status</li><li>• Values:<ul style="list-style-type: none"><li>– 0x00 - No service</li><li>– 0x01 - Limited service</li><li>– 0x02 - Service available</li><li>– 0x03 - Limited regional service</li><li>– 0x04 - MS in power save or deep sleep</li></ul></li></ul>

<i>hdrHybrid</i>	<ul style="list-style-type: none"> <li>• HDR hybrid information</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0x00 - System is not hybrid</li> <li>– 0x01 - System is hybrid</li> </ul> </li> </ul>
<i>isSysForbidden</i>	<ul style="list-style-type: none"> <li>• Forbidden system information</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0x00 - System is not a forbidden system</li> <li>– 0x01 - System is a forbidden system</li> </ul> </li> </ul>

## 8.131.2 Field Documentation

8.131.2.1 **BYTE** detailSvcInfo::hdrHybrid

8.131.2.2 **BYTE** detailSvcInfo::hdrSrvStatus

8.131.2.3 **BYTE** detailSvcInfo::isSysForbidden

8.131.2.4 **BYTE** detailSvcInfo::srvCapability

8.131.2.5 **BYTE** detailSvcInfo::srvStatus

## 8.132 DeviceConfigDetail Struct Reference

### Data Fields

- [BYTE Technology](#)
- [BYTE QLIC](#)
- [BYTE Chipset](#)
- [BYTE HWVersion](#)

### 8.132.1 Detailed Description

This structure contains Device Configuration Details

## Parameters

<i>Technology</i>	<ul style="list-style-type: none"> <li>• Current technology in use <ul style="list-style-type: none"> <li>– 0x00 - 1x RTT</li> <li>– 0x01 - EVDO Rev 0</li> <li>– 0x02 - EVDO Rev A</li> <li>– 0xFF - NA</li> </ul> </li> </ul>
<i>QLIC</i>	<ul style="list-style-type: none"> <li>• Quasi Linear Interference Cancellation <ul style="list-style-type: none"> <li>– 0x00 - Not supported</li> <li>– 0x01 - Supported</li> </ul> </li> </ul>
<i>Chipset</i>	<ul style="list-style-type: none"> <li>• Qualcomm platform <ul style="list-style-type: none"> <li>– 0x4E - MDM6200</li> <li>– 0x4F - MDM6600</li> <li>– 0xFF - NA</li> </ul> </li> </ul>
<i>HWVersion</i>	<ul style="list-style-type: none"> <li>• Hardware version <ul style="list-style-type: none"> <li>– 0x00 - BSHWREV0</li> <li>– 0x01 - BSHWREV1</li> <li>– 0x02 - BSHWREV2</li> <li>– 0x03 - BSHWREV3</li> <li>– 0x04 - BSHWREVMAX</li> <li>– 0xFF - BSHWREVUNKNOWN</li> </ul> </li> </ul>

## 8.132.2 Field Documentation

8.132.2.1 BYTE DeviceConfigDetail::Chipset

8.132.2.2 BYTE DeviceConfigDetail::HWVersion

8.132.2.3 BYTE DeviceConfigDetail::QLIC

8.132.2.4 BYTE DeviceConfigDetail::Technology

## 8.133 diagInfo Struct Reference

## Data Fields

- [BYTE diagInfoLen](#)
- [BYTE diagnosticInfo](#) [255]

### 8.133.1 Detailed Description

This structure contains Diagnostic Information

#### Parameters

<i>diagInfoLen</i>	<ul style="list-style-type: none"> <li>Provides the length of information which follow.</li> <li>If zero(0) then no further information exists.</li> </ul>
<i>diagnosticInfo[M-AX_DESCRIPTOR_LENGTH]</i>	<ul style="list-style-type: none"> <li>Diagnostic information.</li> </ul>

### 8.133.2 Field Documentation

8.133.2.1 **BYTE** diagInfo::diagInfoLen

8.133.2.2 **BYTE** diagInfo::diagnosticInfo[255]

## 8.134 dirNum Struct Reference

### Data Fields

- [BYTE](#) dirNumLen
- [BYTE](#) dirNum [255]

### 8.134.1 Detailed Description

This structure contains the parameters for Directory Number Information

#### Parameters

<i>dirNumLen</i>	<ul style="list-style-type: none"> <li>Number of sets of the following elements: <ul style="list-style-type: none"> <li>– dir_num</li> </ul> </li> <li>If zero(0), then no information follows.</li> </ul>
<i>dirNum</i>	<ul style="list-style-type: none"> <li>Directory number in ASCII characters.</li> </ul>

### 8.134.2 Field Documentation

8.134.2.1 **BYTE** dirNum::dirNum[255]

8.134.2.2 **BYTE** dirNum::dirNumLen

## 8.135 dmsCurrentPRLInfo Struct Reference

## Data Fields

- [WORD](#) \* [pPRLVersion](#)
- [BYTE](#) \* [pPRLPreference](#)

### 8.135.1 Detailed Description

This structure contains GetCurrentPRLInfo response parameter

#### Parameters

<i>pPRLVersion</i> [O-UT]	- Optional <ul style="list-style-type: none"><li>• PRL version of device.</li></ul>
<i>pPRLPreference</i>	[OUT]- Optional <ul style="list-style-type: none"><li>• PRL Preference<ul style="list-style-type: none"><li>– 0 - Unset</li><li>– 1 - Set</li></ul></li></ul>

### 8.135.2 Field Documentation

8.135.2.1 [BYTE](#)\* dmsCurrentPRLInfo::pPRLPreference

8.135.2.2 [WORD](#)\* dmsCurrentPRLInfo::pPRLVersion

## 8.136 Domain Struct Reference

## Data Fields

- [WORD](#) [domainLen](#)
- [CHAR](#) [domainName](#) [256]

### 8.136.1 Detailed Description

This structure contains the DomainName Information

#### Parameters

<i>domainLen</i>	<ul style="list-style-type: none"><li>• length of the received <a href="#">Domain</a> name</li></ul>
<i>domainName</i>	<ul style="list-style-type: none"><li>• <a href="#">Domain</a> name(Max 256 characters)</li></ul>

### 8.136.2 Field Documentation

8.136.2.1 [WORD](#) Domain::domainLen

8.136.2.2 [CHAR](#) Domain::domainName[256]

## 8.137 DomainNameList Struct Reference

### Data Fields

- [BYTE numInstances](#)
- struct [Domain domain](#) [10]

### 8.137.1 Detailed Description

This structure contains the [DomainNameList](#) Information

#### Parameters

<i>numInstances</i>	<ul style="list-style-type: none"> <li>• Number of <a href="#">Domain</a> name received</li> </ul>
<i>domain</i>	<ul style="list-style-type: none"> <li>• <a href="#">Domain</a> name information(Max 10 Domain names)</li> </ul>

### 8.137.2 Field Documentation

8.137.2.1 struct [Domain](#) [DomainNameList::domain](#)[10]

8.137.2.2 [BYTE](#) [DomainNameList::numInstances](#)

## 8.138 DRCPParams Struct Reference

### Data Fields

- [BYTE DRCValue](#)
- [BYTE DRCCover](#)

### 8.138.1 Detailed Description

This structure contains Data Rate Channel parameters

#### Parameters

<i>DRCValue</i>	<ul style="list-style-type: none"> <li>• Current Data Rate Channel</li> </ul>
<i>DRCCover</i>	<ul style="list-style-type: none"> <li>• Current Data Rate Channel cover</li> </ul>

### 8.138.2 Field Documentation

8.138.2.1 [BYTE](#) [DRCPParams::DRCCover](#)

8.138.2.2 [BYTE](#) [DRCPParams::DRCValue](#)



## 8.139 DTMFInfo Struct Reference

### Data Fields

- [BYTE callID](#)
- [BYTE DTMFEvent](#)
- [BYTE digitCnt](#)
- [BYTE digitBuff](#) [255]

### 8.139.1 Detailed Description

This structure contains information about the DTMF (Dual Tone Multi-Frequency).

#### Parameters

<i>callID</i>	<ul style="list-style-type: none"> <li>• Call identifier for the current call.</li> </ul>
<i>DTMFEvent</i>	<ul style="list-style-type: none"> <li>• DTMF event           <ul style="list-style-type: none"> <li>– 0x00 - DTMF_EVENT_REV_BURST - Sends a CDMA-burst DTMF</li> <li>– 0x01 - DTMF_EVENT_REV_START_CONT - Starts a continuous DTMF tone</li> <li>– 0x03 - DTMF_EVENT_REV_STOP_CONT - Stops a continuous DTMF tone</li> <li>– 0x05 - DTMF_EVENT_FWD_BURST - Received a CDMA-burst DTMF message</li> <li>– 0x06 - DTMF_EVENT_FWD_START_CONT - Received a start-continuous DTMF tone order</li> <li>– 0x07 - DTMF_EVENT_FWD_STOP_CONT - Received a stop-continuous DTMF tone order</li> </ul> </li> </ul>
<i>digitCnt</i>	<ul style="list-style-type: none"> <li>• Number of set of following element i.e. digitBuff.</li> </ul>
<i>digitBuff</i> [MAX_DESCRIPTION_LENGTH]	<ul style="list-style-type: none"> <li>• DTMF digit buffer in ASCII string which is NULL terminated</li> </ul>

### 8.139.2 Field Documentation

8.139.2.1 [BYTE DTMFInfo::callID](#)

8.139.2.2 [BYTE DTMFInfo::digitBuff](#)[255]

8.139.2.3 [BYTE DTMFInfo::digitCnt](#)

8.139.2.4 [BYTE DTMFInfo::DTMFEvent](#)

## 8.140 DTMFLengths Struct Reference

### Data Fields

- [BYTE DTMFPulseWidth](#)

- [BYTE DTMFInterdigitInterval](#)

### 8.140.1 Detailed Description

This structure contains Voice Burst DTMF pulse length information

Parameters

<i>DTMFPulse-Width</i>	<ul style="list-style-type: none"> <li>• DTMF pulse width. Values: <ul style="list-style-type: none"> <li>– 0x00 - DTMF_ONLENGTH_95MS - 95 ms</li> <li>– 0x01 - DTMF_ONLENGTH_150MS - 150 ms</li> <li>– 0x02 - DTMF_ONLENGTH_200MS - 200 ms</li> <li>– 0x03 - DTMF_ONLENGTH_250MS - 250 ms</li> <li>– 0x04 - DTMF_ONLENGTH_300MS - 300 ms</li> <li>– 0x05 - DTMF_ONLENGTH_350MS - 350 ms</li> <li>– 0x06 - DTMF_ONLENGTH_SMS SMS Tx special pulse width</li> </ul> </li> </ul>
<i>DTMFInterdigit-Interval</i>	<ul style="list-style-type: none"> <li>• DTMF interdigit interval Values: <ul style="list-style-type: none"> <li>– 0x00 - DTMF_OFFLENGTH_60MS - 60 ms</li> <li>– 0x01 - DTMF_OFFLENGTH_100MS - 100 ms</li> <li>– 0x02 - DTMF_OFFLENGTH_150MS - 150 ms</li> <li>– 0x03 - DTMF_OFFLENGTH_200MS - 200 ms</li> </ul> </li> </ul>

### 8.140.2 Field Documentation

8.140.2.1 [BYTE DTMFLengths::DTMFInterdigitInterval](#)

8.140.2.2 [BYTE DTMFLengths::DTMFPulseWidth](#)

## 8.141 DUNCallInfoInd Struct Reference

Data Fields

- [BYTE MdmConnStatus](#)
- [WORD CallEndReason](#)
- [ULONGLONG TXOKBytesCount](#)
- [ULONGLONG RXOKBytesCount](#)
- [BYTE DormancyStatus](#)
- [BYTE DataBearerTech](#)
- [channelRate ChannelRate](#)

### 8.141.1 Field Documentation

8.141.1.1 [WORD DUNCallInfoInd::CallEndReason](#)

8.141.1.2 **channelRate** DUNCallInfoInd::ChannelRate

8.141.1.3 **BYTE** DUNCallInfoInd::DataBearerTech

8.141.1.4 **BYTE** DUNCallInfoInd::DormancyStatus

8.141.1.5 **BYTE** DUNCallInfoInd::MdmConnStatus

8.141.1.6 **ULONGLONG** DUNCallInfoInd::RXOKBytesCount

8.141.1.7 **ULONGLONG** DUNCallInfoInd::TXOKBytesCount

## 8.142 ecioListElement Struct Reference

### Data Fields

- [SHORT](#) *ecio*
- [BYTE](#) *radiolf*

### 8.142.1 Detailed Description

This structure contains the ECIO Information

#### Parameters

<i>ecio</i>	<ul style="list-style-type: none"><li>• ECIO value in dBm</li></ul>
<i>radiolf</i>	<ul style="list-style-type: none"><li>• Radio interface technology of the signal being measured<ul style="list-style-type: none"><li>– 0x00 - RADIO_IF_NO_SVC - None (no service)</li><li>– 0x01 - RADIO_IF_CDMA_1X - cdma2000 1X</li><li>– 0x02 - RADIO_IF_CDMA_1XEVD0 - cdma2000 HRPD (1xEV-DO)</li><li>– 0x03 - RADIO_IF_AMPS - AMPS</li><li>– 0x04 - RADIO_IF_GSM - GSM</li><li>– 0x05 - RADIO_IF_UMTS - UMTS</li></ul></li></ul>

### 8.142.2 Field Documentation

8.142.2.1 **SHORT** *ecioListElement::ecio*

8.142.2.2 **BYTE** *ecioListElement::radiolf*

## 8.143 ECIOThresh Struct Reference

### Data Fields

- [BYTE](#) *ECIOThresListLen*
- [SHORT](#) \* *pECIOThresList*

### 8.143.1 Detailed Description

This structure contains ECIO threshold related parameters.

## Parameters

<i>ECIOThresListLen</i>	<ul style="list-style-type: none"> <li>Length of the ECIO threshold list parameter to follow</li> </ul>
<i>pECIOThresList</i>	<ul style="list-style-type: none"> <li>Sequence of thresholds delimiting ECIO event reporting bands</li> <li>Every time a new ECIO value crosses a threshold value, an event report indication message with the new ECIO value is sent to the requesting control point. For this field <ul style="list-style-type: none"> <li>Each ECIO threshold value is a signed 2 byte value</li> <li>Each ECIO threshold value increments in negative 0.5 dB, e.g., an ECIO threshold value of 2 means -1dB</li> <li>Maximum number of threshold values is 16</li> <li>At least one value must be specified</li> <li>Threshold values specified above are used for all RATs</li> </ul> </li> </ul>

## 8.143.2 Field Documentation

8.143.2.1 BYTE ECIOThresh::ECIOThresListLen

8.143.2.2 SHORT\* ECIOThresh::pECIOThresList

## 8.144 ECTNum Struct Reference

## Data Fields

- [BYTE ECTCallState](#)
- [BYTE presentationInd](#)
- [BYTE number](#) [81]

## 8.144.1 Detailed Description

Contains the parameters passed for Explicit Communication Transfer by the device.

## Parameters

<i>ECTCallState</i>	<ul style="list-style-type: none"> <li>ECT call state: <ul style="list-style-type: none"> <li>0x00 - ECT_CALL_STATE_NONE - None</li> <li>0x01 - ECT_CALL_STATE_ALERTING - Alerting</li> <li>0x02 - ECT_CALL_STATE_ACTIVE - Active</li> </ul> </li> </ul>
---------------------	--

<i>presentationInd</i>	<ul style="list-style-type: none"> <li>• Presentation indicator <ul style="list-style-type: none"> <li>– 0x00 - presentationAllowedAddress</li> <li>– 0x01 - presentationRestricted</li> <li>– 0x02 - numberNotAvailable</li> <li>– 0x04 - presentationRestrictedAddress</li> </ul> </li> </ul>
<i>number</i>	<ul style="list-style-type: none"> <li>• Number in ASCII characters terminated by NULL</li> </ul>

### 8.144.2 Field Documentation

8.144.2.1 **BYTE** ECTNum::ECTCallState

8.144.2.2 **BYTE** ECTNum::number[81]

8.144.2.3 **BYTE** ECTNum::presentationInd

## 8.145 encryptedPIN1 Struct Reference

### Data Fields

- [BYTE](#) pin1Len
- [BYTE](#) pin1Val [255]

### 8.145.1 Detailed Description

This structure contains the encrypted PIN1 Information.

#### Parameters

<i>pin1Len</i>	<ul style="list-style-type: none"> <li>• Number of sets of the following elements ie encrypted PIN1 value.</li> <li>• If zero(0), no information follows.</li> </ul>
<i>pin1Val</i>	<ul style="list-style-type: none"> <li>• Encrypted PIN1 value.</li> </ul>

#### Note

This value is returned only when PIN1 is enabled successfully and the feature is supported.

### 8.145.2 Field Documentation

8.145.2.1 **BYTE** encryptedPIN1::pin1Len

8.145.2.2 **BYTE** encryptedPIN1::pin1Val[255]

## 8.146 ERIFileparams Struct Reference

### Data Fields

- [WORD](#) \* [pFileSize](#)
- [BYTE](#) \* [pFile](#)

### 8.146.1 Detailed Description

This structure contains Extended Roaming Indicator(ERI) file parameters

#### Parameters

<i>pFileSize</i> [IN/OUT]	<ul style="list-style-type: none"><li>• Upon input, the maximum number of bytes that file contents array can contain.</li><li>• Upon successful output, actual number of bytes written to file contents array</li></ul>
<i>pFile</i> [OUT]	<ul style="list-style-type: none"><li>• ERI data read from persistent storage( Max size is 1024 )</li></ul>

### 8.146.2 Field Documentation

8.146.2.1 [BYTE](#)\* [ERIFileparams::pFile](#)

8.146.2.2 [WORD](#)\* [ERIFileparams::pFileSize](#)

## 8.147 errorRateListElement Struct Reference

### Data Fields

- [USHORT](#) [errorRate](#)
- [BYTE](#) [radiolf](#)

### 8.147.1 Detailed Description

This structure contains the Error Rate Information

## Parameters

<i>errorRate</i>	<ul style="list-style-type: none"> <li>• Error rate value corresponds to the RAT that is currently registered.             <ul style="list-style-type: none"> <li>– For CDMA, the error rate reported is Frame Error Rate:                 <ul style="list-style-type: none"> <li>* Valid error rate values between 1 and 10000 are returned to indicate percentage, e.g., a value of 300 means the error rate is 3%</li> <li>* A value of 0xFFFF indicates that the error rate is unknown or unavailable</li> </ul> </li> <li>– For HDR, the error rate reported is Packet Error Rate:                 <ul style="list-style-type: none"> <li>* Valid error rate values between 1 and 10000 are returned to indicate percentage, e.g., a value of 300 means the error rate is 3%</li> <li>* A value of 0xFFFF indicates that the error rate is unknown or unavailable</li> </ul> </li> <li>– For GSM, the error rate reported is Bit Error Rate:                 <ul style="list-style-type: none"> <li>* Valid values are 0, 100, 200, 300, 400, 500, 600, and 700 The reported value divided by 100 gives the error rate as an RxQual value, e.g., a value of 300 represents an RxQual value of 3.</li> <li>* A value of 25500 indicates No Data</li> </ul> </li> <li>– For WCDMA, the error rate reported is Block Error Rate (BLER):                 <ul style="list-style-type: none"> <li>* Valid values are 1 to 10000</li> <li>* The reported value divided by 100 provides the error rate in percentages, e.g., a value of 300 represents a BLER of 3%.</li> <li>* A value of 0 indicates No Data</li> </ul> </li> </ul> </li> </ul>
<i>radioIf</i>	<ul style="list-style-type: none"> <li>• Radio interface technology of the signal being measured             <ul style="list-style-type: none"> <li>– 0x00 - RADIO_IF_NO_SVC - None (no service)</li> <li>– 0x01 - RADIO_IF_CDMA_1X - cdma2000 1X</li> <li>– 0x02 - RADIO_IF_CDMA_1xEVDO - cdma2000 HRPD (1xEV-DO)</li> <li>– 0x03 - RADIO_IF_AMPS - AMPS</li> <li>– 0x04 - RADIO_IF_GSM - GSM</li> <li>– 0x05 - RADIO_IF_UMTS - UMTS</li> </ul> </li> </ul>



### 8.147.2 Field Documentation

8.147.2.1 USHORT errorRateListElement::errorRate

8.147.2.2 BYTE errorRateListElement::radiolf

## 8.148 extDispRecInfo Struct Reference

### Data Fields

- [BYTE dispType](#)
- [BYTE extDispInfoLen](#)
- [BYTE extDispInfo](#) [255]

### 8.148.1 Detailed Description

This structure contains Line Control Information

#### Parameters

<i>dispType</i>	<ul style="list-style-type: none"><li>• Values are per [S1, Table 3.7.5.16-1].</li></ul>
<i>extDispInfoLen</i>	<ul style="list-style-type: none"><li>• Number of sets of the following elements:<ul style="list-style-type: none"><li>– ext_display_info</li></ul></li></ul>
<i>extDispInfo</i>	<ul style="list-style-type: none"><li>• Extended display information buffer containing the display record; refer to [S1, Section 3.7.5.16] for the format information of the buffer contents.</li></ul>

### 8.148.2 Field Documentation

8.148.2.1 BYTE extDispRecInfo::dispType

8.148.2.2 BYTE extDispRecInfo::extDispInfo[255]

8.148.2.3 BYTE extDispRecInfo::extDispInfoLen

## 8.149 FactorySequenceNumber Struct Reference

### Data Fields

- [BYTE FSNumber](#) [255]

### 8.149.1 Detailed Description

This structure used to store Factory Sequence Number parameter

## Parameters

<i>FSNumber</i> [OUT]	<ul style="list-style-type: none"><li>• Factory Sequence Number</li><li>• Maximum Length is 255 Bytes</li></ul>
-----------------------	---

## 8.149.2 Field Documentation

8.149.2.1 BYTE FactorySequenceNumber::FSNumber[255]

## 8.150 fileAttributes Struct Reference

## Data Fields

- WORD fileSize
- WORD fileID
- BYTE fileType
- WORD recordSize
- WORD recordCount
- BYTE secRead
- WORD secReadMask
- BYTE secWrite
- WORD secWriteMask
- BYTE secIncrease
- WORD secIncreaseMask
- BYTE secDeactivate
- WORD secDeactivateMask
- BYTE secActivate
- WORD secActivateMask
- WORD rawLen
- BYTE rawValue [255]

## 8.150.1 Detailed Description

This structure contains the information about the File Attributes.

## Parameters

<i>fileSize</i>	<ul style="list-style-type: none"> <li>Indicates the size of the file.</li> </ul>
<i>fileID</i>	<ul style="list-style-type: none"> <li>Indicates the ID of the file.</li> </ul>
<i>fileType</i>	<ul style="list-style-type: none"> <li>Indicates the type of the file. <ul style="list-style-type: none"> <li>0 - Transparent</li> <li>1 - Cyclic</li> <li>2 - Linear fixed</li> <li>3 - Dedicated file</li> <li>4 - Master file</li> </ul> </li> </ul>
<i>recordSize</i>	<ul style="list-style-type: none"> <li>Indicates the size of the records.</li> <li>Only for cyclic and linear fixed files</li> </ul>
<i>recordCount</i>	<ul style="list-style-type: none"> <li>Indicates the total no. of the records.</li> <li>Only for linear fixed files</li> </ul>
<i>secRead</i>	<ul style="list-style-type: none"> <li>Read security attributes. <ul style="list-style-type: none"> <li>0 - Always</li> <li>1 - Never</li> <li>2 - AND condition</li> <li>3 - OR condition</li> <li>4 - Single condition</li> </ul> </li> </ul>
<i>secReadMask</i>	<ul style="list-style-type: none"> <li>Mask with read security attributes.</li> <li>This field is valid only when required by security attributes. <ul style="list-style-type: none"> <li>Bit 0 - PIN1</li> <li>Bit 1 - PIN2</li> <li>Bit 2 - UPIN</li> <li>Bit 3 - ADM</li> </ul> </li> </ul>

<i>secWrite</i>	<ul style="list-style-type: none"><li>• Write security attributes.<ul style="list-style-type: none"><li>– 0 - Always</li><li>– 1 - Never</li><li>– 2 - AND condition</li><li>– 3 - OR condition</li><li>– 4 - Single condition</li></ul></li></ul>
<i>secWriteMask</i>	<ul style="list-style-type: none"><li>• Mask with write security attributes.</li><li>• This field is valid only when required by security attributes.<ul style="list-style-type: none"><li>– Bit 0 - PIN1</li><li>– Bit 1 - PIN2</li><li>– Bit 2 - UPIN</li><li>– Bit 3 - ADM</li></ul></li></ul>
<i>secIncrease</i>	<ul style="list-style-type: none"><li>• Increase security attributes.<ul style="list-style-type: none"><li>– 0 - Always</li><li>– 1 - Never</li><li>– 2 - AND condition</li><li>– 3 - OR condition</li><li>– 4 - Single condition</li></ul></li></ul>
<i>secIncrease-Mask</i>	<ul style="list-style-type: none"><li>• Mask with increase security attributes.</li><li>• This field is valid only when required by security attributes.<ul style="list-style-type: none"><li>– Bit 0 - PIN1</li><li>– Bit 1 - PIN2</li><li>– Bit 2 - UPIN</li><li>– Bit 3 - ADM</li></ul></li></ul>

<i>secDeactivate</i>	<ul style="list-style-type: none"> <li>• Deactivate security attributes. <ul style="list-style-type: none"> <li>– 0 - Always</li> <li>– 1 - Never</li> <li>– 2 - AND condition</li> <li>– 3 - OR condition</li> <li>– 4 - Single condition</li> </ul> </li> </ul>
<i>secDeactivateMask</i>	<ul style="list-style-type: none"> <li>• Mask with deactivate security attributes.</li> <li>• This field is valid only when required by security attributes. <ul style="list-style-type: none"> <li>– Bit 0 - PIN1</li> <li>– Bit 1 - PIN2</li> <li>– Bit 2 - UPIN</li> <li>– Bit 3 - ADM</li> </ul> </li> </ul>
<i>secActivate</i>	<ul style="list-style-type: none"> <li>• Activate security attributes. <ul style="list-style-type: none"> <li>– 0 - Always</li> <li>– 1 - Never</li> <li>– 2 - AND condition</li> <li>– 3 - OR condition</li> <li>– 4 - Single condition</li> </ul> </li> </ul>
<i>secActivateMask</i>	<ul style="list-style-type: none"> <li>• Mask with activate security attributes.</li> <li>• This field is valid only when required by security attributes. <ul style="list-style-type: none"> <li>– Bit 0 - PIN1</li> <li>– Bit 1 - PIN2</li> <li>– Bit 2 - UPIN</li> <li>– Bit 3 - ADM</li> </ul> </li> </ul>

<i>rawLen</i>	<ul style="list-style-type: none"> <li>Length of the following elements i.e. raw value.</li> </ul>
<i>rawValue</i> [MAX_DESCRIPTION_LENGTH]	<ul style="list-style-type: none"> <li>Raw value of file attributes.</li> </ul>

## 8.150.2 Field Documentation

8.150.2.1 WORD fileAttributes::fileID

8.150.2.2 WORD fileAttributes::fileSize

8.150.2.3 BYTE fileAttributes::fileType

8.150.2.4 WORD fileAttributes::rawLen

8.150.2.5 BYTE fileAttributes::rawValue[255]

8.150.2.6 WORD fileAttributes::recordCount

8.150.2.7 WORD fileAttributes::recordSize

8.150.2.8 BYTE fileAttributes::secActivate

8.150.2.9 WORD fileAttributes::secActivateMask

8.150.2.10 BYTE fileAttributes::secDeactivate

8.150.2.11 WORD fileAttributes::secDeactivateMask

8.150.2.12 BYTE fileAttributes::secIncrease

8.150.2.13 WORD fileAttributes::secIncreaseMask

8.150.2.14 BYTE fileAttributes::secRead

8.150.2.15 WORD fileAttributes::secReadMask

8.150.2.16 BYTE fileAttributes::secWrite

8.150.2.17 WORD fileAttributes::secWriteMask

## 8.151 fileInfo Struct Reference

### Data Fields

- [WORD fileID](#)
- [BYTE pathLen](#)
- [WORD path](#) [255]

### 8.151.1 Detailed Description

This structure contains paramaters for file Information



## Parameters

<i>fileID</i>	<ul style="list-style-type: none"><li>• This is Identifier to SIM files; e.g. in UIM "6F07" is Identifier of IMSI File</li></ul>
<i>pathLen</i>	<ul style="list-style-type: none"><li>• Length of file Path</li></ul>
<i>path</i>	<ul style="list-style-type: none"><li>• Path value. This value must be the complete path of the file, which is a sequence block of 2 bytes (e.g., 0x3F00 0x7FFF).</li></ul>

## 8.151.2 Field Documentation

8.151.2.1 WORD fileInfo::fileID

8.151.2.2 WORD fileInfo::path[255]

8.151.2.3 BYTE fileInfo::pathLen

## 8.152 FirmwareUpdatStat Struct Reference

## Data Fields

- [ULONG ResCode](#)
- [BYTE \\* pImgType](#)
- [ULONG \\* pRefData](#)
- [BYTE \\* pRefStringLen](#)
- [BYTE \\* pRefString](#)

## 8.152.1 Detailed Description

This structure is used to store Firmware Update Status

## Parameters

<i>ResCode</i> [OUT]	<ul style="list-style-type: none"> <li>FW Update Result Code</li> <li>Values: <ul style="list-style-type: none"> <li>0x00000001 - Successful</li> <li>0xFFFFFFFF - Unknown (due to power off reset after firmware update )</li> <li>0x100000nn - File update errors while nn will be the exact error number: <ul style="list-style-type: none"> <li>* 00 - General error</li> </ul> </li> <li>0x200000nn - NVUP update errors while nn will be the exact error number: <ul style="list-style-type: none"> <li>* 00 - General error</li> </ul> </li> <li>0x400000nnn - FOTA update agent errors while nnn will be the exact error number: <ul style="list-style-type: none"> <li>* 000 ~ 0FF - Insignia defined error code</li> <li>* 100 ~ 1FF - Sierra defined error code</li> <li>* See qaGobiApiTableFwDldErrorCodes.h for more detailed information</li> </ul> </li> <li>0x800000nn - FDT/SSDP reported errors while nn will be the exact error number <ul style="list-style-type: none"> <li>* See qaGobiApiTableFwDldErrorCodes.h for more detailed information</li> </ul> </li> </ul> </li> </ul>
<i>plmgType</i> [OUT]	<ul style="list-style-type: none"> <li>Firmware image type that failed the update</li> </ul>
<i>pRefData</i> [OUT]	<ul style="list-style-type: none"> <li>Failed image reference data</li> <li>This is normally the offset of the image that caused the failure</li> </ul>
<i>pRefStringLen</i> [I- N/OUT]	<ul style="list-style-type: none"> <li>Length of Reference String parameter to follow</li> <li>As input parameter specifies length assigned to pRefString parameter</li> <li>As output parameter specifies length of actual value retrieved from the device</li> </ul>
<i>pRefString</i> [OUT]	<ul style="list-style-type: none"> <li>Failed image reference string. This is normally the partition name of the image that caused the failure if applicable.</li> </ul>

## 8.152.2 Field Documentation

8.152.2.1 BYTE\* FirmwareUpdatStat::plmgType

8.152.2.2 ULONG\* FirmwareUpdatStat::pRefData

8.152.2.3 BYTE\* FirmwareUpdatStat::pRefString

8.152.2.4 BYTE\* FirmwareUpdatStat::pRefStringLen

8.152.2.5 ULONG FirmwareUpdatStat::ResCode

## 8.153 fwinfo\_s Struct Reference

### Data Fields

- [ULONG FirmwareID](#)
- [ULONG Technology](#)
- [ULONG Carrier](#)
- [ULONG Region](#)
- [ULONG GPSCapability](#)

### 8.153.1 Detailed Description

Gobi firmware image info structure

#### Parameters

<i>FirmwareID</i>	<ul style="list-style-type: none"><li>• Firmware ID obtained from the firmware image</li></ul>
<i>Technology</i>	<ul style="list-style-type: none"><li>• Technology (0xFFFFFFFF if unknown)</li></ul>
<i>Carrier</i>	<ul style="list-style-type: none"><li>• Carrier (0xFFFFFFFF if unknown)</li></ul>
<i>Region</i>	<ul style="list-style-type: none"><li>• Region (0xFFFFFFFF if unknown)</li></ul>
<i>GPSCapability</i>	<ul style="list-style-type: none"><li>• GPS capability (0xFFFFFFFF if unknown)</li></ul>

### 8.153.2 Field Documentation

8.153.2.1 **ULONG** fwinfo\_s::Carrier

8.153.2.2 **ULONG** fwinfo\_s::FirmwareID

8.153.2.3 **ULONG** fwinfo\_s::GPSCapability

8.153.2.4 **ULONG** fwinfo\_s::Region

8.153.2.5 **ULONG** fwinfo\_s::Technology

## 8.154 GERANInfo Struct Reference

### Data Fields

- [ULONG cellID](#)
- [BYTE plmn](#) [3]
- [WORD lac](#)
- [WORD arfcn](#)
- [BYTE bsic](#)

- [ULONG timingAdvance](#)
- [WORD rxLev](#)
- [BYTE nmrInst](#)
- [nmrCellInfo](#) [insNmrCellInfo](#) [255]

### 8.154.1 Detailed Description

This structure contains information about the GERAN Network.

#### Parameters

<i>cellID</i>	<ul style="list-style-type: none"> <li>• Cell ID.</li> <li>• 0xFFFFFFFF indicates cell ID information is not present.</li> </ul>
<i>plmn[PLMN_LENGTH]</i>	<ul style="list-style-type: none"> <li>• MCC/MNC information coded as octet 3, 4, and 5.</li> <li>• This field is ignored when nmrCellID is not present.</li> </ul>
<i>lac</i>	<ul style="list-style-type: none"> <li>• Location area code.</li> <li>• This field is ignored when nmrCellID is not present. <ul style="list-style-type: none"> <li>– 0xFFFF - Not Available</li> </ul> </li> </ul>
<i>arfcn</i>	<ul style="list-style-type: none"> <li>• Absolute RF channel number. <ul style="list-style-type: none"> <li>– 0xFFFF - Not Available</li> </ul> </li> </ul>
<i>bsic</i>	<ul style="list-style-type: none"> <li>• Base station identity code. <ul style="list-style-type: none"> <li>– 0xFF - Not Available</li> </ul> </li> </ul>
<i>timingAdvance</i>	<ul style="list-style-type: none"> <li>• Measured delay (in bit periods; 1 bit period = 48/13 microsecond) of access burst transmission on RACH or PRACH to the expected signal from an MS at zero distance under static channel conditions. <ul style="list-style-type: none"> <li>– 0xFFFFFFFF - Not Available</li> </ul> </li> </ul>

<i>rxLev</i>	<ul style="list-style-type: none"> <li>• Serving Cell Rx measurement.</li> <li>• Values range between 0 and 63.</li> <li>• Mapped to a measured signal level: <ul style="list-style-type: none"> <li>– Rxlev 0 is a signal strength less than -110 dBm</li> <li>– Rxlev 1 is -110 dBm to -109 dBm</li> <li>– Rxlev 2 is -109 dBm to -108 dBm</li> <li>– ...</li> <li>– Rxlev 62 is -49 dBm to -48 dBm</li> <li>– Rxlev 63 is greater than -48 dBm</li> <li>– 0xFFFF - Not Available</li> </ul> </li> </ul>
<i>nmrInst</i>	<ul style="list-style-type: none"> <li>• Provides the number of set of instances which follow.</li> <li>• If 0(zero), then no information follows it.</li> </ul>
<i>insNmrCellInfo[ MAX_DESCRIPTION_LENGTH]</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nmrCellInfo</a> for more information.</li> </ul>

## 8.154.2 Field Documentation

8.154.2.1 WORD GERANInfo::arfcn

8.154.2.2 BYTE GERANInfo::bsic

8.154.2.3 ULONG GERANInfo::cellID

8.154.2.4 nmrCellInfo GERANInfo::insNmrCellInfo[255]

8.154.2.5 WORD GERANInfo::lac

8.154.2.6 BYTE GERANInfo::nmrInst

8.154.2.7 BYTE GERANInfo::plmn[3]

8.154.2.8 WORD GERANInfo::rxLev

8.154.2.9 ULONG GERANInfo::timingAdvance

## 8.155 geranInstInfo Struct Reference

### Data Fields

- [WORD geranArfcn](#)
- [BYTE geranBsicNcc](#)
- [BYTE geranBsicBcc](#)
- [SHORT geranRssi](#)

### 8.155.1 Detailed Description

This structure contains information about the GERAN Instances in UMTS Network.

## Parameters

<i>geranArfcn</i>	<ul style="list-style-type: none"> <li>• Absolute RF channel number.</li> </ul>
<i>geranBsicNcc</i>	<ul style="list-style-type: none"> <li>• Base station identity code network color code.</li> <li>• 0xFF indicates information is not present.</li> </ul>
<i>geranBsicBcc</i>	<ul style="list-style-type: none"> <li>• Base station identity code base station color code.</li> <li>• 0xFF indicates information is not present.</li> </ul>
<i>geranRssi</i>	<ul style="list-style-type: none"> <li>• Received signal strength indicator.</li> </ul>

## 8.155.2 Field Documentation

8.155.2.1 WORD `geranInstInfo::geranArfcn`8.155.2.2 BYTE `geranInstInfo::geranBsicBcc`8.155.2.3 BYTE `geranInstInfo::geranBsicNcc`8.155.2.4 SHORT `geranInstInfo::geranRssi`

## 8.156 getAllCallInformation Struct Reference

## Data Fields

- [callInfo](#) `Callinfo`
- [BYTE](#) `isEmpty`
- [BYTE](#) `ALS`

## 8.156.1 Detailed Description

This structure contains information related to call state change.

## Parameters

<i>Callinfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">callInfo</a> for more information.</li> </ul>
<i>isEmpty</i>	<ul style="list-style-type: none"> <li>• Multiparty indicator. <ul style="list-style-type: none"> <li>– 0x00 - False</li> <li>– 0x01 - True</li> </ul> </li> </ul>

<i>ALS</i>	<ul style="list-style-type: none"> <li>• Alternate Line Service line indicator.</li> <li>• Feature for supporting two different phone numbers on the same mobile device. <ul style="list-style-type: none"> <li>– 0x00 - ALS_LINE1 - Line 1 (default)</li> <li>– 0x01 - ALS_LINE2 - Line 2</li> </ul> </li> </ul>
------------	---

## 8.156.2 Field Documentation

8.156.2.1 **BYTE** `getAllCallInformation::ALS`

8.156.2.2 **callInfo** `getAllCallInformation::Callinfo`

8.156.2.3 **BYTE** `getAllCallInformation::isEmpty`

## 8.157 `getAllCallRmtPtyName` Struct Reference

### Data Fields

- [BYTE](#) `callID`
- [remotePartyName](#) `RemotePartyName`

### 8.157.1 Detailed Description

This structure contains information for All Call Remote Party Names

#### Parameters

<i>callID</i>	<ul style="list-style-type: none"> <li>• Unique call identifier for the call.</li> </ul>
<i>RemoteParty-Name</i>	<ul style="list-style-type: none"> <li>• See <a href="#">remotePartyName</a> for more information.</li> </ul>

## 8.157.2 Field Documentation

8.157.2.1 **BYTE** `getAllCallRmtPtyName::callID`

8.157.2.2 **remotePartyName** `getAllCallRmtPtyName::RemotePartyName`

## 8.158 `getAllCallRmtPtyNum` Struct Reference

### Data Fields

- [BYTE](#) `callID`
- [remotePartyNum](#) `RemotePartyNum`



### 8.158.1 Detailed Description

This structure contains information for All Call Remote Party Numbers

## Parameters

<i>callID</i>	<ul style="list-style-type: none"><li>• Unique call identifier for the call.</li></ul>
<i>RemoteParty-Num</i>	<ul style="list-style-type: none"><li>• See <a href="#">remotePartyNum</a> for more information.</li></ul>

## 8.158.2 Field Documentation

### 8.158.2.1 BYTE getAllCallRmtPtyNum::callID

### 8.158.2.2 remotePartyNum getAllCallRmtPtyNum::RemotePartyNum

## 8.159 GetAudioPathConfigReq Struct Reference

## Data Fields

- [BYTE Profile](#)
- [BYTE Item](#)

### 8.159.1 Detailed Description

This structure contains the SLQSGetAudioPathConfig request parameters

## Parameters

<i>Profile</i>	<ul style="list-style-type: none"><li>• Audio Profile<ul style="list-style-type: none"><li>– 0-9</li></ul></li></ul>
<i>Item</i>	<ul style="list-style-type: none"><li>• Item<ul style="list-style-type: none"><li>– 0 - AV_EC</li><li>– 1 - AV_NS</li><li>– 2 - AV_TXVOL</li><li>– 3 - AV_DTMFTXG</li><li>– 4 - AV_CODECSTG</li><li>– 5 - AV_TXPCMIIRFLTR</li><li>– 6 - AV_RXPCMIIRFLTR</li><li>– 7 - AV_MICGAIN</li><li>– 8 - AV_RXAGC</li><li>– 9 - AV_TXAGC</li><li>– 10 - AV_RXAGCLIST</li><li>– 11 - AV_RXAVCLIST</li><li>– 12 - AV_TXAGCLIST</li></ul></li></ul>

### 8.159.2 Field Documentation

8.159.2.1 **BYTE** GetAudioPathConfigReq::Item

8.159.2.2 **BYTE** GetAudioPathConfigReq::Profile

## 8.160 GetAudioPathConfigResp Struct Reference

### Data Fields

- **BYTE** \* pECMode
- **BYTE** \* pNSEnable
- **WORD** \* pTXGain
- **WORD** \* pDTMFTXGain
- **WORD** \* pCodecSTGain
- **TXPCMIIRFiltr** \* pTXPCMIIRFiltr
- **RXPCMIIRFiltr** \* pRXPCMIIRFiltr
- **BYTE** \* pMICGainSelect
- **BYTE** \* pRXAVCAGCSwitch
- **BYTE** \* pTXAVCSwitch
- **RXAGCList** \* pRXAGCList
- **RXAVCList** \* pRXAVCList
- **TXAGCList** \* pTXAGCList

### 8.160.1 Detailed Description

This structure contains the SLQSGetAudioPathConfig response parameters.

#### Parameters

<i>pECMode</i>	[Optional] <ul style="list-style-type: none"><li>• AV_EC<ul style="list-style-type: none"><li>– 0 - Echo cancellation off</li><li>– 1 - Handset mode</li><li>– 2 - Headset mode</li><li>– 3 - Car kit mode</li><li>– 4 - Speaker Mode</li></ul></li></ul>
----------------	---

<i>pNSEnable</i>	[Optional] <ul style="list-style-type: none"> <li>• AV_NS             <ul style="list-style-type: none"> <li>– 0 - Noise suppression off</li> <li>– 1 - Noise suppression on</li> </ul> </li> </ul>
<i>pTXGain</i>	[Optional] <ul style="list-style-type: none"> <li>• AV_TXVOL             <ul style="list-style-type: none"> <li>– 0x0000 - 0xffff</li> </ul> </li> </ul>
<i>pDTMFTXGain</i>	[Optional] <ul style="list-style-type: none"> <li>• AV_DTMFTXG             <ul style="list-style-type: none"> <li>– 0x0000 - 0xffff</li> </ul> </li> </ul>
<i>pCodecSTGain</i>	[Optional] <ul style="list-style-type: none"> <li>• AV_CODECSTG             <ul style="list-style-type: none"> <li>– 0x0000 - 0xffff</li> </ul> </li> </ul>
<i>pTXPCMIIRFtr</i>	[Optional] <ul style="list-style-type: none"> <li>• See <a href="#">TXPCMIIRFtr</a> for more information</li> </ul>
<i>pRXPCMIIRFtr</i>	[Optional] <ul style="list-style-type: none"> <li>• See <a href="#">RXPCMIIRFtr</a> for more information</li> </ul>
<i>pMICGainSelect</i>	[Optional] <ul style="list-style-type: none"> <li>• AV_MICGAIN</li> </ul>
<i>pRXAVCAGC-Switch</i>	[Optional] <ul style="list-style-type: none"> <li>• RX AVC/AGC Switch</li> </ul>
<i>pTXAVCSwitch</i>	[Optional] <ul style="list-style-type: none"> <li>• TX AVC Switch</li> </ul>
<i>pRXAGCList</i>	[Optional] <ul style="list-style-type: none"> <li>• See <a href="#">RXAGCList</a> for more information</li> </ul>
<i>pRXAVCList</i>	[Optional] <ul style="list-style-type: none"> <li>• See <a href="#">RXAVCList</a> for more information</li> </ul>
<i>pTXAGCList</i>	[Optional] <ul style="list-style-type: none"> <li>• See <a href="#">TXAGCList</a> for more information</li> </ul>

## 8.160.2 Field Documentation

### 8.160.2.1 WORD\* GetAudioPathConfigResp::pCodecSTGain

- 8.160.2.2 **WORD\*** GetAudioPathConfigResp::pDTMFTXGain
- 8.160.2.3 **BYTE\*** GetAudioPathConfigResp::pECMode
- 8.160.2.4 **BYTE\*** GetAudioPathConfigResp::pMICGainSelect
- 8.160.2.5 **BYTE\*** GetAudioPathConfigResp::pNSEnable
- 8.160.2.6 **RXAGCList\*** GetAudioPathConfigResp::pRXAGCList
- 8.160.2.7 **BYTE\*** GetAudioPathConfigResp::pRXAVCAGCSwitch
- 8.160.2.8 **RXAVCList\*** GetAudioPathConfigResp::pRXAVCList
- 8.160.2.9 **RXPCMIIRFitr\*** GetAudioPathConfigResp::pRXPCMIIRFitr
- 8.160.2.10 **TXAGCList\*** GetAudioPathConfigResp::pTXAGCList
- 8.160.2.11 **BYTE\*** GetAudioPathConfigResp::pTXAVCSwitch
- 8.160.2.12 **WORD\*** GetAudioPathConfigResp::pTXGain
- 8.160.2.13 **TXPCMIIRFitr\*** GetAudioPathConfigResp::pTXPCMIIRFitr

## 8.161 GetAudioProfileReq Struct Reference

### Data Fields

- [BYTE Generator](#)

### 8.161.1 Detailed Description

This structure contains the SLQSGetAudioProfile request parameters

#### Parameters

<i>Generator</i>	<ul style="list-style-type: none"><li>• Audio Generator<ul style="list-style-type: none"><li>– 0 - Voice</li><li>– 1 - Key Beep</li><li>– 2 - MIDI</li></ul></li></ul>
------------------	--

### 8.161.2 Field Documentation

- 8.161.2.1 **BYTE** GetAudioProfileReq::Generator

## 8.162 GetAudioProfileResp Struct Reference

### Data Fields

- [BYTE Profile](#)

- [BYTE EarMute](#)
- [BYTE MicMute](#)
- [BYTE Volume](#)

### 8.162.1 Detailed Description

This structure contains the SLQSGetAudioProfile response parameters.

Parameters

<i>Profile</i>	<ul style="list-style-type: none"> <li>• Audio Profile           <ul style="list-style-type: none"> <li>– 0 - Handset</li> <li>– 1 - Headset</li> <li>– 2 - Car Kit</li> <li>– 3 - Speaker phone</li> <li>– 4 - Auxiliary</li> <li>– 5 - TTY</li> <li>– 6 - Auxiliary external PCM</li> <li>– 7 - Primary external PCM</li> <li>– 8 - External slave PCM</li> <li>– 9 - I2S</li> </ul> </li> </ul>
<i>EarMute</i>	<ul style="list-style-type: none"> <li>• Ear Mute Setting           <ul style="list-style-type: none"> <li>– 0 - unmuted</li> <li>– 1 - muted</li> </ul> </li> </ul>
<i>MicMute</i>	<ul style="list-style-type: none"> <li>• MIC Mute Setting           <ul style="list-style-type: none"> <li>– 0 - unmuted</li> <li>– 1 - muted</li> </ul> </li> </ul>
<i>Volume</i>	<ul style="list-style-type: none"> <li>• Audio Volume Level           <ul style="list-style-type: none"> <li>– 0 to 7</li> </ul> </li> </ul>

### 8.162.2 Field Documentation

8.162.2.1 **BYTE** GetAudioProfileResp::EarMute

8.162.2.2 **BYTE** GetAudioProfileResp::MicMute

8.162.2.3 **BYTE** GetAudioProfileResp::Profile

## 8.162.2.4 BYTE GetAudioProfileResp::Volume

## 8.163 GetAudioVolTLBConfigReq Struct Reference

## Data Fields

- [BYTE Profile](#)
- [BYTE Generator](#)
- [BYTE Volume](#)
- [BYTE Item](#)

## 8.163.1 Detailed Description

This structure contains the SLQSGetAudioVolTLBConfig request parameters

## Parameters

<i>Profile</i>	<ul style="list-style-type: none"><li>• Audio Profile<ul style="list-style-type: none"><li>– 0-9</li></ul></li></ul>
<i>Generator</i>	<ul style="list-style-type: none"><li>• Audio Generator<ul style="list-style-type: none"><li>– 0-2</li></ul></li></ul>
<i>Volume</i>	<ul style="list-style-type: none"><li>• Audio Volume Level<ul style="list-style-type: none"><li>– 0-7</li></ul></li></ul>
<i>Item</i>	<ul style="list-style-type: none"><li>• Item<ul style="list-style-type: none"><li>– 13 - AV_RXVOLDB</li><li>– 14 - AV_DTMFVOLDB</li><li>– 15 - AV_PAD</li></ul></li></ul>

## 8.163.2 Field Documentation

## 8.163.2.1 BYTE GetAudioVolTLBConfigReq::Generator

## 8.163.2.2 BYTE GetAudioVolTLBConfigReq::Item

## 8.163.2.3 BYTE GetAudioVolTLBConfigReq::Profile

## 8.163.2.4 BYTE GetAudioVolTLBConfigReq::Volume

## 8.164 GetAudioVolTLBConfigResp Struct Reference

## Data Fields

- [WORD ResCode](#)

### 8.164.1 Detailed Description

This structure contains the SLQSGetAudioVoITLBConfig response parameters.

#### Parameters

<i>ResCode</i>	<ul style="list-style-type: none"> <li>• Result of requested item</li> </ul>
----------------	--

### 8.164.2 Field Documentation

#### 8.164.2.1 WORD GetAudioVoITLBConfigResp::ResCode

## 8.165 getCallFWExtInfo Struct Reference

## Data Fields

- [BYTE numInstances](#)
- [callFWExtInfo CallFWExtInfo](#) [20]

### 8.165.1 Detailed Description

This structure contains an array of Call Forwarded Extended Information.

#### Parameters

<i>numInstances</i>	<ul style="list-style-type: none"> <li>• Number of <a href="#">callFWExtInfo</a> that follow.</li> <li>• If zero(0) then no further information exists.</li> </ul>
<i>CallFWExtInfo[-MAX_NO_OF_CALLS]</i>	<ul style="list-style-type: none"> <li>• Array of CallFWExtInfo. <ul style="list-style-type: none"> <li>– See <a href="#">CallFWExtInfo</a> for more information.</li> </ul> </li> </ul>

### 8.165.2 Field Documentation

#### 8.165.2.1 callFWExtInfo getCallFWExtInfo::CallFWExtInfo[20]

#### 8.165.2.2 BYTE getCallFWExtInfo::numInstances

## 8.166 getCallFWInfo Struct Reference

## Data Fields

- [BYTE numInstances](#)



- [callFWInfo](#) [CallFWInfo](#) [20]

### 8.166.1 Detailed Description

This structure contains an array of Call Forwarded Information.

#### Parameters

<i>numInstances</i>	<ul style="list-style-type: none"> <li>• Number of <a href="#">callFWInfo</a> that follow.</li> <li>• If zero(0) then no further information exists.</li> </ul>
<i>CallFWInfo[MA-X_NO_OF_CALLS]</i>	<ul style="list-style-type: none"> <li>• Array of <a href="#">callFWInfo</a>. <ul style="list-style-type: none"> <li>– See <a href="#">callFWInfo</a> for more information.</li> </ul> </li> </ul>

### 8.166.2 Field Documentation

8.166.2.1 [callFWInfo](#) [getCallFWInfo::CallFWInfo](#)[20]

8.166.2.2 [BYTE](#) [getCallFWInfo::numInstances](#)

## 8.167 getCustomFeatureV2 Struct Reference

### Data Fields

- [getCustomInput](#) \* [pGetCustomInput](#)
- [custSettingInfo](#) \* [pCustSettingInfo](#)
- [custSettingList](#) \* [pCustSettingList](#)

### 8.167.1 Detailed Description

This struture contains the TLV required to get the Customization Info and customization list.

#### Parameters

<i>IN]</i>	<p><a href="#">pGetCustomInput</a></p> <ul style="list-style-type: none"> <li>• See <a href="#">getCustomInput</a> for more information</li> </ul>
<i>OUT]</i>	<p><a href="#">pCustSettingInfo</a></p> <ul style="list-style-type: none"> <li>• See <a href="#">custSettingInfo</a> for more information</li> </ul>
<i>OUT]</i>	<p><a href="#">pCustSettingList</a></p> <ul style="list-style-type: none"> <li>• See <a href="#">custSettingList</a> for more information</li> </ul>

### 8.167.2 Field Documentation

8.167.2.1 **custSettingInfo\*** `getCustomFeatureV2::pCustSettingInfo`

8.167.2.2 **custSettingList\*** `getCustomFeatureV2::pCustSettingList`

8.167.2.3 **getCustomInput\*** `getCustomFeatureV2::pGetCustomInput`

## 8.168 getCustomInput Struct Reference

### Data Fields

- [CHAR cust\\_id](#) [64+1]
- [BYTE list\\_type](#)

### 8.168.1 Detailed Description

This structure contains which customization id or the list type want to retrieve from modem. This TLV is only applicable for 9x30 modules so far

#### Parameters

<i>cust_id</i>	<ul style="list-style-type: none"> <li>• Customization ID (Maximum 64 bytes)</li> </ul>
<i>list_type</i>	<ul style="list-style-type: none"> <li>• list type requested</li> </ul>

### 8.168.2 Field Documentation

8.168.2.1 **CHAR** `getCustomInput::cust_id`[64+1]

8.168.2.2 **BYTE** `getCustomInput::list_type`

## 8.169 getDUNCallInfoReq Struct Reference

### Data Fields

- [ULONG Mask](#)
- [BYTE \\*](#) `pReportConnStatus`
- [TransferStatInd \\*](#) `pTransferStatInd`
- [BYTE \\*](#) `pReportDormStatus`
- [BYTE \\*](#) `pReportDataBearerTech`
- [BYTE \\*](#) `pReportChannelRate`

### 8.169.1 Detailed Description

This structure contains the DUN Call Info Request parameters.

#### Parameters

<i>Mask</i>	<ul style="list-style-type: none"> <li>• Mandatory parameter</li> <li>• Set the bits corresponding to the information requested to 1</li> <li>• All other bits must be set to 0.</li> <li>• If any values are not available or applicable, the corresponding TLVs are not returned in the response. <ul style="list-style-type: none"> <li>– Bit 0 - Connection Status</li> <li>– Bit 1 - Last call end reason</li> <li>– Bit 2 - Tx/Rx bytes OK</li> <li>– Bit 3 - Dormancy status</li> <li>– Bit 4 - Data bearer</li> <li>– Bit 5 - Channel rate</li> <li>– Bit 6 - Call active duration</li> </ul> </li> </ul>
<i>pReportConn-Status</i>	<ul style="list-style-type: none"> <li>• Connect Status Indicator <ul style="list-style-type: none"> <li>– 0 - Do not report</li> <li>– 1 - Report connection status and call end reason</li> </ul> </li> </ul>
<i>pTransferStatInd</i>	<ul style="list-style-type: none"> <li>• See <a href="#">TransferStatInd</a> for more information</li> </ul>
<i>pReportDorm-Status</i>	<ul style="list-style-type: none"> <li>• Dormancy Status Indicator <ul style="list-style-type: none"> <li>– 0 - Do not report</li> <li>– 1 - Report traffic channel state of interface used for data connection</li> </ul> </li> </ul>
<i>pReportData-BearerTech</i>	<ul style="list-style-type: none"> <li>• Current Data Bearer Technology Indicator <ul style="list-style-type: none"> <li>– 0 - Do not report</li> <li>– 1 - Report radio interface used for data transfer when it changes</li> </ul> </li> </ul>
<i>pReport-ChannelRate</i>	<ul style="list-style-type: none"> <li>• Channel Rate Indicator <ul style="list-style-type: none"> <li>– 0 - Do not report</li> <li>– 1 - Report channel rate</li> </ul> </li> </ul>

## 8.169.2 Field Documentation

### 8.169.2.1 ULONG getDUNCallInfoReq::Mask

8.169.2.2 **BYTE\*** `getDUNCallInfoReq::pReportChannelRate`

8.169.2.3 **BYTE\*** `getDUNCallInfoReq::pReportConnStatus`

8.169.2.4 **BYTE\*** `getDUNCallInfoReq::pReportDataBearerTech`

8.169.2.5 **BYTE\*** `getDUNCallInfoReq::pReportDormStatus`

8.169.2.6 **TransferStatInd\*** `getDUNCallInfoReq::pTransferStatInd`

## 8.170 `getDUNCallInfoResp` Struct Reference

### Data Fields

- [ConnectionStatus](#) \* `pConnectionStatus`
- [WORD](#) \* `pCallEndReason`
- [ULONGLONG](#) \* `pTXOKBytesCount`
- [ULONGLONG](#) \* `pRXOKBytesCount`
- [BYTE](#) \* `pDormancyStatus`
- [BYTE](#) \* `pDataBearerTech`
- [ChannelRate](#) \* `pChannelRate`
- [ULONGLONG](#) \* `pLastCallTXOKBytesCnt`
- [ULONGLONG](#) \* `pLastCallRXOKBytesCnt`
- [ULONGLONG](#) \* `pMdmCallDurationActive`
- [BYTE](#) \* `pLastCallDataBearerTech`

### 8.170.1 Detailed Description

This structure contains the DUN Call Info response parameters

#### Parameters

<i>pConnection-Status</i>	<ul style="list-style-type: none"> <li>• See <a href="#">ConnectionStatus</a> for more information</li> </ul>
<i>pCallEndReason</i>	<ul style="list-style-type: none"> <li>• Last modem call end reason</li> <li>• See <a href="#">qaGobiApiTableCallEndReasons.h</a> for Call End Reason</li> <li>• Only valid if the last call made was DUN, else zero is returned</li> </ul>
<i>pTXOKBytes-Count</i>	<ul style="list-style-type: none"> <li>• Number of bytes transmitted without error</li> <li>• Returned only if a data call is up</li> </ul>
<i>pRXOKBytes-Count</i>	<ul style="list-style-type: none"> <li>• Number of bytes received without error</li> <li>• Returned only if a data call is up</li> </ul>

<i>pDormancy- Status</i>	<ul style="list-style-type: none"><li>• Current traffic channel status</li><li>• Returned if a data call is up<ul style="list-style-type: none"><li>– 0x01 - Traffic channel dormant</li><li>– 0x02 - Traffic channel active</li></ul></li></ul>
<i>pDataBearer- Tech</i>	<ul style="list-style-type: none"><li>• Current data bearer technology</li><li>• Returned only if a data call is up<ul style="list-style-type: none"><li>– 0x01 - cdma2000 1X</li><li>– 0x02 - cdma2000 HRPD (1xEV-DO)</li><li>– 0x03 - GSM</li><li>– 0x04 - UMTS</li><li>– 0x05 - cdma200 HRPD ( 1xEV-DO RevA)</li><li>– 0x06 - EDGE</li><li>– 0x07 - HSDPA and WCDMA</li><li>– 0x08 - WCDMA and HSUPA</li><li>– 0x09 - HSDPA and HSUPA</li><li>– 0x0A - LTE</li><li>– 0x0B - cdma2000 EHRPD</li><li>– 0x0C - HSDPA+ and WCDMA</li><li>– 0x0D - HSDPA+ and HSUPA</li><li>– 0x0E - DC_HSDPA+ and WCDMA</li><li>– 0x0F - DC_HSDPA+ and HSUPA</li><li>– 0x10 - HSDPA+ and 64QAM</li><li>– 0x11 - HSDPA+, 64QAM and HSUPA</li><li>– 0x12 - TDSCDMA</li><li>– 0x13 - TDSCDMA and HSDPA</li><li>– 0xFF - Unknown</li></ul></li></ul>

<i>pChannelRate</i>	<ul style="list-style-type: none"> <li>• See <a href="#">ChannelRate</a> for more information</li> </ul>
<i>pLastCallTXOK-BytesCnt</i>	<ul style="list-style-type: none"> <li>• Number of bytes transmitted without error during the last data call ( 0 if no call was made ).</li> <li>• Return only if not in a call and the previous call was made using DUN.</li> </ul>
<i>pLastCallRXOK-BytesCnt</i>	<ul style="list-style-type: none"> <li>• Number of bytes transmitted without error during the last data call ( 0 if no call was made ).</li> <li>• Return only if not in a call and the previous call was made using DUN.</li> </ul>
<i>pMdmCall-DurationActive</i>	<ul style="list-style-type: none"> <li>• Duration that the call is active in milliseconds</li> <li>• If the modem connection status is connected, this represents the active duration of the current DUN call</li> <li>• If the modem connection status is disconnected, this represents the duration of the last DUN call since the device was powered up (0 if no call has been made or if the last call was not DUN)</li> </ul>
<i>pLastCallData-BearerTech</i>	<ul style="list-style-type: none"> <li>• Last Call Data Bearer Technology</li> <li>• Returned only if not in a call and when the previous call was made using DUN <ul style="list-style-type: none"> <li>– 0x01 - cdma2000 1X</li> <li>– 0x02 - cdma2000 HRPD (1xEV-DO)</li> <li>– 0x03 - GSM</li> <li>– 0x04 - UMTS</li> <li>– 0x05 - cdma200 HRPD (1xEV-DO Rev A)</li> <li>– 0x06 - EDGE</li> <li>– 0x07 - HSDPA and WCDMA</li> <li>– 0x08 - WCDMA and HSUPA</li> <li>– 0x09 - HSDPA and HSUPA</li> <li>– 0x0A - LTE</li> <li>– 0x0B - cdma2000 EHRPD</li> <li>– 0x0C - HSDPA+ and WCDMA</li> <li>– 0x0D - HSDPA+ and HSUPA</li> <li>– 0x0E - DC_HSDPA+ and WCDMA</li> <li>– 0x0F - DC_HSDPA+ and HSUPA</li> <li>– 0x10 - HSDPA+ and 64QAM</li> <li>– 0x11 - HSDPA+, 64QAM and HSUPA</li> <li>– 0x12 - TDSCDMA</li> <li>– 0x13 - TDSCDMA and HSDPA</li> <li>– 0xFF - Unknown</li> </ul> </li> </ul>

## 8.170.2 Field Documentation

- 8.170.2.1 **WORD\*** `getDUNCallInfoResp::pCallEndReason`
- 8.170.2.2 **ChannelRate\*** `getDUNCallInfoResp::pChannelRate`
- 8.170.2.3 **ConnectionStatus\*** `getDUNCallInfoResp::pConnectionStatus`
- 8.170.2.4 **BYTE\*** `getDUNCallInfoResp::pDataBearerTech`
- 8.170.2.5 **BYTE\*** `getDUNCallInfoResp::pDormancyStatus`
- 8.170.2.6 **BYTE\*** `getDUNCallInfoResp::pLastCallDataBearerTech`
- 8.170.2.7 **ULONGLONG\*** `getDUNCallInfoResp::pLastCallRXOKBytesCnt`
- 8.170.2.8 **ULONGLONG\*** `getDUNCallInfoResp::pLastCallTXOKBytesCnt`
- 8.170.2.9 **ULONGLONG\*** `getDUNCallInfoResp::pMdmCallDurationActive`
- 8.170.2.10 **ULONGLONG\*** `getDUNCallInfoResp::pRXOKBytesCount`
- 8.170.2.11 **ULONGLONG\*** `getDUNCallInfoResp::pTXOKBytesCount`

## 8.171 GetErrRateResp Struct Reference

### Data Fields

- **WORD \*** `pCDMAFrameErrRate`
- **WORD \*** `pHDRPackErrRate`
- **BYTE \*** `pGSMBER`
- **BYTE \*** `pWCDMABER`

### 8.171.1 Detailed Description

This structure contains information about the SLQSGetErrorRate response parameters.

#### Parameters

<i>pCDMAFrame-ErrRate[Out]</i>	<ul style="list-style-type: none"><li>• CDMA Frame Error Rate</li><li>• Valid error rate values between 1 and 10000 are returned to indicate the percentage, e.g., a value of 300 means the error rate is 3%.</li><li>• A value of 0xFFFF indicates that the error rate is unknown/unavailable.</li></ul>
--------------------------------	---

<i>pHDRPackErrRate[Out]</i>	<ul style="list-style-type: none"> <li>• HDR Packet Error Rate</li> <li>• Valid error rate values between 1 and 10000 are returned to indicate the percentage, e.g., a value of 300 means the error rate is 3%.</li> <li>• A value of 0xFFFF indicates that the error rate is unknown/unavailable.</li> </ul>
<i>pGSMBER[Out]</i>	<ul style="list-style-type: none"> <li>• GSM Bit Error Rate</li> <li>• Valid error rate values between 1 and 100 are returned to indicate the percentage value.</li> <li>• A 0% block error rate (BLER) indicates No Data.</li> </ul>
<i>pWCDMABER[Out]</i>	<ul style="list-style-type: none"> <li>• WCDMA Block Error Rate</li> <li>• Valid error rate values between 1 and 100 are returned to indicate the percentage value.</li> <li>• A value of 0xFF indicates that the error rate is unknown/unavailable.</li> </ul>

## 8.171.2 Field Documentation

8.171.2.1 **WORD\*** GetErrRateResp::pCDMAFrameErrRate

8.171.2.2 **BYTE\*** GetErrRateResp::pGSMBER

8.171.2.3 **WORD\*** GetErrRateResp::pHDRPackErrRate

8.171.2.4 **BYTE\*** GetErrRateResp::pWCDMABER

## 8.172 GetHRPDStatsResp Struct Reference

### Data Fields

- [DRCParams](#) \* [pDRCParams](#)
- [BYTE](#) \* [pUATI](#)
- [PilotSetData](#) \* [pPilotSetData](#)

### 8.172.1 Detailed Description

This structure contains information about the SLQSSwiGetHRPDStats response parameters.

#### Parameters

<i>pDRCParams[Out]</i>	<ul style="list-style-type: none"> <li>• See <a href="#">DRCParams</a> for more information.</li> </ul>
------------------------	---



<i>pUATI[Out]</i>	<ul style="list-style-type: none"> <li>• A 128-bit address that includes the access terminal identifier and subnet ID</li> <li>• Size must be 16 bytes</li> </ul>
<i>pPilotSetData[-Out]</i>	<ul style="list-style-type: none"> <li>• See <a href="#">PilotSetData</a> for more information.</li> </ul>

## 8.172.2 Field Documentation

8.172.2.1 **DRCParams\*** GetHRPDStatsResp::pDRCParams

8.172.2.2 **PilotSetData\*** GetHRPDStatsResp::pPilotSetData

8.172.2.3 **BYTE\*** GetHRPDStatsResp::pUATI

## 8.173 GetIMSSMSConfigParams Struct Reference

### Data Fields

- [BYTE](#) \* [pSettingResp](#)
- [BYTE](#) \* [pSMSFormat](#)
- [BYTE](#) \* [pSMSOverIPNwInd](#)
- [BYTE](#) \* [pPhoneCtxtURLen](#)
- [BYTE](#) \* [pPhoneCtxtURL](#)

### 8.173.1 Detailed Description

This structure contains the SLQSGetIMSSMSConfig response parameters.

#### Parameters

<i>pSettingResp</i>	<ul style="list-style-type: none"> <li>• Settings Response</li> </ul>
<i>pSMSFormat</i>	<ul style="list-style-type: none"> <li>• SMS format <ul style="list-style-type: none"> <li>– 0 - 3GPP</li> <li>– 1 - 3GPP2</li> </ul> </li> </ul>
<i>pSMSOverIPNwInd</i>	<ul style="list-style-type: none"> <li>• SMS over IP Network Indication Flag <ul style="list-style-type: none"> <li>– TRUE - Turn on mobile-originated SMS</li> <li>– FALSE - Turn off mobile-originated SMS</li> </ul> </li> </ul>

<i>pPhoneCtxtURLen</i>	<ul style="list-style-type: none"> <li>Size in bytes assigned to the Phone context Universal Resource Identifier to follow</li> </ul>
<i>pPhoneCtxtURI</i>	<ul style="list-style-type: none"> <li>Phone context universal resource identifier</li> <li>Length of this string must be specified in pPhoneCtxtURLen parameter</li> </ul>

### 8.173.2 Field Documentation

8.173.2.1 **BYTE\*** GetIMSSMSConfigParams::pPhoneCtxtURI

8.173.2.2 **BYTE\*** GetIMSSMSConfigParams::pPhoneCtxtURLen

8.173.2.3 **BYTE\*** GetIMSSMSConfigParams::pSettingResp

8.173.2.4 **BYTE\*** GetIMSSMSConfigParams::pSMSFormat

8.173.2.5 **BYTE\*** GetIMSSMSConfigParams::pSMSOverIPNwInd

## 8.174 GetIMSUserConfigParams Struct Reference

### Data Fields

- BYTE \*** [pSettingResp](#)
- BYTE \*** [pIMSDomainLen](#)
- BYTE \*** [pIMSDomain](#)

### 8.174.1 Detailed Description

This structure contains the SLQSGetIMSUserConfig response parameters.

#### Parameters

<i>pSettingResp</i>	<ul style="list-style-type: none"> <li>Settings Response</li> </ul>
<i>pIMSDomainLen</i>	<ul style="list-style-type: none"> <li>Length of IMS <a href="#">Domain</a> Name to follow</li> </ul>
<i>pIMSDomain</i>	<ul style="list-style-type: none"> <li>IMS domain name</li> <li>Length of this string must be specified in pIMSDomainLen parameter</li> </ul>

### 8.174.2 Field Documentation

8.174.2.1 **BYTE\*** GetIMSUserConfigParams::pIMSDomain

8.174.2.2 **BYTE\*** GetIMSUserConfigParams::pIMSDomainLen

8.174.2.3 BYTE\* GetIMSUserConfigParams::pSettingResp

## 8.175 GetIMSVoIPConfigResp Struct Reference

### Data Fields

- BYTE \* pSettingResp
- WORD \* pSessionExpiryTimer
- WORD \* pMinSessionExpiryTimer
- BYTE \* pAmrWbEnable
- BYTE \* pScrAmrEnable
- BYTE \* pScrAmrWbEnable
- BYTE \* pAmrMode
- WORD \* pAmrWBMode
- BYTE \* pAmrOctetAligned
- BYTE \* pAmrWBOctetAligned
- WORD \* pRingingTimer
- WORD \* pRingBackTimer
- WORD \* pRTPRTCPInactTimer

### 8.175.1 Detailed Description

This structure contains the SLQSGetIMSVoIPConfig request parameters.

#### Parameters

<i>pSettingResp</i>	<ul style="list-style-type: none"> <li>• Settings Response. A settings specific error code is returned when the standard response error type is QMI_ERR_CAUSE_CODE</li> </ul>
<i>pSessionExpiryTimer</i>	<ul style="list-style-type: none"> <li>• Session duration, in seconds</li> </ul>
<i>pMinSessionExpiryTimer</i>	<ul style="list-style-type: none"> <li>• Minimum allowed value for session expiry timer, in seconds</li> </ul>
<i>pAmrWbEnable</i>	<ul style="list-style-type: none"> <li>• Flag to enable/disable Adaptive Multirate Codec(AMR) WideBand(WB) audio</li> <li>• Values: <ul style="list-style-type: none"> <li>– True - Enable</li> <li>– False - Disable</li> </ul> </li> </ul>

<i>pScrAmrEnable</i>	<ul style="list-style-type: none"> <li>• Flag to enable/disable Source Control Rate(SCR) for AMR NarrowBand (NB)</li> <li>• Values: <ul style="list-style-type: none"> <li>– True - Enable</li> <li>– False - Disable</li> </ul> </li> </ul>
<i>pScrAmrWb-Enable</i>	<ul style="list-style-type: none"> <li>• Flag to enable/disable SCR for AMR WB Audio</li> <li>• Values: <ul style="list-style-type: none"> <li>– True - Enable</li> <li>– False - Disable</li> </ul> </li> </ul>
<i>pAmrMode</i>	<ul style="list-style-type: none"> <li>• BitMask for AMR NB modes allowed</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0x1 - 4.75 kbps</li> <li>– 0x2 - 5.15 kbps</li> <li>– 0x4 - 5.9 kbps</li> <li>– 0x8 - 6.17 kbps</li> <li>– 0x10 - 7.4 kbps</li> <li>– 0x20 - 7.95 kbps</li> <li>– 0x40 - 10.2 kbps</li> <li>– 0x80 - 12.2 kbps</li> </ul> </li> </ul>
<i>pAmrWBMode</i>	<ul style="list-style-type: none"> <li>• BitMask for AMR WB modes allowed</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0x1 - 6.60 kbps</li> <li>– 0x2 - 8.85 kbps</li> <li>– 0x4 - 12.65 kbps</li> <li>– 0x8 - 14.25 kbps</li> <li>– 0x10 - 15.85 kbps</li> <li>– 0x20 - 18.25 kbps</li> <li>– 0x40 - 19.85 kbps</li> <li>– 0x80 - 23.05 kbps</li> <li>– 0x100 - 23.85 kbps</li> </ul> </li> </ul>

<i>pAmrOctet-Aligned</i>	<ul style="list-style-type: none"> <li>• Flag to indicate if the octet is aligned for AMR NB Audio</li> <li>• Values: <ul style="list-style-type: none"> <li>– True - Aligned</li> <li>– False - Not aligned, Bandwidth Efficient mode</li> </ul> </li> </ul>
<i>pAmrWBOctet-Aligned</i>	<ul style="list-style-type: none"> <li>• Flag to indicate if the octet is aligned for AMR WB Audio</li> <li>• Values: <ul style="list-style-type: none"> <li>– True - Aligned</li> <li>– False - Not aligned, Bandwidth Efficient mode</li> </ul> </li> </ul>
<i>pRingingTimer</i>	<ul style="list-style-type: none"> <li>• Duration of ringing timer, in seconds. The ringing timer starts on the ringing event. If the call is not answered within the duration of this timer, the call is disconnected.</li> </ul>
<i>pRingBackTimer</i>	<ul style="list-style-type: none"> <li>• Duration of ringback timer, in seconds. The ringback timer starts on the ringback event. If the call is not answered within the duration of this timer, the call is disconnected.</li> </ul>
<i>pRTPRTCP-InactTimer</i>	<ul style="list-style-type: none"> <li>• Duration of RTP/RTCP inactivity timer, in seconds. If no RTP/RTCP packet is received prior to the expiry of this timer, the call is disconnected.</li> </ul>

## 8.175.2 Field Documentation

8.175.2.1 **BYTE\*** GetIMSVoIPConfigResp::pAmrMode

8.175.2.2 **BYTE\*** GetIMSVoIPConfigResp::pAmrOctetAligned

8.175.2.3 **BYTE\*** GetIMSVoIPConfigResp::pAmrWbEnable

8.175.2.4 **WORD\*** GetIMSVoIPConfigResp::pAmrWBMode

8.175.2.5 **BYTE\*** GetIMSVoIPConfigResp::pAmrWBOctetAligned

8.175.2.6 **WORD\*** GetIMSVoIPConfigResp::pMinSessionExpiryTimer

8.175.2.7 **WORD\*** GetIMSVoIPConfigResp::pRingBackTimer

8.175.2.8 **WORD\*** GetIMSVoIPConfigResp::pRingingTimer

8.175.2.9 **WORD\*** GetIMSVoIPConfigResp::pRTPRTCPInactTimer

8.175.2.10 **BYTE\*** GetIMSVoIPConfigResp::pScrAmrEnable

8.175.2.11 **BYTE\*** GetIMSVoIPConfigResp::pScrAmrWbEnable

8.175.2.12 WORD\* GetIMSVoIPConfigResp::pSessionExpiryTimer

8.175.2.13 BYTE\* GetIMSVoIPConfigResp::pSettingResp

## 8.176 GetInstIDResp Struct Reference

### Data Fields

- [BYTE \\* pInstanceID](#)
- [BYTE \\* pIPFamily](#)

### 8.176.1 Field Documentation

8.176.1.1 BYTE\* GetInstIDResp::pInstanceID

8.176.1.2 BYTE\* GetInstIDResp::pIPFamily

## 8.177 GetM2MAudioProfileReq Struct Reference

### Data Fields

- [BYTE \\* pGenerator](#)

### 8.177.1 Detailed Description

This structure contains the SLQSGetM2MAudioProfile request parameters.

#### Parameters

<i>p-Generator[optional]</i>	<ul style="list-style-type: none"> <li>• Generator             <ul style="list-style-type: none"> <li>– 0 - Voice</li> </ul> </li> </ul>
------------------------------	--

### 8.177.2 Field Documentation

8.177.2.1 BYTE\* GetM2MAudioProfileReq::pGenerator

## 8.178 GetM2MAudioProfileResp Struct Reference

### Data Fields

- [BYTE Profile](#)
- [BYTE EarMute](#)
- [BYTE MicMute](#)
- [BYTE Generator](#)
- [BYTE Volume](#)
- [BYTE CwtMute](#)

### 8.178.1 Detailed Description

This structure contains the SLQSGetM2MAudioProfile response parameters.

## Parameters

<i>Profile</i>	<ul style="list-style-type: none"> <li>• Audio Profile <ul style="list-style-type: none"> <li>– 0-5</li> </ul> </li> </ul>
<i>EarMute</i>	<ul style="list-style-type: none"> <li>• Ear Mute <ul style="list-style-type: none"> <li>– 0 - Mute</li> <li>– 1 - Unmute</li> </ul> </li> </ul>
<i>MicMute</i>	<ul style="list-style-type: none"> <li>• MIC Mute <ul style="list-style-type: none"> <li>– 0 - Mute</li> <li>– 1 - Unmute</li> </ul> </li> </ul>
<i>Generator</i>	<ul style="list-style-type: none"> <li>• Generator <ul style="list-style-type: none"> <li>– 0 - Voice</li> </ul> </li> </ul>
<i>Volume</i>	<ul style="list-style-type: none"> <li>• RX volume level <ul style="list-style-type: none"> <li>– 0-5</li> </ul> </li> </ul>
<i>CwtMute</i>	<ul style="list-style-type: none"> <li>• Call waiting tone Mute <ul style="list-style-type: none"> <li>– 0 - Mute</li> <li>– 1 - Unmute</li> </ul> </li> </ul>

## 8.178.2 Field Documentation

8.178.2.1 BYTE GetM2MAudioProfileResp::CwtMute

8.178.2.2 BYTE GetM2MAudioProfileResp::EarMute

8.178.2.3 BYTE GetM2MAudioProfileResp::Generator

8.178.2.4 BYTE GetM2MAudioProfileResp::MicMute

8.178.2.5 BYTE GetM2MAudioProfileResp::Profile

8.178.2.6 BYTE GetM2MAudioProfileResp::Volume



## 8.179 GetM2MAudioVolumeReq Struct Reference

### Data Fields

- [BYTE Profile](#)
- [BYTE Generator](#)

#### 8.179.1 Detailed Description

This structure contains the SLQSGetM2MAudioVolume request parameters.

##### Parameters

<i>Profile</i>	<ul style="list-style-type: none"><li>• Audio Profile<ul style="list-style-type: none"><li>– 0-5</li></ul></li></ul>
<i>Generator</i>	<ul style="list-style-type: none"><li>• Generator<ul style="list-style-type: none"><li>– 0 - Voice</li></ul></li></ul>

#### 8.179.2 Field Documentation

##### 8.179.2.1 BYTE GetM2MAudioVolumeReq::Generator

##### 8.179.2.2 BYTE GetM2MAudioVolumeReq::Profile

## 8.180 GetM2MAudioVolumeResp Struct Reference

### Data Fields

- [BYTE Level](#)

#### 8.180.1 Detailed Description

This structure contains the SLQSGetM2MAudioVolume response parameters.

##### Parameters

<i>Level</i>	<ul style="list-style-type: none"><li>• The RX Volume Level<ul style="list-style-type: none"><li>– 0-5</li></ul></li></ul>
--------------	--

#### 8.180.2 Field Documentation

##### 8.180.2.1 BYTE GetM2MAudioVolumeResp::Level

## 8.181 GetM2MAVMuteReq Struct Reference

### Data Fields

- [BYTE Profile](#)

### 8.181.1 Detailed Description

This structure contains the SLQSGetM2MAVMute request parameters.

#### Parameters

<i>Profile</i>	<ul style="list-style-type: none"> <li>• Audio Profile Number <ul style="list-style-type: none"> <li>– 0-5</li> </ul> </li> </ul>
----------------	---

### 8.181.2 Field Documentation

#### 8.181.2.1 BYTE GetM2MAVMuteReq::Profile

## 8.182 GetM2MAVMuteResp Struct Reference

### Data Fields

- [BYTE EarMute](#)
- [BYTE MicMute](#)
- [BYTE CwtMute](#)

### 8.182.1 Detailed Description

This structure contains the SLQSGetM2MAVMute response parameters.

#### Parameters

<i>pEarMute</i>	<ul style="list-style-type: none"> <li>• Ear Mute <ul style="list-style-type: none"> <li>– 0-Mute</li> <li>– 1-UnMute</li> </ul> </li> </ul>
<i>pMicMute</i>	<ul style="list-style-type: none"> <li>• Mic Mute <ul style="list-style-type: none"> <li>– 0-Mute</li> <li>– 1-unmute</li> </ul> </li> </ul>

<i>CwtMute</i>	<ul style="list-style-type: none"><li>• Waiting tone Mute<ul style="list-style-type: none"><li>– 0-5</li></ul></li></ul>
----------------	--

### 8.182.2 Field Documentation

8.182.2.1 **BYTE** GetM2MAVMuteResp::CwtMute

8.182.2.2 **BYTE** GetM2MAVMuteResp::EarMute

8.182.2.3 **BYTE** GetM2MAVMuteResp::MicMute

## 8.183 GetM2MSpkrGainReq Struct Reference

### Data Fields

- [BYTE Profile](#)

### 8.183.1 Detailed Description

This structure contains the SLQSGetM2MSpkrGain request parameters.

#### Parameters

<i>pProfile</i>	<ul style="list-style-type: none"><li>• Audio Profile Number<ul style="list-style-type: none"><li>– 0-5</li></ul></li></ul>
-----------------	---

### 8.183.2 Field Documentation

8.183.2.1 **BYTE** GetM2MSpkrGainReq::Profile

## 8.184 GetM2MSpkrGainResp Struct Reference

### Data Fields

- [WORD Value](#)

### 8.184.1 Detailed Description

This structure contains the SLQSGetM2MSpkrGain response parameters.

#### Parameters

<i>Value</i>	<ul style="list-style-type: none"> <li>• RX speakerphone gain <ul style="list-style-type: none"> <li>– 0x0 - 0x7fff</li> </ul> </li> </ul>
--------------	--

## 8.184.2 Field Documentation

### 8.184.2.1 WORD GetM2MSpkrGainResp::Value

## 8.185 getMsgWaitingInfo Struct Reference

### Data Fields

- [BYTE numInstances](#)
- [messageWaitingInfoContent msgWaitInfo \[0xFF\]](#)

### 8.185.1 Detailed Description

This structure contains Get Message Waiting Info Response parameters

#### Parameters

<i>numInstances</i>	<ul style="list-style-type: none"> <li>• Number of sets of the elements in structure <a href="#">messageWaitingInfoContent</a></li> </ul>
<i>pMsgWaitInfo</i>	<ul style="list-style-type: none"> <li>• Pointer to structure of <a href="#">messageWaitingInfoContent</a>. <ul style="list-style-type: none"> <li>– See <a href="#">messageWaitingInfoContent</a> for more information.</li> </ul> </li> </ul>

## 8.185.2 Field Documentation

### 8.185.2.1 messageWaitingInfoContent getMsgWaitingInfo::msgWaitInfo[0xFF]

### 8.185.2.2 BYTE getMsgWaitingInfo::numInstances

## 8.186 GetRegMgrConfigParams Struct Reference

### Data Fields

- [BYTE \\* pSettingResp](#)
- [WORD \\* pPCSCFPort](#)
- [BYTE \\* pPriCSCFPortNameLen](#)
- [BYTE \\* pPriCSCFPortName](#)
- [BYTE \\* pIMSTestMode](#)

### 8.186.1 Detailed Description

This structure contains the SLQSGetRegMgrConfig response parameters.

## Parameters

<i>pSettingResp</i>	<ul style="list-style-type: none"> <li>• Settings Response</li> </ul>
<i>pPCSCFPort</i>	<ul style="list-style-type: none"> <li>• Proxy call session control function port</li> </ul>
<i>pPriCSCFPort-NameLen(IN/OUT)</i>	<ul style="list-style-type: none"> <li>• Size in bytes assigned to the primary CSCF <a href="#">Port</a> name parameter to follow</li> </ul>
<i>pPriCSCFPort-Name</i>	<ul style="list-style-type: none"> <li>• Call Session control port, fully qualified domain name</li> <li>• Length of this string must be specified in pPriCSCFPortNameLen parameter</li> </ul>
<i>pIMSTestMode</i>	<ul style="list-style-type: none"> <li>• IMS Test mode Enabled. <ul style="list-style-type: none"> <li>– TRUE - Enabled</li> <li>– FALSE - Disabled</li> </ul> </li> </ul>

## Note

pPriCSCFPortNameLen must be set to a valid value during API call to retrieve pPriCSCFPortName.

## 8.186.2 Field Documentation

8.186.2.1 **BYTE\*** GetRegMgrConfigParams::pIMSTestMode

8.186.2.2 **WORD\*** GetRegMgrConfigParams::pPCSCFPort

8.186.2.3 **BYTE\*** GetRegMgrConfigParams::pPriCSCFPortName

8.186.2.4 **BYTE\*** GetRegMgrConfigParams::pPriCSCFPortNameLen

8.186.2.5 **BYTE\*** GetRegMgrConfigParams::pSettingResp

## 8.187 GetSessionIDResp Struct Reference

## Data Fields

- [ULONG](#) \* pSessionIDv4
- [ULONG](#) \* pSessionIDv6

## 8.187.1 Field Documentation

8.187.1.1 **ULONG\*** GetSessionIDResp::pSessionIDv4

8.187.1.2 **ULONG\*** GetSessionIDResp::pSessionIDv6

## 8.188 GetSIPConfigResp Struct Reference

### Data Fields

- [BYTE](#) \* [pSettingResp](#)
- [WORD](#) \* [pSIPLocalPort](#)
- [ULONG](#) \* [pTimerSIPReg](#)
- [ULONG](#) \* [pSubscribeTimer](#)
- [ULONG](#) \* [pTimerT1](#)
- [ULONG](#) \* [pTimerT2](#)
- [ULONG](#) \* [pTimerTf](#)
- [BYTE](#) \* [pSigCompEnabled](#)

### 8.188.1 Detailed Description

This structure contains the SLQSGetSIPConfig response parameters.

#### Parameters

<i>pSettingResp</i>	<ul style="list-style-type: none"> <li>• Settings Response</li> </ul>
<i>pSIPLocalPort</i>	<ul style="list-style-type: none"> <li>• Primary call session control function SIP port number</li> </ul>
<i>pTimerSIPReg</i>	<ul style="list-style-type: none"> <li>• Initial SIP registration duration from the User equipment, in seconds</li> </ul>
<i>pSubscribeTimer</i>	<ul style="list-style-type: none"> <li>• Duration of the subscription by the UE for IMS registration notifications, in seconds</li> </ul>
<i>pTimerT1</i>	<ul style="list-style-type: none"> <li>• RTT estimate, in milliseconds</li> </ul>
<i>pTimerT2</i>	<ul style="list-style-type: none"> <li>• The maximum retransmit interval for non-invite requests and invite responses, in milliseconds</li> </ul>
<i>pTimerTf</i>	<ul style="list-style-type: none"> <li>• Non-invite transaction timeout timer, in milliseconds</li> </ul>
<i>pSigComp-Enabled</i>	<ul style="list-style-type: none"> <li>• Sig Comp Status               <ul style="list-style-type: none"> <li>– TRUE - Sig Comp Enabled</li> <li>– FALSE - Sig Comp Disabled</li> </ul> </li> </ul>

### 8.188.2 Field Documentation

#### 8.188.2.1 [BYTE](#)\* GetSIPConfigResp::pSettingResp

8.188.2.2 **BYTE\*** GetSIPConfigResp::pSigCompEnabled

8.188.2.3 **WORD\*** GetSIPConfigResp::pSIPLocalPort

8.188.2.4 **ULONG\*** GetSIPConfigResp::pSubscribeTimer

8.188.2.5 **ULONG\*** GetSIPConfigResp::pTimerSIPReg

8.188.2.6 **ULONG\*** GetSIPConfigResp::pTimerT1

8.188.2.7 **ULONG\*** GetSIPConfigResp::pTimerT2

8.188.2.8 **ULONG\*** GetSIPConfigResp::pTimerTf

## 8.189 GnssData Struct Reference

### Data Fields

- [ULONGLONG mask](#)

### 8.189.1 Detailed Description

This structure contains the GNSS data

## Parameters

<i>mask</i>	<ul style="list-style-type: none"> <li>• Mask for the GNSS data that is to be deleted</li> <li>• Valid values: <ul style="list-style-type: none"> <li>– QMI_LOC_MASK_DELETE_GPS_SVDIR (0x00000001) - Mask to delete GPS SVDIR</li> <li>– QMI_LOC_MASK_DELETE_GPS_SVSTEER (0x00000002) - Mask to delete GPS SVSTEER</li> <li>– QMI_LOC_MASK_DELETE_GPS_TIME (0x00000004) - Mask to delete GPS time</li> <li>– QMI_LOC_MASK_DELETE_GPS_ALM_CORR (0x00000008) - Mask to delete almanac correlation</li> <li>– QMI_LOC_MASK_DELETE_GLO_SVDIR (0x00000010) - Mask to delete GLO-NASS SVDIR</li> <li>– QMI_LOC_MASK_DELETE_GLO_SVSTEER (0x00000020) - Mask to delete G-LONASS SVSTEER</li> <li>– QMI_LOC_MASK_DELETE_GLO_TIME (0x00000040) - Mask to delete GLONASS time</li> <li>– QMI_LOC_MASK_DELETE_GLO_ALM_CORR (0x00000080) - Mask to delete GLONASS almanac correlation</li> <li>– QMI_LOC_MASK_DELETE_SBAS_SVDIR (0x00000100) - Mask to delete SBAS SVDIR</li> <li>– QMI_LOC_MASK_DELETE_SBAS_SVSTEER (0x00000200) - Mask to delete SBAS SVSTEER</li> <li>– QMI_LOC_MASK_DELETE_POSITION (0x00000400) - Mask to delete position estimate</li> <li>– QMI_LOC_MASK_DELETE_TIME (0x00000800) - Mask to delete time estimate</li> <li>– QMI_LOC_MASK_DELETE_IONO (0x00001000) - Mask to delete IONO</li> <li>– QMI_LOC_MASK_DELETE_UTC (0x00002000) - Mask to delete UTC estimate</li> <li>– QMI_LOC_MASK_DELETE_HEALTH (0x00004000) - Mask to delete <a href="#">SV</a> health record</li> <li>– QMI_LOC_MASK_DELETE_SADATA (0x00008000) - Mask to delete SADATA</li> <li>– QMI_LOC_MASK_DELETE_RTI (0x00010000) - Mask to delete RTI</li> <li>– QMI_LOC_MASK_DELETE_SV_NO_EXIST (0x00020000) - Mask to delete SV_NO_EXIST</li> <li>– QMI_LOC_MASK_DELETE_FREQ_BIAS_EST (0x00040000) - Mask to delete frequency bias estimate</li> <li>– QMI_LOC_MASK_DELETE_BDS_SVDIR (0x00080000) - Mask to delete BDS SVDIR</li> <li>– QMI_LOC_MASK_DELETE_BDS_SVSTEER (0x00100000) - Mask to delete BDS SVSTEER</li> <li>– QMI_LOC_MASK_DELETE_BDS_TIME (0x00200000) - Mask to delete BDS time</li> <li>– QMI_LOC_MASK_DELETE_BDS_ALM_CORR (0x00400000) - Mask to delete BDS almanac correlation</li> <li>– QMI_LOC_MASK_DELETE_GNSS_SV_BLACKLIST_GPS (0x00800000) - Mask to delete GNSS <a href="#">SV</a> blacklist GPS</li> <li>– QMI_LOC_MASK_DELETE_GNSS_SV_BLACKLIST_GLO (0x01000000) - Mask to delete GNSS <a href="#">SV</a> blacklist GLO</li> <li>– QMI_LOC_MASK_DELETE_GNSS_SV_BLACKLIST_BDS (0x02000000) - Mask to delete GNSS <a href="#">SV</a> blacklist BDS</li> </ul> </li> </ul>
	<p>Generated on Wed Oct 28 2015 00:40:12 for LinuxQMI SDK by Doxygen</p>



## 8.189.2 Field Documentation

### 8.189.2.1 ULONGLONG GnssData::mask

## 8.190 gnssSvInfoNotification Struct Reference

### Data Fields

- [BYTE bAltitudeAssumed](#)
- [satelliteInfo](#) \* [pSatelliteInfo](#)

### 8.190.1 Detailed Description

Contain the parameters passed for SetLocGnssSvInfoCallback by the device.

#### Parameters

<i>bAltitudeAssumed</i>	<ul style="list-style-type: none"> <li>• Indicates whether altitude is assumed or calculated <ul style="list-style-type: none"> <li>– 0x00 (FALSE) - Valid altitude is calculated</li> <li>– 0x01 (TRUE) - Valid altitude is assumed; there may not be enough satellites to determine precise altitude</li> </ul> </li> </ul>
<i>pSatelliteInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">satelliteInfo</a> for more information.</li> </ul>

#### Note

None

## 8.190.2 Field Documentation

### 8.190.2.1 BYTE gnssSvInfoNotification::bAltitudeAssumed

### 8.190.2.2 satelliteInfo\* gnssSvInfoNotification::pSatelliteInfo

## 8.191 GPRSQoS Struct Reference

### Data Fields

- [ULONG precedenceClass](#)
- [ULONG delayClass](#)
- [ULONG reliabilityClass](#)
- [ULONG peakThroughputClass](#)
- [ULONG meanThroughputClass](#)

### 8.191.1 Detailed Description

This structure contains the GPRS Quality Of Service Information

- Parameter values default to their data type's maximum unsigned value unless explicitly stated otherwise.

## Parameters

<i>precedence-Class</i>	<ul style="list-style-type: none"> <li>• Precedence class</li> </ul>
<i>delayClass</i>	<ul style="list-style-type: none"> <li>• Delay class</li> </ul>
<i>reliabilityClass</i>	<ul style="list-style-type: none"> <li>• Reliability class</li> </ul>
<i>peak-Throughput-Class</i>	<ul style="list-style-type: none"> <li>• Peak throughput class</li> </ul>
<i>mean-Throughput-Class</i>	<ul style="list-style-type: none"> <li>• Mean throughput class</li> </ul>

## 8.191.2 Field Documentation

8.191.2.1 **ULONG** GPRSQoS::delayClass8.191.2.2 **ULONG** GPRSQoS::meanThroughputClass8.191.2.3 **ULONG** GPRSQoS::peakThroughputClass8.191.2.4 **ULONG** GPRSQoS::precedenceClass8.191.2.5 **ULONG** GPRSQoS::reliabilityClass

## 8.192 GPRSRequestedQoS Struct Reference

## Data Fields

- [ULONG precedenceClass](#)
- [ULONG delayClass](#)
- [ULONG reliabilityClass](#)
- [ULONG peakThroughputClass](#)
- [ULONG meanThroughputClass](#)

## 8.192.1 Detailed Description

This structure contains the GPRS Quality Of Service Information

## Parameters

<i>precedence-Class</i>	<ul style="list-style-type: none"> <li>• Precedence class</li> </ul>
-------------------------	--

<i>delayClass</i>	<ul style="list-style-type: none"> <li>• Delay class</li> </ul>
<i>reliabilityClass</i>	<ul style="list-style-type: none"> <li>• Reliability class</li> </ul>
<i>peak-Throughput-Class</i>	<ul style="list-style-type: none"> <li>• Peak throughput class</li> </ul>
<i>mean-Throughput-Class</i>	<ul style="list-style-type: none"> <li>• Mean throughput class</li> </ul>

## 8.192.2 Field Documentation

8.192.2.1 **ULONG** GPRSRequestedQoS::delayClass

8.192.2.2 **ULONG** GPRSRequestedQoS::meanThroughputClass

8.192.2.3 **ULONG** GPRSRequestedQoS::peakThroughputClass

8.192.2.4 **ULONG** GPRSRequestedQoS::precedenceClass

8.192.2.5 **ULONG** GPRSRequestedQoS::reliabilityClass

## 8.193 GPSSStateInfo Struct Reference

### Data Fields

- [BYTE](#) EngineState
- [ULONG](#) ValidMask
- [ULONGLONG](#) Latitude
- [ULONGLONG](#) Longitude
- [ULONG](#) HorizontalUncertainty
- [ULONG](#) Altitude
- [ULONG](#) VerticalUncertainty
- [ULONG](#) TimeStmp\_tow\_ms
- [WORD](#) TimeStmp\_gps\_week
- [ULONG](#) Time\_uncert\_ms
- [BYTE](#) Iono\_valid
- [ULONG](#) gps\_ephemeris\_sv\_msk
- [ULONG](#) gps\_almanac\_sv\_msk
- [ULONG](#) gps\_health\_sv\_msk
- [ULONG](#) gps\_visible\_sv\_msk
- [ULONG](#) glo\_ephemeris\_sv\_msk
- [ULONG](#) glo\_almanac\_sv\_msk
- [ULONG](#) glo\_health\_sv\_msk
- [ULONG](#) glo\_visible\_sv\_msk
- [ULONG](#) sbas\_ephemeris\_sv\_msk
- [ULONG](#) sbas\_almanac\_sv\_msk
- [ULONG](#) sbas\_health\_sv\_msk
- [ULONG](#) sbas\_visible\_sv\_msk
- [WORD](#) xtra\_start\_gps\_week

- [WORD xtra\\_start\\_gps\\_minutes](#)
- [WORD xtra\\_valid\\_duration\\_hours](#)

### 8.193.1 Detailed Description

GPS state Info.

Parameters

<i>EngineState</i>	<ul style="list-style-type: none"> <li>• Values: <ul style="list-style-type: none"> <li>– 0 - OFF</li> <li>– 1 - ON</li> </ul> </li> <li>• This field is always valid</li> </ul>
<i>ValidMask</i>	<ul style="list-style-type: none"> <li>• Mask of valid state information data.</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0x00000001 - Position(latitude/longitude/horizontal uncertainty)</li> <li>– 0x00000002 - Altitude and vertical uncertainty</li> <li>– 0x00000004 - Time ms</li> <li>– 0x00000008 - Time week number</li> <li>– 0x00000010 - Time uncertainty</li> <li>– 0x00000020 - Iono validity</li> <li>– 0x00000040 - GPS ephemeris</li> <li>– 0x00000080 - GPS almanac</li> <li>– 0x00000100 - GPS health</li> <li>– 0x00000200 - GPS visible SVs</li> <li>– 0x00000400 - GLONASS ephemeris</li> <li>– 0x00000800 - GLONASS almanac</li> <li>– 0x00001000 - GLONASS health</li> <li>– 0x00002000 - GLONASS visible SVs</li> <li>– 0x00004000 - SBAS ephemeris</li> <li>– 0x00008000 - SBAS almanac</li> <li>– 0x00010000 - SBAS health</li> <li>– 0x00020000 - SBAS visible SVs</li> <li>– 0x00040000 - XTRA information</li> </ul> </li> </ul>

<i>Latitude</i>	<ul style="list-style-type: none"> <li>Latitude position referenced to the WGS-84 reference ellipsoid, counting positive angles north of the equator and negative angles south of the equator.</li> <li>Units: Decimal degrees</li> <li>Range: -90 to +90 degrees.</li> <li>Value is in double float format (refer to IEEE Std 754-1985)</li> </ul>
<i>Longitude</i>	<ul style="list-style-type: none"> <li>Longitude position referenced to the WGS-84 reference ellipsoid, counting positive angles east of the Greenwich Meridian and negative angles west of Greenwich meridian.</li> <li>Units: Decimal degrees</li> <li>Range: -180 to +180 degrees</li> <li>Value is in double float format (refer to IEEE Std 754-1985)</li> </ul>
<i>Horizontal-Uncertainty</i>	<ul style="list-style-type: none"> <li>Circular horizontal uncertainty (in meters). The uncertainty is provided at 63 percent confidence.</li> <li>Value is in single float format (refer to IEEE Std 754-1985)</li> </ul>
<i>Altitude</i>	<ul style="list-style-type: none"> <li>Height above the WGS-84 reference ellipsoid. Value conveys height (in meters) plus 500 m</li> <li>Range -500 to 15883</li> <li>Value in single float format (refer to IEEE Std 754-1985)</li> </ul>
<i>Vertical-Uncertainty</i>	<ul style="list-style-type: none"> <li>Vertical uncertainty (in meters). The uncertainty is provided at 68 percent confidence.</li> <li>Value in single float format (refer to IEEE Std 754-1985)</li> </ul>
<i>TimeStmp_tow_-ms</i>	<ul style="list-style-type: none"> <li>Time stamp in GPS time of week( in milliseconds)</li> </ul>
<i>TimeStmp_gps-_week</i>	<ul style="list-style-type: none"> <li>GPS week number</li> </ul>
<i>Time_uncert_ms</i>	<ul style="list-style-type: none"> <li>Time uncertainty (in milliseconds). The uncertainty is provided at 99 percent confidence.</li> </ul>
<i>Iono_valid</i>	<ul style="list-style-type: none"> <li>Iono validity.</li> <li>Values: <ul style="list-style-type: none"> <li>0 - Invalid</li> <li>1 - Valid</li> </ul> </li> </ul>

<i>gps_ephemeris_sv_msk</i>	<ul style="list-style-type: none"> <li>GPS <b>SV</b> mask for ephemeris; if the bit is set, ephemeris for that <b>SV</b> is available.</li> </ul>
<i>gps_almanac_sv_msk</i>	<ul style="list-style-type: none"> <li>GPS <b>SV</b> mask for almanac; if the bit is set, almanac for that <b>SV</b> is available.</li> </ul>
<i>gps_health_sv_msk</i>	<ul style="list-style-type: none"> <li>GPS <b>SV</b> mask for health; if the bit is set, health for that <b>SV</b> is available.</li> </ul>
<i>gps_visible_sv_msk</i>	<ul style="list-style-type: none"> <li>GPS <b>SV</b> mask for visible Svcs; if the bit is set, the <b>SV</b> is available.</li> </ul>
<i>glo_ephemeris_sv_msk</i>	<ul style="list-style-type: none"> <li>GLONASS <b>SV</b> mask for ephemeris; if the bit is set, ephemeris for that <b>SV</b> is available.</li> </ul>
<i>glo_almanac_sv_msk</i>	<ul style="list-style-type: none"> <li>GLONASS <b>SV</b> mask for almanac; if the bit is set, almanac for that <b>SV</b> is available.</li> </ul>
<i>glo_health_sv_msk</i>	<ul style="list-style-type: none"> <li>GLONASS <b>SV</b> mask for health; if the bit is set, health for that <b>SV</b> is available.</li> </ul>
<i>glo_visible_sv_msk</i>	<ul style="list-style-type: none"> <li>GLONASS <b>SV</b> mask for visible SVs; if the bit is set, the <b>SV</b> is available.</li> </ul>
<i>sbas_ephemeris_sv_msk</i>	<ul style="list-style-type: none"> <li>SBAS <b>SV</b> mask for ephemeris; if the bit is set, ephemeris for that <b>SV</b> is available.</li> </ul>
<i>sbas_almanac_sv_msk</i>	<ul style="list-style-type: none"> <li>SBAS <b>SV</b> mask for almanac; if the bit is set, almanac for that <b>SV</b> is available.</li> </ul>
<i>sbas_health_sv_msk</i>	<ul style="list-style-type: none"> <li>SBAS <b>SV</b> mask for health; if the bit is set, health for that <b>SV</b> is available.</li> </ul>
<i>sbas_visible_sv_msk</i>	<ul style="list-style-type: none"> <li>SBAS <b>SV</b> mask for visible SVs; if the bit is set, the <b>SV</b> is available.</li> </ul>
<i>xtra_start_gps_week</i>	<ul style="list-style-type: none"> <li>Current XTRA information is valid starting from this GPS week number</li> </ul>
<i>xtra_start_gps_minutes</i>	<ul style="list-style-type: none"> <li>Current XTRA information is valid starting from the GPS minutes with the GPS week</li> </ul>
<i>xtra_valid_duration_hours</i>	<ul style="list-style-type: none"> <li>XTRA information is valid for this many hours starting from the specified GPS week/minutes</li> </ul>

## 8.193.2 Field Documentation

### 8.193.2.1 ULONG GPSSateInfo::Altitude

- 8.193.2.2 **BYTE** GPSSStateInfo::EngineState
- 8.193.2.3 **ULONG** GPSSStateInfo::glo\_almanac\_sv\_msk
- 8.193.2.4 **ULONG** GPSSStateInfo::glo\_ephemeris\_sv\_msk
- 8.193.2.5 **ULONG** GPSSStateInfo::glo\_health\_sv\_msk
- 8.193.2.6 **ULONG** GPSSStateInfo::glo\_visible\_sv\_msk
- 8.193.2.7 **ULONG** GPSSStateInfo::gps\_almanac\_sv\_msk
- 8.193.2.8 **ULONG** GPSSStateInfo::gps\_ephemeris\_sv\_msk
- 8.193.2.9 **ULONG** GPSSStateInfo::gps\_health\_sv\_msk
- 8.193.2.10 **ULONG** GPSSStateInfo::gps\_visible\_sv\_msk
- 8.193.2.11 **ULONG** GPSSStateInfo::HorizontalUncertainty
- 8.193.2.12 **BYTE** GPSSStateInfo::lono\_valid
- 8.193.2.13 **ULONGLONG** GPSSStateInfo::Latitude
- 8.193.2.14 **ULONGLONG** GPSSStateInfo::Longitude
- 8.193.2.15 **ULONG** GPSSStateInfo::sbas\_almanac\_sv\_msk
- 8.193.2.16 **ULONG** GPSSStateInfo::sbas\_ephemeris\_sv\_msk
- 8.193.2.17 **ULONG** GPSSStateInfo::sbas\_health\_sv\_msk
- 8.193.2.18 **ULONG** GPSSStateInfo::sbas\_visible\_sv\_msk
- 8.193.2.19 **ULONG** GPSSStateInfo::Time\_uncert\_ms
- 8.193.2.20 **WORD** GPSSStateInfo::TimeStmp\_gps\_week
- 8.193.2.21 **ULONG** GPSSStateInfo::TimeStmp\_tow\_ms
- 8.193.2.22 **ULONG** GPSSStateInfo::ValidMask
- 8.193.2.23 **ULONG** GPSSStateInfo::VerticalUncertainty
- 8.193.2.24 **WORD** GPSSStateInfo::xtra\_start\_gps\_minutes
- 8.193.2.25 **WORD** GPSSStateInfo::xtra\_start\_gps\_week
- 8.193.2.26 **WORD** GPSSStateInfo::xtra\_valid\_duration\_hours

## 8.194 gpsTime\_s Struct Reference

### Data Fields

- [WORD](#) `gpsWeek`
- [ULONG](#) `gpsTimeOfWeekMs`

### 8.194.1 Detailed Description

This structure contains GPS Time info.

#### Parameters

<i>gpsWeek</i>	<ul style="list-style-type: none"> <li>• Current GPS week as calculated from midnight, Jan. 6, 1980.</li> <li>• Units - Weeks</li> </ul>
<i>gpsTimeOf-WeekMs</i>	<ul style="list-style-type: none"> <li>• Amount of time into the current GPS week.</li> <li>• Units - Milliseconds</li> </ul>

### 8.194.2 Field Documentation

8.194.2.1 **ULONG** *gpsTime\_s::gpsTimeOfWeekMs*

8.194.2.2 **WORD** *gpsTime\_s::gpsWeek*

## 8.195 gsmCellInfo Struct Reference

### Data Fields

- [WORD](#) *arfcn*
- [BYTE](#) *band1900*
- [BYTE](#) *cellIdValid*
- [BYTE](#) *bsicId*
- [SHORT](#) *rsi*
- [SHORT](#) *srxlev*

### 8.195.1 Detailed Description

This structure contains information about the GSM Cell.

#### Parameters

<i>arfcn</i>	<ul style="list-style-type: none"> <li>• GSM frequency being reported.</li> <li>• Range: 0 to 1023.</li> </ul>
<i>band1900</i>	<ul style="list-style-type: none"> <li>• Band indicator for the GSM ARFCN</li> <li>• This field is only valid if <i>arfcn</i> is in the overlapping region.</li> <li>• If TRUE and the cell is in the overlapping region, the ARFCN is on the 1900 band.</li> <li>• If FALSE, it is on the 1800 band.</li> </ul>



<i>cellIdValid</i>	<ul style="list-style-type: none"> <li>Flag indicating whether the base station identity code ID is valid.</li> </ul>
<i>bsicId</i>	<ul style="list-style-type: none"> <li>Base station identity code ID, including base station color code and network color code.</li> <li>The lower 6 bits can be set to any value.</li> </ul>
<i>rsSI</i>	<ul style="list-style-type: none"> <li>Measured RSSI value in 1/10 dB.</li> <li>Range: -200.0 dB to 0</li> </ul>
<i>srxlev</i>	<ul style="list-style-type: none"> <li>Cell selection Rx level (Srxlev) value.</li> <li>Range: -128 to 128.</li> <li>This field is only valid when ue_in_idle is TRUE.</li> </ul>

## 8.195.2 Field Documentation

8.195.2.1 WORD gsmCellInfo::arfcn

8.195.2.2 BYTE gsmCellInfo::band1900

8.195.2.3 BYTE gsmCellInfo::bsicId

8.195.2.4 BYTE gsmCellInfo::cellIdValid

8.195.2.5 SHORT gsmCellInfo::rsSI

8.195.2.6 SHORT gsmCellInfo::srxlev

## 8.196 GSMRSSIThresh Struct Reference

### Data Fields

- [BYTE GSMRSSIThreshListLen](#)
- [WORD \\* pGSMRSSIThreshList](#)

### 8.196.1 Detailed Description

This structure contains GSM RSSI threshold related parameters.

#### Parameters

<i>GSMRSSI- ThreshListLen</i>	<ul style="list-style-type: none"> <li>Length of the GSM RSSI threshold list parameter to follow</li> </ul>
-----------------------------------	---

<i>pGSMRSSI- ThreshList</i>	<ul style="list-style-type: none"> <li>• Array of RSSI thresholds (in units of 0.1 dBm)</li> <li>• Maximum of 32 values</li> <li>• Range for RSSI values: -111 to -48 (in dBm)</li> </ul>
---------------------------------	---

## 8.196.2 Field Documentation

8.196.2.1 **BYTE** GSMRSSIthresh::GSMRSSIthreshListLen

8.196.2.2 **WORD\*** GSMRSSIthresh::pGSMRSSIthreshList

## 8.197 GSMSrvStatusInfo Struct Reference

### Data Fields

- [BYTE srvStatus](#)
- [BYTE trueSrvStatus](#)
- [BYTE isPrefDataPath](#)

### 8.197.1 Detailed Description

Structure for storing the service status information for GSM, WCDMA and LTE networks.

#### Parameters

<i>srvStatus</i>	<ul style="list-style-type: none"> <li>• Service status of the system. <ul style="list-style-type: none"> <li>– 0x00 - No service</li> <li>– 0x01 - Limited service</li> <li>– 0x02 - Service</li> <li>– 0x03 - Limited regional service</li> <li>– 0x04 - Power save</li> <li>– 0xFF - Not Available</li> </ul> </li> </ul>
------------------	--

<i>trueSrvStatus</i>	<ul style="list-style-type: none"> <li>• True service status of the system.</li> <li>• Not applicable to CDMA/HDR. <ul style="list-style-type: none"> <li>– 0x00 - No service</li> <li>– 0x01 - Limited service</li> <li>– 0x02 - Service</li> <li>– 0x03 - Limited regional service</li> <li>– 0x04 - Power save</li> <li>– 0xFF - Not Available</li> </ul> </li> </ul>
<i>isPrefDataPath</i>	<ul style="list-style-type: none"> <li>• Whether the RAT is the preferred data path. <ul style="list-style-type: none"> <li>– 0x00 - Not preferred</li> <li>– 0x01 - Preferred</li> <li>– 0xFF - Not Available</li> </ul> </li> </ul>

## 8.197.2 Field Documentation

8.197.2.1 **BYTE** GSMSrvStatusInfo::isPrefDataPath

8.197.2.2 **BYTE** GSMSrvStatusInfo::srvStatus

8.197.2.3 **BYTE** GSMSrvStatusInfo::trueSrvStatus

## 8.198 GSMSysInfo Struct Reference

### Data Fields

- [sysInfoCommon sysInfoGSM](#)
- [BYTE lacValid](#)
- [WORD lac](#)
- [BYTE cellIdValid](#)
- [ULONG cellId](#)
- [BYTE regRejectInfoValid](#)
- [BYTE rejectSrvDomain](#)
- [BYTE rejCause](#)
- [BYTE networkIdValid](#)
- [BYTE MCC \[3\]](#)
- [BYTE MNC \[3\]](#)
- [BYTE egprsSuppValid](#)
- [BYTE egprsSupp](#)
- [BYTE dtmSuppValid](#)
- [BYTE dtmSupp](#)

### 8.198.1 Detailed Description

Structure for storing the GSM System Information.

## Parameters

<i>sysInfoGSM</i>	<ul style="list-style-type: none"> <li>• See <a href="#">sysInfoCommon</a> for more information.</li> </ul>
<i>lacValid</i>	<ul style="list-style-type: none"> <li>• Indicates whether the location area code is valid.. <ul style="list-style-type: none"> <li>– 0x00 - Invalid</li> <li>– 0x01 - Valid</li> <li>– 0xFF - Not Available</li> </ul> </li> </ul>
<i>lac</i>	<ul style="list-style-type: none"> <li>• Location area code.</li> <li>• Only applies to 3GPP. <ul style="list-style-type: none"> <li>– 0xFFFF - Not Available</li> </ul> </li> </ul>
<i>cellIdValid</i>	<ul style="list-style-type: none"> <li>• Indicates whether the cell ID is valid. <ul style="list-style-type: none"> <li>– 0x00 - Invalid</li> <li>– 0x01 - Valid</li> <li>– 0xFF - Not Available</li> </ul> </li> </ul>
<i>cellId</i>	<ul style="list-style-type: none"> <li>• Cell ID. <ul style="list-style-type: none"> <li>– 0xFFFFFFFF - Not Available</li> </ul> </li> </ul>
<i>regRejectInfo-Valid</i>	<ul style="list-style-type: none"> <li>• Indicates whether the registration reject information is valid. <ul style="list-style-type: none"> <li>– 0x00 - Invalid</li> <li>– 0x01 - Valid</li> <li>– 0xFF - Not Available</li> </ul> </li> </ul>
<i>rejectSrvDomain</i>	<ul style="list-style-type: none"> <li>• Type of service domain in which the registration is rejected. <ul style="list-style-type: none"> <li>– 0x00 - SYS_SRV_DOMAIN_NO_SRV - No service</li> <li>– 0x01 - Circuit-switched only</li> <li>– 0x02 - Packet-switched only</li> <li>– 0x03 - Circuit-switched and packet-switched</li> <li>– 0x04 - Camped</li> <li>– 0xFF - Not Available</li> </ul> </li> </ul>
<i>rejCause</i>	<ul style="list-style-type: none"> <li>• Reject cause values sent are specified in [3GPP TS 24.008, Section 10.5.3.6]. <ul style="list-style-type: none"> <li>– 0xFF - Not Available</li> </ul> </li> </ul>
	Generated on Wed Oct 28 2015 00:40:12 for LinuxQMI SDK by Doxygen

<i>networkIdValid</i>	<ul style="list-style-type: none"> <li>Indicates whether the network ID is valid. <ul style="list-style-type: none"> <li>0x00 - Invalid</li> <li>0x01 - Valid</li> <li>0xFF - Not Available</li> </ul> </li> </ul>
<i>MCC[PLMN_LE-NGTH]</i>	<ul style="list-style-type: none"> <li>Mobile Country Code.</li> <li>MCC digits in ASCII characters</li> </ul>
<i>MNC[PLMN_LE-NGTH]</i>	<ul style="list-style-type: none"> <li>Mobile Network Code.</li> <li>MNC digits in ASCII characters</li> <li>An unused byte is set to 0xFF.</li> </ul>
<i>egprsSuppValid</i>	<ul style="list-style-type: none"> <li>Indicates whether the EGPRS support is valid. <ul style="list-style-type: none"> <li>0x00 - Invalid</li> <li>0x01 - Valid</li> <li>0xFF - Not Available</li> </ul> </li> </ul>
<i>egprsSupp</i>	<ul style="list-style-type: none"> <li>EGPRS support indication.</li> <li>Only applicable for GSM. <ul style="list-style-type: none"> <li>0x00 - Not available</li> <li>0x01 - Available</li> <li>0xFF - Not Available</li> </ul> </li> </ul>
<i>dtmSuppValid</i>	<ul style="list-style-type: none"> <li>Indicates whether Dual Transfer mode support is valid. <ul style="list-style-type: none"> <li>0x00 - Invalid</li> <li>0x01 - Valid</li> <li>0xFF - Not Available</li> </ul> </li> </ul>
<i>dtmSupp</i>	<ul style="list-style-type: none"> <li>Dual Transfer mode support indication.</li> <li>Only applicable for GSM. <ul style="list-style-type: none"> <li>0x00 - Not available</li> <li>0x01 - Available</li> <li>0xFF - Not Available</li> </ul> </li> </ul>

## 8.198.2 Field Documentation

- 8.198.2.1 **ULONG** GSMSysInfo::cellId
- 8.198.2.2 **BYTE** GSMSysInfo::cellIdValid
- 8.198.2.3 **BYTE** GSMSysInfo::dtmSupp
- 8.198.2.4 **BYTE** GSMSysInfo::dtmSuppValid
- 8.198.2.5 **BYTE** GSMSysInfo::egprsSupp
- 8.198.2.6 **BYTE** GSMSysInfo::egprsSuppValid
- 8.198.2.7 **WORD** GSMSysInfo::lac
- 8.198.2.8 **BYTE** GSMSysInfo::lacValid
- 8.198.2.9 **BYTE** GSMSysInfo::MCC[3]
- 8.198.2.10 **BYTE** GSMSysInfo::MNC[3]
- 8.198.2.11 **BYTE** GSMSysInfo::networkIdValid
- 8.198.2.12 **BYTE** GSMSysInfo::regRejectInfoValid
- 8.198.2.13 **BYTE** GSMSysInfo::rejCause
- 8.198.2.14 **BYTE** GSMSysInfo::rejectSrvDomain
- 8.198.2.15 **sysInfoCommon** GSMSysInfo::sysInfoGSM

## 8.199 gyroAcceptReady\_s Struct Reference

### Data Fields

- [BYTE injectEnable](#)
- [WORD samplesPerBatch](#)
- [WORD batchPerSec](#)

### 8.199.1 Detailed Description

This structure contains Gyroscope Accept Ready Info

## Parameters

<i>injectEnable</i>	<ul style="list-style-type: none"> <li>GNSS location engine is ready to accept data from sensor.</li> <li>Values</li> <li>0x01 - Ready to accept sensor data</li> <li>0x00 - Not ready to accept sensor data</li> </ul>
<i>samplesPerBatch</i>	<ul style="list-style-type: none"> <li>number of samples per batch the GNSS location engine is to receive.</li> <li>samplingFrequency = samplesPerBatch * batchesPerSecond</li> <li>samplesPerBatch must be a nonzero positive value.</li> </ul>
<i>batchPerSec</i>	<ul style="list-style-type: none"> <li>LTE NAS version minor</li> <li>Number of sensor-data batches the GNSS location engine is to receive per second.</li> <li>BatchesPerSecond must be a nonzero positive value.</li> </ul>

## 8.199.2 Field Documentation

8.199.2.1 WORD gyroAcceptReady\_s::batchPerSec

8.199.2.2 BYTE gyroAcceptReady\_s::injectEnable

8.199.2.3 WORD gyroAcceptReady\_s::samplesPerBatch

## 8.200 gyroTempAcceptReady\_s Struct Reference

## Data Fields

- [BYTE injectEnable](#)
- [WORD samplesPerBatch](#)
- [WORD batchPerSec](#)

## 8.200.1 Detailed Description

This structure contains Gyroscope Temperature Accept Ready Info

## Parameters

<i>injectEnable</i>	<ul style="list-style-type: none"> <li>GNSS location engine is ready to accept data from sensor.</li> <li>Values</li> <li>0x01 - Ready to accept sensor data</li> <li>0x00 - Not ready to accept sensor data</li> </ul>
<i>samplesPerBatch</i>	<ul style="list-style-type: none"> <li>number of samples per batch the GNSS location engine is to receive.</li> <li>samplingFrequency = samplesPerBatch * batchesPerSecond</li> <li>samplesPerBatch must be a nonzero positive value.</li> </ul>
<i>batchPerSec</i>	<ul style="list-style-type: none"> <li>LTE NAS version minor</li> <li>Number of sensor-data batches the GNSS location engine is to receive per second.</li> <li>BatchesPerSecond must be a nonzero positive value.</li> </ul>

## 8.200.2 Field Documentation

8.200.2.1 WORD gyroTempAcceptReady\_s::batchPerSec

8.200.2.2 BYTE gyroTempAcceptReady\_s::injectEnable

8.200.2.3 WORD gyroTempAcceptReady\_s::samplesPerBatch

## 8.201 HDRECIOThresh Struct Reference

## Data Fields

- BYTE HDRECIOThreshListLen
- WORD \* pHRECIOThreshList

## 8.201.1 Detailed Description

This structure contains HDR ECIO threshold related parameters.

## Parameters

<i>HDRECIO- ThreshListLen</i>	<ul style="list-style-type: none"> <li>Length of the HDR ECIO threshold list parameter to follow</li> </ul>
<i>pHDRECIO- ThreshList</i>	<ul style="list-style-type: none"> <li>Array of ECIO thresholds (in units of 0.1 dB)</li> <li>Maximum of 32 values</li> <li>Range for ECIO values: -31.5 to 0 (in dB).</li> </ul>



### 8.201.2 Field Documentation

8.201.2.1 **BYTE** HDRECIOTthresh::HDRECIOTthreshListLen

8.201.2.2 **WORD\*** HDRECIOTthresh::pHDRECIOTthreshList

## 8.202 HDRIOTthresh Struct Reference

### Data Fields

- [BYTE HDRIOTthreshListLen](#)
- [WORD \\* pHDRIOThreshList](#)

### 8.202.1 Detailed Description

This structure contains HDR IO threshold related parameters.

#### Parameters

<i>HDRIOTthresh- ListLen</i>	<ul style="list-style-type: none"> <li>• Length of the HDR IO threshold list parameter to follow</li> </ul>
<i>pHDRIOTthresh- List</i>	<ul style="list-style-type: none"> <li>• Array of IO thresholds (in units of 0.1 dBm)</li> <li>• Maximum of 32 values</li> <li>• Range for IO values: -128 to -13 (in dBm).</li> </ul>

### 8.202.2 Field Documentation

8.202.2.1 **BYTE** HDRIOTthresh::HDRIOTthreshListLen

8.202.2.2 **WORD\*** HDRIOTthresh::pHDRIOTthreshList

## 8.203 HDRPersonalityInd Struct Reference

### Data Fields

- [WORD \\* pCurrentPersonality](#)
- [BYTE \\* pPersonalityListLength](#)
- [protocolSubtypeElement \\* pProtocolSubtypeElement](#)

### 8.203.1 Field Documentation

8.203.1.1 **WORD\*** HDRPersonalityInd::pCurrentPersonality

8.203.1.2 **BYTE\*** HDRPersonalityInd::pPersonalityListLength

8.203.1.3 **protocolSubtypeElement\*** HDRPersonalityInd::pProtocolSubtypeElement

## 8.204 HDRPersonalityResp Struct Reference

### Data Fields

- [WORD](#) \* [pCurrentPersonality](#)
- [BYTE](#) \* [pPersonalityListLength](#)
- [protocolSubtypeElement](#) \* [pProtocolSubtypeElement](#)

### 8.204.1 Detailed Description

This structure contains information about the SLQSSwiGetHDRPersonality response parameters.

#### Parameters

<i>pCurrent-Personality[Out]</i>	<ul style="list-style-type: none"> <li>• Current active personality index.</li> </ul>
<i>pPersonalityList-Length[In/Out]</i>	<ul style="list-style-type: none"> <li>• Number of Personality Protocol Subtype contains in this response.</li> <li>• maximum input value is 3</li> </ul>
<i>pProtocol-Subtype-Element[Out]</i>	<ul style="list-style-type: none"> <li>• See <a href="#">protocolSubtypeElement</a> for more information.</li> </ul>

### 8.204.2 Field Documentation

8.204.2.1 [WORD](#)\* HDRPersonalityResp::pCurrentPersonality

8.204.2.2 [BYTE](#)\* HDRPersonalityResp::pPersonalityListLength

8.204.2.3 [protocolSubtypeElement](#)\* HDRPersonalityResp::pProtocolSubtypeElement

## 8.205 HDRProtSubtypResp Struct Reference

### Data Fields

- [WORD](#) \* [pCurrentPrsnlty](#)
- [BYTE](#) \* [pPersonalityListLength](#)
- [protocolSubtypeElement](#) \* [pProtoSubTypElmnt](#)
- [ULONGLONG](#) \* [pAppSubType](#)

### 8.205.1 Detailed Description

This structure contains information about the SLQSSwiGetHDRProtSubtype response parameters.

#### Parameters

<i>pCurrent-Personality[Out]</i>	<ul style="list-style-type: none"> <li>• Current active personality index.</li> </ul>
----------------------------------	---

<i>pPersonalityListLength</i> [In/Out]	<ul style="list-style-type: none"> <li>Number of Personality Protocol Subtype contains in this response.</li> <li>maximum input value is 4</li> </ul>
<i>pProtocolSubtypeElement</i> [Out]	<ul style="list-style-type: none"> <li>See <a href="#">protocolSubtypeElement</a> for more information.</li> </ul>
<i>pAppSubType</i> [Out]	<ul style="list-style-type: none"> <li>Stream application subtype</li> <li>Application subtype for each stream,</li> </ul>

## 8.205.2 Field Documentation

8.205.2.1 **ULONGLONG\*** HDRProtSubtypResp::pAppSubType

8.205.2.2 **WORD\*** HDRProtSubtypResp::pCurrentPrsnlty

8.205.2.3 **BYTE\*** HDRProtSubtypResp::pPersonalityListLength

8.205.2.4 **protocolSubtypeElement\*** HDRProtSubtypResp::pProtoSubTypeElmnt

## 8.206 HDRRSSIThresh Struct Reference

### Data Fields

- [BYTE HDRRSSIThreshListLen](#)
- [WORD \\* pHDRRSSIThreshList](#)

## 8.206.1 Detailed Description

This structure contains HDR RSSI threshold related parameters.

### Parameters

<i>HDRRSSIThreshListLen</i>	<ul style="list-style-type: none"> <li>Length of the HDR RSSI threshold list parameter to follow</li> </ul>
<i>pHDRRSSIThreshList</i>	<ul style="list-style-type: none"> <li>Array of RSSI thresholds (in units of 0.1 dBm)</li> <li>Maximum of 32 values.</li> <li>Range for RSSI values: -118 to -13 (in dBm).</li> </ul>

## 8.206.2 Field Documentation

8.206.2.1 **BYTE** HDRRSSIThresh::HDRRSSIThreshListLen

8.206.2.2 **WORD\*** HDRRSSIThresh::pHDRRSSIThreshList

## 8.207 HDRSINRThresh Struct Reference

### Data Fields

- [BYTE HDRSINRThresListLen](#)
- [BYTE \\* pHDRSINRThresList](#)

### 8.207.1 Detailed Description

This structure contains HDR SINR threshold related parameters.

#### Parameters

<i>HDRSINRThres- ListLen</i>	<ul style="list-style-type: none"> <li>• Length of the HDR SINR threshold list parameter to follow</li> </ul>
<i>pHDRSINR- ThresList</i>	<ul style="list-style-type: none"> <li>• Sequence of thresholds delimiting SINR event reporting bands</li> <li>• Every time a new SINR value crosses a threshold value, an event report indication message with the new SINR value is sent to the requesting control point. For this field               <ul style="list-style-type: none"> <li>– SINR is reported only for HDR</li> <li>– Each SINR threshold value is an unsigned 1 byte value</li> <li>– Maximum number of threshold values is 16</li> <li>– At least one value must be specified</li> </ul> </li> </ul>

### 8.207.2 Field Documentation

8.207.2.1 [BYTE HDRSINRThresh::HDRSINRThresListLen](#)

8.207.2.2 [BYTE\\* HDRSINRThresh::pHDRSINRThresList](#)

## 8.208 HDRSINRThreshold Struct Reference

### Data Fields

- [BYTE HDRSINRThreshListLen](#)
- [WORD \\* pHDRSINRThreshList](#)

### 8.208.1 Detailed Description

This structure contains HDR SINR threshold related parameters.

#### Parameters

<i>HDRSINR- ThreshListLen</i>	<ul style="list-style-type: none"> <li>• Length of the HDR ECIO threshold list parameter to follow</li> </ul>
-----------------------------------	---

<i>pHDRSINR- ThreshList</i>	<ul style="list-style-type: none"> <li>• Array of SINR level thresholds (in units of 1)</li> <li>• maximum of 32 values.</li> <li>• Valid levels are 0 to 8 <ul style="list-style-type: none"> <li>– 0x00 - SINR_LEVEL_0 is -9 dB</li> <li>– 0x01 - SINR_LEVEL_1 is -6 dB</li> <li>– 0x02 - SINR_LEVEL_2 is -4.5 dB</li> <li>– 0x03 - SINR_LEVEL_3 is -3 dB</li> <li>– 0x04 - SINR_LEVEL_4 is -2 dB</li> <li>– 0x05 - SINR_LEVEL_5 is +1 dB</li> <li>– 0x06 - SINR_LEVEL_6 is +3 dB</li> <li>– 0x07 - SINR_LEVEL_7 is +6 dB</li> <li>– 0x08 - SINR_LEVEL_8 is +9 dB</li> </ul> </li> </ul>
---------------------------------	--

## 8.208.2 Field Documentation

8.208.2.1 **BYTE** HDRSINRThreshold::HDRSINRThreshListLen

8.208.2.2 **WORD\*** HDRSINRThreshold::pHDRSINRThreshList

## 8.209 HDRSSInfo Struct Reference

### Data Fields

- [INT8](#) rssi
- [SHORT](#) ecio
- [BYTE](#) sinr
- [INT32](#) io

### 8.209.1 Detailed Description

This structure contains the parameters for HDR Signal Strength Information

#### Parameters

<i>rssi</i>	<ul style="list-style-type: none"> <li>• RSSI in dBm (signed value).</li> <li>• A value of -125 dBm or lower is used to indicate No Signal.</li> </ul>
<i>ecio</i>	<ul style="list-style-type: none"> <li>• ECIO value representing negative 0.5 dBm increments, i.e., 2 means -1 dBm (14 means -7 dBm, 63 means -31.5 dBm).</li> </ul>

<i>sinr</i>	<ul style="list-style-type: none"> <li>• SINR level.</li> <li>• SINR is only applicable for 1xEV-DO.</li> <li>• Valid levels are 0 to 8, where the maximum value for: <ul style="list-style-type: none"> <li>– 0 - SINR_LEVEL_0 is -9 dB</li> <li>– 1 - SINR_LEVEL_1 is -6 dB</li> <li>– 2 - SINR_LEVEL_2 is -4.5 dB</li> <li>– 3 - SINR_LEVEL_3 is -3 dB</li> <li>– 4 - SINR_LEVEL_4 is -2 dB</li> <li>– 5 - SINR_LEVEL_5 is +1 dB</li> <li>– 6 - SINR_LEVEL_6 is +3 dB</li> <li>– 7 - SINR_LEVEL_7 is +6 dB</li> <li>– 8 - SINR_LEVEL_8 is +9 dB</li> <li>– 0xFF - Not Available</li> </ul> </li> </ul>
<i>io</i>	<ul style="list-style-type: none"> <li>• Received IO in dBm.</li> <li>• IO is only applicable for 1xEV-DO.</li> </ul>

## 8.209.2 Field Documentation

### 8.209.2.1 SHORT HDRSSInfo::ecio

### 8.209.2.2 INT32 HDRSSInfo::io

### 8.209.2.3 INT8 HDRSSInfo::rssi

### 8.209.2.4 BYTE HDRSSInfo::sinr

## 8.210 HDRSysInfo Struct Reference

### Data Fields

- [sysInfoCommon sysInfoHDR](#)
- [BYTE isSysPrIMatchValid](#)
- [BYTE isSysPrIMatch](#)
- [BYTE hdrPersonalityValid](#)
- [BYTE hdrPersonality](#)
- [BYTE hdrActiveProtValid](#)
- [BYTE hdrActiveProt](#)
- [BYTE is856SysIdValid](#)
- [BYTE is856SysId \[16\]](#)

### 8.210.1 Detailed Description

Structure for storing the HDR System Information.

## Parameters

<i>sysInfoHDR</i>	<ul style="list-style-type: none"> <li>• See <a href="#">sysInfoCommon</a> for more information.</li> </ul>
<i>isSysPrIMatch-Valid</i>	<ul style="list-style-type: none"> <li>• Indicates whether the system PRL match is valid. <ul style="list-style-type: none"> <li>– 0x00 - Invalid</li> <li>– 0x01 - Valid</li> <li>– 0xFF - Not Available</li> </ul> </li> </ul>
<i>isSysPrIMatch</i>	<ul style="list-style-type: none"> <li>• Indicates whether the system is in a PRL.</li> <li>• Only applies to CDMA/HDR. <ul style="list-style-type: none"> <li>– 0x00 - System is not in a PRL</li> <li>– 0x01 - System is in a PRL</li> <li>– 0xFF - Not Available</li> </ul> </li> <li>• If the system is not in a PRL, roam_status carries the value from the default roaming indicator in the PRL.</li> <li>• If the system is in a PRL, roam_status is set to the value based on the standard specification.</li> </ul>
<i>hdrPersonality-Valid</i>	<ul style="list-style-type: none"> <li>• Indicates whether the HDR personality is valid. <ul style="list-style-type: none"> <li>– 0x00 - Invalid</li> <li>– 0x01 - Valid</li> <li>– 0xFF - Not Available</li> </ul> </li> </ul>
<i>hdrPersonality</i>	<ul style="list-style-type: none"> <li>• HDR personality information.</li> <li>• Only applicable for HDR. <ul style="list-style-type: none"> <li>– 0x00 - None</li> <li>– 0x02 - HRPD</li> <li>– 0x03 - eHRPD</li> <li>– 0xFF - Not Available</li> </ul> </li> </ul>

<i>hdrActiveProt-Valid</i>	<ul style="list-style-type: none"> <li>Indicates whether the HDR active protocol revision information is valid. <ul style="list-style-type: none"> <li>0x00 - Invalid</li> <li>0x01 - Valid</li> <li>0xFF - Not Available</li> </ul> </li> </ul>
<i>hdrActiveProt</i>	<ul style="list-style-type: none"> <li>HDR active protocol revision information .</li> <li>Only applicable for HDR. <ul style="list-style-type: none"> <li>0x00 - None</li> <li>0x02 - HDR Rel 0</li> <li>0x03 - HDR Rel A</li> <li>0x04 - HDR Rel B</li> <li>0xFF - Not Available</li> </ul> </li> </ul>
<i>is856SysIdValid</i>	<ul style="list-style-type: none"> <li>Indicates whether the IS-856 system ID is valid. <ul style="list-style-type: none"> <li>0x00 - Invalid</li> <li>0x01 - Valid</li> <li>0xFF - Not Available</li> </ul> </li> </ul>
<i>is856SysId[SLQ-S_SYSTEM_ID_SIZE]</i>	<ul style="list-style-type: none"> <li>IS-856 system ID.</li> <li>Only applicable for HDR.</li> </ul>

## 8.210.2 Field Documentation

8.210.2.1 **BYTE** HDRSysInfo::hdrActiveProt

8.210.2.2 **BYTE** HDRSysInfo::hdrActiveProtValid

8.210.2.3 **BYTE** HDRSysInfo::hdrPersonality

8.210.2.4 **BYTE** HDRSysInfo::hdrPersonalityValid

8.210.2.5 **BYTE** HDRSysInfo::is856SysId[16]

8.210.2.6 **BYTE** HDRSysInfo::is856SysIdValid

8.210.2.7 **BYTE** HDRSysInfo::isSysPrIMatch

8.210.2.8 **BYTE** HDRSysInfo::isSysPrIMatchValid

8.210.2.9 **sysInfoCommon** HDRSysInfo::sysInfoHDR



## 8.211 homeSIDNID Struct Reference

### Data Fields

- [BYTE numInstances](#)
- [sidNid SidNid](#) [255]

### 8.211.1 Detailed Description

This structure contains the parameters for Home SID/NID Information

#### Parameters

<i>numInstances</i>	<ul style="list-style-type: none"><li>• Number of sets of the following elements:<ul style="list-style-type: none"><li>– sid</li><li>– nid</li></ul></li><li>• If zero(0), then no information follows.</li></ul>
<i>SidNid</i>	<ul style="list-style-type: none"><li>• See <a href="#">sidNid</a> for more information</li></ul>

### 8.211.2 Field Documentation

8.211.2.1 **BYTE** homeSIDNID::numInstances

8.211.2.2 **sidNid** homeSIDNID::SidNid[255]

## 8.212 hotSwapStatus Struct Reference

### Data Fields

- [BYTE hotSwapLength](#)
- [BYTE hotSwap](#) [255]

### 8.212.1 Detailed Description

This structure contains Hot Swap Status Information.

## Parameters

<i>hotSwapLength</i>	<ul style="list-style-type: none"> <li>• Number of sets of the following elements. i.e. hot_swap</li> </ul>
<i>hotSwap[MAX_DESCRIPTION_LENGTH]</i>	<ul style="list-style-type: none"> <li>• Indicates the status of the hot-swap switch. <ul style="list-style-type: none"> <li>– 0 - Hot-swap is not supported</li> <li>– 1 - Hot-swap is supported, but the status of the switch is not supported</li> <li>– 2 - Switch indicates that the card is present</li> <li>– 3 - Switch indicates that the card is not present</li> </ul> </li> </ul>

## 8.212.2 Field Documentation

8.212.2.1 BYTE hotSwapStatus::hotSwap[255]

8.212.2.2 BYTE hotSwapStatus::hotSwapLength

## 8.213 ImageElement Struct Reference

## Data Fields

- [BYTE imageType](#)
- [BYTE imageId](#) [16]
- [BYTE buildIdLength](#)
- [CHAR buildId](#) [100]

## 8.213.1 Detailed Description

push current alignment to stack set alignment to 1 byte boundary This structure contains the Image Element information

## Parameters

<i>imageType</i>	<ul style="list-style-type: none"> <li>• Type of image 0 - Modem 1 - PRI</li> </ul>
<i>imageId</i>	<ul style="list-style-type: none"> <li>• Unique image identifier</li> </ul>
<i>buildIdLength</i>	<ul style="list-style-type: none"> <li>• Length of the build ID string (may be zero)</li> </ul>
<i>pBuildId</i>	<ul style="list-style-type: none"> <li>• Build ID ANSI string with length provided by the previous field</li> </ul>

## 8.213.2 Field Documentation

8.213.2.1 CHAR ImageElement::buildId[100]

8.213.2.2 BYTE ImageElement::buildIdLength

8.213.2.3 BYTE ImageElement::imageId[16]

8.213.2.4 BYTE ImageElement::imageType

## 8.214 ImageIdElement Struct Reference

### Data Fields

- [BYTE storageIndex](#)
- [BYTE failureCount](#)
- [BYTE imageID \[16\]](#)
- [BYTE buildIDLength](#)
- [CHAR buildID \[100\]](#)

### 8.214.1 Detailed Description

push current alignment to stack set alignment to 1 byte boundary This structure contains the Image ID list element Information

#### Parameters

<i>storageIndex</i>	<ul style="list-style-type: none"> <li>• Index in storage where the image is located(a value of 0xFF indicates that the storage for this type of image is not relevant)</li> </ul>
<i>failureCount</i>	<ul style="list-style-type: none"> <li>• Number of consecutive write attempts to this storage index that have failed(a value of 0xFF indicates unspecified)</li> </ul>
<i>imageID</i>	<ul style="list-style-type: none"> <li>• Image unique identifier(max 16 chars.)</li> </ul>
<i>buildIDLength</i>	<ul style="list-style-type: none"> <li>• Length of the build ID string. If there is no build ID, this field will be 0 and no data will follow.</li> </ul>
<i>buildID</i>	<ul style="list-style-type: none"> <li>• String containing image build information( Max 100 characters )</li> </ul>

### 8.214.2 Field Documentation

8.214.2.1 CHAR ImageIdElement::buildID[100]

8.214.2.2 BYTE ImageIdElement::buildIDLength

8.214.2.3 BYTE ImageIdElement::failureCount

8.214.2.4 BYTE ImageIdElement::imageID[16]

8.214.2.5 **BYTE** ImageIDElement::storageIndex

## 8.215 ImageIDEntries Struct Reference

### Data Fields

- [BYTE](#) imageType
- [BYTE](#) maxImages
- [BYTE](#) executingImage
- [BYTE](#) imageIDSize
- struct [ImageIDElement](#) imageIDElement [50]

### 8.215.1 Detailed Description

This structure contains the list entry Information

#### Parameters

<i>imageType</i>	<ul style="list-style-type: none"> <li>• Type of image <ul style="list-style-type: none"> <li>– 0 - Modem</li> <li>– 1 - PRI</li> </ul> </li> </ul>
<i>maxImages</i>	<ul style="list-style-type: none"> <li>• Maximum number of images of this type that may be stored concurrently on the device</li> </ul>
<i>executingImage</i>	<ul style="list-style-type: none"> <li>• Index (into the next array) of image that is currently executing</li> </ul>
<i>imageIDSize</i>	<ul style="list-style-type: none"> <li>• The number of elements in the image ID list</li> </ul>
<i>imageIDElement</i>	<ul style="list-style-type: none"> <li>• Array of ImageIDElement Structure ( Max 50 elements )</li> </ul>

### 8.215.2 Field Documentation

8.215.2.1 **BYTE** ImageIDEntries::executingImage

8.215.2.2 **struct** ImageIDElement ImageIDEntries::imageIDElement[50]

8.215.2.3 **BYTE** ImageIDEntries::imageIDSize

8.215.2.4 **BYTE** ImageIDEntries::imageType

8.215.2.5 **BYTE** ImageIDEntries::maxImages

## 8.216 ImageList Struct Reference

**Data Fields**

- [BYTE](#) *listSize*
- struct [ImageIDEntries](#) *imageIDEntries* [2]

**8.216.1 Detailed Description**

This structure contains the Get Stored Images List

**Parameters**

<i>listSize</i>	<ul style="list-style-type: none"> <li>• The number of elements in the image list</li> </ul>
<i>imageIDEntries</i>	<ul style="list-style-type: none"> <li>• Array of <a href="#">ImageIDEntries</a> Structure ( Max 2 entries )</li> </ul>

**8.216.2 Field Documentation**

8.216.2.1 struct [ImageIDEntries](#) *ImageList::imageIDEntries*[2]

8.216.2.2 **BYTE** *ImageList::listSize*

**8.217 IMSAIndRegisterInfo Struct Reference****Data Fields**

- **BYTE** \* [pRegStatusConfig](#)
- **BYTE** \* [pServiceStatusConfig](#)
- **BYTE** \* [pRatHandoverStatusConfig](#)
- **BYTE** \* [pPdpStatusConfig](#)

**8.217.1 Detailed Description**

This structure contains parameters of IMSA Config Indication Register

**Parameters**

<i>pRegStatus-Config(optional)</i>	<ul style="list-style-type: none"> <li>• Register Indication For Registration status.</li> <li>• When this registration is enabled, the device learns of Registration status via the QMI- _IMSA_REGISTRATION_STATUS_IND indication. <ul style="list-style-type: none"> <li>– 0x00 - Disable</li> <li>– 0x01 - Enable</li> </ul> </li> </ul>
------------------------------------	---

<i>pServiceStatus-Config(optional)</i>	<ul style="list-style-type: none"> <li>• Register Indication For Service status Events.</li> <li>• When this registration is enabled, the device learns of Service status via the QMI_IMSA_SERVICE_STATUS_IND indication. <ul style="list-style-type: none"> <li>– 0x00 - Disable</li> <li>– 0x01 - Enable</li> </ul> </li> </ul>
<i>pRatHandover-Status-Config(optional)</i>	<ul style="list-style-type: none"> <li>• Registration Indication For RAT handover status.</li> <li>• When this registration is enabled, the device learns of RAT handover status via the QMI_IMSA_RAT_HANDOVER_STATUS_IND indication. <ul style="list-style-type: none"> <li>– 0x00 - Disable</li> <li>– 0x01 - Enable</li> </ul> </li> </ul>
<i>pPdpStatus-Config(optional)</i>	<ul style="list-style-type: none"> <li>• PDP Status Configuration. <ul style="list-style-type: none"> <li>– 0x00 - Disable</li> <li>– 0x01 - Enable</li> </ul> </li> </ul>

**Note**

One of the optional parameter is mandatory to be present in the request.

**8.217.2 Field Documentation**

8.217.2.1 **BYTE\*** IMSAIndRegisterInfo::pPdpStatusConfig

8.217.2.2 **BYTE\*** IMSAIndRegisterInfo::pRatHandoverStatusConfig

8.217.2.3 **BYTE\*** IMSAIndRegisterInfo::pRegStatusConfig

8.217.2.4 **BYTE\*** IMSAIndRegisterInfo::pServiceStatusConfig

**8.218 imsaPdpStatusInfo Struct Reference****Data Fields**

- [BYTE](#) connetionState
- [ULONG \\*](#) pFailErrorCode

**8.218.1 Detailed Description**

Contains the parameters passed for SLQSSetIMSAPdpStatusCallback by the device.

## Parameters

<i>connetionState</i>	<ul style="list-style-type: none"> <li>IMS PDP connection state information.</li> <li>Values <ul style="list-style-type: none"> <li>TRUE – IMS PDP is connected</li> <li>FALSE – IMS PDP is not connected</li> </ul> </li> </ul>
<i>pFailErrorCode</i>	<ul style="list-style-type: none"> <li>IMS PDP connection failure error reason code when the IMS PDP Connection State TLV is FALSE.</li> <li>Values <ul style="list-style-type: none"> <li>0 - Generic failure reason for other than specified</li> <li>1 - Option is unsubscribed.</li> <li>2 - PDP status was unknown.</li> </ul> </li> </ul>

## 8.218.2 Field Documentation

8.218.2.1 **BYTE** imsaPdpStatusInfo::connetionState8.218.2.2 **ULONG\*** imsaPdpStatusInfo::pFailErrorCode

## 8.219 imsaRatStatusInfo Struct Reference

## Data Fields

- ULONG \*** pRATStatus
- ULONG \*** pSrcRAT
- ULONG \*** pTgtRAT
- BYTE \*** pErrorCodeStr

## 8.219.1 Detailed Description

Contains the parameters passed for SLQSSetIMSAStatusCallback by the device.

## Parameters

<i>pRATStatus</i>	<ul style="list-style-type: none"> <li>• RAT handover Status</li> </ul>
<i>pSrcRAT</i>	<ul style="list-style-type: none"> <li>• Source RAT</li> </ul>
<i>pTgtRAT</i>	<ul style="list-style-type: none"> <li>• Target RAT</li> </ul>
<i>pErrorCodeStr</i>	<ul style="list-style-type: none"> <li>• Error Code String</li> </ul>

## 8.219.2 Field Documentation

8.219.2.1 **BYTE\*** `imsaRatStatusInfo::pErrorCodeStr`8.219.2.2 **ULONG\*** `imsaRatStatusInfo::pRATStatus`8.219.2.3 **ULONG\*** `imsaRatStatusInfo::pSrcRAT`8.219.2.4 **ULONG\*** `imsaRatStatusInfo::pTgtRAT`

## 8.220 IMSRegistrationStatus Struct Reference

## Data Fields

- **BYTE** \* `plmsRegStatus`
- **WORD** \* `plmsRegErrCode`
- **ULONG** \* `pNewImsRegStatus`

## 8.220.1 Detailed Description

This structure contains response parameters of registration status.

## Parameters

<i>plmsRegStatus</i>	<ul style="list-style-type: none"> <li>• IMS Registration Status (Deprecated).</li> <li>• Values <ul style="list-style-type: none"> <li>– TRUE - UE is registered on the IMS network</li> <li>– FALSE - UE is not registered on the IMS network</li> </ul> </li> </ul>
----------------------	--



<i>plmsRegErr-Code</i>	<ul style="list-style-type: none"> <li>• IMS Registration Error Code.</li> <li>• An error code is returned when the IMS registration status is IMSA_STATUS_NOT_REGISTERED. -Values <ul style="list-style-type: none"> <li>– 3xx – Redirection responses</li> <li>– 4xx – Client failure responses</li> <li>– 5xx – Server failure responses</li> <li>– 6xx – Global failure responses</li> </ul> </li> </ul>
<i>pNewImReg-Status</i>	<ul style="list-style-type: none"> <li>• New IMS Registration Status</li> <li>• Values <ul style="list-style-type: none"> <li>– 0 - Not registered for IMS</li> <li>– 1 - Registering for IMS</li> <li>– 2 - Registered for IMS</li> </ul> </li> </ul>

## 8.220.2 Field Documentation

8.220.2.1 WORD\* IMSARegistrationStatus::plmsRegErrCode

8.220.2.2 BYTE\* IMSARegistrationStatus::plmsRegStatus

8.220.2.3 ULONG\* IMSARegistrationStatus::pNewImRegStatus

## 8.221 imsaRegStatusInfo Struct Reference

### Data Fields

- [BYTE](#) \* [pbIMSRegistered](#)
- [WORD](#) \* [pRegStatusErrorCode](#)
- [ULONG](#) \* [plmsRegStatus](#)

### 8.221.1 Detailed Description

Contains the parameters passed for SLQSSetIMSARegStatusCallback by the device.

#### Parameters

<i>pbIMS-Registered</i>	<ul style="list-style-type: none"> <li>• TRUE/FALSE</li> </ul>
<i>pRegStatus-ErrorCode</i>	<ul style="list-style-type: none"> <li>• if IMSA_STATUS_NOT_REGISTERED. Values: 3xx – Redirection responses 4xx – Client failure responses 5xx – Server failure responses 6xx – Global failure responses</li> </ul>

<i>pImsRegStatus</i>	IMS registration status. Values: IMSA_STATUS_NOT_REGISTERED - 0 IMSA_STATUS_REGISTERING - 1 IMSA_STATUS_REGISTERED -2
----------------------	---

### 8.221.2 Field Documentation

8.221.2.1 **BYTE\*** *imsaRegStatusInfo::pbIMSRegistered*

8.221.2.2 **ULONG\*** *imsaRegStatusInfo::pImsRegStatus*

8.221.2.3 **WORD\*** *imsaRegStatusInfo::pRegStatusErrorCode*

## 8.222 IMSAServiceStatus Struct Reference

### Data Fields

- [ULONG \\* pSmsServiceStatus](#)
- [ULONG \\* pVoipServiceStatus](#)
- [ULONG \\* pVtServiceStatus](#)
- [ULONG \\* pSmsServiceRat](#)
- [ULONG \\* pVoipServiceRat](#)
- [ULONG \\* pVtServiceRat](#)
- [ULONG \\* pUtServiceStatus](#)
- [ULONG \\* pUtServiceRat](#)
- [ULONG \\* pVsServiceStatus](#)
- [ULONG \\* pVsServiceRat](#)

### 8.222.1 Detailed Description

This structure contains response parameters of service status for various IMS services.

#### Parameters

<i>pSmsServiceStatus</i>	<ul style="list-style-type: none"> <li>• SMS Service Status.</li> <li>• Values <ul style="list-style-type: none"> <li>– 0 - IMS SMS service is not available</li> <li>– 1 - IMS SMS is in limited service</li> <li>– 2 - IMS SMS is in full service</li> </ul> </li> </ul>
--------------------------	--

<i>pVoipServiceStatus</i>	<ul style="list-style-type: none"><li>• VoIP Service Status. -Values<ul style="list-style-type: none"><li>– 0 - IMS VoIP service is not available</li><li>– 2 - IMS VoIP is in full service</li></ul></li></ul>
<i>pVtServiceStatus</i>	<ul style="list-style-type: none"><li>• VT Service Status</li><li>• Values<ul style="list-style-type: none"><li>– 0 - IMS VT service is not available</li><li>– 2 - IMS VT is in full service</li></ul></li></ul>
<i>pSmsServiceRat</i>	<ul style="list-style-type: none"><li>• SMS service RAT</li><li>• Values<ul style="list-style-type: none"><li>– 0 - IMS service is registered on WLAN</li><li>– 1 - IMS service is registered on WWAN</li><li>– 2 - IMS service is registered on interworking WLAN</li></ul></li></ul>
<i>pVoipServiceRat</i>	<ul style="list-style-type: none"><li>• VoIP service RAT.</li><li>• Values<ul style="list-style-type: none"><li>– 0 - IMS service is registered on WLAN</li><li>– 1 - IMS service is registered on WWAN</li><li>– 2 - IMS service is registered on interworking WLAN</li></ul></li></ul>
<i>pVtServiceRat</i>	<ul style="list-style-type: none"><li>• VT service RAT.</li><li>• Values<ul style="list-style-type: none"><li>– 0 - IMS service is registered on WLAN</li><li>– 1 - IMS service is registered on WWAN</li><li>– 2 - IMS service is registered on interworking WLAN</li></ul></li></ul>

<i>pUtService-Status</i>	<ul style="list-style-type: none"> <li>• UT service Status.</li> <li>• Values <ul style="list-style-type: none"> <li>– 0 - IMS UT service is not available</li> <li>– 2 - IMS UT is in full service</li> </ul> </li> </ul>
<i>pUtServiceRat</i>	<ul style="list-style-type: none"> <li>• UT service RAT.</li> <li>• Values <ul style="list-style-type: none"> <li>– 0 - IMS service is registered on WLAN</li> <li>– 1 - IMS service is registered on WWAN</li> <li>– 2 - IMS service is registered on interworking WLAN</li> </ul> </li> </ul>
<i>pVsService-Status</i>	<ul style="list-style-type: none"> <li>• VS service Status.</li> <li>• Values <ul style="list-style-type: none"> <li>– 0 - IMS UT service is not available</li> <li>– 2 - IMS UT is in full service</li> </ul> </li> </ul>
<i>pVsServiceRat</i>	<ul style="list-style-type: none"> <li>• VS service RAT.</li> <li>• Values <ul style="list-style-type: none"> <li>– 0 - IMS service is registered on WLAN</li> <li>– 1 - IMS service is registered on WWAN</li> <li>– 2 - IMS service is registered on interworking WLAN</li> </ul> </li> </ul>

## 8.222.2 Field Documentation

8.222.2.1 **ULONG\*** IMSAServiceStatus::pSmsServiceRat

8.222.2.2 **ULONG\*** IMSAServiceStatus::pSmsServiceStatus

8.222.2.3 **ULONG\*** IMSAServiceStatus::pUtServiceRat

8.222.2.4 **ULONG\*** IMSAServiceStatus::pUtServiceStatus

8.222.2.5 **ULONG\*** IMSAServiceStatus::pVoipServiceRat

8.222.2.6 **ULONG\*** IMSAServiceStatus::pVoipServiceStatus

8.222.2.7 **ULONG\*** IMSAServiceStatus::pVsServiceRat

8.222.2.8 **ULONG\*** IMSAServiceStatus::pVsServiceStatus

8.222.2.9 `ULONG*` `IMSAServiceStatus::pVtServiceRat`

8.222.2.10 `ULONG*` `IMSAServiceStatus::pVtServiceStatus`

## 8.223 IMSASupportedFieldsResp Struct Reference

### Data Fields

- struct [ReqFieldsList](#) \* `pReqFieldsList`
- struct [RespFieldsList](#) \* `pRespFieldsList`
- struct [IndFieldsList](#) \* `pIndFieldsList`

### 8.223.1 Detailed Description

This structure contains response of supported fields by the currently running software.

#### Parameters

<i>pReqFieldsList</i>	<ul style="list-style-type: none"> <li>• List of Supported Request Fields.</li> <li>• See <a href="#">ReqFieldsList</a> for more information</li> </ul>
<i>pRespFieldsList</i>	<ul style="list-style-type: none"> <li>• List of Supported Request Fields.</li> <li>• See <a href="#">RespFieldsList</a> for more information</li> </ul>
<i>pIndFieldsList</i>	<ul style="list-style-type: none"> <li>• List of Supported Request Fields.</li> <li>• See <a href="#">IndFieldsList</a> for more information</li> </ul>

### 8.223.2 Field Documentation

8.223.2.1 `struct IndFieldsList*` `IMSASupportedFieldsResp::pIndFieldsList`

8.223.2.2 `struct ReqFieldsList*` `IMSASupportedFieldsResp::pReqFieldsList`

8.223.2.3 `struct RespFieldsList*` `IMSASupportedFieldsResp::pRespFieldsList`

## 8.224 IMSASupportedMsgInfo Struct Reference

### Data Fields

- struct [SupportedMsgList](#) \* `pSupportedMsgList`

### 8.224.1 Detailed Description

This structure contains Queries the set of messages implemented by the currently running software.

## Parameters

<i>pSupportedMsgList</i>	<ul style="list-style-type: none"> <li>List of Supported Messages.</li> <li>See <a href="#">SupportedMsgList</a> for more information</li> </ul>
--------------------------	--

## 8.224.2 Field Documentation

8.224.2.1 struct SupportedMsgList\* IMSASupportedMsgInfo::pSupportedMsgList

## 8.225 imsaSvcStatusInfo Struct Reference

## Data Fields

- [ULONG](#) \* [pSMSSvcStatus](#)
- [ULONG](#) \* [pVOIPSvcStatus](#)
- [ULONG](#) \* [pVTSvcStatus](#)
- [ULONG](#) \* [pSMSSvcRAT](#)
- [ULONG](#) \* [pVOIPSvcRAT](#)
- [ULONG](#) \* [pVTSvcRAT](#)
- [ULONG](#) \* [pUTSvcStatus](#)
- [ULONG](#) \* [pUTSvcRAT](#)

## 8.225.1 Detailed Description

Contains the parameters passed for SLQSSetIMSASvcStatusCallback by the device.

## Parameters

<i>pSMSSvcStatus</i>	IMS SMS NOT AVAILABLE - 0 IMS SMS LIMITED SERVICE - 1 IMS SMS FULL SERVICE - 2
<i>pVOIPSvcStatus</i>	IMS VOIP NOT AVAILABLE - 0 IMS SMS FULL SERVICE - 2
<i>pVTSvcStatus</i>	IMS VT NOT AVAILABLE - 0 IMS VT FULL SERVICE - 2
<i>pSMSSvcRAT</i>	IMS service is registered on WLAN - 0 IMS service is registered on WWAN - 1 IMS service is registered on interworking WLAN - 2
<i>pVOIPSvcRAT</i>	IMS service is registered on WLAN - 0 IMS service is registered on WWAN - 1 IMS service is registered on interworking WLAN - 2 *
<i>pVTSvcRAT</i>	IMS service is registered on WLAN - 0 IMS service is registered on WWAN - 1 IMS service is registered on interworking WLAN - 2
<i>pUTSvcStatus</i>	IMS VOIP NOT AVAILABLE - 0 IMS SMS FULL SERVICE - 2
<i>pUTSvcRAT</i>	IMS service is registered on WLAN - 0 IMS service is registered on WWAN - 1 IMS service is registered on interworking WLAN - 2

## 8.225.2 Field Documentation

8.225.2.1 [ULONG](#)\* imsaSvcStatusInfo::pSMSSvcRAT8.225.2.2 [ULONG](#)\* imsaSvcStatusInfo::pSMSSvcStatus8.225.2.3 [ULONG](#)\* imsaSvcStatusInfo::pUTSvcRAT8.225.2.4 [ULONG](#)\* imsaSvcStatusInfo::pUTSvcStatus

8.225.2.5 **ULONG\*** imsaSvcStatusInfo::pVOIPSvcRAT8.225.2.6 **ULONG\*** imsaSvcStatusInfo::pVOIPSvcStatus8.225.2.7 **ULONG\*** imsaSvcStatusInfo::pVTSvcRAT8.225.2.8 **ULONG\*** imsaSvcStatusInfo::pVTSvcStatus

## 8.226 imsCfgIndRegisterInfo Struct Reference

### Data Fields

- **BYTE \*** pSIPConfigEvents
- **BYTE \*** pRegMgrConfigEvents
- **BYTE \*** pSMSCConfigEvents
- **BYTE \*** pUserConfigEvents
- **BYTE \*** pVoIPConfigEvents

### 8.226.1 Detailed Description

This structure contains parameters of IMS Config Indication Register

#### Parameters

<i>pSIPConfig-Events(optional)</i>	<ul style="list-style-type: none"> <li>• Registration Indication For SIP Configuration Events.</li> <li>• When this registration is enabled, the device learns of SIP config events via the QMI_ - IMS_SIP_CONFIG_IND indication. <ul style="list-style-type: none"> <li>– 0x00 - Disable</li> <li>– 0x01 - Enable</li> </ul> </li> </ul>
<i>pRegMgrConfig-Events(optional)</i>	<ul style="list-style-type: none"> <li>• Registration Indication For Registration Manager Configuration Events.</li> <li>• When this registration is enabled, the device learns of Reg Mgr config events via the QMI_ IMS_REG_MGR_CONFIG_IND indication. <ul style="list-style-type: none"> <li>– 0x00 - Disable</li> <li>– 0x01 - Enable</li> </ul> </li> </ul>

<i>pSMSConfig-Events(optional)</i>	<ul style="list-style-type: none"> <li>• Registration Indication For SMS Configuration Events.</li> <li>• When this registration is enabled, the device learns of SMS config events via the QMI-IMS_SMS_CONFIG_IND indication. <ul style="list-style-type: none"> <li>– 0x00 - Disable</li> <li>– 0x01 - Enable</li> </ul> </li> </ul>
<i>pUserConfig-Events(optional)</i>	<ul style="list-style-type: none"> <li>• Registration Indication For User Configuration Events.</li> <li>• When this registration is enabled, the device learns of user config events via the QMI-IMS_USER_CONFIG_IND indication. <ul style="list-style-type: none"> <li>– 0x00 - Disable</li> <li>– 0x01 - Enable</li> </ul> </li> </ul>
<i>pVoIPConfig-Events(optional)</i>	<ul style="list-style-type: none"> <li>• Registration Indication For VoIP Configuration Events.</li> <li>• When this registration is enabled, the device learns of VOIP config events via the QMI-IMS_VOIP_CONFIG_IND indication. <ul style="list-style-type: none"> <li>– 0x00 - Disable</li> <li>– 0x01 - Enable</li> </ul> </li> </ul>

**Note**

One of the optional parameter is mandatory to be present in the request.

**8.226.2 Field Documentation**

8.226.2.1 **BYTE\*** imsCfgIndRegisterInfo::pRegMgrConfigEvents

8.226.2.2 **BYTE\*** imsCfgIndRegisterInfo::pSIPConfigEvents

8.226.2.3 **BYTE\*** imsCfgIndRegisterInfo::pSMSConfigEvents

8.226.2.4 **BYTE\*** imsCfgIndRegisterInfo::pUserConfigEvents

8.226.2.5 **BYTE\*** imsCfgIndRegisterInfo::pVoIPConfigEvents

**8.227 imsRegMgrConfigInfo Struct Reference****Data Fields**

- **WORD \*** pPriCSCFPort
- **BYTE \*** pCSCFPortName
- **BYTE \*** pIMSTestMode



### 8.227.1 Detailed Description

Contains the parameters passed for SLQSSetRegMgrConfigCallback by the device.

## Parameters

<i>pPriCSCFPort</i>	<ul style="list-style-type: none"> <li>Primary call session control function port</li> </ul>
<i>pCSCFPort-Name</i>	<ul style="list-style-type: none"> <li>Call Session control port, fully qualified domain name</li> <li>Length of this string can be of maximum 255 bytes</li> </ul>
<i>pIMSTestMode</i>	<ul style="list-style-type: none"> <li>IMS Test mode Enabled. <ul style="list-style-type: none"> <li>TRUE - Enable, no IMS registration</li> <li>FALSE - Disable, IMS registration is initiated</li> </ul> </li> </ul>

## 8.227.2 Field Documentation

8.227.2.1 **BYTE\*** imsRegMgrConfigInfo::pCSCFPortName8.227.2.2 **BYTE\*** imsRegMgrConfigInfo::pIMSTestMode8.227.2.3 **WORD\*** imsRegMgrConfigInfo::pPriCSCFPort

## 8.228 imsSIPConfigInfo Struct Reference

## Data Fields

- WORD \*** pSIPLocalPort
- ULONG \*** pTimerSIPReg
- ULONG \*** pSubscribeTimer
- ULONG \*** pTimerT1
- ULONG \*** pTimerT2
- ULONG \*** pTimerTf
- BYTE \*** pSigCompEnabled

## 8.228.1 Detailed Description

Contains the parameters passed for SLQSSetSIPConfigCallback by the device.

## Parameters

<i>pSIPLocalPort</i>	<ul style="list-style-type: none"> <li>Primary call session control function SIP port number</li> </ul>
<i>pTimerSIPReg</i>	<ul style="list-style-type: none"> <li>Initial SIP registration duration from the User equipment, in seconds</li> </ul>

<i>pSubscribeTimer</i>	<ul style="list-style-type: none"> <li>Duration of the subscription by the UE for IMS registration notifications, in seconds</li> </ul>
<i>pTimerT1</i>	<ul style="list-style-type: none"> <li>RTT estimate, in milliseconds</li> </ul>
<i>pTimerT2</i>	<ul style="list-style-type: none"> <li>The maximum retransmit interval for non-invite requests and invite responses, in milliseconds</li> </ul>
<i>pTimerTf</i>	<ul style="list-style-type: none"> <li>Non-invite transaction timeout timer, in milliseconds</li> </ul>
<i>pSigCompEnabled</i>	<ul style="list-style-type: none"> <li>Sig Comp Status <ul style="list-style-type: none"> <li>TRUE - Enable</li> <li>FALSE - Disable</li> </ul> </li> </ul>

**Note**

None

**8.228.2 Field Documentation****8.228.2.1** **BYTE\*** **imsSIPConfigInfo::pSigCompEnabled****8.228.2.2** **WORD\*** **imsSIPConfigInfo::pSIPLocalPort****8.228.2.3** **ULONG\*** **imsSIPConfigInfo::pSubscribeTimer****8.228.2.4** **ULONG\*** **imsSIPConfigInfo::pTimerSIPReg****8.228.2.5** **ULONG\*** **imsSIPConfigInfo::pTimerT1****8.228.2.6** **ULONG\*** **imsSIPConfigInfo::pTimerT2****8.228.2.7** **ULONG\*** **imsSIPConfigInfo::pTimerTf****8.229 imsSMSConfigInfo Struct Reference****Data Fields**

- [BYTE \\*](#) [pSMSFormat](#)
- [BYTE \\*](#) [pSMSOverIPNwInd](#)
- [BYTE \\*](#) [pPhoneCtxtURI](#)

**8.229.1 Detailed Description**

Contains the parameters passed for SLQSSetIMSSMSConfigCallback by the device.

## Parameters

<i>pSMSFormat</i>	<ul style="list-style-type: none"> <li>• SMS format <ul style="list-style-type: none"> <li>– 0 - 3GPP</li> <li>– 1 - 3GPP2</li> </ul> </li> </ul>
<i>pSMSOverIPNwInd</i>	<ul style="list-style-type: none"> <li>• SMS over IP Network Indication Flag <ul style="list-style-type: none"> <li>– TRUE - Mobile-Originated(MO) SMS turned on</li> <li>– FALSE - MO SMS turned off</li> </ul> </li> </ul>
<i>pPhoneCtxtURI</i>	<ul style="list-style-type: none"> <li>• Phone context universal resource identifier</li> <li>• Length of this string can be of maximum 255 bytes</li> </ul>

## 8.229.2 Field Documentation

8.229.2.1 **BYTE\*** imsSMSConfigInfo::pPhoneCtxtURI8.229.2.2 **BYTE\*** imsSMSConfigInfo::pSMSFormat8.229.2.3 **BYTE\*** imsSMSConfigInfo::pSMSOverIPNwInd

## 8.230 imsUserConfigInfo Struct Reference

## Data Fields

- **BYTE \*** [pIMSDomain](#)

## 8.230.1 Detailed Description

Contains the parameters passed for SLQSSetIMSUserConfigCallback by the device.

## Parameters

<i>pIMSDomain</i>	<ul style="list-style-type: none"> <li>• IMS domain name</li> <li>• Length of this string can be of maximum 255 bytes</li> </ul>
-------------------	--

## 8.230.2 Field Documentation

8.230.2.1 **BYTE\*** imsUserConfigInfo::pIMSDomain

## 8.231 imsVoIPConfigInfo Struct Reference

## Data Fields

- WORD \* pSessionExpiryTimer
- WORD \* pMinSessionExpiryTimer
- BYTE \* pAmrWbEnable
- BYTE \* pScrAmrEnable
- BYTE \* pScrAmrWbEnable
- BYTE \* pAmrMode
- WORD \* pAmrWBMode
- BYTE \* pAmrOctetAligned
- BYTE \* pAmrWBOctetAligned
- WORD \* pRingingTimer
- WORD \* pRingBackTimer
- WORD \* pRTPRTCPInactTimer

## 8.231.1 Detailed Description

Contains the parameters passed for SLQSSetIMSVoIPConfigCallback by the device.

## Parameters

<i>pSessionExpiryTimer</i>	<ul style="list-style-type: none"> <li>• Session duration, in seconds</li> </ul>
<i>pMinSessionExpiryTimer</i>	<ul style="list-style-type: none"> <li>• Minimum allowed value for session expiry timer, in seconds</li> </ul>
<i>pAmrWbEnable</i>	<ul style="list-style-type: none"> <li>• Flag to enable/disable Adaptive Multirate Codec(AMR) WideBand(WB) audio</li> <li>• Values: <ul style="list-style-type: none"> <li>– True - Enabled</li> <li>– False - Disabled</li> </ul> </li> </ul>
<i>pScrAmrEnable</i>	<ul style="list-style-type: none"> <li>• Flag to enable/disable Source Control Rate(SCR) for AMR NarrowBand (NB)</li> <li>• Values: <ul style="list-style-type: none"> <li>– True - Enabled</li> <li>– False - Disabled</li> </ul> </li> </ul>
<i>pScrAmrWbEnable</i>	<ul style="list-style-type: none"> <li>• Flag to enable/disable SCR for AMR WB Audio</li> <li>• Values: <ul style="list-style-type: none"> <li>– True - Enabled</li> <li>– False - Disabled</li> </ul> </li> </ul>

<i>pAmrMode</i>	<ul style="list-style-type: none"> <li>• BitMask for AMR NB modes allowed</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0x1 - 4.75 kbps</li> <li>– 0x2 - 5.15 kbps</li> <li>– 0x4 - 5.9 kbps</li> <li>– 0x8 - 6.17 kbps</li> <li>– 0x10 - 7.4 kbps</li> <li>– 0x20 - 7.95 kbps</li> <li>– 0x40 - 10.2 kbps</li> <li>– 0x80 - 12.2 kbps</li> </ul> </li> </ul>
<i>pAmrWBMode</i>	<ul style="list-style-type: none"> <li>• BitMask for AMR WB modes allowed</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0x1 - 6.60 kbps</li> <li>– 0x2 - 8.85 kbps</li> <li>– 0x4 - 12.65 kbps</li> <li>– 0x8 - 14.25 kbps</li> <li>– 0x10 - 15.85 kbps</li> <li>– 0x20 - 18.25 kbps</li> <li>– 0x40 - 19.85 kbps</li> <li>– 0x80 - 23.05 kbps</li> <li>– 0x100 - 23.85 kbps</li> </ul> </li> </ul>
<i>pAmrOctet-Aligned</i>	<ul style="list-style-type: none"> <li>• Flag to indicate if the octet is aligned for AMR NB Audio</li> <li>• Values: <ul style="list-style-type: none"> <li>– True - Aligned</li> <li>– False - Not aligned, Bandwidth Efficient mode</li> </ul> </li> </ul>
<i>pAmrWBOctet-Aligned</i>	<ul style="list-style-type: none"> <li>• Flag to indicate if the octet is aligned for AMR WB Audio</li> <li>• Values: <ul style="list-style-type: none"> <li>– True - Aligned</li> <li>– False - Not aligned, Bandwidth Efficient mode</li> </ul> </li> </ul>

<i>pRingingTimer</i>	<ul style="list-style-type: none"> <li>Duration of ringing timer, in seconds. The ringing timer starts on the ringing event. If the call is not answered within the duration of this timer, the call is disconnected.</li> </ul>
<i>pRingBackTimer</i>	<ul style="list-style-type: none"> <li>Duration of ringback timer, in seconds. The ringback timer starts on the ringback event. If the call is not answered within the duration of this timer, the call is disconnected.</li> </ul>
<i>pRTPRTCP-InactTimer</i>	<ul style="list-style-type: none"> <li>Duration of RTP/RTCP inactivity timer, in seconds. If no RTP/RTCP packet is received prior to the expiry of this timer, the call is disconnected.</li> </ul>

## 8.231.2 Field Documentation

8.231.2.1 **BYTE\*** `imsVoIPConfigInfo::pAmrMode`

8.231.2.2 **BYTE\*** `imsVoIPConfigInfo::pAmrOctetAligned`

8.231.2.3 **BYTE\*** `imsVoIPConfigInfo::pAmrWbEnable`

8.231.2.4 **WORD\*** `imsVoIPConfigInfo::pAmrWBMode`

8.231.2.5 **BYTE\*** `imsVoIPConfigInfo::pAmrWBOctetAligned`

8.231.2.6 **WORD\*** `imsVoIPConfigInfo::pMinSessionExpiryTimer`

8.231.2.7 **WORD\*** `imsVoIPConfigInfo::pRingBackTimer`

8.231.2.8 **WORD\*** `imsVoIPConfigInfo::pRingingTimer`

8.231.2.9 **WORD\*** `imsVoIPConfigInfo::pRTPRTCPInactTimer`

8.231.2.10 **BYTE\*** `imsVoIPConfigInfo::pScrAmrEnable`

8.231.2.11 **BYTE\*** `imsVoIPConfigInfo::pScrAmrWbEnable`

8.231.2.12 **WORD\*** `imsVoIPConfigInfo::pSessionExpiryTimer`

## 8.232 IndFieldsList Struct Reference

### Data Fields

- [BYTE](#) `indicationFieldsLen`
- [BYTE](#) `indicationFields` [256]

### 8.232.1 Detailed Description

This structure contains the Supported Indication Fields List Information

## Parameters

<i>indicationFieldsLen</i>	<ul style="list-style-type: none"> <li>• Number of sets of the indication fields.</li> </ul>
<i>indicationFields</i>	<ul style="list-style-type: none"> <li>• Describes which optional field IDs are supported in QMI indication.</li> <li>• Format is same as request field.</li> </ul>

## 8.232.2 Field Documentation

8.232.2.1 BYTE IndFieldsList::indicationFields[256]

8.232.2.2 BYTE IndFieldsList::indicationFieldsLen

## 8.233 infoInterFreq Struct Reference

## Data Fields

- [WORD earfcn](#)
- [BYTE threshXLow](#)
- [BYTE threshXHigh](#)
- [BYTE cell\\_resel\\_priority](#)
- [BYTE cells\\_len](#)
- [cellParams cellInterFreqParams](#) [255]

## 8.233.1 Detailed Description

This structure contains information about the inter-frequency.

## Parameters

<i>earfcn</i>	<ul style="list-style-type: none"> <li>• E-UTRA absolute radio frequency channel number of the serving cell.</li> <li>• Range: 0 to 65535.</li> </ul>
<i>threshXLow</i>	<ul style="list-style-type: none"> <li>• Cell Srxlev low threshold.</li> <li>• Range: 0 to 31.</li> <li>• When the serving cell does not exceed thresh_serving_low, the value of an evaluated cell must be smaller than this value to be considered for re-selection.</li> </ul>



<i>threshXHigh</i>	<ul style="list-style-type: none"> <li>• Cell Srxlev high threshold.</li> <li>• Range: 0 to 31.</li> <li>• When the serving cell exceeds thresh_serving_low, the value of an evaluated cell must be greater than this value to be considered for re-selection.</li> </ul>
<i>cell_resel_priority</i>	<ul style="list-style-type: none"> <li>• Cell re-selection priority</li> <li>• Range: 0 to 7.</li> <li>• This field is only valid when ue_in_idle is TRUE.</li> </ul>
<i>cells_len</i>	<ul style="list-style-type: none"> <li>• Provides the number of set of cell params.</li> </ul>
<i>cellInterFreqParams[MAX_DESCRIPTION_LENGTH]</i>	<ul style="list-style-type: none"> <li>• See <a href="#">cellParams</a> for more information.</li> </ul>

### 8.233.2 Field Documentation

8.233.2.1 **BYTE** infoInterFreq::cell\_resel\_priority

8.233.2.2 **cellParams** infoInterFreq::cellInterFreqParams[255]

8.233.2.3 **BYTE** infoInterFreq::cells\_len

8.233.2.4 **WORD** infoInterFreq::earfcn

8.233.2.5 **BYTE** infoInterFreq::threshXHigh

8.233.2.6 **BYTE** infoInterFreq::threshXLow

## 8.234 IOTresh Struct Reference

### Data Fields

- [BYTE](#) IOTreshListLen
- [INT32](#) \* pIOTreshList

### 8.234.1 Detailed Description

This structure contains IO threshold related parameters.

#### Parameters

<i>IOTreshListLen</i>	<ul style="list-style-type: none"> <li>• Length of the LTE SNR threshold list parameter to follow</li> </ul>
-----------------------	--

<i>plIOThresList</i>	<ul style="list-style-type: none"> <li>• Sequence of thresholds delimiting IO event reporting bands</li> <li>• Every time a new IO value crosses a threshold value, an event report indication message with the new IO value is sent to the requesting control point. For this field <ul style="list-style-type: none"> <li>– IO is applicable only for HDR</li> <li>– Each IO threshold value is a signed 4 byte value</li> <li>– Maximum number of threshold values is 16</li> <li>– At least one value must be specified</li> </ul> </li> </ul>
----------------------	--

## 8.234.2 Field Documentation

8.234.2.1 BYTE IOThresh::IOThresListLen

8.234.2.2 INT32\* IOThresh::plIOThresList

## 8.235 IPv4Addr Struct Reference

### Data Fields

- [ULONG addr](#)
- [ULONG subnetMask](#)

### 8.235.1 Detailed Description

This structure contains the IPv4 filter address

#### Parameters

<i>addr</i>	IPv4 address
<i>subnetMask</i>	A packet matches if: <ul style="list-style-type: none"> <li>• (addr and subnetMask) == (IP pkt addr &amp; subnetMask) Callers to set up a filter with a range of source addresses, if needed; subnet mask of all 1s (255.255.255.255) specifies a single address value</li> </ul>

## 8.235.2 Field Documentation

8.235.2.1 ULONG IPv4Addr::addr

8.235.2.2 ULONG IPv4Addr::subnetMask

## 8.236 IPv6Addr Struct Reference

### Data Fields

- [BYTE addr](#) [16]
- [BYTE prefixLen](#)

### 8.236.1 Detailed Description

This structure contains the IPv6 filter address

## Parameters

<i>addr</i>	IPv6 address (in network byte order); this is a 16-byte byte array (in Big-endian format)
<i>prefixLen</i>	IPv6 filter prefix length; can take a value between 0 and 128 Note: A packet matches if the IPv6 source address bytes until the prefix lengths are equal. Therefore prefix length can be used to set a filter with a range of source addresses. A prefix length of 128 specifies a single address value.

## 8.236.2 Field Documentation

8.236.2.1 BYTE IPv6Addr::addr[16]

8.236.2.2 BYTE IPv6Addr::prefixLen

## 8.237 IPV6AddressInfo Struct Reference

## Data Fields

- [BYTE IPV6PrefixLen](#)
- [USHORT IPAddressV6](#) [8]

## 8.237.1 Detailed Description

This structure contains the IPV6 Address Information

## Parameters

<i>IPV6PrefixLen</i>	<ul style="list-style-type: none"> <li>Length of the received IPv6 address in no. of bits; can take value between 0 and 128 <ul style="list-style-type: none"> <li>– 0xFF - Not Available</li> </ul> </li> </ul>
<i>IPAddressV6</i>	<ul style="list-style-type: none"> <li>IPv6 address(in network byte order); This is an 8-element array of 16 bit numbers, each of which is in big endian format.</li> </ul>

## 8.237.2 Field Documentation

8.237.2.1 USHORT IPV6AddressInfo::IPAddressV6[8]

8.237.2.2 BYTE IPV6AddressInfo::IPV6PrefixLen

## 8.238 IPV6GWAddressInfo Struct Reference

## Data Fields

- [BYTE gwV6PrefixLen](#)
- [USHORT gwAddressV6](#) [8]

## 8.238.1 Detailed Description

This structure contains the IPV6 Gateway Address Information

## Parameters

<i>gwV6PrefixLen</i>	<ul style="list-style-type: none"> <li>Length of the received IPv6 Gateway address in no. of bits; can take value between 0 and 128</li> </ul>
<i>IPAddressV6</i>	<ul style="list-style-type: none"> <li>IPv6 Gateway address(in network byte order); This is an 8-element array of 16 bit numbers, each of which is in big endian format.</li> </ul>

## 8.238.2 Field Documentation

8.238.2.1 USHORT IPV6GWAddressInfo::gwAddressV6[8]

8.238.2.2 BYTE IPV6GWAddressInfo::gwV6PrefixLen

## 8.239 IPv6TrafCls Struct Reference

## Data Fields

- [BYTE val](#)
- [BYTE mask](#)

## 8.239.1 Detailed Description

This structure contains the IPv6 filter traffic class

## Parameters

<i>val</i>	The traffic class value
<i>mask</i>	<p>The packet matches the traffic class filter if: (IPv6_filter_traffic_class_val and IPv6_filter_traffic_class_mask) == (Traffic class value in the IP packet &amp; IPv6_filter_traffic_class_mask)</p> <p>Example:</p> <ul style="list-style-type: none"> <li>IPv6_filter_tc_val = 00101000</li> <li>IPv6_filter_tc_mask = 11111100 Filter will compare only the first 6 bits in IPv6_filter_traffic_class with the first 6 bits in the traffic class field of the IP packet; first 6 bits in the traffic class field of the IP packet must be 001010 to match filter; last 2 bits can be anything, since they are ignored by filtering</li> </ul>

## 8.239.2 Field Documentation

8.239.2.1 BYTE IPv6TrafCls::mask

8.239.2.2 BYTE IPv6TrafCls::val

## 8.240 lineCtrlInfo Struct Reference

## Data Fields

- [BYTE polarityIncluded](#)

- [BYTE toggleMode](#)
- [BYTE revPolarity](#)
- [BYTE pwrDenialTime](#)

### 8.240.1 Detailed Description

This structure contains Line Control Information

#### Parameters

<i>polarityIncluded</i>	<ul style="list-style-type: none"><li>• Included Polarity; Boolean Value</li></ul>
<i>toggleMode</i>	<ul style="list-style-type: none"><li>• Toggle mode; Boolean Value</li></ul>
<i>revPolarity</i>	<ul style="list-style-type: none"><li>• Reverse Polarity; Boolean Value</li></ul>
<i>pwrDenialTime</i>	<ul style="list-style-type: none"><li>• Power denial time; refer to [S1, Section 3.7.5.15 Line Control] for valid values</li></ul>

### 8.240.2 Field Documentation

8.240.2.1 [BYTE lineCtrlInfo::polarityIncluded](#)

8.240.2.2 [BYTE lineCtrlInfo::pwrDenialTime](#)

8.240.2.3 [BYTE lineCtrlInfo::revPolarity](#)

8.240.2.4 [BYTE lineCtrlInfo::toggleMode](#)

## 8.241 LocApplicationInfo Struct Reference

### Data Fields

- [BYTE appProviderLength](#)
- [CHAR \\* pAppProvider](#)
- [BYTE appNameLength](#)
- [CHAR \\* pAppName](#)
- [BYTE appVersionValid](#)
- [CHAR appVersionLength](#)
- [CHAR \\* pAppVersion](#)

### 8.241.1 Detailed Description

This structure contains the Application Information

## Parameters

<i>appProviderLength</i>	<ul style="list-style-type: none"> <li>Length of the Application Provider</li> </ul>
<i>pAppProvider</i>	<ul style="list-style-type: none"> <li>Application Provider</li> <li>Depends upon the Length of application Provider</li> </ul>
<i>appNameLength</i>	<ul style="list-style-type: none"> <li>Length of Application Name</li> </ul>
<i>pAppName</i>	<ul style="list-style-type: none"> <li>Application Name</li> <li>Depends upon the Length of application Name</li> </ul>
<i>appVersionValid</i>	<ul style="list-style-type: none"> <li>Specifies whether the application version string contains a valid value</li> <li>0x00 (FALSE) – Application version string is invalid</li> <li>0x01 (TRUE) – Application version string is valid</li> </ul>
<i>appVersionLength</i>	<ul style="list-style-type: none"> <li>Length of Application Version</li> </ul>
<i>pAppVersion</i>	<ul style="list-style-type: none"> <li>Application Version</li> <li>Depends upon the Length of application Version</li> </ul>

## 8.241.2 Field Documentation

8.241.2.1 BYTE LocApplicationInfo::appNameLength

8.241.2.2 BYTE LocApplicationInfo::appProviderLength

8.241.2.3 CHAR LocApplicationInfo::appVersionLength

8.241.2.4 BYTE LocApplicationInfo::appVersionValid

8.241.2.5 CHAR\* LocApplicationInfo::pAppName

8.241.2.6 CHAR\* LocApplicationInfo::pAppProvider

8.241.2.7 CHAR\* LocApplicationInfo::pAppVersion

## 8.242 LocDelAssDataReq Struct Reference

## Data Fields

- [SVInfo](#) \* [pSVInfo](#)

- [GnssData](#) \* [pGnssData](#)
- [CellDb](#) \* [pCellDb](#)
- [ClkInfo](#) \* [pClkInfo](#)
- [BdsSVInfo](#) \* [pBdsSVInfo](#)

### 8.242.1 Detailed Description

This structure contains LOC delete assist data request

#### Parameters

<i>pSVInfo</i>	<ul style="list-style-type: none"> <li>• Pointer to struct <a href="#">SVInfo</a>. See <a href="#">SVInfo</a> for more information</li> </ul>
<i>pGnssData</i>	<ul style="list-style-type: none"> <li>• Pointer to struct <a href="#">GnssData</a>. See <a href="#">GnssData</a> for more information</li> </ul>
<i>pCellDb</i>	<ul style="list-style-type: none"> <li>• Pointer to struct <a href="#">CellDb</a>. See <a href="#">CellDb</a> for more information</li> </ul>
<i>pClkInfo</i>	<ul style="list-style-type: none"> <li>• Pointer to struct <a href="#">ClkInfo</a>. See <a href="#">ClkInfo</a> for more information</li> </ul>
<i>pBdsSVInfo</i>	<ul style="list-style-type: none"> <li>• Pointer to struct <a href="#">BdsSVInfo</a>. See <a href="#">BdsSVInfo</a> for more information</li> </ul>

### 8.242.2 Field Documentation

8.242.2.1 **BdsSVInfo\*** `LocDelAssDataReq::pBdsSVInfo`

8.242.2.2 **CellDb\*** `LocDelAssDataReq::pCellDb`

8.242.2.3 **ClkInfo\*** `LocDelAssDataReq::pClkInfo`

8.242.2.4 **GnssData\*** `LocDelAssDataReq::pGnssData`

8.242.2.5 **SVInfo\*** `LocDelAssDataReq::pSVInfo`

## 8.243 LOCEventRegisterReqResp Struct Reference

### Data Fields

- [ULONGLONG](#) `eventRegister`

### 8.243.1 Detailed Description

This structure contains the Parameter for RegisterEvents



## Parameters

<i>pEventRegMask</i>	<ul style="list-style-type: none"> <li>• Specifies the events that the control point is interested in receiving. -Values             <ul style="list-style-type: none"> <li>– 0x00000001 - to receive position report event indications</li> <li>– 0x00000002 - to receive satellite report event indications</li> <li>– 0x00000004 - to receive NMEA reports for position and satellites in view</li> <li>– 0x00000008 - to receive NI Notify/Verify request event indications</li> <li>– 0x00000010 - to receive time injection request event indications.</li> <li>– 0x00000020 - to receive predicted orbits request event indications.</li> <li>– 0x00000040 - to receive position injection request event indications.</li> <li>– 0x00000080 - to receive fix session status report event indications.</li> <li>– 0x00000200 - to receive Wi-Fi position request event indications.</li> <li>– 0x00000400 - to receive notifications from the location engine indicating its readiness to accept data from the sensors</li> <li>– 0x00000800 - to receive time sync requests from the GPS engine. Time sync enables the GPS engine to synchronize its clock with the sensor processor's clock.</li> <li>– 0x00001000 - to receive Stationary Position Indicator (SPI) streaming report indications.</li> <li>– 0x00002000 - to receive location server requests. These requests are generated when the service wishes to establish a connection with a location server.</li> <li>– 0x00004000 - to receive notifications related to network-initiated Geofences. These events notify the client when a network-initiated Geofence is added, deleted, or edited.</li> <li>– 0x00008000 - to receive Geofence alerts. These alerts are generated to inform the client of the changes that may affect a Geofence, e.g., if GPS is turned off or if the network is unavailable.</li> <li>– 0x00010000 - to receive notifications when a Geofence is breached. These events are generated when a UE enters or leaves the perimeter of a Geofence. This breach report is for a single Geofence.</li> <li>– 0x00020000 - to register for pedometer control requests from the location engine. The location engine sends this event to control the injection of pedometer reports.</li> <li>– 0x00040000 - to register for motion data control requests from the location engine. The location engine sends this event to control the injection of motion data.</li> <li>– 0x00080000 - to receive notification when a batch is full. The location engine sends this event to notify of Batch Full for ongoing batching session.</li> <li>– 0x00100000 - to receive position report indications along with an ongoing batching session. The location engine sends this event to notify the batched position report while a batching session is ongoing.</li> <li>– 0x00200000 - to receive Wi-Fi Access Point (AP) data inject request event indications.</li> <li>– 0x00400000 - to receive notifications when a Geofence is breached. These events are generated when a UE enters or leaves the perimeter of a Geofence. This breach notification is for multiple Geofences. Breaches from multiple Geofences are all batched and sent in the same notification.</li> <li>– 0x00800000 - to receive notifications from the location engine indicating its readiness to accept vehicle data (vehicle accelerometer, vehicle angular rate, vehicle odometry, etc.).</li> <li>– 0x01000000 - to receive system clock and satellite measurement report events (system clock, <a href="#">SV</a> time, Doppler, etc.).</li> <li>– 0x02000000 - to receive satellite position reports as polynomials.</li> </ul> </li> </ul>
Generated on Wed Oct 28 2015 00:40:12 for LinuxQMSDK by Doxygen	

### 8.243.2 Field Documentation

#### 8.243.2.1 ULONGLONG LOCEventRegisterReqResp::eventRegister

## 8.244 LOCExtPowerStateReqResp Struct Reference

### Data Fields

- [ULONG extPowerState](#)

### 8.244.1 Detailed Description

This structure contains the Parameter External Power Source State.

#### Parameters

<i>pLOCEvent-RegisterReq-Resp</i>	<ul style="list-style-type: none"> <li>• Specifies the Power state; injected by the control point.</li> <li>• Values             <ul style="list-style-type: none"> <li>– 0 - Device is not connected to an external power source</li> <li>– 1 - Device is connected to an external power source</li> <li>– 2 - Unknown external power state</li> </ul> </li> </ul>
-----------------------------------	---

### 8.244.2 Field Documentation

#### 8.244.2.1 ULONG LOCExtPowerStateReqResp::extPowerState

## 8.245 LOCStartReq Struct Reference

### Data Fields

- [BYTE SessionId](#)
- [ULONG \\* pRecurrenceType](#)
- [ULONG \\* pHorizontalAccuracyLvl](#)
- [ULONG \\* pIntermediateReportState](#)
- [ULONG \\* pMinIntervalTime](#)
- [struct LocApplicationInfo \\* pApplicationInfo](#)
- [ULONG \\* pConfigAltitudeAssumed](#)

### 8.245.1 Detailed Description

This structure contains the LOC Start Request

## Parameters

<i>SessionId</i>	<ul style="list-style-type: none"> <li>• ID of the session as identified by the control point.</li> <li>• Range: 0 to 255</li> </ul>
<i>pRecurrence-Type</i>	<ul style="list-style-type: none"> <li>• Specifies the type of session in which the control point is interested.</li> <li>• Defaults to SINGLE. -Values <ul style="list-style-type: none"> <li>– 1 - Request periodic position fixes</li> <li>– 2 - Request a single position fix</li> </ul> </li> </ul>
<i>pHorizontal-AccuracyLvl</i>	<ul style="list-style-type: none"> <li>• Specifies the horizontal accuracy level required by the control point.</li> <li>• Defaults to LOW</li> <li>• Values <ul style="list-style-type: none"> <li>– 1 - Low accuracy</li> <li>– 2 - Medium accuracy</li> <li>– 3 - High accuracy</li> </ul> </li> </ul>
<i>pIntermediate-ReportState</i>	<ul style="list-style-type: none"> <li>• Specifies if the control point is interested in receiving intermediate reports.</li> <li>• ON by default.</li> <li>• Values <ul style="list-style-type: none"> <li>– 1 - Intermediate reports are turned on</li> <li>– 2 - Intermediate reports are turned off</li> </ul> </li> </ul>
<i>appVersionValid</i>	<ul style="list-style-type: none"> <li>• Specifies whether the application version string contains a valid value</li> <li>• 0x00 (FALSE) – Application version string is invalid</li> <li>• 0x01 (TRUE) – Application version string is valid</li> </ul>
<i>LocApplication-Info</i>	<ul style="list-style-type: none"> <li>• LOC Application Parameters</li> <li>• See <a href="#">LocApplicationInfo</a> for more information</li> </ul>
<i>pConfigAltitude-Assumed</i>	<ul style="list-style-type: none"> <li>• Configuration for Altitude Assumed Info in GNSS <a href="#">SV</a> Info Event</li> <li>• Defaults to ENABLED.</li> <li>• Values <ul style="list-style-type: none"> <li>– 1 - Enable Altitude Assumed information in GNSS <a href="#">SV</a> Info Event</li> <li>– 2 - Disable Altitude Assumed information in GNSS <a href="#">SV</a> Info Event</li> </ul> </li> </ul>

## 8.245.2 Field Documentation

8.245.2.1 struct LocApplicationInfo\* LOCStartReq::pApplicationInfo

8.245.2.2 ULONG\* LOCStartReq::pConfigAltitudeAssumed

8.245.2.3 ULONG\* LOCStartReq::pHorizontalAccuracyLvl

8.245.2.4 ULONG\* LOCStartReq::pIntermediateReportState

8.245.2.5 ULONG\* LOCStartReq::pMinIntervalTime

8.245.2.6 ULONG\* LOCStartReq::pRecurrenceType

8.245.2.7 BYTE LOCStartReq::SessionId

## 8.246 LOCStopReq Struct Reference

### Data Fields

- [BYTE sessionId](#)

### 8.246.1 Detailed Description

This structure contains the LOC Stop Request

#### Parameters

<i>sessionId</i>	<ul style="list-style-type: none"><li>• ID of the session as identified by the control point.</li><li>• Range: 0 to 255</li></ul>
------------------	---

## 8.246.2 Field Documentation

8.246.2.1 BYTE LOCStopReq::sessionId

## 8.247 LteCQIParm Struct Reference

### Data Fields

- [BYTE ValidityCW0](#)
- [BYTE CQIValueCW0](#)
- [BYTE ValidityCW1](#)
- [BYTE CQIValueCW1](#)

### 8.247.1 Detailed Description

This structure contains information about the SLQSSwiGetLteCQI response parameters.

## Parameters

<i>pValidityCW0</i> [O-UT]	<ul style="list-style-type: none"> <li>• Values <ul style="list-style-type: none"> <li>– 0- Invalid.</li> <li>– 1- Valid.</li> </ul> </li> </ul>
<i>pCQIValueCW0</i> [OUT]	<ul style="list-style-type: none"> <li>• Values <ul style="list-style-type: none"> <li>– Range 0~15</li> </ul> </li> </ul>
<i>pValidityCW1</i> [O-UT]	<ul style="list-style-type: none"> <li>• Values <ul style="list-style-type: none"> <li>– 0- Invalid.</li> <li>– 1- Valid.</li> </ul> </li> </ul>
<i>pCQIValueCW1</i> [OUT]	<ul style="list-style-type: none"> <li>• Values <ul style="list-style-type: none"> <li>– Range 0~15</li> </ul> </li> </ul>

## 8.247.2 Field Documentation

8.247.2.1 BYTE lteCQIParm::CQIValueCW0

8.247.2.2 BYTE lteCQIParm::CQIValueCW1

8.247.2.3 BYTE lteCQIParm::ValidityCW0

8.247.2.4 BYTE lteCQIParm::ValidityCW1

## 8.248 lteGsmCellInfo Struct Reference

## Data Fields

- [BYTE cellReselPriority](#)
- [BYTE threshGsmHigh](#)
- [BYTE threshGsmLow](#)
- [BYTE nccPermitted](#)
- [BYTE cells\\_len](#)
- [gsmCellInfo](#) [GsmCellInfo](#) [255]

## 8.248.1 Detailed Description

This structure contains information about the LTE GSM Cell.

## Parameters

<i>cellReselPriority</i>	<ul style="list-style-type: none"> <li>• Priority of this frequency group.</li> <li>• Range: 0 to 7.</li> <li>• This field is only valid when <code>ue_in_idle</code> is TRUE.</li> </ul>
<i>threshGsmHigh</i>	<ul style="list-style-type: none"> <li>• Reselection threshold for high priority layers.</li> <li>• Range: 0 to 31.</li> <li>• This field is only valid when <code>ue_in_idle</code> is TRUE.</li> </ul>
<i>threshGsmLow</i>	<ul style="list-style-type: none"> <li>• Reselection threshold for low priority layers.</li> <li>• Range: 0 to 31.</li> <li>• This field is only valid when <code>ue_in_idle</code> is TRUE.</li> </ul>
<i>nccPermitted</i>	<ul style="list-style-type: none"> <li>• Bitmask specifying whether a neighbor with a specific network color code is to be reported.</li> <li>• Range: 0 to 255.</li> <li>• Bit <code>n</code> set to 1 means a neighbor with NCC <code>n</code> must be included in the report. This flag is synonymous with a blacklist in other RATs.</li> <li>• This field is only valid when <code>ue_in_idle</code> is TRUE.</li> </ul>
<i>cells_len</i>	<ul style="list-style-type: none"> <li>• Provides the number of set of gsm cells.</li> </ul>
<i>GsmCellInfo[MAX_DESCRIPTOR_LENGTH]</i>	<ul style="list-style-type: none"> <li>• See <a href="#">gsmCellInfo</a> for more information.</li> </ul>

## 8.248.2 Field Documentation

8.248.2.1 BYTE lteGsmCellInfo::cellReselPriority

8.248.2.2 BYTE lteGsmCellInfo::cells\_len

8.248.2.3 gsmCellInfo lteGsmCellInfo::GsmCellInfo[255]

8.248.2.4 BYTE lteGsmCellInfo::nccPermitted

8.248.2.5 BYTE lteGsmCellInfo::threshGsmHigh

8.248.2.6 BYTE lteGsmCellInfo::threshGsmLow

## 8.249 LTEInfo Struct Reference

## Data Fields

- [BYTE band](#)
- [BYTE bandwidth](#)
- [WORD RXChan](#)
- [WORD TXChan](#)
- [BYTE emmState](#)
- [BYTE emmSubState](#)
- [BYTE emmConnState](#)

### 8.249.1 Detailed Description

Structure for storing the LTE information for the device.

#### Parameters

<i>band</i>	<ul style="list-style-type: none"><li>• LTE Band<ul style="list-style-type: none"><li>– 1 ~ 40 (Band in decimal)</li><li>– 0xFF - Invalid</li></ul></li></ul>
<i>bandwidth</i>	<ul style="list-style-type: none"><li>• BandWidth.<ul style="list-style-type: none"><li>– 0x00 - 1.4 MHz</li><li>– 0x01 - 3 MHz</li><li>– 0x02 - 5 MHz</li><li>– 0x03 - 10 MHz</li><li>– 0x04 - 15 MHz</li><li>– 0x05 - 20 MHz</li><li>– 0x06 - Invalid</li><li>– 0xFF - Unknown</li></ul></li></ul>

<i>RXChan</i>	<ul style="list-style-type: none"> <li>• RX channel number in decimal <ul style="list-style-type: none"> <li>– 0xFFFF - Not Available</li> </ul> </li> </ul>
<i>TXChan</i>	<ul style="list-style-type: none"> <li>• TX channel number in decimal <ul style="list-style-type: none"> <li>– 0xFFFF - Not Available</li> </ul> </li> </ul>
<i>emmState</i>	<ul style="list-style-type: none"> <li>• EMM State. <ul style="list-style-type: none"> <li>– 0x00 - Deregistered</li> <li>– 0x01 - Reg Initiated</li> <li>– 0x02 - Registered</li> <li>– 0x03 - TAU Initiated</li> <li>– 0x04 - SR Initiated</li> <li>– 0x05 - Dereg Initiated</li> <li>– 0x06 - Invalid</li> <li>– 0xFF - Unknown</li> </ul> </li> </ul>
<i>emmSubState</i>	<ul style="list-style-type: none"> <li>• EMM Sub State. <ul style="list-style-type: none"> <li>– 0xFF - NOT Applicable</li> </ul> </li> <li>• When EMM_state is 0x00: <ul style="list-style-type: none"> <li>– 0x00 - No IMSI</li> <li>– 0x01 - PLMN Search</li> <li>– 0x02 - Attach Needed</li> <li>– 0x03 - No Cell</li> <li>– 0x04 - Attaching</li> <li>– 0x05 - Normal Service</li> <li>– 0x06 - Limited Service</li> <li>– 0x07 - Waiting for PDN</li> </ul> </li> <li>• When EMM_state is 0x01: <ul style="list-style-type: none"> <li>– 0x00 - Waiting for NW</li> <li>– 0x01 - Waiting for ESM</li> </ul> </li> <li>• When EMM_state is 0x02: <ul style="list-style-type: none"> <li>– 0x00 - Normal Service</li> <li>– 0x01 - Update Needed</li> <li>– 0x02 - Attempt Update</li> <li>– 0x03 - No Cell</li> <li>– 0x04 - PLMN Search</li> <li>– 0x05 - Limited Service</li> <li>– 0x06 - MM Update</li> </ul> </li> </ul>
	<ul style="list-style-type: none"> <li>– 0x07 - IMSI Detach</li> <li>– 0x08 - Waiting for ESM</li> </ul>



<i>emmConnState</i>	<ul style="list-style-type: none"><li>• EMM Connected Mode State.<ul style="list-style-type: none"><li>– 0x00 - RRC Idle</li><li>– 0x01 - Waiting RRC Cfm</li><li>– 0x02 - RRC Connected</li><li>– 0x03 - RRC Releasing</li><li>– 0xFF - Unknown</li></ul></li></ul>
---------------------	--

## 8.249.2 Field Documentation

8.249.2.1 BYTE LTEInfo::band

8.249.2.2 BYTE LTEInfo::bandwidth

8.249.2.3 BYTE LTEInfo::emmConnState

8.249.2.4 BYTE LTEInfo::emmState

8.249.2.5 BYTE LTEInfo::emmSubState

8.249.2.6 WORD LTEInfo::RXChan

8.249.2.7 WORD LTEInfo::TXChan

## 8.250 LTEInfoInterfreq Struct Reference

### Data Fields

- [BYTE ueInIdle](#)
- [BYTE freqsLen](#)
- [infoInterFreq InfoInterfreq](#) [255]

### 8.250.1 Detailed Description

This structure contains information about the LTE Inter-Frequency Network.

#### Parameters

<i>ueInIdle</i>	<ul style="list-style-type: none"><li>• TRUE if the UE is in Idle mode, otherwise FALSE.<ul style="list-style-type: none"><li>– 0xFF - Not Available</li></ul></li></ul>
-----------------	--

<i>freqsLen</i>	<ul style="list-style-type: none"> <li>• Provides the number of set of inter frequency information.</li> <li>• If 0(zero), then no information follows it.</li> </ul>
<i>InfoInterfreq[MA- X_DESCRIPTI- ON_LENGTH]</i>	<ul style="list-style-type: none"> <li>• See <a href="#">infoInterFreq</a> for more information.</li> </ul>

## 8.250.2 Field Documentation

### 8.250.2.1 BYTE LTEInfoInterfreq::freqsLen

### 8.250.2.2 infoInterFreq LTEInfoInterfreq::InfoInterfreq[255]

### 8.250.2.3 BYTE LTEInfoInterfreq::ueInIdle

## 8.251 LTEInfoIntrafreq Struct Reference

### Data Fields

- [BYTE ueInIdle](#)
- [BYTE plmn](#) [3]
- [WORD tac](#)
- [ULONG globalCellId](#)
- [WORD earfcn](#)
- [WORD servingCellId](#)
- [BYTE cellReselPriority](#)
- [BYTE sNonIntraSearch](#)
- [BYTE threshServingLow](#)
- [BYTE sIntraSearch](#)
- [BYTE cellsLen](#)
- [cellParams CellParams](#) [255]

### 8.251.1 Detailed Description

This structure contains information about the LTE Intra-Frequency Network.

#### Parameters

<i>ueInIdle</i>	<ul style="list-style-type: none"> <li>• TRUE if the UE is in Idle mode, otherwise FALSE.</li> <li>– 0xFF - Not Available</li> </ul>
-----------------	--

<i>plmn[PLMN_LENGTH]</i>	<ul style="list-style-type: none"> <li>• PLMN ID coded as octet 3, 4, and 5.</li> </ul>
<i>tac</i>	<ul style="list-style-type: none"> <li>• Tracking area code. <ul style="list-style-type: none"> <li>– 0xFFFF - Not Available</li> </ul> </li> </ul>
<i>globalCellId</i>	<ul style="list-style-type: none"> <li>• Global cell ID in the system information block. <ul style="list-style-type: none"> <li>– 0xFFFFFFFF - Not Available</li> </ul> </li> </ul>
<i>earfcn</i>	<ul style="list-style-type: none"> <li>• E-UTRA absolute radio frequency channel number of the serving cell.</li> <li>• Range: 0 to 65535. <ul style="list-style-type: none"> <li>– 0xFFFF - Not Available</li> </ul> </li> </ul>
<i>servingCellId</i>	<ul style="list-style-type: none"> <li>• LTE serving cell ID.</li> <li>• Range: 0 to 503.</li> <li>• This is the cell ID of the serving cell and can be found in the cell list. <ul style="list-style-type: none"> <li>– 0xFFFF - Not Available</li> </ul> </li> </ul>
<i>cellReselPriority</i>	<ul style="list-style-type: none"> <li>• Priority for serving frequency.</li> <li>• Range: 0 to 7.</li> <li>• This field is only valid when <code>ue_in_idle</code> is TRUE. <ul style="list-style-type: none"> <li>– 0xFF - Not Available</li> </ul> </li> </ul>
<i>sNonIntraSearch</i>	<ul style="list-style-type: none"> <li>• S non-intra search threshold to control non-intrafrequency searches.</li> <li>• Range: 0 to 31.</li> <li>• This field is only valid when <code>ue_in_idle</code> is TRUE. <ul style="list-style-type: none"> <li>– 0xFF - Not Available</li> </ul> </li> </ul>

<i>threshServing-Low</i>	<ul style="list-style-type: none"> <li>• Serving cell low threshold.</li> <li>• Range: 0 to 31.</li> <li>• This field is only valid when ue_in_idle is TRUE. <ul style="list-style-type: none"> <li>– 0xFF - Not Available</li> </ul> </li> </ul>
<i>sIntraSearch</i>	<ul style="list-style-type: none"> <li>• S Intra search threshold.</li> <li>• Range: 0 to 31.</li> <li>• The current cell measurement must fall below this threshold to consider intrafrequency for reselection.</li> <li>• This field is only valid when ue_in_idle is TRUE. <ul style="list-style-type: none"> <li>– 0xFF - Not Available</li> </ul> </li> </ul>
<i>cellsLen</i>	<ul style="list-style-type: none"> <li>• Provides the number of set of cell params.</li> <li>• If 0(zero), then no information follows it.</li> </ul>
<i>CellParams[MAX_DESCRIPTOR_LENGTH]</i>	<ul style="list-style-type: none"> <li>• See <a href="#">cellParams</a> for more information.</li> </ul>

## 8.251.2 Field Documentation

8.251.2.1 **cellParams** LTEInfoIntrafreq::CellParams[255]

8.251.2.2 **BYTE** LTEInfoIntrafreq::cellReselPriority

8.251.2.3 **BYTE** LTEInfoIntrafreq::cellsLen

8.251.2.4 **WORD** LTEInfoIntrafreq::earfcn

8.251.2.5 **ULONG** LTEInfoIntrafreq::globalCellId

8.251.2.6 **BYTE** LTEInfoIntrafreq::plmn[3]

8.251.2.7 **WORD** LTEInfoIntrafreq::servingCellId

8.251.2.8 **BYTE** LTEInfoIntrafreq::sIntraSearch

8.251.2.9 **BYTE** LTEInfoIntrafreq::sNonIntraSearch

8.251.2.10 **WORD** LTEInfoIntrafreq::tac

8.251.2.11 **BYTE** LTEInfoIntrafreq::threshServingLow

8.251.2.12 **BYTE** LTEInfoIntrafreq::ueInIdle

## 8.252 LTEInfoNeighboringGSM Struct Reference

### Data Fields

- [BYTE ueIdle](#)
- [BYTE freqsLen](#)
- [lteGsmCellInfo](#) [LteGsmCellInfo](#) [255]

### 8.252.1 Detailed Description

This structure contains information about the LTE Neighboring GSM Network.

#### Parameters

<i>ueIdle</i>	<ul style="list-style-type: none"> <li>• TRUE if the UE is in Idle mode, otherwise FALSE.</li> <li>– 0xFF - Not Available</li> </ul>
<i>freqsLen</i>	<ul style="list-style-type: none"> <li>• Provides the number of set of LTE GSM cell information.</li> <li>• If 0(zero), then no information follows it.</li> </ul>
<i>LteGsmCellInfo[ MAX_DESCRIPTION_LENGTH]</i>	<ul style="list-style-type: none"> <li>• See <a href="#">lteGsmCellInfo</a> for more information.</li> </ul>

### 8.252.2 Field Documentation

8.252.2.1 **BYTE** LTEInfoNeighboringGSM::freqsLen

8.252.2.2 **lteGsmCellInfo** LTEInfoNeighboringGSM::LteGsmCellInfo[255]

8.252.2.3 **BYTE** LTEInfoNeighboringGSM::ueIdle

## 8.253 LTEInfoNeighboringWCDMA Struct Reference

### Data Fields

- [BYTE ueIdle](#)
- [BYTE freqsLen](#)
- [lteWcdmaCellInfo](#) [LTEWCDMACellInfo](#) [255]

### 8.253.1 Detailed Description

This structure contains information about the LTE Neighboring WCDMA Network.

#### Parameters

<i>ueIdle</i>	<ul style="list-style-type: none"> <li>• TRUE if the UE is in Idle mode, otherwise FALSE. <ul style="list-style-type: none"> <li>– 0xFF - Not Available</li> </ul> </li> </ul>
<i>freqsLen</i>	<ul style="list-style-type: none"> <li>• Provides the number of set of LTE WCDMA cell information.</li> <li>• If 0(zero), then no information follows it.</li> </ul>
<i>LTEWCDMA-CellInfo[MAX_DESCRIPTION_LENGTH]</i>	<ul style="list-style-type: none"> <li>• See <a href="#">lteWcdmaCellInfo</a> for more information.</li> </ul>

### 8.253.2 Field Documentation

8.253.2.1 BYTE LTEInfoNeighboringWCDMA::freqsLen

8.253.2.2 lteWcdmaCellInfo LTEInfoNeighboringWCDMA::LTEWCDMACellInfo[255]

8.253.2.3 BYTE LTEInfoNeighboringWCDMA::ueIdle

## 8.254 LteNasReleaseInfo\_s Struct Reference

### Data Fields

- [BYTE nas\\_release](#)
- [BYTE nas\\_major](#)
- [BYTE nas\\_minor](#)

### 8.254.1 Detailed Description

This structure contains LTE Nas Release Information

#### Parameters

<i>nas_release</i>	<ul style="list-style-type: none"> <li>• LTE NAS release</li> </ul>
<i>nas_major</i>	<ul style="list-style-type: none"> <li>• LTE NAS version major</li> </ul>
<i>nas_minor</i>	<ul style="list-style-type: none"> <li>• LTE NAS version minor</li> </ul>

### 8.254.2 Field Documentation

8.254.2.1 BYTE LteNasReleaseInfo\_s::nas\_major

8.254.2.2 BYTE LteNasReleaseInfo\_s::nas\_minor

8.254.2.3 BYTE lteNasReleaseInfo\_s::nas\_release

## 8.255 lteRsrpInformation Struct Reference

### Data Fields

- [SHORT rsrplevel](#)

### 8.255.1 Detailed Description

This structure contains the LTE RSRP Information

#### Parameters

<i>rsrplevel</i>	<ul style="list-style-type: none"><li>• LTE RSRP in dBm as a mesaured by L1. Range: -44 to -140(-44 means -44dBm, -140 means -140dBm).</li></ul>
------------------	--

### 8.255.2 Field Documentation

8.255.2.1 SHORT lteRsrpInformation::rsrplevel

## 8.256 LTERSRPThresh Struct Reference

### Data Fields

- [BYTE LTERSRPThreshListLen](#)
- [WORD \\* pLTERSRPThreshList](#)

### 8.256.1 Detailed Description

This structure contains LTE RSRP threshold related parameters.

#### Parameters

<i>LTERSRP- ThreshListLen</i>	<ul style="list-style-type: none"><li>• Length of the LTE RSRP threshold list parameter to follow</li></ul>
<i>pLTERSRP- ThreshList</i>	<ul style="list-style-type: none"><li>• Array of RSRP thresholds (in units of 0.1 dBm)</li><li>• Maximum of 32 values</li><li>• Range for RSRP values: -140 to -44 (in dBm).</li></ul>

### 8.256.2 Field Documentation

8.256.2.1 BYTE LTERSRPThresh::LTERSRPThreshListLen

8.256.2.2 WORD\* LTERSRPThresh::pLTERSRPThreshList

## 8.257 LTERSRQThresh Struct Reference

### Data Fields

- [BYTE LTERSRQThreshListLen](#)
- [WORD \\* pLTERSRQThreshList](#)

### 8.257.1 Detailed Description

This structure contains LTE RSRQ threshold related parameters.

#### Parameters

<i>LTERSRQ- ThreshListLen</i>	<ul style="list-style-type: none"> <li>• Length of the LTE RSRQ threshold list parameter to follow</li> </ul>
<i>pLTERSRQ- ThreshList</i>	<ul style="list-style-type: none"> <li>• Array of RSRQ thresholds (in units of 0.1 dBm)</li> <li>• Maximum of 32 values.</li> <li>• Range for RSRQ values: -20 to -3 (in dBm)</li> </ul>

### 8.257.2 Field Documentation

8.257.2.1 [BYTE LTERSRQThresh::LTERSRQThreshListLen](#)

8.257.2.2 [WORD\\* LTERSRQThresh::pLTERSRQThreshList](#)

## 8.258 LTERSSIThresh Struct Reference

### Data Fields

- [BYTE LTERSSIThreshListLen](#)
- [WORD \\* pLTERSSIThreshList](#)

### 8.258.1 Detailed Description

This structure contains LTE RSSI threshold related parameters.

#### Parameters

<i>LTERSSI- ThreshListLen</i>	<ul style="list-style-type: none"> <li>• Length of the LTE RSSI threshold list parameter to follow</li> </ul>
<i>pLTERSSI- ThreshList</i>	<ul style="list-style-type: none"> <li>• Array of RSSI thresholds (in units of 0.1 dBm)</li> <li>• Maximum of 32 values.</li> <li>• Range for RSSI values: -120 to 0 (in dBm)</li> </ul>



## 8.258.2 Field Documentation

8.258.2.1 **BYTE** LTERSSIThresh::LTERSSIThreshListLen

8.258.2.2 **WORD\*** LTERSSIThresh::pLTERSSIThreshList

## 8.259 LTESigRptCfg Struct Reference

### Data Fields

- [BYTE rptRate](#)
- [BYTE avgPeriod](#)

### 8.259.1 Detailed Description

This structure contains LTE Signal Report Config parameters.

#### Parameters

<i>rptRate</i>	<ul style="list-style-type: none"> <li>• Rate on how often the LTE signal must be checked for reporting</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0 - Report using the default configuration</li> <li>– 1 - Report every 1 sec</li> <li>– 2 - Report every 2 sec</li> <li>– 3 - Report every 3 sec</li> <li>– 4 - Report every 4 sec</li> <li>– 5 - Report every 5 sec</li> </ul> </li> </ul>
<i>avgPeriod</i>	<ul style="list-style-type: none"> <li>• Averaging period to be used for the LTE signal</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0 - Average using the default configuration</li> <li>– 1 - Average over 1 sec</li> <li>– 2 - Average over 2 sec</li> <li>– 3 - Average over 3 sec</li> <li>– 4 - Average over 4 sec</li> <li>– 5 - Average over 5 sec</li> <li>– 6 - Average over 6 sec</li> <li>– 7 - Average over 7 sec</li> <li>– 8 - Average over 8 sec</li> <li>– 9 - Average over 9 sec</li> <li>– 10 - Average over 10 sec</li> </ul> </li> </ul>

## 8.259.2 Field Documentation

8.259.2.1 BYTE LTESigRptCfg::avgPeriod

8.259.2.2 BYTE LTESigRptCfg::rptRate

## 8.260 LTESigRptConfig Struct Reference

### Data Fields

- [BYTE rptRate](#)
- [BYTE avgPeriod](#)

### 8.260.1 Detailed Description

This structure contains LTE RSRP threshold related parameters.

#### Parameters

<i>rptRate</i>	<ul style="list-style-type: none"> <li>• Rate on how often the LTE signal must be checked for reporting Values</li> <li>• 0 - Report using the default configuration</li> <li>• 1 - Report every 1 sec</li> <li>• 2 - Report every 2 sec</li> <li>• 3 - Report every 3 sec</li> <li>• 4 - Report every 4 sec</li> <li>• 5 - Report every 5 sec</li> </ul>
<i>avgPeriod</i>	<ul style="list-style-type: none"> <li>• Averaging period to be used for the LTE signal.</li> <li>• Values <ul style="list-style-type: none"> <li>– 0 - Average using the default configuration</li> <li>– 1 - Average over 1 sec</li> <li>– 2 - Average over 2 sec</li> <li>– 3 - Average over 3 sec</li> <li>– 4 - Average over 4 sec</li> <li>– 5 - Average over 5 sec</li> <li>– 6 - Average over 6 sec</li> <li>– 7 - Average over 7 sec</li> <li>– 8 - Average over 8 sec</li> <li>– 9 - Average over 9 sec</li> <li>– 10 - Average over 10 sec</li> </ul> </li> </ul>

### 8.260.2 Field Documentation

8.260.2.1 BYTE LTESigRptConfig::avgPeriod

8.260.2.2 BYTE lteSigRptConfig::rptRate

## 8.261 lteSnrinformation Struct Reference

### Data Fields

- [SHORT snrlevel](#)

### 8.261.1 Detailed Description

This structure contains the LTE SNR Information

#### Parameters

<i>snrlevel</i>	<ul style="list-style-type: none"><li>• LTE SNR level as a scaled integer in units of 0.1dB e.g. -16dB has a value of -160 and 24.6dB has value of 246.</li></ul>
-----------------	---

### 8.261.2 Field Documentation

8.261.2.1 SHORT lteSnrinformation::snrlevel

## 8.262 lteSNRThresh Struct Reference

### Data Fields

- [BYTE lteSNRThresListLen](#)
- [SHORT \\* plteSNRThresList](#)

### 8.262.1 Detailed Description

This structure contains LTE SNR threshold related parameters.

#### Parameters

<i>lteSNRThresListLen</i>	<ul style="list-style-type: none"><li>• Length of the LTE SNR threshold list parameter to follow</li></ul>
<i>plteSNRThresList</i>	<ul style="list-style-type: none"><li>• Sequence of thresholds delimiting SNR event reporting bands</li><li>• Every time a SNR value crosses a threshold value, an event report indication message with the new SNR value is sent to the requesting control point. For this field<ul style="list-style-type: none"><li>– For LTE, each SNR threshold value is a signed 2 Byte value</li><li>– Maximum number of threshold values is 16</li><li>– At least one value must be specified</li><li>– SNR level as a scaled integer in units of 0.1 dB; e.g., -16 dB has a value of -160 and 24.6 dB has a value of 246</li></ul></li></ul>

### 8.262.2 Field Documentation

8.262.2.1 **BYTE** LTESNRThresh::LTESNRThresListLen

8.262.2.2 **SHORT\*** LTESNRThresh::pLTESNRThresList

## 8.263 LTESNRThreshold Struct Reference

### Data Fields

- [BYTE LTESNRThreshListLen](#)
- [WORD \\*](#) [pLTESNRThreshList](#)

### 8.263.1 Detailed Description

This structure contains LTE SNR threshold related parameters.

#### Parameters

<i>LTESNRThresh-ListLen</i>	<ul style="list-style-type: none"> <li>• Length of the LTE SNR threshold list parameter to follow</li> </ul>
<i>pLTESNR-ThreshList</i>	<ul style="list-style-type: none"> <li>• Array of SNR thresholds (in units of 0.1 dB)</li> <li>• Maximum of 32 values</li> <li>• Range for SNR values: -20 to 30 (in dB).</li> </ul>

### 8.263.2 Field Documentation

8.263.2.1 **BYTE** LTESNRThreshold::LTESNRThreshListLen

8.263.2.2 **WORD\*** LTESNRThreshold::pLTESNRThreshList

## 8.264 LTESInfo Struct Reference

### Data Fields

- [INT8 rssi](#)
- [INT8 rsrq](#)
- [SHORT rsrp](#)
- [SHORT snr](#)

### 8.264.1 Detailed Description

This structure contains the parameters for LTE Signal Strength Information

#### Parameters

<i>rsqi</i>	<ul style="list-style-type: none"> <li>• RSSI in dBm (signed value).</li> <li>• A value of -125 dBm or lower is used to indicate No Signal. <ul style="list-style-type: none"> <li>– For CDMA and UMTS, this indicates forward link pilot Ec</li> <li>– For GSM, this indicates received signal strength</li> </ul> </li> </ul>
<i>rsrq</i>	<ul style="list-style-type: none"> <li>• RSRQ value in dB (signed integer value) as measured by L1.</li> <li>• Range: -3 to -20 (-3 means -3 dB, -20 means -20 dB).</li> </ul>
<i>rsrp</i>	<ul style="list-style-type: none"> <li>• Current RSRP in dBm as measured by L1.</li> <li>• Range: -44 to -140 (-44 means -44 dBm, -140 means -140 dBm).</li> </ul>
<i>snr</i>	<ul style="list-style-type: none"> <li>• SNR level as a scaled integer in units of 0.1 dB. e.g., -16 dB has a value of -160 and 24.6 dB has a value of 246,</li> </ul>

## 8.264.2 Field Documentation

### 8.264.2.1 SHORT LTESInfo::rsrp

### 8.264.2.2 INT8 LTESInfo::rsrq

### 8.264.2.3 INT8 LTESInfo::rsqi

### 8.264.2.4 SHORT LTESInfo::snr

## 8.265 LTESysInfo Struct Reference

### Data Fields

- [sysInfoCommon](#) [sysInfoLTE](#)
- [BYTE](#) lacValid
- [WORD](#) lac
- [BYTE](#) cellIdValid
- [ULONG](#) cellId
- [BYTE](#) regRejectInfoValid
- [BYTE](#) rejectSrvDomain
- [BYTE](#) rejCause
- [BYTE](#) networkIdValid
- [BYTE](#) MCC [3]
- [BYTE](#) MNC [3]
- [BYTE](#) tacValid
- [WORD](#) tac

### 8.265.1 Detailed Description

Structure for storing the LTE System Information.

## Parameters

<i>sysInfoLTE</i>	<ul style="list-style-type: none"> <li>• See <a href="#">sysInfoCommon</a> for more information.</li> </ul>
<i>lacValid</i>	<ul style="list-style-type: none"> <li>• Indicates whether the location area code is valid.. <ul style="list-style-type: none"> <li>– 0x00 - Invalid</li> <li>– 0x01 - Valid</li> <li>– 0xFF - Not Available</li> </ul> </li> </ul>
<i>lac</i>	<ul style="list-style-type: none"> <li>• Location area code.</li> <li>• Only applies to 3GPP. <ul style="list-style-type: none"> <li>– 0xFFFF - Not Available</li> </ul> </li> </ul>
<i>cellIdValid</i>	<ul style="list-style-type: none"> <li>• Indicates whether the cell ID is valid. <ul style="list-style-type: none"> <li>– 0x00 - Invalid</li> <li>– 0x01 - Valid</li> <li>– 0xFF - Not Available</li> </ul> </li> </ul>
<i>cellId</i>	<ul style="list-style-type: none"> <li>• Cell ID. <ul style="list-style-type: none"> <li>– 0xFFFFFFFF - Not Available</li> </ul> </li> </ul>
<i>regRejectInfoValid</i>	<ul style="list-style-type: none"> <li>• Indicates whether the registration reject information is valid. <ul style="list-style-type: none"> <li>– 0x00 - Invalid</li> <li>– 0x01 - Valid</li> <li>– 0xFF - Not Available</li> </ul> </li> </ul>
<i>rejectSrvDomain</i>	<ul style="list-style-type: none"> <li>• Type of service domain in which the registration is rejected. <ul style="list-style-type: none"> <li>– 0x00 - SYS_SRV_DOMAIN_NO_SRV - No service</li> <li>– 0x01 - Circuit-switched only</li> <li>– 0x02 - Packet-switched only</li> <li>– 0x03 - Circuit-switched and packet-switched</li> <li>– 0x04 - Camped</li> <li>– 0xFF - Not Available</li> </ul> </li> </ul>
<i>rejCause</i>	<ul style="list-style-type: none"> <li>• Reject cause values sent are specified in [3GPP TS 24.008, Section 10.5.3.6]. <ul style="list-style-type: none"> <li>– 0xFF - Not Available</li> </ul> </li> </ul>

<i>networkIdValid</i>	<ul style="list-style-type: none"> <li>Indicates whether the network ID is valid. <ul style="list-style-type: none"> <li>0x00 - Invalid</li> <li>0x01 - Valid</li> <li>0xFF - Not Available</li> </ul> </li> </ul>
<i>MCC[PLMN_LENGTH]</i>	<ul style="list-style-type: none"> <li>Mobile Country Code.</li> <li>MCC digits in ASCII characters</li> </ul>
<i>MNC[PLMN_LENGTH]</i>	<ul style="list-style-type: none"> <li>Mobile Network Code.</li> <li>MNC digits in ASCII characters</li> <li>An unused byte is set to 0xFF.</li> </ul>
<i>tacValid</i>	<ul style="list-style-type: none"> <li>Indicates whether tracking area code is valid. <ul style="list-style-type: none"> <li>0x00 - Invalid</li> <li>0x01 - Valid</li> <li>0xFF - Not Available</li> </ul> </li> </ul>
<i>tac</i>	<ul style="list-style-type: none"> <li>Tracking area code.</li> <li>Only applicable for LTE. <ul style="list-style-type: none"> <li>0xFFFF - Not Available</li> </ul> </li> </ul>

## 8.265.2 Field Documentation

8.265.2.1 **ULONG** `LTESysInfo::cellId`

8.265.2.2 **BYTE** `LTESysInfo::cellIdValid`

8.265.2.3 **WORD** `LTESysInfo::lac`

8.265.2.4 **BYTE** `LTESysInfo::lacValid`

8.265.2.5 **BYTE** `LTESysInfo::MCC[3]`

8.265.2.6 **BYTE** `LTESysInfo::MNC[3]`

8.265.2.7 **BYTE** `LTESysInfo::networkIdValid`

8.265.2.8 **BYTE** `LTESysInfo::regRejectInfoValid`

8.265.2.9 **BYTE** `LTESysInfo::rejCause`



8.265.2.10 **BYTE** LTESysInfo::rejectSrvDomain8.265.2.11 **sysInfoCommon** LTESysInfo::sysInfoLTE8.265.2.12 **WORD** LTESysInfo::tac8.265.2.13 **BYTE** LTESysInfo::tacValid

## 8.266 IteWcdmaCellInfo Struct Reference

### Data Fields

- [WORD](#) *uarfcn*
- [BYTE](#) *cellReselPriority*
- [WORD](#) *threshXhigh*
- [WORD](#) *threshXlow*
- [BYTE](#) *cellsLen*
- [wcdmaCellInfo](#) [WCDMACellInfo](#) [255]

### 8.266.1 Detailed Description

This structure contains information about the LTE WCDMA Cell.

#### Parameters

<i>uarfcn</i>	<ul style="list-style-type: none"> <li>• WCDMA layer frequency.</li> <li>• Range: 0 to 16383.</li> </ul>
<i>cellReselPriority</i>	<ul style="list-style-type: none"> <li>• Cell re-selection priority.</li> <li>• Range: 0 to 7.</li> <li>• This field is only valid when <i>ue_in_idle</i> is TRUE.</li> </ul>
<i>threshXhigh</i>	<ul style="list-style-type: none"> <li>• Re-selection low threshold.</li> <li>• Range: 0 to 31.</li> <li>• This field is only valid when <i>ue_in_idle</i> is TRUE.</li> </ul>
<i>threshXlow</i>	<ul style="list-style-type: none"> <li>• Re-selection high threshold.</li> <li>• Range: 0 to 31.</li> <li>• This field is only valid when <i>ue_in_idle</i> is TRUE.</li> </ul>

<i>cellsLen</i>	<ul style="list-style-type: none"> <li>Provides the number of set of WCDMA cells.</li> </ul>
<i>WCDMACellInfo[MAX_DESCRIPTION_LENGTH]</i>	<ul style="list-style-type: none"> <li>See <a href="#">wcdmaCellInfo</a> for more information.</li> </ul>

## 8.266.2 Field Documentation

8.266.2.1 **BYTE** *IteWcdmaCellInfo::cellReselPriority*

8.266.2.2 **BYTE** *IteWcdmaCellInfo::cellsLen*

8.266.2.3 **WORD** *IteWcdmaCellInfo::threshXhigh*

8.266.2.4 **WORD** *IteWcdmaCellInfo::threshXlow*

8.266.2.5 **WORD** *IteWcdmaCellInfo::uarfcn*

8.266.2.6 **wcdmaCellInfo** *IteWcdmaCellInfo::WCDMACellInfo[255]*

## 8.267 messageWaitingInfoContent Struct Reference

### Data Fields

- [BYTE](#) *msgType*
- [BYTE](#) *activeInd*
- [BYTE](#) *msgCount*

### 8.267.1 Detailed Description

This structure contains message waiting information per instance

#### Parameters

<i>msgType</i>	<ul style="list-style-type: none"> <li>Message type <ul style="list-style-type: none"> <li>0x00 - MWI_MESSAGE_TYPE_VOICEMAIL - Voicemail</li> <li>0x01 - MWI_MESSAGE_TYPE_FAX - Fax</li> <li>0x02 - MWI_MESSAGE_TYPE_EMAIL - Email</li> <li>0x03 - MWI_MESSAGE_TYPE_OTHER - Other</li> <li>0x04 - MWI_MESSAGE_TYPE_VIDEOMAIL - Videomail</li> </ul> </li> </ul>
----------------	---

<i>activeInd</i>	<ul style="list-style-type: none"> <li>Indicates whether the indication is active <ul style="list-style-type: none"> <li>0x00 - Inactive</li> <li>0x01 - Active</li> </ul> </li> </ul>
<i>msgCount</i>	<ul style="list-style-type: none"> <li>Number of messages</li> </ul>

## 8.267.2 Field Documentation

8.267.2.1 **BYTE** messageWaitingInfoContent::activeInd

8.267.2.2 **BYTE** messageWaitingInfoContent::msgCount

8.267.2.3 **BYTE** messageWaitingInfoContent::msgType

## 8.268 minBasedIMSI Struct Reference

### Data Fields

- [BYTE](#) mccM [3]
- [WORD](#) imsiM1112
- [BYTE](#) imsiMS1 [7]
- [BYTE](#) imsiMS2 [3]

### 8.268.1 Detailed Description

This structure contains the parameters for Min based IMSI Information

#### Parameters

<i>mccM</i>	<ul style="list-style-type: none"> <li>ASCII character representation of MCC_M</li> </ul>
<i>imsiM1112</i>	<ul style="list-style-type: none"> <li>ASCII character representation of IMSI_M_11_12 value <ul style="list-style-type: none"> <li>0xFFFF - Not Available</li> </ul> </li> </ul>
<i>imsiMS1</i>	<ul style="list-style-type: none"> <li>ASCII character representation of IMSI_M_S1 value</li> </ul>
<i>imsiMS2</i>	<ul style="list-style-type: none"> <li>ASCII character representation of IMSI_M_S2 value</li> </ul>

## 8.268.2 Field Documentation

8.268.2.1 **WORD** minBasedIMSI::imsiM1112

8.268.2.2 **BYTE** minBasedIMSI::imsiMS1[7]

8.268.2.3 **BYTE** minBasedIMSI::imsiMS2[3]

8.268.2.4 **BYTE** minBasedIMSI::mccM[3]

## 8.269 MNRInfo Struct Reference

### Data Fields

- [WORD](#) `mcc`
- [WORD](#) `mnc`
- [ULONG](#) `rat`

### 8.269.1 Detailed Description

Structure contains Manual Network Register Information parameters

#### Parameters

<i>mcc</i>	<ul style="list-style-type: none"><li>• A 16-bit integer representation of Mobile Country Code. Range - 0 to 999.</li></ul>
<i>mnc</i>	<ul style="list-style-type: none"><li>• A 16-bit integer representation of Mobile Network Code. Range - 0 to 999.</li></ul>
<i>rat</i>	<ul style="list-style-type: none"><li>• Radio access technology for which to register.<ul style="list-style-type: none"><li>– 0x04 - RADIO_IF_GSM</li><li>– 0x05 - RADIO_IF_UMTS</li><li>– 0x08 - RADIO_IF_LTE</li></ul></li></ul>

### 8.269.2 Field Documentation

8.269.2.1 **WORD** MNRInfo::mcc

8.269.2.2 **WORD** MNRInfo::mnc

8.269.2.3 **ULONG** MNRInfo::rat

## 8.270 ModifyProfileIn Struct Reference

### Data Fields

- [BYTE](#) \* `pProfileID`
- [BYTE](#) \* `pProfileType`
- [QmiProfileInfo](#) `curProfile`

### 8.270.1 Detailed Description

This structure contains input parameters for SLQSMModifyProfile

## Parameters

<i>ProfileID</i>	<ul style="list-style-type: none"> <li>• 1 to 16 for 3GPP profile</li> <li>• 101 to 106 for 3GPP2 profile</li> </ul>
<i>ProfileType</i>	<ul style="list-style-type: none"> <li>• Identifies the technology type of the profile <ul style="list-style-type: none"> <li>– 0x00 - 3GPP</li> <li>– 0x01 - 3GPP2</li> <li>– NULL is not allowed</li> </ul> </li> </ul>
<i>curProfile</i>	<ul style="list-style-type: none"> <li>• Contains Union of profile(3GPP/3GPP2) structures</li> </ul>

## 8.270.2 Field Documentation

8.270.2.1 QmiProfileInfo ModifyProfileIn::curProfile

8.270.2.2 BYTE\* ModifyProfileIn::pProfileID

8.270.2.3 BYTE\* ModifyProfileIn::pProfileType

## 8.271 ModifyProfileOut Struct Reference

## Data Fields

- USHORT \* [pExtErrorCode](#)

## 8.271.1 Detailed Description

This structure contains out parameters for SLQSMModifyProfile

## Parameters

<i>pExtErrorCode</i>	<ul style="list-style-type: none"> <li>• The extended error code received from DS Profile subsystem of type eWDS_ERR_PROFILE_REG_XXX.</li> <li>• Error code will only will be present if error code eQCWWAN_ERR_QMI_EXTENDED_INTERNAL is returned by device.</li> <li>• See <a href="#">qm_wds_ds_profile_extended_err_codes</a> enum in <a href="#">qmerrno.h</a> for received error description.</li> </ul>
----------------------	---

## 8.271.2 Field Documentation

8.271.2.1 USHORT\* ModifyProfileOut::pExtErrorCode

## 8.272 msgWaitingInfo Struct Reference

### Data Fields

- [BYTE numInstances](#)
- [messageWaitingInfoContent msgWaitInfo](#) [0xFF]

### 8.272.1 Detailed Description

This structure holds information related to message waiting information

#### Parameters

<i>numInstances</i>	<ul style="list-style-type: none"><li>• Number of sets of the elements in structure <a href="#">messageWaitingInfoContent</a></li></ul>
<i>msgWaitInfo</i>	<ul style="list-style-type: none"><li>• Pointer to structure of <a href="#">messageWaitingInfoContent</a>.<ul style="list-style-type: none"><li>– See <a href="#">messageWaitingInfoContent</a> for more information.</li></ul></li></ul>

### 8.272.2 Field Documentation

8.272.2.1 [messageWaitingInfoContent msgWaitingInfo::msgWaitInfo](#)[0xFF]

8.272.2.2 [BYTE msgWaitingInfo::numInstances](#)

## 8.273 namName Struct Reference

### Data Fields

- [BYTE namNameLen](#)
- [BYTE namName](#) [12]

### 8.273.1 Detailed Description

This structure contains the parameters for NAM Name Information

#### Parameters

<i>namNameLen</i>	<ul style="list-style-type: none"><li>• Number of sets of the following elements:<ul style="list-style-type: none"><li>– <code>nam_name</code></li></ul></li><li>• If zero(0), then no information follows.</li></ul>
<i><a href="#">namName</a></i>	<ul style="list-style-type: none"><li>• Name information in ASCII. The maximum length of <code>nam_name</code> is 12.</li></ul>

## 8.273.2 Field Documentation

8.273.2.1 BYTE `namName::namName[12]`

8.273.2.2 BYTE `namName::namNameLen`

## 8.274 nasCellLocationInfoResp Struct Reference

### Data Fields

- [GERANInfo](#) \* [pGERANInfo](#)
- [UMTSInfo](#) \* [pUMTSInfo](#)
- [CDMAInfo](#) \* [pCDMAInfo](#)
- [LTEInfoIntrafreq](#) \* [pLTEInfoIntrafreq](#)
- [LTEInfoInterfreq](#) \* [pLTEInfoInterfreq](#)
- [LTEInfoNeighboringGSM](#) \* [pLTEInfoNeighboringGSM](#)
- [LTEInfoNeighboringWCDMA](#) \* [pLTEInfoNeighboringWCDMA](#)
- [ULONG](#) \* [pUMTSCellID](#)
- [WCDMAInfoLTENeighborCell](#) \* [pWCDMAInfoLTENeighborCell](#)

### 8.274.1 Detailed Description

This structure contains information about the Get Cell Location response parameters.



## Parameters

<i>pGERANInfo</i>	<ul style="list-style-type: none"> <li>See <a href="#">GERANInfo</a> for more information.</li> </ul>
<i>pUMTSInfo</i>	<ul style="list-style-type: none"> <li>See <a href="#">UMTSInfo</a> for more information.</li> </ul>
<i>pCDMAInfo</i>	<ul style="list-style-type: none"> <li>See <a href="#">CDMAInfo</a> for more information.</li> </ul>
<i>pLTEInfo-Intrafreq</i>	<ul style="list-style-type: none"> <li>See <a href="#">LTEInfoIntrafreq</a> for more information.</li> </ul>
<i>pLTEInfo-Interfreq</i>	<ul style="list-style-type: none"> <li>See <a href="#">LTEInfoInterfreq</a> for more information.</li> </ul>
<i>pLTEInfo-NeighboringGSM</i>	<ul style="list-style-type: none"> <li>See <a href="#">LTEInfoNeighboringGSM</a> for more information.</li> </ul>
<i>pLTEInfo-NeighboringWCDMA</i>	<ul style="list-style-type: none"> <li>See <a href="#">LTEInfoNeighboringWCDMA</a> for more information.</li> </ul>
<i>pUMTSCellID</i>	<ul style="list-style-type: none"> <li>Cell ID.</li> <li>0xFFFFFFFF indicates cell ID information is not present.</li> </ul>
<i>pWCDMAInfoLT-ENeighborCell</i>	<ul style="list-style-type: none"> <li>See <a href="#">WCDMAInfoLTENeighborCell</a> for more information.</li> </ul>

## 8.274.2 Field Documentation

8.274.2.1 **CDMAInfo\*** nasCellLocationInfoResp::pCDMAInfo8.274.2.2 **GERANInfo\*** nasCellLocationInfoResp::pGERANInfo8.274.2.3 **LTEInfoInterfreq\*** nasCellLocationInfoResp::pLTEInfoInterfreq8.274.2.4 **LTEInfoIntrafreq\*** nasCellLocationInfoResp::pLTEInfoIntrafreq8.274.2.5 **LTEInfoNeighboringGSM\*** nasCellLocationInfoResp::pLTEInfoNeighboringGSM8.274.2.6 **LTEInfoNeighboringWCDMA\*** nasCellLocationInfoResp::pLTEInfoNeighboringWCDMA8.274.2.7 **ULONG\*** nasCellLocationInfoResp::pUMTSCellID8.274.2.8 **UMTSInfo\*** nasCellLocationInfoResp::pUMTSInfo8.274.2.9 **WCDMAInfoLTENeighborCell\*** nasCellLocationInfoResp::pWCDMAInfoLTENeighborCell

## 8.275 nasGet3GPP2SubscriptionInfoReq Struct Reference

### Data Fields

- [BYTE namID](#)

### 8.275.1 Detailed Description

This structure contains the Get3GPP2SubscriptionInfo request parameters

#### Parameters

<i>namID</i>	[Mandatory] <ul style="list-style-type: none"><li>• NAM ID of the information to be retrieved. The index starts from 0. A <i>nam_id</i> of 0xFF is used to retrieve information of current NAM.</li></ul>
--------------	---

### 8.275.2 Field Documentation

#### 8.275.2.1 BYTE nasGet3GPP2SubscriptionInfoReq::namID

## 8.276 nasGet3GPP2SubscriptionInfoResp Struct Reference

### Data Fields

- [namName](#) \* [pNAMNameInfo](#)
- [dirNum](#) \* [pDirNum](#)
- [homeSIDNID](#) \* [pHomeSIDNID](#)
- [minBasedIMSI](#) \* [pMinBasedIMSI](#)
- [trueIMSI](#) \* [pTrueIMSI](#)
- [CDMAChannel](#) \* [pCDMAChannel](#)

### 8.276.1 Detailed Description

This structure contains the SLQSNasGet3GPP2Subscription response parameters.

#### Parameters

<i>pNAMNameInfo</i>	[Optional] <ul style="list-style-type: none"><li>• See <a href="#">namName</a> for more information</li></ul>
<i>pDirNum</i>	[Optional] <ul style="list-style-type: none"><li>• See <a href="#">dirNum</a> for more information</li></ul>
<i>pHomeSIDNID</i>	[Optional] <ul style="list-style-type: none"><li>• See <a href="#">homeSIDNID</a> for more information</li></ul>

<i>pMinBasedIMSI</i>	[Optional] <ul style="list-style-type: none"> <li>See <a href="#">minBasedIMSI</a> for more information</li> </ul>
<i>pTrueIMSI</i>	[Optional] <ul style="list-style-type: none"> <li>See <a href="#">trueIMSI</a> for more information</li> </ul>
<i>pCDMAChannel</i>	[Optional] <ul style="list-style-type: none"> <li>See <a href="#">CDMAChannel</a> for more information</li> </ul>

## 8.276.2 Field Documentation

8.276.2.1 **CDMAChannel\*** nasGet3GPP2SubscriptionInfoResp::pCDMAChannel

8.276.2.2 **dirNum\*** nasGet3GPP2SubscriptionInfoResp::pDirNum

8.276.2.3 **homeSIDNID\*** nasGet3GPP2SubscriptionInfoResp::pHomeSIDNID

8.276.2.4 **minBasedIMSI\*** nasGet3GPP2SubscriptionInfoResp::pMinBasedIMSI

8.276.2.5 **namName\*** nasGet3GPP2SubscriptionInfoResp::pNAMNameInfo

8.276.2.6 **trueIMSI\*** nasGet3GPP2SubscriptionInfoResp::pTrueIMSI

## 8.277 nasGetHDRColorCodeResp Struct Reference

### Data Fields

- BYTE \*** [pColorCode](#)

### 8.277.1 Detailed Description

Structure for storing the current preferred system selection settings for the device.

#### Parameters

<i>pColorCode</i>	[Optional] <ul style="list-style-type: none"> <li>Color code value</li> <li>Color code corresponding to the sector to which the AT is sending the access probe</li> <li>See 3GPP2 C.S0024-B V3.0, Section 7.11.6.2.1 for more information. <ul style="list-style-type: none"> <li>0xFF - Not Available</li> </ul> </li> </ul>
-------------------	---

## 8.277.2 Field Documentation

8.277.2.1 **BYTE\*** nasGetHDRColorCodeResp::pColorCode

## 8.278 nasGetLTECphyCa Struct Reference

## Data Fields

- [PhyCaAggScellIndType](#) [sPhyCaAggScellIndType](#)
- [PhyCaAggScellDIBw](#) [sPhyCaAggScellDIBw](#)
- [PhyCaAggScellInfo](#) [sPhyCaAggScellInfo](#)
- [PhyCaAggPcellInfo](#) [sPhyCaAggPcellInfo](#)
- [PhyCaAggScellIndex](#) [sPhyCaAggScellIndex](#)

### 8.278.1 Field Documentation

8.278.1.1 [PhyCaAggPcellInfo](#) [nasGetLTECphyCa::sPhyCaAggPcellInfo](#)

8.278.1.2 [PhyCaAggScellDIBw](#) [nasGetLTECphyCa::sPhyCaAggScellDIBw](#)

8.278.1.3 [PhyCaAggScellIndex](#) [nasGetLTECphyCa::sPhyCaAggScellIndex](#)

8.278.1.4 [PhyCaAggScellIndType](#) [nasGetLTECphyCa::sPhyCaAggScellIndType](#)

8.278.1.5 [PhyCaAggScellInfo](#) [nasGetLTECphyCa::sPhyCaAggScellInfo](#)

## 8.279 nasGetLTECphyCaResp Struct Reference

### Data Fields

- [PhyCaAggScellIndType](#) \* [pPhyCaAggScellIndType](#)
- [PhyCaAggScellDIBw](#) \* [pPhyCaAggScellDIBw](#)
- [PhyCaAggScellInfo](#) \* [pPhyCaAggScellInfo](#)
- [PhyCaAggPcellInfo](#) \* [pPhyCaAggPcellInfo](#)
- [PhyCaAggScellIndex](#) \* [pPhyCaAggScellIndex](#)

### 8.279.1 Field Documentation

8.279.1.1 [PhyCaAggPcellInfo](#)\* [nasGetLTECphyCaResp::pPhyCaAggPcellInfo](#)

8.279.1.2 [PhyCaAggScellDIBw](#)\* [nasGetLTECphyCaResp::pPhyCaAggScellDIBw](#)

8.279.1.3 [PhyCaAggScellIndex](#)\* [nasGetLTECphyCaResp::pPhyCaAggScellIndex](#)

8.279.1.4 [PhyCaAggScellIndType](#)\* [nasGetLTECphyCaResp::pPhyCaAggScellIndType](#)

8.279.1.5 [PhyCaAggScellInfo](#)\* [nasGetLTECphyCaResp::pPhyCaAggScellInfo](#)

## 8.280 nasGetSigInfoResp Struct Reference

### Data Fields

- [CDMASSInfo](#) \* [pCDMASSInfo](#)
- [HDRSSInfo](#) \* [pHDRSSInfo](#)
- [INT8](#) \* [pGSMSSInfo](#)
- [CDMASSInfo](#) \* [pWCDMASSInfo](#)
- [LTESSInfo](#) \* [pLTESSInfo](#)

### 8.280.1 Detailed Description

This structure contains the SLQSNasGetSigInfo response parameters.

## Parameters

<i>pCDMASSInfo</i>	[Optional] <ul style="list-style-type: none"> <li>See <a href="#">CDMASSInfo</a> for more information</li> </ul>
<i>pHDRSSInfo</i>	[Optional] <ul style="list-style-type: none"> <li>See <a href="#">HDRSSInfo</a> for more information</li> </ul>
<i>pGSMSSInfo</i>	[Optional] <ul style="list-style-type: none"> <li>GSM signal strength is the RSSI in dBm (signed value).</li> <li>A value of -125 dBm or lower is used to indicate No Signal.</li> </ul>
<i>pWCDMASSInfo</i>	[Optional] <ul style="list-style-type: none"> <li>See <a href="#">CDMASSInfo</a> for more information</li> </ul>
<i>pLTESSInfo</i>	[Optional] <ul style="list-style-type: none"> <li>See <a href="#">LTESSInfo</a> for more information</li> </ul>

## 8.280.2 Field Documentation

8.280.2.1 CDMASInfo\* nasGetSigInfoResp::pCDMASSInfo

8.280.2.2 INT8\* nasGetSigInfoResp::pGSMSSInfo

8.280.2.3 HDRSSInfo\* nasGetSigInfoResp::pHDRSSInfo

8.280.2.4 LTESSInfo\* nasGetSigInfoResp::pLTESSInfo

8.280.2.5 CDMASInfo\* nasGetSigInfoResp::pWCDMASSInfo

## 8.281 nasGetSysInfoResp Struct Reference

## Data Fields

- SrvStatusInfo \* pCDMASrvStatusInfo
- SrvStatusInfo \* pHDRSrvStatusInfo
- GSMStatusInfo \* pGSMStatusInfo
- GSMStatusInfo \* pWCDMASrvStatusInfo
- GSMStatusInfo \* pLTESrvStatusInfo
- CDMASysInfo \* pCDMASysInfo
- HDRSysInfo \* pHDRSysInfo
- GSMStatusInfo \* pGSMStatusInfo
- WCDMASysInfo \* pWCDMASysInfo
- LTESysInfo \* pLTESysInfo
- AddCDMASysInfo \* pAddCDMASysInfo
- WORD \* pAddHDRSysInfo
- AddSysInfo \* pAddGSMStatusInfo
- AddSysInfo \* pAddWCDMASysInfo
- WORD \* pAddLTESysInfo
- CallBarringSysInfo \* pGSMCallBarringSysInfo

- [CallBarringSysInfo](#) \* [pWCDMACallBarringSysInfo](#)
- [BYTE](#) \* [pLTEVoiceSupportSysInfo](#)
- [BYTE](#) \* [pGSMCipherDomainSysInfo](#)
- [BYTE](#) \* [pWCDMACipherDomainSysInfo](#)

### 8.281.1 Detailed Description

Structure for storing the SLQSNasGetSysInfo response parameters.

#### Parameters

<i>pCDMASrv-StatusInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">SrvStatusInfo</a> for more information.</li> </ul>
<i>pHDRSrvStatus-Info</i>	<ul style="list-style-type: none"> <li>• See <a href="#">SrvStatusInfo</a> for more information.</li> </ul>
<i>pGSMSrvStatus-Info</i>	<ul style="list-style-type: none"> <li>• See <a href="#">GSMSrvStatusInfo</a> for more information.</li> </ul>
<i>pWCDMASrv-StatusInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">GSMSrvStatusInfo</a> for more information.</li> </ul>
<i>pLTESrvStatus-Info</i>	<ul style="list-style-type: none"> <li>• See <a href="#">GSMSrvStatusInfo</a> for more information.</li> </ul>
<i>pCDMASysInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">CDMASysInfo</a> for more information.</li> </ul>
<i>pHDRSysInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">HDRSysInfo</a> for more information.</li> </ul>
<i>pGSMSysInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">GSMSysInfo</a> for more information.</li> </ul>
<i>pWCDMASys-Info</i>	<ul style="list-style-type: none"> <li>• See <a href="#">WCDMASysInfo</a> for more information.</li> </ul>
<i>pLTESysInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">LTESysInfo</a> for more information.</li> </ul>
<i>pAddCDMASys-Info</i>	<ul style="list-style-type: none"> <li>• See <a href="#">AddCDMASysInfo</a> for more information.</li> </ul>
<i>pAddHDRSys-Info</i>	<ul style="list-style-type: none"> <li>• System table index referencing the beginning of the geo in which the current serving system is present.</li> <li>• When the system index is not known, 0xFFFF is used.</li> </ul>

<i>pAddGSMSys-Info</i>	<ul style="list-style-type: none"> <li>• See <a href="#">AddSysInfo</a> for more information.</li> </ul>
<i>pAddWCDMA-SysInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">AddSysInfo</a> for more information.</li> </ul>
<i>pAddLTESysInfo</i>	<ul style="list-style-type: none"> <li>• System table index referencing the beginning of the geo in which the current serving system is present.</li> <li>• When the system index is not known, 0xFFFF is used.</li> </ul>
<i>pGSMCall-BarringSysInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">CallBarringSysInfo</a> for more information.</li> </ul>
<i>pWCDMACall-BarringSysInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">CallBarringSysInfo</a> for more information.</li> </ul>
<i>pLTEVoice-SupportSysInfo</i>	<ul style="list-style-type: none"> <li>• Indicates voice support status on LTE. <ul style="list-style-type: none"> <li>– 0x00 - Voice is not supported</li> <li>– 0x01 - Voice is supported</li> </ul> </li> </ul>
<i>pGSMCipher-DomainSysInfo</i>	<ul style="list-style-type: none"> <li>• Ciphering on the service domain. <ul style="list-style-type: none"> <li>– 0x00 - No service</li> <li>– 0x01 - Circuit-switched only</li> <li>– 0x02 - Packet-switched only</li> <li>– 0x03 - Circuit-switched and packet-switched</li> </ul> </li> </ul>
<i>pWCDMA-CipherDomain-SysInfo</i>	<ul style="list-style-type: none"> <li>• Ciphering on the service domain. <ul style="list-style-type: none"> <li>– 0x00 - No service</li> <li>– 0x01 - Circuit-switched only</li> <li>– 0x02 - Packet-switched only</li> <li>– 0x03 - Circuit-switched and packet-switched</li> </ul> </li> </ul>

## 8.281.2 Field Documentation

8.281.2.1 **AddCDMASysInfo\*** nasGetSysInfoResp::pAddCDMASysInfo

8.281.2.2 **AddSysInfo\*** nasGetSysInfoResp::pAddGSMSysInfo

8.281.2.3 **WORD\*** nasGetSysInfoResp::pAddHDRSysInfo

8.281.2.4 **WORD\*** nasGetSysInfoResp::pAddLTESysInfo



- 8.281.2.5 **AddSysInfo\*** nasGetSysInfoResp::pAddWCDMASysInfo
- 8.281.2.6 **SrvStatusInfo\*** nasGetSysInfoResp::pCDMASrvStatusInfo
- 8.281.2.7 **CDMASysInfo\*** nasGetSysInfoResp::pCDMASysInfo
- 8.281.2.8 **CallBarringSysInfo\*** nasGetSysInfoResp::pGSMCallBarringSysInfo
- 8.281.2.9 **BYTE\*** nasGetSysInfoResp::pGSMCipherDomainSysInfo
- 8.281.2.10 **GSMSrvStatusInfo\*** nasGetSysInfoResp::pGSMSrvStatusInfo
- 8.281.2.11 **GSMSysInfo\*** nasGetSysInfoResp::pGSMSysInfo
- 8.281.2.12 **SrvStatusInfo\*** nasGetSysInfoResp::pHDSrvStatusInfo
- 8.281.2.13 **HDRSysInfo\*** nasGetSysInfoResp::pHDSysInfo
- 8.281.2.14 **GSMSrvStatusInfo\*** nasGetSysInfoResp::pLTSrvStatusInfo
- 8.281.2.15 **LTESysInfo\*** nasGetSysInfoResp::pLTESysInfo
- 8.281.2.16 **BYTE\*** nasGetSysInfoResp::pLTEVoiceSupportSysInfo
- 8.281.2.17 **CallBarringSysInfo\*** nasGetSysInfoResp::pWCDMACallBarringSysInfo
- 8.281.2.18 **BYTE\*** nasGetSysInfoResp::pWCDMACipherDomainSysInfo
- 8.281.2.19 **GSMSrvStatusInfo\*** nasGetSysInfoResp::pWCDMASrvStatusInfo
- 8.281.2.20 **WCDMASysInfo\*** nasGetSysInfoResp::pWCDMASysInfo

## 8.282 nasGetTxRxInfoReq Struct Reference

### Data Fields

- [BYTE](#) `radio_if`

### 8.282.1 Detailed Description

This structure contains the GetTxRxInfoReq request parameters

#### Parameters

<i>radio_if</i>	<div>[Mandatory]</div> <ul style="list-style-type: none"> <li>• Radio interface technology of the signal being measured <ul style="list-style-type: none"> <li>– NAS_RADIO_IF_CDMA_1X - CDMA</li> <li>– NAS_RADIO_IF_CDMA_1XEVD0 - HDR</li> <li>– NAS_RADIO_IF_GSM - GSM</li> <li>– NAS_RADIO_IF_UMTS - UMTS</li> <li>– NAS_RADIO_IF_LTE - LTE</li> </ul> </li> </ul>
-----------------	---

## 8.282.2 Field Documentation

### 8.282.2.1 BYTE nasGetTxRxInfoReq::radio\_if

## 8.283 nasGetTxRxInfoResp Struct Reference

### Data Fields

- rxInfo \* pRXChain0Info
- rxInfo \* pRXChain1Info
- txInfo \* pTXInfo

### 8.283.1 Detailed Description

This structure contains the GetTxRxInfoResp response parameters.

#### Parameters

<i>pRXChain0Info</i>	[Optional] <ul style="list-style-type: none"> <li>• See <a href="#">rxInfo</a> for more information.</li> </ul>
<i>pRXChain1Info</i>	[Optional] <ul style="list-style-type: none"> <li>• See <a href="#">rxInfo</a> for more information.</li> </ul>
<i>pTXInfo</i>	[Optional] <ul style="list-style-type: none"> <li>• See <a href="#">txInfo</a> for more information.</li> </ul>

## 8.283.2 Field Documentation

### 8.283.2.1 rxInfo\* nasGetTxRxInfoResp::pRXChain0Info

### 8.283.2.2 rxInfo\* nasGetTxRxInfoResp::pRXChain1Info

### 8.283.2.3 txInfo\* nasGetTxRxInfoResp::pTXInfo

## 8.284 nasIndicationRegisterReq Struct Reference

### Data Fields

- BYTE \* pSystemSelectionInd
- BYTE \* pDDTMInd
- BYTE \* pServingSystemInd
- BYTE \* pDualStandByPrefInd
- BYTE \* pSubscriptionInfoInd
- BYTE \* pNetworkTimeInd
- BYTE \* pSysInfoInd
- BYTE \* pSignalStrengthInd
- BYTE \* pErrorRateInd
- BYTE \* pHDRNewUATIAssInd
- BYTE \* pHDRSessionCloseInd
- BYTE \* pManagedRoamingInd
- BYTE \* pLTECphyCa

### 8.284.1 Detailed Description

This structure contains the SLQSNasIndicationRegisterExt request parameters.

#### Parameters

<i>pSystem-SelectionInd</i>	<p>[Optional]</p> <ul style="list-style-type: none"> <li>System Selection Preference indication registration. The following callbacks would not be invoked if the indication is disabled.</li> </ul> <p><a href="#">tFNRoamingIndicator</a> <a href="#">tFNDataCapabilities</a> and <a href="#">tFNServingSystem</a></p> <ul style="list-style-type: none"> <li>– 0x00 - Disable</li> <li>– 0x01 - Enable</li> </ul>
<i>pDDTMInd</i>	<p>[Optional]</p> <ul style="list-style-type: none"> <li>DDTM (Data Dedicated Transmission Mode) indication registration. The following callbacks would not be invoked if the indication is disabled.</li> </ul> <p><a href="#">tFNDDTM</a></p> <ul style="list-style-type: none"> <li>– 0x00 - Disable</li> <li>– 0x01 - Enable</li> </ul>
<i>pServing-SystemInd</i>	<p>[Optional]</p> <ul style="list-style-type: none"> <li>Serving System indication registration. The following callbacks would not be invoked if the indication is disabled.</li> </ul> <p><a href="#">tFNBandPreference</a></p> <ul style="list-style-type: none"> <li>– 0x00 - Disable</li> <li>– 0x01 - Enable</li> </ul>
<i>pDualStandBy-PrefInd</i>	<p>[Optional]</p> <ul style="list-style-type: none"> <li>Dual Standby Preference indication registration. The following callbacks would not be invoked if the indication is disabled.</li> </ul> <p><a href="#">tFNDualStandByPref</a></p> <ul style="list-style-type: none"> <li>– 0x00 - Disable</li> <li>– 0x01 - Enable</li> </ul>
<i>pSubscription-InfoInd</i>	<p>[Optional]</p> <ul style="list-style-type: none"> <li>Subscription Information indication registration. The following callbacks would not be invoked if the indication is disabled.</li> </ul> <p><a href="#">tFNSubscriptionInfo</a></p> <ul style="list-style-type: none"> <li>– 0x00 - Disable</li> <li>– 0x01 - Enable</li> </ul>

<i>pNetworkTimeInd</i>	<p>[Optional]</p> <ul style="list-style-type: none"> <li>• Network Time indication registration. The following callbacks would not be invoked if the indication is disabled.</li> </ul> <p><a href="#">tFNNetworkTime</a></p> <ul style="list-style-type: none"> <li>– 0x00 - Disable</li> <li>– 0x01 - Enable</li> </ul>
<i>pSysInfoInd</i>	<p>[Optional]</p> <ul style="list-style-type: none"> <li>• System Information indication registration. The following callbacks would not be invoked if the indication is disabled.</li> </ul> <p><a href="#">tFNSysInfo</a></p> <ul style="list-style-type: none"> <li>– 0x00 - Disable</li> <li>– 0x01 - Enable</li> </ul>
<i>pSignalStrengthInd</i>	<p>[Optional]</p> <ul style="list-style-type: none"> <li>• Signal Strength indication registration. The following callbacks would not be invoked if the indication is disabled.</li> </ul> <p><a href="#">tFNSigInfo</a></p> <ul style="list-style-type: none"> <li>– 0x00 - Disable</li> <li>– 0x01 - Enable</li> </ul>
<i>pErrorRateInd</i>	<p>[Optional]</p> <ul style="list-style-type: none"> <li>• Error Rate indication registration. The following callbacks would not be invoked if the indication is disabled.</li> </ul> <p><a href="#">tFNErrRate</a></p> <ul style="list-style-type: none"> <li>– 0x00 - Disable</li> <li>– 0x01 - Enable</li> </ul>
<i>pHDRNewUATI-AssInd</i>	<p>[Optional]</p> <ul style="list-style-type: none"> <li>• HDR New UATI Assigned indication registration. The following callbacks would not be invoked if the indication is disabled.</li> </ul> <p><a href="#">tFNHDRUATIUpdate</a></p> <ul style="list-style-type: none"> <li>– 0x00 - Disable</li> <li>– 0x01 - Enable</li> </ul>

<i>pHDRSession-CloseInd</i>	[Optional] <ul style="list-style-type: none"> <li>HDR Session Closed indication registration. The following callbacks would not be invoked if the indication is disabled.</li> </ul> tFNHDRSessionClose <ul style="list-style-type: none"> <li>0x00 - Disable</li> <li>0x01 - Enable</li> </ul>
<i>pManaged-RoamingInd</i>	[Optional] <ul style="list-style-type: none"> <li>Managed Roaming indication registration. The following callbacks would not be invoked if the indication is disabled.</li> </ul> tFNManagedRoaming <ul style="list-style-type: none"> <li>0x00 - Disable</li> <li>0x01 - Enable</li> </ul>

**Note**

Atleast one parameter must be provided as request. 'NULL' value confirms that the indication value is not sent.

**8.284.2 Field Documentation**

- 8.284.2.1 **BYTE\*** nasIndicationRegisterReq::pDDTMInd
- 8.284.2.2 **BYTE\*** nasIndicationRegisterReq::pDualStandByPrefInd
- 8.284.2.3 **BYTE\*** nasIndicationRegisterReq::pErrorRateInd
- 8.284.2.4 **BYTE\*** nasIndicationRegisterReq::pHDRNewUATIAssInd
- 8.284.2.5 **BYTE\*** nasIndicationRegisterReq::pHDRSessionCloseInd
- 8.284.2.6 **BYTE\*** nasIndicationRegisterReq::pLTECphyCa
- 8.284.2.7 **BYTE\*** nasIndicationRegisterReq::pManagedRoamingInd
- 8.284.2.8 **BYTE\*** nasIndicationRegisterReq::pNetworkTimeInd
- 8.284.2.9 **BYTE\*** nasIndicationRegisterReq::pServingSystemInd
- 8.284.2.10 **BYTE\*** nasIndicationRegisterReq::pSignalStrengthInd
- 8.284.2.11 **BYTE\*** nasIndicationRegisterReq::pSubscriptionInfoInd
- 8.284.2.12 **BYTE\*** nasIndicationRegisterReq::pSysInfoInd
- 8.284.2.13 **BYTE\*** nasIndicationRegisterReq::pSystemSelectionInd

**8.285 nasInitNetworkReg Struct Reference**

## Data Fields

- [ULONG](#) `regAction`
- [MNRIInfo](#) \* `pMNRIInfo`
- [ULONG](#) \* `pChangeDuration`
- [BOOL](#) \* `pMncPcsDigitStatus`

### 8.285.1 Detailed Description

This structure contains Initiate Network Registration request parameters

#### Parameters

<i>regAction</i>	<ul style="list-style-type: none"> <li>• Specifies one of the following register actions : <ul style="list-style-type: none"> <li>– <code>AUTO_REGISTER</code> - Device registers according to its provisioning and optional parameters supplied with the command are ignored.</li> <li>– <code>MANUAL_REGISTER</code> - Device registers to a specified network and the optional Manual Network Register Information parameter <code>pMNRIInfo</code> must also be included for the command to process successfully and supported only for 3GPP.</li> </ul> </li> </ul>
<i>pMNRIInfo</i>	[Optional] <ul style="list-style-type: none"> <li>• Pointer to structure <a href="#">MNRIInfo</a> <ul style="list-style-type: none"> <li>– See <a href="#">MNRIInfo</a> for more information</li> </ul> </li> </ul>
<i>pChange-Duration</i>	[Optional] <ul style="list-style-type: none"> <li>• Duration of the change. <ul style="list-style-type: none"> <li>– <code>0x00</code> - Power cycle - Remains active until the next device power cycle</li> <li>– <code>0x01</code> - Permanent - Remains active through power cycles until changed by the client</li> </ul> </li> </ul>
<i>pMncPcsDigit-Status</i>	[Optional] <ul style="list-style-type: none"> <li>• MNC PCS Digit Include Status <ul style="list-style-type: none"> <li>– True - MNC is a 3-digit value.</li> <li>– False - MNC is a 2-digit value.</li> </ul> </li> </ul>

### 8.285.2 Field Documentation

8.285.2.1 [ULONG](#)\* `nasInitNetworkReg::pChangeDuration`

8.285.2.2 [BOOL](#)\* `nasInitNetworkReg::pMncPcsDigitStatus`

8.285.2.3 [MNRIInfo](#)\* `nasInitNetworkReg::pMNRIInfo`

8.285.2.4 [ULONG](#) `nasInitNetworkReg::regAction`

### 8.286 nasNetworkTime Struct Reference

## Data Fields

- [UniversalTime](#) universalTime
- [BYTE](#) \* pTimeZone
- [BYTE](#) \* pDayltSavAdj

### 8.286.1 Detailed Description

Structure for storing the [nasSysInfo](#) indication parameters.

#### Parameters

<i>universalTime</i>	<ul style="list-style-type: none"><li>• See <a href="#">UniversalTime</a> for more information.</li></ul>
<i>pTimeZone</i>	<ul style="list-style-type: none"><li>• Time Zone.</li><li>• Offset from Universal time, i.e., the difference between local time and Universal time, in increments of 15 min (signed value).</li></ul>
<i>pDayltSavAdj</i>	<ul style="list-style-type: none"><li>• Daylight Saving Adjustment.</li><li>• Daylight saving adjustment in hr.<ul style="list-style-type: none"><li>– Possible values: 0, 1, and 2.</li></ul></li></ul>

### 8.286.2 Field Documentation

8.286.2.1 [BYTE](#)\* nasNetworkTime::pDayltSavAdj

8.286.2.2 [BYTE](#)\* nasNetworkTime::pTimeZone

8.286.2.3 [UniversalTime](#) nasNetworkTime::universalTime

## 8.287 nasOperatorNameResp Struct Reference

## Data Fields

- [serviceName](#) \* pSvcProviderName
- [operatorPLMNList](#) \* pOperatorPLMNList
- [PLMNNetworkName](#) \* pPLMNNetworkName
- [operatorNameString](#) \* pOperatorNameString
- [PLMNNetworkNameData](#) \* pNITZInformation

### 8.287.1 Detailed Description

This structure contains Operator Name Data related from multiple sources.

## Parameters

<i>pSvcProvider-Name</i>	<ul style="list-style-type: none"> <li>Refer <a href="#">serviceProviderName</a> for details (Optional).</li> <li>Can provide NULL if this parameter is not required.</li> </ul>
<i>pOperatorPLM-NList</i>	<ul style="list-style-type: none"> <li>Refer <a href="#">operatorPLMNList</a> for details (Optional).</li> <li>Can provide NULL if this parameter is not required.</li> </ul>
<i>pPLMNNetwork-Name</i>	<ul style="list-style-type: none"> <li>Refer <a href="#">PLMNNetworkName</a> for details (Optional).</li> <li>Can provide NULL if this parameter is not required.</li> </ul>
<i>pOperatorName-String</i>	<ul style="list-style-type: none"> <li>Refer <a href="#">operatorNameString</a> for details (Optional).</li> <li>Can provide NULL if this parameter is not required.</li> </ul>
<i>pNITZ-Information</i>	<ul style="list-style-type: none"> <li>Refer <a href="#">PLMNNetworkNameData</a> for details (Optional).</li> <li>Can provide NULL if this parameter is not required.</li> </ul>

## 8.287.2 Field Documentation

8.287.2.1 **PLMNNetworkNameData\*** nasOperatorNameResp::pNITZInformation8.287.2.2 **operatorNameString\*** nasOperatorNameResp::pOperatorNameString8.287.2.3 **operatorPLMNList\*** nasOperatorNameResp::pOperatorPLMNList8.287.2.4 **PLMNNetworkName\*** nasOperatorNameResp::pPLMNNetworkName8.287.2.5 **serviceProviderName\*** nasOperatorNameResp::pSvcProviderName

## 8.288 nasPLMNNameReq Struct Reference

## Data Fields

- [WORD mcc](#)
- [WORD mnc](#)

## 8.288.1 Detailed Description

Structure for storing the PLMN Name request parameters



## Parameters

<i>mcc</i>	<ul style="list-style-type: none"><li>• A 16-bit integer representation of MCC. Range: 0 to 999</li></ul>
<i>mnc</i>	<ul style="list-style-type: none"><li>• A 16-bit integer representation of MNC. Range: 0 to 999</li></ul>

## Note

None

## 8.288.2 Field Documentation

8.288.2.1 WORD nasPLMNNameReq::mcc

8.288.2.2 WORD nasPLMNNameReq::mnc

## 8.289 nasPLMNNameResp Struct Reference

## Data Fields

- [BYTE spnEncoding](#)
- [BYTE spnLength](#)
- [BYTE spn \[255\]](#)
- [BYTE shortNameEn](#)
- [BYTE shortNameCI](#)
- [BYTE shortNameSB](#)
- [BYTE shortNameLen](#)
- [BYTE shortName \[255\]](#)
- [BYTE longNameEn](#)
- [BYTE longNameCI](#)
- [BYTE longNameSB](#)
- [BYTE longNameLen](#)
- [BYTE longName \[255\]](#)

## 8.289.1 Detailed Description

Structure for storing the PLMN Name response parameters

## Parameters

<i>spnEncoding</i>	<ul style="list-style-type: none"><li>• Coding scheme used for service provider name. This value is ignored if spn_len is zero Values:<ul style="list-style-type: none"><li>– 0x00 - SMS default 7-bit coded alphabet as defined in 3GPP TS 23.038 with bit 8 set to 0</li><li>– 0x01 - UCS2 (16 bit, little-endian) 3GPP TS 23.038</li></ul></li></ul>
--------------------	---

<i>spnLength</i>	<ul style="list-style-type: none"> <li>Length of SPN which follows</li> </ul>
<i>spn</i>	<ul style="list-style-type: none"> <li>Service Provider name string</li> </ul>
<i>shortNameEn</i>	<ul style="list-style-type: none"> <li>Coding scheme used for PLMN short name. This value is ignored if PLMN short name length is zero Values:             <ul style="list-style-type: none"> <li>0x00 - SMS default 7-bit coded alphabet as defined in 3GPP TS 23.038 with bit 8 set to 0</li> <li>0x01 - UCS2 (16 bit, little-endian) 3GPP TS 23.038</li> </ul> </li> </ul>
<i>shortNameCI</i>	<ul style="list-style-type: none"> <li>Indicates whether the country initials are to be added to the shortName. This value is ignored if shortNameLen is zero. Values:             <ul style="list-style-type: none"> <li>0x00 - Do not add the letters for the countrys initials to the name</li> <li>0x01 - Add the countrys initials and a text string to the name</li> <li>0xFF - Not specified</li> </ul> </li> </ul>
<i>shortNameSB</i>	<ul style="list-style-type: none"> <li>PLMN short name spare bits. This value is ignored if shortNameLen is zero. Values:             <ul style="list-style-type: none"> <li>0x01 - Bit 8 is spare and set to 0 in octet</li> <li>0x02 - Bits 7 and 8 are spare and set to 0 in octet n</li> <li>0x03 - Bits 6 to 8 (inclusive) are spare and set to 0 in octet n</li> <li>0x04 - Bits 5 to 8 (inclusive) are spare and set to 0 in octet n</li> <li>0x05 - Bits 4 to 8 (inclusive) are spare and set to 0 in octet n</li> <li>0x06 - Bits 3 to 8 (inclusive) are spare and set to 0 in octet n</li> <li>0x07 - Bits 2 to 8 (inclusive) are spare and set to 0 in octet n</li> <li>0x00 - Carries no information about the number of spare bits in octet n</li> </ul> </li> </ul>
<i>shortNameLen</i>	<ul style="list-style-type: none"> <li>Length of shortName which follows</li> </ul>
<i>shortName</i>	<ul style="list-style-type: none"> <li>PLMN short name</li> </ul>
<i>longNameEn</i>	<ul style="list-style-type: none"> <li>Coding scheme used for PLMN long name. This value is ignored if PLMN long name length is zero Values:             <ul style="list-style-type: none"> <li>0x00 - SMS default 7-bit coded alphabet as defined in 3GPP TS 23.038 with bit 8 set to 0</li> <li>0x01 - UCS2 (16 bit, little-endian) 3GPP TS 23.038</li> </ul> </li> </ul>

<i>longNameCI</i>	<ul style="list-style-type: none"> <li>Indicates whether the country initials are to be added to the longName. This value is ignored if longNameLen is zero. Values: <ul style="list-style-type: none"> <li>0x00 - Do not add the letters for the countrys initials to the name</li> <li>0x01 - Add the countrys initials and a text string to the name</li> <li>0xFF - Not specified</li> </ul> </li> </ul>
<i>longNameSB</i>	<ul style="list-style-type: none"> <li>PLMN long name spare bits. This value is ignored if longNameLen is zero. Values: <ul style="list-style-type: none"> <li>0x01 - Bit 8 is spare and set to 0 in octet</li> <li>0x02 - Bits 7 and 8 are spare and set to 0 in octet n</li> <li>0x03 - Bits 6 to 8 (inclusive) are spare and set to 0 in octet n</li> <li>0x04 - Bits 5 to 8 (inclusive) are spare and set to 0 in octet n</li> <li>0x05 - Bits 4 to 8 (inclusive) are spare and set to 0 in octet n</li> <li>0x06 - Bits 3 to 8 (inclusive) are spare and set to 0 in octet n</li> <li>0x07 - Bits 2 to 8 (inclusive) are spare and set to 0 in octet n</li> <li>0x00 - Carries no information about the number of spare bits in octet n</li> </ul> </li> </ul>
<i>longNameLen</i>	<ul style="list-style-type: none"> <li>Length of longName which follows</li> </ul>
<i>longName</i>	<ul style="list-style-type: none"> <li>PLMN long name</li> </ul>

## Note

None

## 8.289.2 Field Documentation

8.289.2.1 BYTE nasPLMNNameResp::longName[255]

8.289.2.2 BYTE nasPLMNNameResp::longNameCI

8.289.2.3 BYTE nasPLMNNameResp::longNameEn

8.289.2.4 BYTE nasPLMNNameResp::longNameLen

8.289.2.5 BYTE nasPLMNNameResp::longNameSB

8.289.2.6 BYTE nasPLMNNameResp::shortName[255]

8.289.2.7 BYTE nasPLMNNameResp::shortNameCI

8.289.2.8 BYTE nasPLMNNameResp::shortNameEn

8.289.2.9 BYTE nasPLMNNameResp::shortNameLen

8.289.2.10 BYTE nasPLMNNameResp::shortNameSB

8.289.2.11 BYTE nasPLMNNameResp::spn[255]

8.289.2.12 BYTE nasPLMNNameResp::spnEncoding

8.289.2.13 BYTE nasPLMNNameResp::spnLength

## 8.290 nasSigInfo Struct Reference

### Data Fields

- [CDMASSInfo](#) \* [pCDMASigInfo](#)
- [HDRSSInfo](#) \* [pHDRSigInfo](#)
- [INT8](#) \* [pGSMSigInfo](#)
- [CDMASSInfo](#) \* [pWCDMASigInfo](#)
- [LTESSInfo](#) \* [pLTESigInfo](#)
- [INT8](#) \* [pRscp](#)
- [TDSCDMASigInfoExt](#) \* [pTDSCDMASigInfoExt](#)

### 8.290.1 Detailed Description

Structure for storing the [nasSigInfo](#) indication parameters.

#### Parameters

<i>pCDMASigInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">CDMASSInfo</a> for more information.</li> </ul>
<i>pHDRSigInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">HDRSSInfo</a> for more information.</li> </ul>
<i>pGSMSigInfo</i>	<ul style="list-style-type: none"> <li>• one byte value, GSM signal strength is the RSSI in dBm (signed value). A value of -125 dBm or lower is used to indicate No Signal</li> </ul>
<i>pWCDMASigInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">CDMASSInfo</a> for more information.</li> </ul>
<i>pLTESigInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">LTESSInfo</a> for more information.</li> </ul>
<i>pRscp</i>	<ul style="list-style-type: none"> <li>• RSCP of the Primary Common Control Physical Channel (PCCPCH) in dBm. Measurement range: -120 dBm to -25 dBm.</li> </ul>
<i>pTDSCDMASig-InfoExt</i>	<ul style="list-style-type: none"> <li>• See <a href="#">TDSCDMASigInfoExt</a> for more information.</li> </ul>

### 8.290.2 Field Documentation

8.290.2.1 CDMASigInfo\* nasSigInfo::pCDMASigInfo

8.290.2.2 INT8\* nasSigInfo::pGMSigInfo

8.290.2.3 HDRSSigInfo\* nasSigInfo::pHDRSigInfo

8.290.2.4 LTESigInfo\* nasSigInfo::pLTESigInfo

8.290.2.5 INT8\* nasSigInfo::pRscp

8.290.2.6 TDSCDMASigInfoExt\* nasSigInfo::pTDSCDMASigInfoExt

8.290.2.7 CDMASigInfo\* nasSigInfo::pWCDMASigInfo

## 8.291 NasSwlndReg Struct Reference

### Data Fields

- [BYTE lteEsmUI](#)
- [BYTE lteEsmDI](#)
- [BYTE lteEmmUI](#)
- [BYTE lteEmmDI](#)
- [BYTE gsmUmtsUI](#)
- [BYTE gsmUmtsDI](#)

### 8.291.1 Detailed Description

This structure contains the OTA message indication.

#### Parameters

<i>lteEsmUI</i>	<ul style="list-style-type: none"> <li>• 0 - do not report</li> <li>• 1 - report LTE ESM uplink messages</li> </ul>
<i>lteEsmDI</i>	<ul style="list-style-type: none"> <li>• 0 - do not report</li> <li>• 1 - report LTE ESM downlink messages</li> </ul>
<i>lteEmmUI</i>	<ul style="list-style-type: none"> <li>• 0 - do not report</li> <li>• 1 - report LTE EMM uplink messages</li> </ul>
<i>lteEmmDI</i>	<ul style="list-style-type: none"> <li>• 0 - do not report</li> <li>• 1 - report GSM/UMTS uplink messages</li> </ul>

<i>gsmUmtsUI</i>	<ul style="list-style-type: none"> <li>• 0 - do not report</li> <li>• 1 - report GSM/UMTS uplink messages</li> </ul>
<i>gsmUmtsDI</i>	<ul style="list-style-type: none"> <li>• 0 - do not report</li> <li>• 1 - report GSM/UMTS downlink messages</li> </ul>

## 8.291.2 Field Documentation

8.291.2.1 **BYTE** NasSwlndReg::gsmUmtsDI

8.291.2.2 **BYTE** NasSwlndReg::gsmUmtsUI

8.291.2.3 **BYTE** NasSwlndReg::lteEmmDI

8.291.2.4 **BYTE** NasSwlndReg::lteEmmUI

8.291.2.5 **BYTE** NasSwlndReg::lteEsmDI

8.291.2.6 **BYTE** NasSwlndReg::lteEsmUI

## 8.292 nasSysInfo Struct Reference

### Data Fields

- SrvStatusInfo \* pCDMASrvStatusInfo
- SrvStatusInfo \* pHDRSrvStatusInfo
- GSMSrvStatusInfo \* pGSMSrvStatusInfo
- GSMSrvStatusInfo \* pWCDMASrvStatusInfo
- GSMSrvStatusInfo \* pLTERsrvStatusInfo
- CDMA SysInfo \* pCDMA SysInfo
- HDR SysInfo \* pHDR SysInfo
- GSM SysInfo \* pGSM SysInfo
- WCDMA SysInfo \* pWCDMA SysInfo
- LTE SysInfo \* pLTE SysInfo
- AddCDMA SysInfo \* pAddCDMA SysInfo
- WORD \* pAddHDR SysInfo
- Add SysInfo \* pAddGSM SysInfo
- Add SysInfo \* pAddWCDMA SysInfo
- WORD \* pAddLTE SysInfo
- CallBarring SysInfo \* pGSM CallBarring SysInfo
- CallBarring SysInfo \* pWCDMA CallBarring SysInfo
- BYTE \* pLTEVoiceSupport SysInfo
- BYTE \* pGSM CipherDomain SysInfo
- BYTE \* pWCDMA CipherDomain SysInfo
- BYTE \* pSysInfoNoChange

### 8.292.1 Detailed Description

Structure for storing the [nasSysInfo](#) indication parameters.

## Parameters

<i>pCDMASrvStatusInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">SrvStatusInfo</a> for more information.</li> </ul>
<i>pHDRSrvStatusInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">SrvStatusInfo</a> for more information.</li> </ul>
<i>pGSMSrvStatusInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">GSMSrvStatusInfo</a> for more information.</li> </ul>
<i>pWCDMASrvStatusInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">GSMSrvStatusInfo</a> for more information.</li> </ul>
<i>pLTESrvStatusInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">GSMSrvStatusInfo</a> for more information.</li> </ul>
<i>pCDMASysInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">CDMASysInfo</a> for more information.</li> </ul>
<i>pHDRSysInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">HDRSysInfo</a> for more information.</li> </ul>
<i>pGSMSysInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">GSMSysInfo</a> for more information.</li> </ul>
<i>pWCDMASysInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">WCDMASysInfo</a> for more information.</li> </ul>
<i>pLTESysInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">LTESysInfo</a> for more information.</li> </ul>
<i>pAddCDMASysInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">AddCDMASysInfo</a> for more information.</li> </ul>
<i>pAddHDRSysInfo</i>	<ul style="list-style-type: none"> <li>• System table index referencing the beginning of the geo in which the current serving system is present.</li> <li>• When the system index is not known, 0xFFFF is used.</li> </ul>
<i>pAddGSMSysInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">AddSysInfo</a> for more information.</li> </ul>
<i>pAddWCDMA-SysInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">AddSysInfo</a> for more information.</li> </ul>
<i>pAddLTESysInfo</i>	<ul style="list-style-type: none"> <li>• System table index referencing the beginning of the geo in which the current serving system is present.</li> <li>• When the system index is not known, 0xFFFF is used.</li> </ul>

<i>pGSMCall-BarringSysInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">CallBarringSysInfo</a> for more information.</li> </ul>
<i>pWCDMACall-BarringSysInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">CallBarringSysInfo</a> for more information.</li> </ul>
<i>pLTEVoice-SupportSysInfo</i>	<ul style="list-style-type: none"> <li>• Indicates voice support status on LTE. <ul style="list-style-type: none"> <li>– 0x00 - Voice is not supported</li> <li>– 0x01 - Voice is supported</li> </ul> </li> </ul>
<i>pGSMCipher-DomainSysInfo</i>	<ul style="list-style-type: none"> <li>• Ciphering on the service domain. <ul style="list-style-type: none"> <li>– 0x00 - No service</li> <li>– 0x01 - Circuit-switched only</li> <li>– 0x02 - Packet-switched only</li> <li>– 0x03 - Circuit-switched and packet-switched</li> </ul> </li> </ul>
<i>pWCDMA-CipherDomain-SysInfo</i>	<ul style="list-style-type: none"> <li>• Ciphering on the service domain. <ul style="list-style-type: none"> <li>– 0x00 - No service</li> <li>– 0x01 - Circuit-switched only</li> <li>– 0x02 - Packet-switched only</li> <li>– 0x03 - Circuit-switched and packet-switched</li> </ul> </li> </ul>
<i>pSysInfoNo-Change</i>	<ul style="list-style-type: none"> <li>• System Info No Change.</li> <li>• Flag used to notify clients that a request to select a network ended with no change in the PLMN. <ul style="list-style-type: none"> <li>– 0x01 - No change in system information</li> </ul> </li> </ul>

## 8.292.2 Field Documentation

8.292.2.1 **AddCDMASysInfo\*** nasSysInfo::pAddCDMASysInfo

8.292.2.2 **AddSysInfo\*** nasSysInfo::pAddGSMSysInfo

8.292.2.3 **WORD\*** nasSysInfo::pAddHDRSysInfo

8.292.2.4 **WORD\*** nasSysInfo::pAddLTESysInfo

8.292.2.5 **AddSysInfo\*** nasSysInfo::pAddWCDMASysInfo

8.292.2.6 **SrvStatusInfo\*** nasSysInfo::pCDMASrvStatusInfo



- 8.292.2.7 **CDMASysInfo\*** nasSysInfo::pCDMASysInfo
- 8.292.2.8 **CallBarringSysInfo\*** nasSysInfo::pGSMCallBarringSysInfo
- 8.292.2.9 **BYTE\*** nasSysInfo::pGSMCipherDomainSysInfo
- 8.292.2.10 **GSMSrvStatusInfo\*** nasSysInfo::pGSMSrvStatusInfo
- 8.292.2.11 **GSMSysInfo\*** nasSysInfo::pGSMSysInfo
- 8.292.2.12 **SrvStatusInfo\*** nasSysInfo::pHDRSrvStatusInfo
- 8.292.2.13 **HDRSysInfo\*** nasSysInfo::pHDRSysInfo
- 8.292.2.14 **GSMSrvStatusInfo\*** nasSysInfo::pLTERsrvStatusInfo
- 8.292.2.15 **LTESysInfo\*** nasSysInfo::pLTESysInfo
- 8.292.2.16 **BYTE\*** nasSysInfo::pLTEVoiceSupportSysInfo
- 8.292.2.17 **BYTE\*** nasSysInfo::pSysInfoNoChange
- 8.292.2.18 **CallBarringSysInfo\*** nasSysInfo::pWCDMACallBarringSysInfo
- 8.292.2.19 **BYTE\*** nasSysInfo::pWCDMACipherDomainSysInfo
- 8.292.2.20 **GSMSrvStatusInfo\*** nasSysInfo::pWCDMASrvStatusInfo
- 8.292.2.21 **WCDMASysInfo\*** nasSysInfo::pWCDMASysInfo

## 8.293 netSelectionPref Struct Reference

### Data Fields

- [BYTE](#) netReg
- [WORD](#) mcc
- [WORD](#) mnc

### 8.293.1 Detailed Description

Contain the network selection preference.

## Parameters

<i>netReg</i>	<ul style="list-style-type: none"> <li>• specifies one of the following actions: <ul style="list-style-type: none"> <li>– 0x00 - Automatic registration <ul style="list-style-type: none"> <li>* Device registers according to its provisioning; mcc and mnc fields are ignored</li> </ul> </li> <li>– 0x01 - Manual Registration <ul style="list-style-type: none"> <li>* Device registers to specified network; mcc and mnc must contain valid values</li> </ul> </li> </ul> </li> </ul>
<i>mcc</i>	<ul style="list-style-type: none"> <li>• MCC value. Range 0 to 999</li> </ul>
<i>mnc</i>	<ul style="list-style-type: none"> <li>• MNC value. Range 0 to 999</li> </ul>

## 8.293.2 Field Documentation

8.293.2.1 WORD netSelectionPref::mcc

8.293.2.2 WORD netSelectionPref::mnc

8.293.2.3 BYTE netSelectionPref::netReg

## 8.294 NetStats Struct Reference

## Data Fields

- [ULONG rx\\_packets](#)
- [ULONG tx\\_packets](#)
- [ULONGLONG rx\\_bytes](#)
- [ULONGLONG tx\\_bytes](#)
- [ULONG rx\\_errors](#)
- [ULONG tx\\_errors](#)
- [ULONG rx\\_overflows](#)
- [ULONG tx\\_overflows](#)

## 8.294.1 Detailed Description

This structure contains the SLQSGetNetStatistic Information

## Parameters

<i>rx_packets</i>	<ul style="list-style-type: none"> <li>• Number of received Packets without error</li> </ul>
-------------------	--

<i>tx_packets</i>	<ul style="list-style-type: none"> <li>• Number of transmitted Packets without error</li> </ul>
<i>rx_bytes</i>	<ul style="list-style-type: none"> <li>• Number of bytes recieved without error</li> </ul>
<i>tx_bytes</i>	<ul style="list-style-type: none"> <li>• NNumero of bytes transmitted without error</li> </ul>
<i>rx_error</i>	<ul style="list-style-type: none"> <li>• Number of incoming packets with framing errors</li> </ul>
<i>tx_error</i>	<ul style="list-style-type: none"> <li>• Number of outgoing packets with framing errors</li> </ul>
<i>rx_overflows</i>	<ul style="list-style-type: none"> <li>• Number of packets dropped because Rx buffer overflowed</li> </ul>
<i>tx_overflows</i>	<ul style="list-style-type: none"> <li>• Number of packets dropped because Tx buffer overflowed</li> </ul>

## 8.294.2 Field Documentation

8.294.2.1 **ULONGLONG** NetStats::rx\_bytes

8.294.2.2 **ULONG** NetStats::rx\_errors

8.294.2.3 **ULONG** NetStats::rx\_overflows

8.294.2.4 **ULONG** NetStats::rx\_packets

8.294.2.5 **ULONGLONG** NetStats::tx\_bytes

8.294.2.6 **ULONG** NetStats::tx\_errors

8.294.2.7 **ULONG** NetStats::tx\_overflows

8.294.2.8 **ULONG** NetStats::tx\_packets

## 8.295 NetworkDebugResp Struct Reference

### Data Fields

- [BYTE](#) \* pObjectVer
- [NetworkStat1x](#) \* pNetworkStat1x
- [NetworkStatEVDO](#) \* pNetworkStatEVDO
- [DeviceConfigDetail](#) \* pDeviceConfigDetail
- [DataStatusDetail](#) \* pDataStatusDetail

### 8.295.1 Detailed Description

This structure contains information about the SLQSSwiNetworkDebug response parameters.

## Parameters

<i>pObjectVer</i>	<ul style="list-style-type: none"> <li>Object's version number for the host to handle <ul style="list-style-type: none"> <li>0xFF - NA</li> <li>Others - shows in decimal</li> </ul> </li> </ul>
<i>pNetworkStat1x</i>	<ul style="list-style-type: none"> <li>See <a href="#">NetworkStat1x</a> for more information</li> </ul>
<i>pNetworkStatEVDO</i>	<ul style="list-style-type: none"> <li>See <a href="#">NetworkStatEVDO</a> for more information.</li> </ul>
<i>pDeviceConfigDetail</i>	<ul style="list-style-type: none"> <li>See <a href="#">DeviceConfigDetail</a> for more information.</li> </ul>
<i>pDataStatusDetail</i>	<ul style="list-style-type: none"> <li>See <a href="#">DataStatusDetail</a> for more information.</li> </ul>

## 8.295.2 Field Documentation

8.295.2.1 [DataStatusDetail](#)\* [NetworkDebugResp::pDataStatusDetail](#)8.295.2.2 [DeviceConfigDetail](#)\* [NetworkDebugResp::pDeviceConfigDetail](#)8.295.2.3 [NetworkStat1x](#)\* [NetworkDebugResp::pNetworkStat1x](#)8.295.2.4 [NetworkStatEVDO](#)\* [NetworkDebugResp::pNetworkStatEVDO](#)8.295.2.5 [BYTE](#)\* [NetworkDebugResp::pObjectVer](#)

## 8.296 NetworkStat1x Struct Reference

## Data Fields

- [BYTE](#) State
- [WORD](#) SO
- [ULONG](#) RX\_PWR
- [WORD](#) RX\_EC\_IO
- [ULONG](#) TX\_PWR
- [BYTE](#) ActSetCnt
- [ActPilotPNElement](#) \* [pActPilotPNElements](#)
- [BYTE](#) NeighborSetCnt
- [WORD](#) \* [pNeighborSetPilotPN](#)

## 8.296.1 Detailed Description

This structure contains 1x network status details

## Parameters

<i>State</i>	<ul style="list-style-type: none"> <li>• CDMA current access state               <ul style="list-style-type: none"> <li>– 0x01 - Searching</li> <li>– 0x02 - Idle</li> <li>– 0x03 - Traffic</li> <li>– Others - NA</li> </ul> </li> </ul>
<i>SO</i>	<ul style="list-style-type: none"> <li>• CDMA service option               <ul style="list-style-type: none"> <li>– 0xFFFF - Not in a call</li> <li>– 0x0001 - Basic Variable Rate Voice Service(8kbps)</li> <li>– 0x0002 - Mobile Station Loopback(8kbps)</li> <li>– 0x0003 - Enhanced Variable Rate Codec(EVRC) Voice Service(8kbps)</li> <li>– 0x0006 - Short message Services(Rate Set 1)</li> <li>– 0x0009 - Mobile Station Loopback(13kbps)</li> <li>– 0x000E - Short Message Service (Rate Set 2)</li> <li>– 0x0011 - High Rate Voice Service(13kbps)</li> <li>– 0x0020 - Test Data Service Option(TDSO)</li> <li>– 0x0021 - cdma2000 High Speed Packet Data Service, Internet or ISO Protocol Stack</li> <li>– 0x0044 - EVRC-B Voice Service(8 kbps)</li> <li>– 0x0046 - EVRC-WB Voice Service(8 kbps)</li> <li>– 0x0049 - Voice Echo mode supports smart blanking(EVRC-NW)</li> <li>– 0x004B - Enhanced loopback</li> <li>– 0x8000 - Proprietary Service Option (Qualcomm Inc.)</li> </ul> </li> </ul>

<i>RX_PWR</i>	<ul style="list-style-type: none"> <li>• RX Pwr(dBm) <ul style="list-style-type: none"> <li>– 0xABCD00EF - -ABCD.EF dBm</li> <li>– ABCD00EF should be transferred to decimal while displaying</li> <li>– Example: 0x12340056 - -4660.86dBm 0x1234 = 4660, 0x0056 = 86</li> <li>– 0xFFFFFFFF - NA</li> </ul> </li> </ul>
<i>RX_EC_IO</i>	<ul style="list-style-type: none"> <li>• RX EC/IO(dB) <ul style="list-style-type: none"> <li>– 0xABCD - -AB.CD dB</li> <li>– ABCD should be transferred to decimal while displaying</li> <li>– Example: 0x1234 - -18.52dB 0x12 = 18, 0x34 = 52</li> <li>– 0xFFFF - NA</li> </ul> </li> </ul>
<i>TX_PWR</i>	<ul style="list-style-type: none"> <li>• TX PWR(dBm) <ul style="list-style-type: none"> <li>– 0xFFFFFFFF - NA</li> <li>– Others - display actual value in decimal</li> <li>– Example: 0x1234 - -4660dBm 0x1234 = 4660</li> </ul> </li> </ul>
<i>ActSetCnt(</i>	IN/OUT ) <ul style="list-style-type: none"> <li>• Count of active pilot PN elements</li> <li>• As input specifies number of sets of parameter pActPilotElements for which memory has been assigned</li> <li>• As output specifies the actual number of sets of parameter pActPilotElements returned by device</li> </ul>
<i>pActPilotPN-Elements</i>	<ul style="list-style-type: none"> <li>• See <a href="#">ActPilotPNElement</a> for more information</li> </ul>
<i>NeighborSetCnt(</i>	IN/OUT ) <ul style="list-style-type: none"> <li>• Count of neighbor pilot PN elements</li> <li>• As input specifies number of sets of parameter pNeighborSetPilotPN for which memory has been assigned</li> <li>• As output specifies the actual number of sets of parameter pNeighborSetPilotPN returned by device</li> </ul>

<i>pNeighborSet-PilotPN</i>	<ul style="list-style-type: none"><li>• Neighbor pilot PN</li></ul>
-----------------------------	---

## 8.296.2 Field Documentation

8.296.2.1 **BYTE** NetworkStat1x::ActSetCnt

8.296.2.2 **BYTE** NetworkStat1x::NeighborSetCnt

8.296.2.3 **ActPilotPNElement\*** NetworkStat1x::pActPilotPNElements

8.296.2.4 **WORD\*** NetworkStat1x::pNeighborSetPilotPN

8.296.2.5 **WORD** NetworkStat1x::RX\_EC\_IO

8.296.2.6 **ULONG** NetworkStat1x::RX\_PWR

8.296.2.7 **WORD** NetworkStat1x::SO

8.296.2.8 **BYTE** NetworkStat1x::State

8.296.2.9 **ULONG** NetworkStat1x::TX\_PWR

## 8.297 NetworkStatEVDO Struct Reference

### Data Fields

- [BYTE State](#)
- [BYTE MACIndex](#)
- [BYTE SectorIDLen](#)
- [WORD \\* pSectorID](#)
- [WORD RX\\_PWR](#)
- [WORD PER](#)
- [WORD PilotEnergy](#)
- [BYTE SNR](#)

### 8.297.1 Detailed Description

This structure contains EVDO network status details



## Parameters

<i>State</i>	<ul style="list-style-type: none"> <li>• EVDO network access state <ul style="list-style-type: none"> <li>– 0x00 - Sleep</li> <li>– 0x01 - Searching</li> <li>– 0x02 - Idle</li> <li>– 0x03 - Active</li> <li>– 0xFF - NA</li> </ul> </li> </ul>
<i>MACIndex</i>	<ul style="list-style-type: none"> <li>• HDR Mac index <ul style="list-style-type: none"> <li>– 0xFF - NA</li> <li>– Others - Display the actual value in decimal</li> <li>– Example: 0x12 - 18 0x12 = 18</li> </ul> </li> </ul>
<i>SectorIDLen</i>	(IN/OUT) <ul style="list-style-type: none"> <li>• Sector ID length</li> <li>• As input specifies length of parameter pSectorID for which memory has been assigned</li> <li>• As output specifies the actual length of parameter pSectorID returned by device</li> </ul>
<i>pSectorID</i>	<ul style="list-style-type: none"> <li>• Sector ID</li> </ul>
<i>RX_PWR</i>	<ul style="list-style-type: none"> <li>• TX PWR(dBm) <ul style="list-style-type: none"> <li>– 0xABCD - -ABCD dBm</li> <li>– ABCD should be transferred to decimal while displaying</li> <li>– Example: 0x1234 - -4660dBm 0x1234 = 4660</li> <li>– 0xFFFF - NA</li> </ul> </li> </ul>

<i>PER</i>	<ul style="list-style-type: none"> <li>• HDR Packet Error Rate <ul style="list-style-type: none"> <li>– 0xFFFF - Unknown</li> <li>– Others - display the actual value in decimal</li> <li>– Example: 0x1234 - -4660dBm 0x1234 = 4660</li> </ul> </li> </ul>
<i>PilotEnergy</i>	<ul style="list-style-type: none"> <li>• Pilt Energy (dB) <ul style="list-style-type: none"> <li>– 0xFFFF - NA</li> <li>– 0xABCD should be transferred to decimal while displaying</li> <li>– Example: 0x1234 - -4660dBm 0x1234 = 4660</li> </ul> </li> </ul>
<i>SNR</i>	<ul style="list-style-type: none"> <li>• Signal to Noise ratio (dB)</li> </ul>

## 8.297.2 Field Documentation

8.297.2.1 **BYTE** NetworkStatEVDO::MACIndex

8.297.2.2 **WORD** NetworkStatEVDO::PER

8.297.2.3 **WORD** NetworkStatEVDO::PilotEnergy

8.297.2.4 **WORD\*** NetworkStatEVDO::pSectorID

8.297.2.5 **WORD** NetworkStatEVDO::RX\_PWR

8.297.2.6 **BYTE** NetworkStatEVDO::SectorIDLen

8.297.2.7 **BYTE** NetworkStatEVDO::SNR

8.297.2.8 **BYTE** NetworkStatEVDO::State

## 8.298 newPwdData Struct Reference

### Data Fields

- [BYTE newPwd](#) [4]
- [BYTE newPwdAgain](#) [4]

### 8.298.1 Detailed Description

This structure contains New Password Data.

## Parameters

<i>newPwd</i> [PASS-WORD_LENGTH]	<ul style="list-style-type: none"> <li>New password. <ul style="list-style-type: none"> <li>Password consists of 4 ASCII digits.</li> <li>Range: 0000 to 9999.</li> </ul> </li> </ul>
<i>newPwdAgain</i> [PASSWORD_LENGTH]	<ul style="list-style-type: none"> <li>New password again. <ul style="list-style-type: none"> <li>Password consists of 4 ASCII digits.</li> <li>Range: 0000 to 9999.</li> </ul> </li> </ul>

## 8.298.2 Field Documentation

8.298.2.1 BYTE newPwdData::newPwd[4]

8.298.2.2 BYTE newPwdData::newPwdAgain[4]

## 8.299 nmrCellInfo Struct Reference

## Data Fields

- [ULONG nmrCellID](#)
- [BYTE nmrPlmn](#) [3]
- [WORD nmrLac](#)
- [WORD nmrArfcn](#)
- [BYTE nmrBsic](#)
- [WORD nmrRxLev](#)

## 8.299.1 Detailed Description

This structure contains information about the Network Measurement Report (NMR) Cell Information.

## Parameters

<i>nmrCellID</i>	<ul style="list-style-type: none"> <li>Cell ID.</li> <li>0xFFFFFFFF indicates cell ID information is not present.</li> </ul>
<i>nmrPlmn</i> [PLMN_LENGTH]	<ul style="list-style-type: none"> <li>MCC/MNC information coded as octet 3, 4, and 5.</li> <li>This field is ignored when nmrCellID is not present.</li> </ul>

<i>nmrLac</i>	<ul style="list-style-type: none"> <li>• Location area code.</li> <li>• This field is ignored when nmrCellID is not present. <ul style="list-style-type: none"> <li>– 0xFFFF - Not Available</li> </ul> </li> </ul>
<i>nmrArfcn</i>	<ul style="list-style-type: none"> <li>• Absolute RF channel number. <ul style="list-style-type: none"> <li>– 0xFFFF - Not Available</li> </ul> </li> </ul>
<i>nmrBsic</i>	<ul style="list-style-type: none"> <li>• Base station identity code. <ul style="list-style-type: none"> <li>– 0xFF - Not Available</li> </ul> </li> </ul>
<i>nmrRxLev</i>	<ul style="list-style-type: none"> <li>• Cell Rx measurement.</li> <li>• Values range between 0 and 63.</li> <li>• Mapped to a measured signal level: <ul style="list-style-type: none"> <li>– Rxlev 0 is a signal strength less than -110 dBm</li> <li>– Rxlev 1 is -110 dBm to -109 dBm</li> <li>– Rxlev 2 is -109 dBm to -108 dBm</li> <li>– ...</li> <li>– Rxlev 62 is -49 dBm to -48 dBm</li> <li>– Rxlev 63 is greater than -48 dBm</li> <li>– 0xFFFF - Not Available</li> </ul> </li> </ul>

## 8.299.2 Field Documentation

8.299.2.1 WORD nmrCellInfo::nmrArfcn

8.299.2.2 BYTE nmrCellInfo::nmrBsic

8.299.2.3 ULONG nmrCellInfo::nmrCellID

8.299.2.4 WORD nmrCellInfo::nmrLac

8.299.2.5 BYTE nmrCellInfo::nmrPlmn[3]

8.299.2.6 WORD nmrCellInfo::nmrRxLev

## 8.300 NSSAudioCtrl Struct Reference

### Data Fields

- [BYTE upLink](#)

- [BYTE downLink](#)

### 8.300.1 Detailed Description

This structure contains National Supplementary Services - Audio Control Information

Parameters

<i>upLink</i>	<ul style="list-style-type: none"><li>• Values as per[ S24, 4.10 Reservation response].</li></ul>
<i>downLink</i>	<ul style="list-style-type: none"><li>• Values as per[ S24, 4.10 Reservation response].</li></ul>

### 8.300.2 Field Documentation

8.300.2.1 **BYTE** NSSAudioCtrl::downLink

8.300.2.2 **BYTE** NSSAudioCtrl::upLink

## 8.301 NWProfile Struct Reference

Data Fields

- [WORD tech](#)
- [BYTE \\* pProfSz](#)
- [WORD \\* pProfValues](#)

### 8.301.1 Detailed Description

This structure contains Network supported QoS profile

Parameters

<i>tech</i>	Technology on which the network supported QoS profiles are being returned: <ul style="list-style-type: none"><li>• CDMA – 0x8001</li></ul>
<i>exponent</i>	

### 8.301.2 Field Documentation

8.301.2.1 **BYTE\*** NWProfile::pProfSz

8.301.2.2 **WORD\*** NWProfile::pProfValues

8.301.2.3 **WORD** NWProfile::tech

## 8.302 omaDmConfigTlv Struct Reference

## Data Fields

- [BYTE](#) `state`
- [BYTE](#) `userInputReq`
- [USHORT](#) `userInputTimeout`
- [USHORT](#) `alertmsglength`
- [BYTE](#) `alertmsg` [256]

### 8.302.1 Detailed Description

This structure will hold the SwiOmaDmConfig session parameters information.

#### Parameters

<i>state</i>	<ul style="list-style-type: none"> <li>• 0x01 - OMA-DM Read Request</li> <li>• 0x02 - OMA-DM Change Request</li> <li>• 0x03 - OMA-DM Config Complete</li> </ul>
<i>user_input_req</i>	- Bit mask of available user inputs <ul style="list-style-type: none"> <li>• 0x00 - No user input required. Informational indication</li> <li>• 0x01 - Accept</li> <li>• 0x02 - Reject</li> </ul>
<i>user_input_timeout</i>	<ul style="list-style-type: none"> <li>• Timeout for user input in minutes. A value of 0 means no time-out</li> </ul>
<i>alertmsglength</i>	<ul style="list-style-type: none"> <li>• Length of Alert message string in bytes</li> </ul>
<i>alertmsg</i>	<ul style="list-style-type: none"> <li>• Alert message in UCS2 (Max 256 characters)</li> </ul>

### 8.302.2 Field Documentation

8.302.2.1 [BYTE](#) `omaDmConfigTlv::alertmsg[256]`

8.302.2.2 [USHORT](#) `omaDmConfigTlv::alertmsglength`

8.302.2.3 [BYTE](#) `omaDmConfigTlv::state`

8.302.2.4 [BYTE](#) `omaDmConfigTlv::userInputReq`

8.302.2.5 [USHORT](#) `omaDmConfigTlv::userInputTimeout`

## 8.303 omaDmConfigTlvExt Struct Reference

## Data Fields

- [BYTE](#) `state`

- [BYTE](#) `userInputReq`
- [USHORT](#) `userInputTimeout`
- [USHORT](#) `alertmsglength`
- [BYTE](#) `alertmsg` [256]

#### 8.303.1 Detailed Description

This structure will hold the SwiOmaDmConfig session parameters information.

## Parameters

<i>state</i>	<ul style="list-style-type: none"> <li>• 1 - reserved</li> <li>• 2 - reserved</li> <li>• 3 - reserved</li> <li>• 4 - CI DC Success</li> <li>• 5 - CI DC Failure</li> <li>• 6 - User/device initiated PRL update success.</li> <li>• 7 - User/device initiated PRL update failure.</li> <li>• 8 - HFA DC session start</li> <li>• 9 - HFA DC success.</li> <li>• 10 - HFA is cancelled.</li> <li>• 11 - HFA retry. UI Screen 13[1] with 0 percent progress bar should be shown.</li> <li>• 12 - HFA fail after 5 retries. UI Screen 2[1] should be displayed.</li> <li>• 13 - HFA retry down counter. Used to update the process bar of UI Screen 13[1].</li> <li>• 14 - HFA PRL session start, UI screen 4[1] should be displayed.</li> <li>• 15 - HFA PRL update success.</li> <li>• 16 - Device is launching a NI session. UI Screen 1[1] should be displayed.</li> <li>• 17 - An empty session. UI Screen 2[1] should be displayed.</li> <li>• 18 - No network coverage.</li> <li>• 19 - HFA is not enabled.</li> <li>• 20 - CI DC Start, UI Screen 1[1] should be displayed.</li> <li>• 21 - CI PRL start, UI screen 4[1] should be displayed.</li> <li>• 22 - HFA PRL updates fail.</li> <li>• 23 - Device reboot.</li> <li>• 24 - CI DC is cancelled.</li> <li>• 25 - User/device initiated PRL update is cancelled.</li> <li>• 26 - NI session is cancelled.</li> <li>• 27 - Current NI session is not enabled.</li> <li>• 28 - NI DC success.</li> <li>• 29 - NI DC Fail.</li> <li>• 30 - NI PRL success</li> <li>• 31 - NI PRL fail.</li> <li>• 32 - Reserved</li> <li>• 33 - NI fumo fail</li> <li>• 34 - NI session fail, unable to point out the session type.</li> </ul>
<i>user_input_req</i>	<p>- OMA task stop to wait user's input if this field is valid until user input selection or after "UI Timer out (next field). In the case of timeout, a default selection of "YES/OK" is accepted. Note that this option is valid when DM state is 4/6/12/28/30. 0 - user/host doesn't need to input anything, and OMA task doesn't blocked by UI. 1 - user/host must input "YES/OK/CANCEL". 2 - User/host must input "NO/CANCEL". 3 - user/host must input "YES/OK/NO/CANCEL".</p>



<i>user_input_timeout</i>	<ul style="list-style-type: none"> <li>Timeout for user input in seconds. This indicates how many seconds OMA task stop to wait for host/user's response.</li> </ul>
<i>alertmsglength</i>	<ul style="list-style-type: none"> <li>Length of Alert message string in word(16-bit)</li> </ul>
<i>alertmsg</i>	<ul style="list-style-type: none"> <li>Alert message in UCS2 (Max 256 characters)</li> <li>This string is printed by host</li> </ul>

### 8.303.2 Field Documentation

8.303.2.1 **BYTE** omaDmConfigTlvExt::alertmsg[256]

8.303.2.2 **USHORT** omaDmConfigTlvExt::alertmsglength

8.303.2.3 **BYTE** omaDmConfigTlvExt::state

8.303.2.4 **BYTE** omaDmConfigTlvExt::userInputReq

8.303.2.5 **USHORT** omaDmConfigTlvExt::userInputTimeout

## 8.304 omaDmFotaTlv Struct Reference

### Data Fields

- [BYTE](#) state
- [BYTE](#) userInputReq
- [USHORT](#) userInputTimeout
- [ULONG](#) fwdloadsize
- [ULONG](#) fwloadComplete
- [USHORT](#) updateCompleteStatus
- [BYTE](#) severity
- [USHORT](#) versionlength
- [BYTE](#) version [256]
- [USHORT](#) namelength
- [BYTE](#) package\_name [256]
- [USHORT](#) descriptionlength
- [BYTE](#) description [256]
- [BYTE](#) sessionType

### 8.304.1 Detailed Description

This structure will hold the SwiOmaDmFota session parameters information.

#### Parameters

---

<i>state</i>	<ul style="list-style-type: none"> <li>• 0x01 - No Firmware available</li> <li>• 0x02 - Query Firmware Download</li> <li>• 0x03 - Firmware Downloading</li> <li>• 0x04 - Firmware downloaded</li> <li>• 0x05 - Query Firmware Update</li> <li>• 0x06 - Firmware updating</li> <li>• 0x07 - Firmware updated</li> </ul>
<i>user_input_req</i>	<ul style="list-style-type: none"> <li>- Bit mask of available user inputs</li> <li>• 0x00 - No user input required. Informational indication</li> <li>• 0x01 - Accept</li> <li>• 0x02 - Reject</li> </ul>
<i>user_input_timeout</i>	<ul style="list-style-type: none"> <li>• Timeout for user input in minutes. A value of 0 means no time-out</li> </ul>
<i>fw_dload_size</i>	<ul style="list-style-type: none"> <li>• The size (in bytes) of the firmware update package</li> </ul>
<i>fw_dload_complete</i>	<ul style="list-style-type: none"> <li>• The number of bytes downloaded. Need to determine how often to send this message for progress bar notification. Every 500ms or 5% increment.</li> </ul>
<i>update_complete_status</i>	<ul style="list-style-type: none"> <li>• See table below.</li> </ul>
<i>severity</i>	<ul style="list-style-type: none"> <li>• 0x01 - Mandatory</li> <li>• 0x02 - Optional</li> </ul>
<i>versionlength</i>	<ul style="list-style-type: none"> <li>• Length of FW Version string in bytes</li> </ul>
<i>version</i>	<ul style="list-style-type: none"> <li>• FW Version string in ASCII (Max 256 characters)</li> </ul>
<i>namelength</i>	<ul style="list-style-type: none"> <li>• Length Package Name string in bytes</li> </ul>
<i>package_name</i>	<ul style="list-style-type: none"> <li>• Package Name in UCS2 (Max 256 characters)</li> </ul>
<i>descriptionlength</i>	<ul style="list-style-type: none"> <li>• Length of description in bytes</li> </ul>

<i>description</i>	<ul style="list-style-type: none"> <li>• Description of Update Package in USC2 (Max 256 characters)</li> </ul>
<i>sessionType</i>	<ul style="list-style-type: none"> <li>• 0x00 - Client initiated</li> <li>• 0x01 - Network initiated</li> </ul>

## 8.304.2 Field Documentation

8.304.2.1 **BYTE** omaDmFotaTlv::description[256]

8.304.2.2 **USHORT** omaDmFotaTlv::descriptionlength

8.304.2.3 **ULONG** omaDmFotaTlv::fwddownloadsize

8.304.2.4 **ULONG** omaDmFotaTlv::fwloadComplete

8.304.2.5 **USHORT** omaDmFotaTlv::namelength

8.304.2.6 **BYTE** omaDmFotaTlv::package\_name[256]

8.304.2.7 **BYTE** omaDmFotaTlv::sessionType

8.304.2.8 **BYTE** omaDmFotaTlv::severity

8.304.2.9 **BYTE** omaDmFotaTlv::state

8.304.2.10 **USHORT** omaDmFotaTlv::updateCompleteStatus

8.304.2.11 **BYTE** omaDmFotaTlv::userInputReq

8.304.2.12 **USHORT** omaDmFotaTlv::userInputTimeout

8.304.2.13 **BYTE** omaDmFotaTlv::version[256]

8.304.2.14 **USHORT** omaDmFotaTlv::versionlength

## 8.305 omaDmFotaTlvExt Struct Reference

### Data Fields

- [BYTE](#) state
- [BYTE](#) reserved
- [USHORT](#) userInputTimeout
- [ULONG](#) packageSize
- [ULONG](#) receivedBytes
- [USHORT](#) fumoResultCode
- [USHORT](#) versionlength
- [BYTE](#) version [256]
- [USHORT](#) namelength
- [BYTE](#) package\_name [256]
- [USHORT](#) descriptionlength
- [BYTE](#) description [256]

### 8.305.1 Detailed Description

This structure will hold the SwiOmaDmFota session parameters information.

#### Parameters

<i>state</i>	<ul style="list-style-type: none"> <li>• 0x01 - No Firmware available</li> <li>• 0x02 - reserved</li> <li>• 0x03 - Update progress bar, UI screen 7[1] should be displayed</li> <li>• 0x04 - reserved</li> <li>• 0x05 - FUMO image download success, UI screen 8[1] should be displayed</li> <li>• 0x06 - reserved</li> <li>• 0x07 - FUMO image installation result, used to return error code.</li> <li>• 0x08 - FUMO session start</li> <li>• 0x09 - UI Screen 7[1] with 0 percent bar should be displayed</li> <li>• 0x0A - FUMO image installation is cancelled by user.</li> <li>• 0x0B - FUMO session fail</li> <li>• 0x0C - Device is sending a report to OMA Server.</li> <li>• 0x0D - Report to server success.</li> <li>• 0x0E - Report to server fails.</li> <li>• 0x0F - reserved</li> <li>• 0x10 - FUMO session is cancelled before image download success.</li> <li>• 0x11 - UI Screen 16[1] should be displayed, FUMO delay option, OMA task is blocked until a valid response is received.</li> </ul>
--------------	---

<i>reserved</i>	- For sprint reserved
<i>user_input_timeout</i>	<ul style="list-style-type: none"> <li>How many seconds OMA task stop to wait for user/host response.</li> </ul>
<i>packageSize</i>	<ul style="list-style-type: none"> <li>The size (in bytes) of the firmware update package (only valid for states 3/5/7).</li> </ul>
<i>receivedBytes</i>	<ul style="list-style-type: none"> <li>The number of bytes downloaded. Useful for FUMO state 3.</li> </ul>
<i>fumoResultCode</i>	<ul style="list-style-type: none"> <li>Used when fumo state is 7/11. REsult code of FUMO image installation <ul style="list-style-type: none"> <li>200 image install success</li> </ul> </li> <li>Others: image install fail</li> </ul>
<i>versionlength</i>	<ul style="list-style-type: none"> <li>Length of FW Version string in bytes</li> </ul>
<i>version</i>	<ul style="list-style-type: none"> <li>FW Version string in ASCII (Max 256 characters)</li> </ul>
<i>namelength</i>	<ul style="list-style-type: none"> <li>Length Package Name string in bytes</li> </ul>
<i>package_name</i>	<ul style="list-style-type: none"> <li>Package Name in UCS2 (Max 256 characters)</li> </ul>
<i>descriptionlength</i>	<ul style="list-style-type: none"> <li>Length of description in bytes</li> </ul>
<i>description</i>	<ul style="list-style-type: none"> <li>Description of Update Package in USC2 (Max 256 characters)</li> </ul>

## 8.305.2 Field Documentation

8.305.2.1 **BYTE** omaDmFotaTlvExt::description[256]

8.305.2.2 **USHORT** omaDmFotaTlvExt::descriptionlength

8.305.2.3 **USHORT** omaDmFotaTlvExt::fumoResultCode

8.305.2.4 **USHORT** omaDmFotaTlvExt::namelength

8.305.2.5 **BYTE** omaDmFotaTlvExt::package\_name[256]

8.305.2.6 **ULONG** omaDmFotaTlvExt::packageSize

8.305.2.7 **ULONG** omaDmFotaTlvExt::receivedBytes

- 8.305.2.8 **BYTE** omaDmFotaTlvExt::reserved
- 8.305.2.9 **BYTE** omaDmFotaTlvExt::state
- 8.305.2.10 **USHORT** omaDmFotaTlvExt::userInputTimeout
- 8.305.2.11 **BYTE** omaDmFotaTlvExt::version[256]
- 8.305.2.12 **USHORT** omaDmFotaTlvExt::versionlength

## 8.306 omaDmNotificationsTlv Struct Reference

### Data Fields

- [BYTE](#) notification
- [USHORT](#) sessionStatus

### 8.306.1 Field Documentation

- 8.306.1.1 **BYTE** omaDmNotificationsTlv::notification
- 8.306.1.2 **USHORT** omaDmNotificationsTlv::sessionStatus

## 8.307 operatorNameString Struct Reference

### Data Fields

- [BYTE](#) [PLMNName](#) [255]

### 8.307.1 Detailed Description

This structure contains Operator Name String as defined in CPHS4\_2.WW6(Feb 27, 1997) (Section B.4.1.2)from multiple sources.

#### Parameters

<i>PLMNName</i>	<ul style="list-style-type: none"><li>• PLMN name must be coded in a default 7-bit alphabet with b8 set to 0.</li></ul>
-----------------	---

### 8.307.2 Field Documentation

- 8.307.2.1 **BYTE** operatorNameString::PLMNName[255]

## 8.308 OperatorPLMNData Struct Reference

### Data Fields

- [BYTE](#) mcc [3]
- [BYTE](#) mnc [3]
- [WORD](#) lac1
- [WORD](#) lac2
- [BYTE](#) PLMNRecID

### 8.308.1 Detailed Description

This structure contains Operator PLMN Data from multiple sources.

#### Parameters

<i>mcc</i>	<ul style="list-style-type: none"><li>MCC in ASCII string (a value of D in any of the digits is to be used to indicate a "wild" value for that corresponding digit).</li></ul>
<i>mnc</i>	<ul style="list-style-type: none"><li>MNC in ASCII string (a value of D in any of the digits is to be used to indicate a "wild" value for that corresponding digit; digit 3 in MNC is optional and when not present, will be set as ASCII F).</li></ul>
<i>lac1</i>	<ul style="list-style-type: none"><li>Location area code 1.</li></ul>
<i>lac2</i>	<ul style="list-style-type: none"><li>Location area code 1.</li></ul>
<i>PLMNRecID</i>	<ul style="list-style-type: none"><li>PLMN network name record identifier.</li></ul>

### 8.308.2 Field Documentation

8.308.2.1 WORD OperatorPLMNData::lac1

8.308.2.2 WORD OperatorPLMNData::lac2

8.308.2.3 BYTE OperatorPLMNData::mcc[3]

8.308.2.4 BYTE OperatorPLMNData::mnc[3]

8.308.2.5 BYTE OperatorPLMNData::PLMNRecID

## 8.309 operatorPLMNList Struct Reference

#### Data Fields

- [WORD numInstance](#)
- [OperatorPLMNData PLMNData \[255\]](#)

### 8.309.1 Detailed Description

This structure contains Operator PLMN List as defined in 3GPP TS 31.102 (Section 4.2.59) from multiple sources.

## Parameters

<i>numInstance</i>	<ul style="list-style-type: none"> <li>• Number of sets of the elements.</li> </ul>
<i>PLMNData</i>	<ul style="list-style-type: none"> <li>• Refer OperatorPLMNData for details (Optional).</li> </ul>

## 8.309.2 Field Documentation

8.309.2.1 WORD operatorPLMNList::numInstance

8.309.2.2 OperatorPLMNData operatorPLMNList::PLMNData[255]

## 8.310 PCMparams Struct Reference

## Data Fields

- [BYTE iFaceTabLen](#)
- [BYTE iFaceTab](#) [255]

## 8.310.1 Detailed Description

This structure contains the PCM parameters.

## Parameters

<i>iFaceTabLen</i>	<ul style="list-style-type: none"> <li>• Number of sets of iface table</li> </ul>
<i>iFaceTab</i>	<ul style="list-style-type: none"> <li>• Physical Interface Parameters</li> <li>• See <a href="#">qaGobiApiTableSwiAudio.h</a> for more information on physical interface parameters</li> </ul>

## 8.310.2 Field Documentation

8.310.2.1 BYTE PCMparams::iFaceTab[255]

8.310.2.2 BYTE PCMparams::iFaceTabLen

## 8.311 PCSCFFQDNAddress Struct Reference

## Data Fields

- [WORD fqdnLen](#)
- [CHAR fqdnAddr](#) [256]



### 8.311.1 Detailed Description

This structure contains the [PCSCFFQDNAddress](#) Information

## Parameters

<i>fqdnLen</i>	<ul style="list-style-type: none"> <li>length of the received FQDN address</li> </ul>
<i>fqdnAddr</i>	<ul style="list-style-type: none"> <li>FQDN address(Max 256 characters)</li> </ul>

## 8.311.2 Field Documentation

8.311.2.1 CHAR PCSCFFQDNAddress::fqdnAddr[256]

8.311.2.2 WORD PCSCFFQDNAddress::fqdnLen

## 8.312 PCSCFFQDNAddressList Struct Reference

## Data Fields

- [BYTE numInstances](#)
- struct [PCSCFFQDNAddress pcsfFQDNAddress](#) [10]

## 8.312.1 Detailed Description

This structure contains the [PCSCFFQDNAddressList](#) Information

## Parameters

<i>numInstances</i>	<ul style="list-style-type: none"> <li>Number of FQDN addresses received</li> </ul>
<i>pcsfFQDN-Address</i>	<ul style="list-style-type: none"> <li>FQDN address information(Max 10 addresses)</li> </ul>

## 8.312.2 Field Documentation

8.312.2.1 BYTE PCSCFFQDNAddressList::numInstances

8.312.2.2 struct PCSCFFQDNAddress PCSCFFQDNAddressList::pcsfFQDNAddress[10]

## 8.313 PCSCFIPv4ServerAddressList Struct Reference

## Data Fields

- [BYTE numInstances](#)
- [ULONG pcsfIPv4Addr](#) [64]

## 8.313.1 Detailed Description

This structure contains the [PCSCFIPv4ServerAddressList](#) Information

## Parameters

<i>numInstances</i>	<ul style="list-style-type: none"> <li>number of address following</li> </ul>
<i>pcscfIPv4Addr</i>	<ul style="list-style-type: none"> <li>P-CSCF IPv4 server addresses(Max 16 address, 4 bytes each)</li> </ul>

## 8.313.2 Field Documentation

8.313.2.1 BYTE PCSCFIPv4ServerAddressList::numInstances

8.313.2.2 ULONG PCSCFIPv4ServerAddressList::pcscfIPv4Addr[64]

## 8.314 PDSPositionData Struct Reference

## Data Fields

- [ULONGLONG \\* pTimeStamp](#)
- [ULONGLONG \\* pLatitude](#)
- [ULONGLONG \\* pLongitude](#)
- [ULONG \\* pAltitudeWrtEllipsoid](#)
- [ULONG \\* pAltitudeWrtSealevel](#)
- [ULONG \\* pHorizontalUncCircular](#)
- [ULONG \\* pVerticalUnc](#)
- [BYTE \\* pHorizontalConfidence](#)
- [BYTE \\* pVerticalConfidence](#)
- [BYTE \\* pPositionSource](#)
- [BYTE \\* pTimeType](#)

## 8.314.1 Detailed Description

Position Data Parameters from the external source to be injected to PDS engine.

## Parameters

<i>pTimeStamp</i>	<ul style="list-style-type: none"> <li>Timestamp of the injected position in msec. The time can be of type UTC, GPS, or Age and is defined in the pTimeType parameter. If the pTimeType is not present, the timestamp shall be assumed to be UTC time</li> </ul>
<i>pLatitude</i>	<ul style="list-style-type: none"> <li>Latitude position referenced to the WGS-84 reference ellipsoid, counting positive angles north of the equator and negative angles south of the equator. Value (in decimal degrees) in the range from -90 degrees to +90 degrees.Value in double float format (refer toIEEE Std 754-1985)</li> </ul>

<i>pLongitude</i>	<ul style="list-style-type: none"> <li>Longitude position referenced to the WGS-84 reference ellipsoid, counting positive angles east of the Greenwich Meridian and negative angles west of Greenwich meridian. Value (in decimal degrees) in the range from -180 degrees to +180 degrees.</li> </ul>
<i>pAltitudeWrt-Ellipsoid</i>	<ul style="list-style-type: none"> <li>Height above the WGS-84 reference ellipsoid. Value conveys height (in meters). When injecting altitude information, the control point should include either this parameter or the <i>pAltitudeWrtSealevel</i> parameter. Value in single float format (refer to IEEE Std 754-1985)</li> </ul>
<i>pAltitudeWrt-Sealevel</i>	<ul style="list-style-type: none"> <li>Height of MS above the mean sea level in units (in meters). When injecting altitude information, the control point should include either this parameter or the <i>pAltitudeWrt-Ellipsoid</i> parameter. Value in single float format (refer to IEEE Std 754-1985)</li> </ul>
<i>pHorizontalUnc-Circular</i>	<ul style="list-style-type: none"> <li>Circular horizontal uncertainty (in meters). This parameter must be included if the latitude and longitude parameters are specified. Value in single float format (refer to IEEE Std 754-1985)</li> </ul>
<i>pVerticalUnc</i>	<ul style="list-style-type: none"> <li>Vertical uncertainty (in meters). This parameter must be included if one of the altitude parameter are specified. Value in single float format (refer to IEEE Std 754-1985)</li> </ul>
<i>pHorizontal-Confidence</i>	<ul style="list-style-type: none"> <li>Confidence value of the location horizontal uncertainty, specified as percentage, 1 to 100. This parameter must be included if the latitude and longitude parameters are specified.</li> </ul>
<i>pVertical-Confidence</i>	<ul style="list-style-type: none"> <li>Confidence value of the location vertical uncertainty, specified as percentage, 1 to 100. This parameter must be included if one of the altitude paramters are specified.</li> </ul>
<i>pPositionSource</i>	<ul style="list-style-type: none"> <li>Source of injected position: <ul style="list-style-type: none"> <li>0x00 - Unknown</li> <li>0x01 - GPS</li> <li>0x02 - Cell ID</li> <li>0x03 - Enhanced cell ID</li> <li>0x04 - WiFi</li> <li>0x05 - Terrestrial</li> <li>0x06 - Terrestrial hybrid</li> <li>0x07 - Other</li> </ul> </li> </ul>

<i>pTimeType</i>	<ul style="list-style-type: none"> <li>Defines the time value set in the pTimeStamp parameter. <ul style="list-style-type: none"> <li>0x00 - UTC Time: starting Jan 1, 1970</li> <li>0x01 - GPS Time: starting Jan 6, 1980</li> <li>0x02 - Age: Age of position information</li> </ul> </li> </ul>
------------------	--

## 8.314.2 Field Documentation

8.314.2.1 **ULONG\*** PDSPositionData::pAltitudeWrtEllipsoid

8.314.2.2 **ULONG\*** PDSPositionData::pAltitudeWrtSealevel

8.314.2.3 **BYTE\*** PDSPositionData::pHorizontalConfidence

8.314.2.4 **ULONG\*** PDSPositionData::pHorizontalUncCircular

8.314.2.5 **ULONGLONG\*** PDSPositionData::pLatitude

8.314.2.6 **ULONGLONG\*** PDSPositionData::pLongitude

8.314.2.7 **BYTE\*** PDSPositionData::pPositionSource

8.314.2.8 **ULONGLONG\*** PDSPositionData::pTimeStamp

8.314.2.9 **BYTE\*** PDSPositionData::pTimeType

8.314.2.10 **BYTE\*** PDSPositionData::pVerticalConfidence

8.314.2.11 **ULONG\*** PDSPositionData::pVerticalUnc

## 8.315 PDSPosMethodStateReq Struct Reference

### Data Fields

- BYTE \*** [pXtraTimeState](#)
- BYTE \*** [pXtraDataState](#)
- BYTE \*** [pWifiState](#)

### 8.315.1 Detailed Description

Parameters to Set state of positioning method for a device.

## Parameters

<i>pXtraTimeState</i>	<ul style="list-style-type: none"> <li>• XTRA Time Position Method State.</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0x00 - Disable</li> <li>– 0x01 - Enable</li> </ul> </li> </ul>
<i>pXtraDataState</i>	<ul style="list-style-type: none"> <li>• XTRA Data Position Method State.</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0x00 - Disable</li> <li>– 0x01 - Enable</li> </ul> </li> </ul>
<i>Latitude</i>	<ul style="list-style-type: none"> <li>• WiFi Position Method State</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0x00 - Disable</li> <li>– 0x01 - Enable</li> </ul> </li> </ul>

## 8.315.2 Field Documentation

8.315.2.1 **BYTE\*** PDSPosMethodStateReq::pWifiState8.315.2.2 **BYTE\*** PDSPosMethodStateReq::pXtraDataState8.315.2.3 **BYTE\*** PDSPosMethodStateReq::pXtraTimeState

## 8.316 peerNumberInfo Struct Reference

## Data Fields

- [BYTE](#) callID
- [BYTE](#) numPI
- [BYTE](#) numSI
- [BYTE](#) numType
- [BYTE](#) numPlan
- [BYTE](#) numLen
- [BYTE](#) number [81]

## 8.316.1 Detailed Description

This structure contains information for Connected Peer Numbers.

## Parameters

<i>callID</i>	<ul style="list-style-type: none"> <li>• Unique call identifier for the call.</li> </ul>
<i>numPI</i>	<ul style="list-style-type: none"> <li>• Number presentation indicator. <ul style="list-style-type: none"> <li>– 0x00 - PRESENTATION_ALLOWED - Allowed presentation</li> <li>– 0x01 - PRESENTATION_RESTRICTED - Restricted presentation</li> <li>– 0x02 - PRESENTATION_NUM_UNAVAILABLE - Unavailable presentation</li> <li>– 0x04 - PRESENTATION_PAYPHONE - Payphone presentation (GSM/UMTS specific)</li> <li>– 0xFF - Not Available</li> </ul> </li> </ul>
<i>numSI</i>	<ul style="list-style-type: none"> <li>• Number screening indicator. <ul style="list-style-type: none"> <li>– 0x00 - QMI_VOICE_SI_USER_PROVIDED_NOT_SCREENED - Provided user is not screened</li> <li>– 0x01 - QMI_VOICE_SI_USER_PROVIDED_VERIFIED_PASSED - Provided user passed verification</li> <li>– 0x02 - QMI_VOICE_SI_USER_PROVIDED_VERIFIED_FAILED - Provided user failed verification</li> <li>– 0x03 - QMI_VOICE_SI_NETWORK_PROVIDED - Provided network</li> </ul> </li> </ul>
<i>numType</i>	<ul style="list-style-type: none"> <li>• Number type. <ul style="list-style-type: none"> <li>– 0x00 - QMI_VOICE_NUM_TYPE_UNKNOWN - Unknown</li> <li>– 0x01 - QMI_VOICE_NUM_TYPE_INTERNATIONAL - International</li> <li>– 0x02 - QMI_VOICE_NUM_TYPE_NATIONAL - National</li> <li>– 0x03 - QMI_VOICE_NUM_TYPE_NETWORK_SPECIFIC - Network-specific</li> <li>– 0x04 - QMI_VOICE_NUM_TYPE_SUBSCRIBER - Subscriber</li> <li>– 0x05 - QMI_VOICE_NUM_TYPE_RESERVED - Reserved</li> <li>– 0x06 - QMI_VOICE_NUM_TYPE_ABBREVIATED - Abbreviated</li> <li>– 0x07 - QMI_VOICE_NUM_TYPE_RESERVED_EXTENSION - Reserved extension</li> </ul> </li> </ul>

<i>numPlan</i>	<ul style="list-style-type: none"> <li>• Number plan. <ul style="list-style-type: none"> <li>– 0x00 - QMI_VOICE_NUM_PLAN_UNKNOWN - Unknown</li> <li>– 0x01 - QMI_VOICE_NUM_PLAN_ISDN - ISDN</li> <li>– 0x03 - QMI_VOICE_NUM_PLAN_DATA - Data</li> <li>– 0x04 - QMI_VOICE_NUM_PLAN_TELEX - Telex</li> <li>– 0x08 - QMI_VOICE_NUM_PLAN_NATIONAL - National</li> <li>– 0x09 - QMI_VOICE_NUM_PLAN_PRIVATE - Private</li> <li>– 0x0B - QMI_VOICE_NUM_PLAN_RESERVED_CTS - Reserved cordless telephony system</li> <li>– 0x0F - QMI_VOICE_NUM_PLAN_RESERVED_EXTENSION - Reserved extension</li> </ul> </li> </ul>
<i>numLen</i>	<ul style="list-style-type: none"> <li>• Provides the length of number which follow.</li> </ul>
<i>number[<b>MAX_CALL_NO_LEN</b>]</i>	<ul style="list-style-type: none"> <li>• number of numLen length, NULL terminated.</li> </ul>

## 8.316.2 Field Documentation

8.316.2.1 **BYTE** peerNumberInfo::callID

8.316.2.2 **BYTE** peerNumberInfo::number[81]

8.316.2.3 **BYTE** peerNumberInfo::numLen

8.316.2.4 **BYTE** peerNumberInfo::numPl

8.316.2.5 **BYTE** peerNumberInfo::numPlan

8.316.2.6 **BYTE** peerNumberInfo::numSI

8.316.2.7 **BYTE** peerNumberInfo::numType

## 8.317 PhyCaAggPcellInfo Struct Reference

### Data Fields

- int [pci](#)
- int [freq](#)
- [NAS\\_LTE\\_CPHY\\_CA\\_BW\\_NRB dl\\_bw\\_value](#)
- int [iLTEbandValue](#)
- [BYTE](#) [TlvPresent](#)

### 8.317.1 Detailed Description

This structure contains the parameters for Physical Carrier aggregation of Pcell Information.



## Parameters

<i>pci</i>	<ul style="list-style-type: none"> <li>Physical cell ID of the SCell Range.</li> <li>Range for ID values: 0 to 503.</li> </ul>
<i>freq</i>	<ul style="list-style-type: none"> <li>Frequency of the absolute cell Range.</li> <li>Range for ID values: 0 to 65535.</li> </ul>
<i>dl_bw_value</i>	<ul style="list-style-type: none"> <li>Downlink Bandwidth Values.</li> <li>See <a href="#">NAS_LTE_CPHY_CA_BW_NRB</a> for more information.</li> </ul>
<i>scell_state</i>	<ul style="list-style-type: none"> <li>Scell state Values.</li> <li>See <a href="#">NAS_LTE_CPHY_SCELL_STATE</a> for more information.</li> </ul>
<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>Tlv Present.</li> </ul>

## 8.317.2 Field Documentation

8.317.2.1 NAS\_LTE\_CPHY\_CA\_BW\_NRB PhyCaAggPcellInfo::dl\_bw\_value

8.317.2.2 int PhyCaAggPcellInfo::freq

8.317.2.3 int PhyCaAggPcellInfo::lTEbandValue

8.317.2.4 int PhyCaAggPcellInfo::pci

8.317.2.5 BYTE PhyCaAggPcellInfo::TlvPresent

## 8.318 PhyCaAggScellIDIBw Struct Reference

## Data Fields

- [NAS\\_LTE\\_CPHY\\_CA\\_BW\\_NRB dl\\_bw\\_value](#)
- [BYTE TlvPresent](#)

## 8.318.1 Detailed Description

This structure contains the parameters for Physical Carrier aggregation Downlink Bandwidth of Scell.

## Parameters

<i>dl_bw_value</i>	<ul style="list-style-type: none"> <li>• Downlink Bandwidth Values.</li> <li>• See <a href="#">NAS_LTE_CPHY_CA_BW_NRB</a> for more information.</li> </ul>
--------------------	--

## 8.318.2 Field Documentation

8.318.2.1 **NAS\_LTE\_CPHY\_CA\_BW\_NRB** `PhyCaAggScellIDIBw::dl_bw_value`

8.318.2.2 **BYTE** `PhyCaAggScellIDIBw::TlvPresent`

## 8.319 PhyCaAggScellIndex Struct Reference

### Data Fields

- [BYTE](#) `scell_idx`
- [BYTE](#) `TlvPresent`

### 8.319.1 Detailed Description

This structure contains the parameters for Physical Carrier aggregation of Scell Index.

#### Parameters

<i>scell_idx</i>	<ul style="list-style-type: none"> <li>• Physical cell ID of the SCell Range.</li> <li>• Range for ID values: 0 to 503.</li> </ul>
<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>• Tlv Present.</li> </ul>

## 8.319.2 Field Documentation

8.319.2.1 **BYTE** `PhyCaAggScellIndex::scell_idx`

8.319.2.2 **BYTE** `PhyCaAggScellIndex::TlvPresent`

## 8.320 PhyCaAggScellIndType Struct Reference

### Data Fields

- int `pci`
- int `freq`
- [NAS\\_LTE\\_CPHY\\_SCELL\\_STATE](#) `scell_state`
- [BYTE](#) `TlvPresent`

### 8.320.1 Detailed Description

This structure contains the parameters for Physical Carrier aggregation of Scell Indicator Type.

## Parameters

<i>pci</i>	<ul style="list-style-type: none"> <li>Physical cell ID of the SCell Range.</li> <li>Range for ID values: 0 to 503.</li> </ul>
<i>freq</i>	<ul style="list-style-type: none"> <li>Frequency of the absolute cell Range.</li> <li>Range for ID values: 0 to 65535.</li> </ul>
<i>scell_state</i>	<ul style="list-style-type: none"> <li>Scell state Values.</li> <li>See <a href="#">NAS_LTE_CPHY_SCELL_STATE</a> for more information.</li> </ul>
<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>Tlv Present.</li> </ul>

## 8.320.2 Field Documentation

8.320.2.1 int PhyCaAggScellIndType::freq

8.320.2.2 int PhyCaAggScellIndType::pci

8.320.2.3 NAS\_LTE\_CPHY\_SCELL\_STATE PhyCaAggScellIndType::scell\_state

8.320.2.4 BYTE PhyCaAggScellIndType::TlvPresent

## 8.321 PhyCaAggScellInfo Struct Reference

## Data Fields

- int [pci](#)
- int [freq](#)
- [NAS\\_LTE\\_CPHY\\_CA\\_BW\\_NRB](#) dl\_bw\_value
- int [iLTEbandValue](#)
- [NAS\\_LTE\\_CPHY\\_SCELL\\_STATE](#) scell\_state
- BYTE [TlvPresent](#)

## 8.321.1 Detailed Description

This structure contains the parameters for Physical Carrier aggregation of Scell Information.

## Parameters

<i>pci</i>	<ul style="list-style-type: none"> <li>Physical cell ID of the SCell Range.</li> <li>Range for ID values: 0 to 503.</li> </ul>
------------	--

<i>freq</i>	<ul style="list-style-type: none"> <li>• Frequency of the absolute cell Range.</li> <li>• Range for ID values: 0 to 65535.</li> </ul>
<i>dl_bw_value</i>	<ul style="list-style-type: none"> <li>• Downlink Bandwidth Values.</li> <li>• See <a href="#">NAS_LTE_CPHY_CA_BW_NRB</a> for more information.</li> </ul>
<i>iLTEbandValue</i>	<ul style="list-style-type: none"> <li>• Band value.</li> <li>• Range for LTE Band class 120 to 160.</li> </ul>
<i>scell_state</i>	<ul style="list-style-type: none"> <li>• Scell state Values.</li> <li>• See <a href="#">NAS_LTE_CPHY_SCELL_STATE</a> for more information.</li> </ul>
<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>• Tlv Present.</li> </ul>

## 8.321.2 Field Documentation

8.321.2.1 **NAS\_LTE\_CPHY\_CA\_BW\_NRB** `PhyCaAggScellInfo::dl_bw_value`

8.321.2.2 `int` `PhyCaAggScellInfo::freq`

8.321.2.3 `int` `PhyCaAggScellInfo::iLTEbandValue`

8.321.2.4 `int` `PhyCaAggScellInfo::pci`

8.321.2.5 **NAS\_LTE\_CPHY\_SCELL\_STATE** `PhyCaAggScellInfo::scell_state`

8.321.2.6 **BYTE** `PhyCaAggScellInfo::TlvPresent`

## 8.322 PilotSetData Struct Reference

### Data Fields

- [BYTE](#) `NumPilots`
- [PilotSetParams](#) \* `pPilotSetInfo`

### 8.322.1 Detailed Description

This structure contains Pilot Set Data

#### Parameters

---

<i>NumPilots(IN/O-UT)</i>	<ul style="list-style-type: none"> <li>• Number of Pilot Sets</li> <li>• As input specifies number of sets of parameter <code>pPilotSetInfo</code> for which memory has been assigned</li> <li>• As output specifies the actual number of sets of parameter <code>pPilotSetInfo</code> returned by device</li> </ul>
<i>pPilotSetInfo</i>	<ul style="list-style-type: none"> <li>• Pilot Set Parameters</li> <li>• See <a href="#">PilotSetParams</a> for more information.</li> </ul>

note A buffer under sized error is returned if the number of sets of `pPilotSetInfo` returned by the device is greater than the value in `NumPilots` input parameter.

### 8.322.2 Field Documentation

#### 8.322.2.1 BYTE PilotSetData::NumPilots

#### 8.322.2.2 PilotSetParams\* PilotSetData::pPilotSetInfo

## 8.323 PilotSetParams Struct Reference

### Data Fields

- [ULONG PilotType](#)
- [WORD PilotPN](#)
- [WORD PilotStrength](#)

### 8.323.1 Detailed Description

This structure contains Pilot Set parameters

#### Parameters

<i>PilotType</i>	<ul style="list-style-type: none"> <li>• 0x00 - NAS_HRPD_PILOT_CURR_ACT_PLT Current Active Pilot</li> <li>• 0x01 - NAS_HRPD_PILOT_NEIGHBOR_PLT Neighbor pilot information</li> </ul>
<i>PilotPN</i>	<ul style="list-style-type: none"> <li>• Pilot PN sequence offset index</li> </ul>
<i>PilotStrength</i>	<ul style="list-style-type: none"> <li>• Strength of the pilot (in dB)</li> </ul>

### 8.323.2 Field Documentation

#### 8.323.2.1 WORD PilotSetParams::PilotPN

8.323.2.2 WORD PilotSetParams::PilotStrength

8.323.2.3 ULONG PilotSetParams::PilotType

## 8.324 pktErrRate Struct Reference

### Data Fields

- [WORD multiplier](#)
- [WORD exponent](#)

### 8.324.1 Detailed Description

This structure contains the IP flow packet error rate

#### Parameters

<i>multiplier</i>	Factor m in calculating packet error rate: $E = m \cdot 10^{**}(-p)$
<i>exponent</i>	Factor p in calculating packet error rate (see above)

### 8.324.2 Field Documentation

8.324.2.1 WORD pktErrRate::exponent

8.324.2.2 WORD pktErrRate::multiplier

## 8.325 PLMNNetworkName Struct Reference

### Data Fields

- [BYTE numInstance](#)
- [PLMNNetworkNameData PLMNNetName](#) [255]

### 8.325.1 Detailed Description

This structure contains PLMN Network Name as defined in 3GPP TS 24.008 (Section 10.5.3.5a) from multiple sources.

#### Parameters

<i>numInstance</i>	<ul style="list-style-type: none"> <li>• Number of sets of the elements.</li> </ul>
<i>PLMNNetName</i>	<ul style="list-style-type: none"> <li>• Refer <a href="#">PLMNNetworkNameData</a> for details (Optional).</li> </ul>

### 8.325.2 Field Documentation

8.325.2.1 BYTE PLMNNetworkName::numInstance

8.325.2.2 PLMNNetworkNameData PLMNNetworkName::PLMNNetName[255]

## 8.326 PLMNNetworkNameData Struct Reference

### Data Fields

- [BYTE codingScheme](#)
- [BYTE countryInitials](#)
- [BYTE longNameSpareBits](#)
- [BYTE shortNameSpareBits](#)
- [BYTE longNameLen](#)
- [BYTE longName](#) [255]
- [BYTE shortNameLen](#)
- [BYTE shortName](#) [255]

### 8.326.1 Detailed Description

This structure contains PLMN Network Name Data from multiple sources.

## Parameters

<i>codingScheme</i>	<ul style="list-style-type: none"> <li>• Coding scheme: <ul style="list-style-type: none"> <li>– 0 - CODING_SCHEME_CELL_BROADCAST_GSM - Cell broadcast data coding scheme, GSM default alphabet, language unspecified;defined in 3GPP TS 23.-038.</li> <li>– 1 - CODING_SCHEME_UCS2 - UCS2 (16 bit);defined in ISO/IEC 10646</li> </ul> </li> </ul>
<i>countryInitials</i>	<ul style="list-style-type: none"> <li>• Country's initials: <ul style="list-style-type: none"> <li>– 0 - COUNTRY_INITIALS_DO_NOT_ADD - MS should not add the letters for the country's initials to the text string.</li> <li>– 1 - COUNTRY_INITIALS_ADD - MS should add the letters for the country's initials and a separator, e.g., a space, to the text string.</li> </ul> </li> </ul>
<i>longNameSpare-Bits</i>	<ul style="list-style-type: none"> <li>• Long Name Spare Bits: <ul style="list-style-type: none"> <li>– 1 - SPARE_BITS_8 - Bit 8 is spare and set to 0 in octet n</li> <li>– 2 - SPARE_BITS_7_TO_8 - Bits 7 and 8 are spare and set to 0 in octet n.</li> <li>– 3 - SPARE_BITS_6_TO_8 - Bits 6 to 8 (inclusive) are spare and set to 0 in octet n.</li> <li>– 4 - SPARE_BITS_5_TO_8 - Bits 5 to 8 (inclusive) are spare and set to 0 in octet n.</li> <li>– 5 - SPARE_BITS_4_TO_8 - Bits 4 to 8 (inclusive) are spare and set to 0 in octet n.</li> <li>– 6 - SPARE_BITS_3_TO_8 - Bits 3 to 8 (inclusive) are spare and set to 0 in octet n.</li> <li>– 7 - SPARE_BITS_2_TO_8 - Bits 2 to 8 (inclusive) are spare and set to 0 in octet n.</li> <li>– 0 - SPARE_BITS_UNKNOWN - Carries no information about the number of spare bits in octet n.</li> </ul> </li> </ul>
<i>shortName-SpareBits</i>	<ul style="list-style-type: none"> <li>• Short Name Spare Bits: <ul style="list-style-type: none"> <li>– 1 - SPARE_BITS_8 - Bit 8 is spare and set to 0 in octet n.</li> <li>– 2 - SPARE_BITS_7_TO_8 - Bits 7 and 8 are spare and set to 0 in octet n.</li> <li>– 3 - SPARE_BITS_6_TO_8 - Bits 6 to 8 (inclusive) are spare and set to 0 in octet n.</li> <li>– 4 - SPARE_BITS_5_TO_8 - Bits 5 to 8 (inclusive) are spare and set to 0 in octet n.</li> <li>– 5 - SPARE_BITS_4_TO_8 - Bits 4 to 8 (inclusive) are spare and set to 0 in octet n.</li> <li>– 6 - SPARE_BITS_3_TO_8 - Bits 3 to 8 (inclusive) are spare and set to 0 in octet n.</li> <li>– 7 - SPARE_BITS_2_TO_8 - Bits 2 to 8 (inclusive) are spare and set to 0 in octet n.</li> <li>– 0 - SPARE_BITS_UNKNOWN - Carries no information about the number of spare bits in octet n.</li> </ul> </li> </ul>
	Generated on Wed Oct 28 2015 00:40:12 for LinuxQMISDK by Doxygen



<i>longNameLen</i>	<ul style="list-style-type: none"><li>• It provides the length of long name.</li></ul>
<i>longName</i>	<ul style="list-style-type: none"><li>• Long name string in coding_scheme.</li></ul>
<i>shortNameLen</i>	<ul style="list-style-type: none"><li>• It provides the length of short name.</li></ul>
<i>shortName</i>	<ul style="list-style-type: none"><li>• Short name string in coding_scheme.</li></ul>

## 8.326.2 Field Documentation

8.326.2.1 BYTE PLMNNetworkNameData::codingScheme

8.326.2.2 BYTE PLMNNetworkNameData::countryInitials

8.326.2.3 BYTE PLMNNetworkNameData::longName[255]

8.326.2.4 BYTE PLMNNetworkNameData::longNameLen

8.326.2.5 BYTE PLMNNetworkNameData::longNameSpareBits

8.326.2.6 BYTE PLMNNetworkNameData::shortName[255]

8.326.2.7 BYTE PLMNNetworkNameData::shortNameLen

8.326.2.8 BYTE PLMNNetworkNameData::shortNameSpareBits

## 8.327 Port Struct Reference

### Data Fields

- [WORD port](#)
- [WORD range](#)

### 8.327.1 Detailed Description

This structure contains the [Port](#) Filter

#### Parameters

<i>port</i>	port value of the filter
<i>range</i>	range specifies the number of ports to be included in the filter starting from port; filter will match if port in the IP packet lies between port and (port + range ) Range value of 0 implies that only one value of the port is valid, as specified by the port

## 8.327.2 Field Documentation

8.327.2.1 WORD Port::port

8.327.2.2 WORD Port::range

## 8.328 precisionDilution\_s Struct Reference

### Data Fields

- [ULONG PDOP](#)
- [ULONG HDOP](#)
- [ULONG VDOP](#)

### 8.328.1 Detailed Description

This structure contains Dilution of precision associated with this position.

#### Parameters

<i>PDOP</i>	<ul style="list-style-type: none"><li>• Position dilution of precision.</li><li>• Range - 1 (highest accuracy) to 50 (lowest accuracy)</li><li>• PDOP = square root of (Square of HDOP + Square of VDOP<sup>2</sup>)</li></ul>
<i>HDOP</i>	<ul style="list-style-type: none"><li>• Horizontal dilution of precision.</li><li>• Range - 1 (highest accuracy) to 50 (lowest accuracy)</li></ul>
<i>VDOP</i>	<ul style="list-style-type: none"><li>• Vertical dilution of precision.</li><li>• Range- 1 (highest accuracy) to 50 (lowest accuracy)</li></ul>

### 8.328.2 Field Documentation

8.328.2.1 ULONG precisionDilution\_s::HDOP

8.328.2.2 ULONG precisionDilution\_s::PDOP

8.328.2.3 ULONG precisionDilution\_s::VDOP

## 8.329 PrefImageList Struct Reference

### Data Fields

- [BYTE listSize](#)
- struct [ImageElement listEntries](#) [2]

### 8.329.1 Detailed Description

This structure contains the Preference Image List information

## Parameters

<i>listSize</i>	<ul style="list-style-type: none"><li>• The number of elements in the image list</li></ul>
<i>pListEntries</i>	<ul style="list-style-type: none"><li>• Array of Image entries with size provided by previous field</li><li>• See <a href="#">ImageElement</a></li></ul>

## 8.329.2 Field Documentation

8.329.2.1 struct ImageElement PrefImageList::listEntries[2]

8.329.2.2 BYTE PrefImageList::listSize

## 8.330 prefVoiceSO Struct Reference

## Data Fields

- [BYTE](#) *namID*
- [BYTE](#) *evrcCapability*
- [WORD](#) *homePageVoiceSO*
- [WORD](#) *homeOrigVoiceSO*
- [WORD](#) *roamOrigVoiceSO*

## 8.330.1 Detailed Description

This structure contains information about the Preferred Voice Service Options.

## Parameters

<i>namID</i>	<ul style="list-style-type: none"><li>• Index of the NAM(Number Assignment Module) to be configured.</li><li>• Range 0 to 3.</li><li>• Some modems support only 1 or 2 NAMs.</li><li>• 0xFF,if not available.</li></ul>
<i>evrcCapability</i>	<ul style="list-style-type: none"><li>• EVRC capability.</li><li>• Values:<ul style="list-style-type: none"><li>– 0x00 - Disable</li><li>– 0x01 - Enable</li><li>– 0xFF - Not Available</li></ul></li></ul>

<i>homePageVoice-</i> SO	<ul style="list-style-type: none"> <li>• Home page voice SO; most preferred CDMA SO to be requested from the network when receiving an incoming (MT) voice call within the home network.</li> <li>• Values:             <ul style="list-style-type: none"> <li>– 0x0000 - VOICE_SO_WILD - Any service option</li> <li>– 0x0001 - VOICE_SO_IS_96A - IS-96A</li> <li>– 0x0003 - VOICE_SO_EVRC - EVRC</li> <li>– 0x0011 - VOICE_SO_13K_IS733 - 13K_IS733</li> <li>– 0x0038 - VOICE_SO_SELECTABLE_MODE_VOCODER - Selectable mode vocoder</li> <li>– 0x0044 - VOICE_SO_4GV_NARROW_BAND - 4GV narrowband</li> <li>– 0x0046 - VOICE_SO_4GV_WIDE_BAND - 4GV wideband</li> <li>– 0x8000 - VOICE_SO_13K - 13K</li> <li>– 0x8001 - VOICE_SO_IS_96 - IS-96</li> <li>– 0x8023 - VOICE_SO_WVRC - WVRC</li> <li>– 0xFFFF - Not Available</li> </ul> </li> </ul>
<i>homeOrigVoice-</i> SO	<ul style="list-style-type: none"> <li>• Home origination voice SO; most preferred CDMA SO to be requested from the network when receiving an incoming (MT) voice call within the home network.</li> <li>• Values:             <ul style="list-style-type: none"> <li>– 0x0000 - VOICE_SO_WILD - Any service option</li> <li>– 0x0001 - VOICE_SO_IS_96A - IS-96A</li> <li>– 0x0003 - VOICE_SO_EVRC - EVRC</li> <li>– 0x0011 - VOICE_SO_13K_IS733 - 13K_IS733</li> <li>– 0x0038 - VOICE_SO_SELECTABLE_MODE_VOCODER - Selectable mode vocoder</li> <li>– 0x0044 - VOICE_SO_4GV_NARROW_BAND - 4GV narrowband</li> <li>– 0x0046 - VOICE_SO_4GV_WIDE_BAND - 4GV wideband</li> <li>– 0x8000 - VOICE_SO_13K - 13K</li> <li>– 0x8001 - VOICE_SO_IS_96 - IS-96</li> <li>– 0x8023 - VOICE_SO_WVRC - WVRC</li> <li>– 0xFFFF - Not Available</li> </ul> </li> </ul>

<i>roamOrigVoiceSO</i>	<ul style="list-style-type: none"> <li>• Roaming origination voice SO; most preferred CDMA SO to be requested from the network when receiving an incoming (MT) voice call within the home network.</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0x0000 - VOICE_SO_WILD - Any service option</li> <li>– 0x0001 - VOICE_SO_IS_96A - IS-96A</li> <li>– 0x0003 - VOICE_SO_EVRC - EVRC</li> <li>– 0x0011 - VOICE_SO_13K_IS733 - 13K_IS733</li> <li>– 0x0038 - VOICE_SO_SELECTABLE_MODE_VOCODER - Selectable mode vocoder</li> <li>– 0x0044 - VOICE_SO_4GV_NARROW_BAND - 4GV narrowband</li> <li>– 0x0046 - VOICE_SO_4GV_WIDE_BAND - 4GV wideband</li> <li>– 0x8000 - VOICE_SO_13K - 13K</li> <li>– 0x8001 - VOICE_SO_IS_96 - IS-96</li> <li>– 0x8023 - VOICE_SO_WVRC - WVRC</li> <li>– 0xFFFF - Not Available</li> </ul> </li> </ul>
------------------------	--

## 8.330.2 Field Documentation

8.330.2.1 **BYTE** `prefVoiceSO::evrcCapability`

8.330.2.2 **WORD** `prefVoiceSO::homeOrigVoiceSO`

8.330.2.3 **WORD** `prefVoiceSO::homePageVoiceSO`

8.330.2.4 **BYTE** `prefVoiceSO::namID`

8.330.2.5 **WORD** `prefVoiceSO::roamOrigVoiceSO`

## 8.331 Profile3GPP Struct Reference

### Data Fields

- **CHAR** \* `pProfilename`
- **WORD** \* `pProfilenameSize`
- **BYTE** \* `pPDType`
- **BYTE** \* `pPdpHdrCompType`
- **BYTE** \* `pPdpDataCompType`
- **CHAR** \* `pAPNName`
- **WORD** \* `pAPNNameSize`
- **ULONG** \* `pPriDNSIPv4AddPref`
- **ULONG** \* `pSecDNSIPv4AddPref`
- **struct** `UMTSQoS` \* `pUMTSReqQoS`
- **struct** `UMTSQoS` \* `pUMTSMinQoS`
- **struct** `GPRSRequestedQoS` \* `pGPRSRequestedQoS`
- **struct** `GPRSRequestedQoS` \* `pGPRSMinimumQoS`
- **CHAR** \* `pUsername`
- **WORD** \* `pUsernameSize`

- CHAR \* pPassword
- WORD \* pPasswordSize
- BYTE \* pAuthenticationPref
- ULONG \* pIPv4AddrPref
- BYTE \* pPcscfAddrUsingPCO
- BYTE \* pPdpAccessConFlag
- BYTE \* pPcscfAddrUsingDhcp
- BYTE \* plmCnFlag
- struct TFTIDParams \* pTFTID1Params
- struct TFTIDParams \* pTFTID2Params
- BYTE \* pPdpContext
- BYTE \* pSecondaryFlag
- BYTE \* pPrimaryID
- USHORT \* pIPv6AddPref
- struct UMTSReqQoSsigInd \* pUMTSReqQoSsigInd
- struct UMTSReqQoSsigInd \* pUMTSMinQoSsigInd
- USHORT \* pPriDNSIPv6addpref
- USHORT \* pSecDNSIPv6addpref
- BYTE \* pAddrAllocPref
- struct QoSClassID \* pQoSClassID
- BYTE \* pAPNDisabledFlag
- ULONG \* pPDNInactivTimeout
- BYTE \* pAPNClass

### 8.331.1 Detailed Description

This structure contains Input parameters of SLQSCreateProfile and SLQSModifyProfile and output parameters of SLQSGetProfileSettings

- Parameter values default to their data type's maximum unsigned value unless explicitly stated otherwise.

#### Parameters

<i>pProfileName</i>	<ul style="list-style-type: none"> <li>• One or more bytes describing the profile</li> </ul>
<i>pProfilename-Size;</i>	<ul style="list-style-type: none"> <li>• This parameter is an input parameter and should be initialised to the size of pProfile-Name field. Size of this parameter is 2 bytes.</li> </ul>
<i>pPDPTType</i>	<ul style="list-style-type: none"> <li>• Packet Data Protocol (PDP) type specifies the type of data payload exchanged over the air link when the packet data session is established with this profile <ul style="list-style-type: none"> <li>– 0x00 - PDP-IP (IPv4)</li> <li>– 0x01 - PDP-PPP</li> <li>– 0x02 - PDP-IPV6</li> <li>– 0x03 - PDP-IPV4V6</li> </ul> </li> </ul>

<i>pPdpHdrComp-Type</i>	<ul style="list-style-type: none"> <li>• PDP header compression type <ul style="list-style-type: none"> <li>– 0 - PDP header compression is OFF</li> <li>– 1 - Manufacturer preferred compression</li> <li>– 2 - PDP header compression based on RFC 1144</li> <li>– 3 - PDP header compression based on RFC 25074 PDP header compression based on RFC 3095</li> </ul> </li> </ul>
<i>pPdpDataComp-Type</i>	<ul style="list-style-type: none"> <li>• PDP data compression type <ul style="list-style-type: none"> <li>– 0 - PDP data compression is OFF</li> <li>– 1 - Manufacturer preferred compression</li> <li>– 2 - V.42BIS data compression</li> <li>– 3 - V.44 data compression</li> </ul> </li> </ul>
<i>pAPNName</i>	<ul style="list-style-type: none"> <li>• Access point name</li> </ul>
<i>pAPNnameSize;</i>	<ul style="list-style-type: none"> <li>• This parameter is an input parameter and should be initialised to the size of pAPN-Name field. Size of this parameter is 2 bytes.</li> </ul>
<i>pPriDNSIPv4-AddPref</i>	<ul style="list-style-type: none"> <li>• Primary DNS IPv4 Address Preference</li> </ul>
<i>pSecDNSIPv4-AddPref</i>	<ul style="list-style-type: none"> <li>• Secondary DNS IPv4 Address Preference</li> </ul>
<i>pUMTSReqQoS</i>	<ul style="list-style-type: none"> <li>• UMTS Requested QoS</li> </ul>
<i>pUMTSMinQoS</i>	<ul style="list-style-type: none"> <li>• UMTS Minimum QoS</li> </ul>
<i>pGPRS-RequestedQoS</i>	<ul style="list-style-type: none"> <li>• GPRS Minimum QoS</li> </ul>
<i>pUsername</i>	<ul style="list-style-type: none"> <li>• User name</li> </ul>
<i>pUsernameSize;</i>	<ul style="list-style-type: none"> <li>• This parameter is an input parameter and should be initialised to the size of pUsername field. Size of this parameter is 2 bytes.</li> </ul>
<i>pPassword</i>	<ul style="list-style-type: none"> <li>• Password</li> </ul>
<i>pPasswordSize;</i>	<ul style="list-style-type: none"> <li>• This parameter is an input parameter and should be initialised to the size of pPassword field. Size of this parameter is 2 bytes.</li> </ul>

<i>pAuthentication-Pref</i>	<ul style="list-style-type: none"> <li>• Authentication Preference           <ul style="list-style-type: none"> <li>– Bit map that indicates the authentication algorithm preference               <ul style="list-style-type: none"> <li>* Bit 0 - PAP preference                   <ul style="list-style-type: none"> <li>· 0 - PAP is never performed</li> <li>· 1 - PAP may be performed</li> </ul> </li> <li>* Bit 1 - CHAP preference                   <ul style="list-style-type: none"> <li>· 0 - CHAP is never performed</li> <li>· 1 - CHAP may be performed</li> </ul> </li> <li>* If more than one bit is set, then the device decides which authentication procedure is performed while setting up the data session. For example, the device may have a policy to select the most secure authentication mechanism.</li> </ul> </li> </ul> </li> </ul>
<i>pIPv4AddrPref</i>	<ul style="list-style-type: none"> <li>• IPv4 Address Preference</li> </ul>
<i>pPcscfAddr-UsingPCO</i>	<ul style="list-style-type: none"> <li>• P-CSCF Address using PCO Flag           <ul style="list-style-type: none"> <li>– 1 - (TRUE) implies request PCSCF address using PCO</li> <li>– 0 - (FALSE) implies do not request By default, this value is 0</li> </ul> </li> </ul>
<i>pPdpAccess-ConFlag</i>	<ul style="list-style-type: none"> <li>• PDP access control flag           <ul style="list-style-type: none"> <li>– 0 - PDP access control none</li> <li>– 1 - PDP access control reject</li> <li>– 2 - PDP access control permission</li> </ul> </li> </ul>
<i>pPcscfAddr-UsingDhcp</i>	<ul style="list-style-type: none"> <li>• P-CSCF address using DHCP           <ul style="list-style-type: none"> <li>– 1 - (TRUE) implies Request PCSCF address using DHCP</li> <li>– 0 - (FALSE) implies do not request By default, value is 0</li> </ul> </li> </ul>
<i>pImCnFlag</i>	<ul style="list-style-type: none"> <li>• IM CN flag           <ul style="list-style-type: none"> <li>– 1 - (TRUE) implies request IM CN flag for this profile</li> <li>– 0 - (FALSE) implies do not request IM CN flag for this profile</li> </ul> </li> </ul>



<i>pTFTID1Params</i>	<ul style="list-style-type: none"> <li>• Traffic Flow Template</li> </ul>
<i>pTFTID2Params</i>	<ul style="list-style-type: none"> <li>• Traffic Flow Template</li> </ul>
<i>pPdpContext</i>	<ul style="list-style-type: none"> <li>• PDP context number</li> </ul>
<i>pSecondaryFlag</i>	<ul style="list-style-type: none"> <li>• PDP context secondary flag <ul style="list-style-type: none"> <li>– 1 - (TRUE) implies this is secondary profile</li> <li>– 0 - (FALSE) implies this is not secondary profile</li> </ul> </li> </ul>
<i>pPrimaryID</i>	<ul style="list-style-type: none"> <li>• PDP context primary ID</li> <li>• function <a href="#">SLQSGetProfileSettings()</a> returns a default value 0xFF if this parameter is not returned by the device</li> </ul>
<i>pIPv6AddPref</i>	<ul style="list-style-type: none"> <li>• IPv6 address preference Preferred IPv6 address to be assigned to the TE; actual assigned address is negotiated with the network and may differ from this value; if not specified, the IPv6 address is obtained automatically from the network</li> </ul>
<i>pUMTSReqQoS-SigInd</i>	<ul style="list-style-type: none"> <li>• UMTS requested QoS with Signalling Indication flag</li> </ul>
<i>pUMTSMInQoS-SigInd</i>	<ul style="list-style-type: none"> <li>• UMTS minimum QoS with Signalling Indication flag</li> </ul>
<i>pPrimaryDNSIPv6addpref</i>	<ul style="list-style-type: none"> <li>• Primary DNS IPv6 address preference <ul style="list-style-type: none"> <li>– The value may be used as a preference during negotiation with the network; if not specified, the wireless device will attempt to obtain the DNS address automatically from the network; the negotiated value is provided to the host via DHCP</li> </ul> </li> </ul>
<i>pSecondaryDNSIPv6addpref</i>	<ul style="list-style-type: none"> <li>• Secondary DNS IPv6 address preference</li> </ul>
<i>paddrAllocation-Pref</i>	<ul style="list-style-type: none"> <li>• DHCP/NAS preference <ul style="list-style-type: none"> <li>– This enumerated value may be used to indicate the address allocation preference <ul style="list-style-type: none"> <li>* 0 - NAS signaling is used for address allocation</li> <li>* 1 - DHCP is used for address allocation</li> </ul> </li> </ul> </li> </ul>

<i>pQosClassID</i>	<ul style="list-style-type: none"> <li>• 3GPP LTE QoS parameters</li> </ul>
<i>pAPNDisabled-Flag</i>	<ul style="list-style-type: none"> <li>• Optional 1 Byte Flag indicating if the APN is disabled/enabled</li> <li>• If set, the profile can not be used for making data calls</li> <li>• Any data call is failed locally</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0 - FALSE(default)</li> <li>– 1 - True</li> </ul> </li> <li>• This parameter is currently read only and can be read by using the function <a href="#">SLQSGet-ProfileSettings()</a>.</li> </ul>
<i>pPDNInactiv-Timeout</i>	<ul style="list-style-type: none"> <li>• Optional 4 Bytes indicating the duration of inactivity timer in seconds</li> <li>• If the PDP context/PDN connection is inactive for this duration i.e. No data Tx/Rx occurs, the PDP context/PDN connection is disconnected</li> <li>• Default value of zero indicates infinite value</li> <li>• This parameter is currently read only and can be read by using the function <a href="#">SLQSGet-ProfileSettings()</a>.</li> </ul>
<i>pAPNClass</i>	<ul style="list-style-type: none"> <li>• Optional 1 Byte numeric identifier representing the APN in profile</li> <li>• Can be set and queried but is not used by the modem</li> <li>• This parameter is currently read only and can be read by using the function <a href="#">SLQSGet-ProfileSettings()</a>.</li> </ul>

## 8.331.2 Field Documentation

8.331.2.1 **BYTE\*** Profile3GPP::pAddrAllocPref

8.331.2.2 **BYTE\*** Profile3GPP::pAPNClass

8.331.2.3 **BYTE\*** Profile3GPP::pAPNDisabledFlag

8.331.2.4 **CHAR\*** Profile3GPP::pAPNName

8.331.2.5 **WORD\*** Profile3GPP::pAPNnameSize

8.331.2.6 **BYTE\*** Profile3GPP::pAuthenticationPref

8.331.2.7 **struct GPRSRequestedQoS\*** Profile3GPP::pGPRSMinimumQoS

8.331.2.8 **struct GPRSRequestedQoS\*** Profile3GPP::pGPRSRequestedQoS

8.331.2.9 **BYTE\*** Profile3GPP::plmCnFlag

- 8.331.2.10 **ULONG\*** Profile3GPP::pIPv4AddrPref
- 8.331.2.11 **USHORT\*** Profile3GPP::pIPv6AddPref
- 8.331.2.12 **CHAR\*** Profile3GPP::pPassword
- 8.331.2.13 **WORD\*** Profile3GPP::pPasswordSize
- 8.331.2.14 **BYTE\*** Profile3GPP::pPcscfAddrUsingDhcp
- 8.331.2.15 **BYTE\*** Profile3GPP::pPcscfAddrUsingPCO
- 8.331.2.16 **ULONG\*** Profile3GPP::pPDNInactivTimeout
- 8.331.2.17 **BYTE\*** Profile3GPP::pPdpAccessConFlag
- 8.331.2.18 **BYTE\*** Profile3GPP::pPdpContext
- 8.331.2.19 **BYTE\*** Profile3GPP::pPdpDataCompType
- 8.331.2.20 **BYTE\*** Profile3GPP::pPdpHdrCompType
- 8.331.2.21 **BYTE\*** Profile3GPP::pPDType
- 8.331.2.22 **ULONG\*** Profile3GPP::pPriDNSIPv4AddPref
- 8.331.2.23 **USHORT\*** Profile3GPP::pPriDNSIPv6addpref
- 8.331.2.24 **BYTE\*** Profile3GPP::pPrimaryID
- 8.331.2.25 **CHAR\*** Profile3GPP::pProfilename
- 8.331.2.26 **WORD\*** Profile3GPP::pProfilenameSize
- 8.331.2.27 **struct QosClassID\*** Profile3GPP::pQosClassID
- 8.331.2.28 **ULONG\*** Profile3GPP::pSecDNSIPv4AddPref
- 8.331.2.29 **USHORT\*** Profile3GPP::pSecDNSIPv6addpref
- 8.331.2.30 **BYTE\*** Profile3GPP::pSecondaryFlag
- 8.331.2.31 **struct TFTIDParams\*** Profile3GPP::pTFTID1Params
- 8.331.2.32 **struct TFTIDParams\*** Profile3GPP::pTFTID2Params
- 8.331.2.33 **struct UMTSQoS\*** Profile3GPP::pUMTSMinQoS
- 8.331.2.34 **struct UMTSReqQoSSigInd\*** Profile3GPP::pUMTSMinQoSSigInd
- 8.331.2.35 **struct UMTSQoS\*** Profile3GPP::pUMTSReqQoS
- 8.331.2.36 **struct UMTSReqQoSSigInd\*** Profile3GPP::pUMTSReqQoSsigInd
- 8.331.2.37 **CHAR\*** Profile3GPP::pUsername

8.331.2.38 WORD\* Profile3GPP::pUsernameSize

## 8.332 Profile3GPP2 Struct Reference

### Data Fields

- BYTE \* pNegoDnsSrvrPref
- ULONG \* pPppSessCloseTimerDO
- ULONG \* pPppSessCloseTimer1x
- BYTE \* pAllowLinger
- USHORT \* pLcpAckTimeout
- USHORT \* plpcpAckTimeout
- USHORT \* pAuthTimeout
- BYTE \* pLcpCreqRetryCount
- BYTE \* plpcpCreqRetryCount
- BYTE \* pAuthRetryCount
- BYTE \* pAuthProtocol
- CHAR \* pUserId
- WORD \* pUserIdSize
- CHAR \* pAuthPassword
- WORD \* pAuthPasswordSize
- BYTE \* pDataRate
- ULONG \* pAppType
- BYTE \* pDataMode
- BYTE \* pAppPriority
- CHAR \* pApnString
- WORD \* pApnStringSize
- BYTE \* pPdnType
- BYTE \* plsPcsfAddressNedded
- ULONG \* pPrimaryV4DnsAddress
- ULONG \* pSecondaryV4DnsAddress
- USHORT \* pPriV6DnsAddress
- USHORT \* pSecV6DnsAddress
- BYTE \* pRATType
- BYTE \* pAPNEnabled3GPP2
- ULONG \* pPDNInactivTimeout3GPP2
- BYTE \* pAPNClass3GPP2

### 8.332.1 Detailed Description

This structure contains the 3GPP2 profile parameters

- Parameter values default to their data type's maximum unsigned value unless explicitly stated otherwise.

## Parameters

<i>pNegoDnsSrvr-Pref</i>	<ul style="list-style-type: none"> <li>• Negotiate DNS Server Preference <ul style="list-style-type: none"> <li>– 1 - (TRUE) implies request DNS addresses from the PDSN</li> <li>– 0 - (FALSE) implies do not request DNS addresses from the PDSN</li> <li>– Default value is 1 (TRUE)</li> </ul> </li> </ul>
<i>pPppSessClose-TimerDO</i>	<ul style="list-style-type: none"> <li>• PPP Session Close Timer for DO <ul style="list-style-type: none"> <li>– Timer value (in seconds) on DO indicating how long the PPP Session should linger before closing down</li> </ul> </li> </ul>
<i>pPppSessClose-Timer1x</i>	<ul style="list-style-type: none"> <li>• PPP Session Close Timer for 1X <ul style="list-style-type: none"> <li>– Timer value (in seconds) on 1X indicating how long the PPP session should linger before closing down</li> </ul> </li> </ul>
<i>pAllowLinger</i>	<ul style="list-style-type: none"> <li>• Allow/disallow lingering of interface <ul style="list-style-type: none"> <li>– 1 -(TRUE) implies allow lingering</li> <li>– 0 -(FALSE) implies do not allow lingering</li> </ul> </li> </ul>
<i>pLcpAckTimeout</i>	<ul style="list-style-type: none"> <li>• LCP ACK Timeout <ul style="list-style-type: none"> <li>– Value of LCP ACK Timeout in milliseconds</li> </ul> </li> </ul>
<i>plpcpAck-Timeout</i>	<ul style="list-style-type: none"> <li>• IPCP ACK Timeout <ul style="list-style-type: none"> <li>– Value of IPCP ACK Timeout in milliseconds</li> </ul> </li> </ul>
<i>pAuthTimeout</i>	<ul style="list-style-type: none"> <li>• AUTH Timeout <ul style="list-style-type: none"> <li>– Value of Authentication Timeout in milliseconds</li> </ul> </li> </ul>
<i>pLcpCreqRetry-Count</i>	<ul style="list-style-type: none"> <li>• LCP Configuration Request Retry Count</li> </ul>
<i>plpcpCreqRetry-Count</i>	<ul style="list-style-type: none"> <li>• IPCP Configuration Request Retry Count</li> </ul>

<i>pAuthRetry-Count</i>	<ul style="list-style-type: none"> <li>• Authentication Retry Count value</li> </ul>
<i>pAuthProtocol</i>	<ul style="list-style-type: none"> <li>• Authentication Protocol <ul style="list-style-type: none"> <li>– 1 - PAP</li> <li>– 2 - CHAP</li> <li>– 3 - PAP or CHAP</li> </ul> </li> </ul>
<i>pUserId</i>	<ul style="list-style-type: none"> <li>• User ID to be used during data network authentication</li> <li>• maximum length allowed is 127 bytes;</li> <li>• QMI_ERR_ARG_TOO_LONG will be returned if the storage on the wireless device is insufficient in size to hold the value.</li> </ul>
<i>pUserIdSize;</i>	<ul style="list-style-type: none"> <li>• This parameter is an input parameter and should be initialised to the size of pUserId field. Size of this parameter is 2 bytes.</li> </ul>
<i>pAuthPassword</i>	<ul style="list-style-type: none"> <li>• Password to be used during data network authentication;</li> <li>• maximum length allowed is 127 bytes</li> <li>• QMI_ERR_ARG_TOO_LONG will be returned if the storage on the wireless device is insufficient in size to hold the value.</li> </ul>
<i>pAuthPassword-Size;</i>	<ul style="list-style-type: none"> <li>• This parameter is an input parameter and should be initialised to the size of pAuthPassword field. Size of this parameter is 2 bytes.</li> </ul>
<i>pDataRate</i>	<ul style="list-style-type: none"> <li>• Data Rate Requested <ul style="list-style-type: none"> <li>– 0 - Low (Low speed Service Options (SO15) only)</li> <li>– 1 - Medium (SO33 + low R-SCH)</li> <li>– 2 - High (SO33 + high R-SCH)</li> <li>– Default is 2</li> </ul> </li> </ul>
<i>pAppType</i>	<ul style="list-style-type: none"> <li>• Application Type: <ul style="list-style-type: none"> <li>– 0x00000001 - Default Application Type</li> <li>– 0x00000020 - LBS Application Type</li> <li>– 0x00000040 - Tethered Application Type</li> <li>– This parameter is not used while creating/modifying a profile</li> </ul> </li> </ul>

<i>pDataMode</i>	<ul style="list-style-type: none"> <li>Data Mode to use: <ul style="list-style-type: none"> <li>0 - CDMA or HDR (Hybrid 1X/1xEV-DO)</li> <li>1 - CDMA Only (1X only)</li> <li>2 - HDR Only (1xEV-DO only)</li> <li>Default is 0</li> </ul> </li> </ul>
<i>pAppPriority</i>	<ul style="list-style-type: none"> <li>Application Priority <ul style="list-style-type: none"> <li>Numerical 1 byte value defining the application priority; higher value implies higher priority</li> <li>This parameter is not used while creating/modifying a profile</li> </ul> </li> </ul>
<i>pApnString</i>	<ul style="list-style-type: none"> <li>String representing the Access Point Name</li> <li>maximum length allowed is 100 bytes</li> <li>QMI_ERR_ARG_TOO_LONG will be returned if the APN name is too long.</li> </ul>
<i>pApnStringSize;</i>	<ul style="list-style-type: none"> <li>This parameter is an input parameter and should be initialised to the size of pApnString field. Size of this parameter is 2 bytes.</li> </ul>
<i>pPdnType</i>	<ul style="list-style-type: none"> <li>Packed Data Network Type Requested: <ul style="list-style-type: none"> <li>0 - IPv4 PDN Type</li> <li>1 - IPv6 PDN Type</li> <li>2 - IPv4 or IPv6 PDN Type</li> <li>3 - Unspecified PDN Type (implying no preference)</li> </ul> </li> </ul>
<i>plsPcscf-AddressNedded</i>	<ul style="list-style-type: none"> <li>This boolean value is used to control if PCSCF address is requested from PDSN <ul style="list-style-type: none"> <li>1 -(TRUE) implies request for PCSCF value from the PDSN</li> <li>0 -(FALSE) implies do not request for PCSCF value from the PDSN</li> </ul> </li> </ul>
<i>pPrimaryV4Dns-Address</i>	<ul style="list-style-type: none"> <li>IPv4 Primary DNS address <ul style="list-style-type: none"> <li>The Primary IPv4 DNS address that can be statically assigned to the UE</li> </ul> </li> </ul>

<i>pSecondaryV4-DnsAddress</i>	<ul style="list-style-type: none"> <li>IPv4 Secondary DNS address <ul style="list-style-type: none"> <li>The Secondary IPv4 DNS address that can be statically assigned to the UE</li> </ul> </li> </ul>
<i>pPriV6Dns-Address</i>	<ul style="list-style-type: none"> <li>Primary IPv6 DNS address <ul style="list-style-type: none"> <li>The Primary IPv6 DNS address that can be statically assigned to the UE</li> </ul> </li> </ul>
<i>pSecV6Dns-Address</i>	<ul style="list-style-type: none"> <li>Secondary IPv6 DNS address <ul style="list-style-type: none"> <li>The Secondary IPv6 DNS address that can be statically assigned to the UE</li> </ul> </li> </ul>
<i>pRATType</i>	<ul style="list-style-type: none"> <li>Optional 1 Byte Flag indicating RAT Type</li> <li>Values: <ul style="list-style-type: none"> <li>1 - HRPD</li> <li>2 - EHRPD</li> <li>3 - HRPD_EHRPD</li> </ul> </li> <li>This parameter is currently read only and can be read by using the function <a href="#">SLQSGet-ProfileSettings()</a>.</li> </ul>
<i>pAPNEnabled3-GPP2</i>	<ul style="list-style-type: none"> <li>Optional 1 Byte Flag indicating if the APN is disabled/enabled</li> <li>If disabled, the profile can not be used for making data calls</li> <li>Values: <ul style="list-style-type: none"> <li>0 - Disabled</li> <li>1 - Enabled(default value)</li> </ul> </li> <li>This parameter is currently read only and can be read by using the function <a href="#">SLQSGet-ProfileSettings()</a>.</li> </ul>
<i>pPDNInactiv-Timeout3GPP2</i>	<ul style="list-style-type: none"> <li>Optional 4 Bytes indicating the duration of inactivity timer in seconds</li> <li>If the PDP context/PDN connection is inactive for this duration i.e. No data Tx/Rx occurs, the PDP context/PDN connection is disconnected</li> <li>Default value of zero indicates infinite value</li> <li>This parameter is currently read only and can be read by using the function <a href="#">SLQSGet-ProfileSettings()</a>.</li> </ul>



<i>pAPNClass3GPP2</i>	<ul style="list-style-type: none"> <li>• Optional 1 Byte numeric identifier representing the APN in profile</li> <li>• Can be set and queried but is not used by the modem</li> <li>• This parameter is currently read only and can be read by using the function <a href="#">SLQSGetProfileSettings()</a>.</li> </ul>
-----------------------	--

## 8.332.2 Field Documentation

8.332.2.1 **BYTE\*** Profile3GPP2::pAllowLinger

8.332.2.2 **BYTE\*** Profile3GPP2::pAPNClass3GPP2

8.332.2.3 **BYTE\*** Profile3GPP2::pAPNEnabled3GPP2

8.332.2.4 **CHAR\*** Profile3GPP2::pApnString

8.332.2.5 **WORD\*** Profile3GPP2::pApnStringSize

8.332.2.6 **BYTE\*** Profile3GPP2::pAppPriority

8.332.2.7 **ULONG\*** Profile3GPP2::pAppType

8.332.2.8 **CHAR\*** Profile3GPP2::pAuthPassword

8.332.2.9 **WORD\*** Profile3GPP2::pAuthPasswordSize

8.332.2.10 **BYTE\*** Profile3GPP2::pAuthProtocol

8.332.2.11 **BYTE\*** Profile3GPP2::pAuthRetryCount

8.332.2.12 **USHORT\*** Profile3GPP2::pAuthTimeout

8.332.2.13 **BYTE\*** Profile3GPP2::pDataMode

8.332.2.14 **BYTE\*** Profile3GPP2::pDataRate

8.332.2.15 **USHORT\*** Profile3GPP2::pIpcpAckTimeout

8.332.2.16 **BYTE\*** Profile3GPP2::pIpcpCreqRetryCount

8.332.2.17 **BYTE\*** Profile3GPP2::pIscscfAddressNedded

8.332.2.18 **USHORT\*** Profile3GPP2::pLcpAckTimeout

8.332.2.19 **BYTE\*** Profile3GPP2::pLcpCreqRetryCount

8.332.2.20 **BYTE\*** Profile3GPP2::pNegoDnsSrvrPref

8.332.2.21 **ULONG\*** Profile3GPP2::pPDNInactivTimeout3GPP2

8.332.2.22 **BYTE\*** Profile3GPP2::pPdnType

- 8.332.2.23 **ULONG\*** Profile3GPP2::pPppSessCloseTimer1x
- 8.332.2.24 **ULONG\*** Profile3GPP2::pPppSessCloseTimerDO
- 8.332.2.25 **ULONG\*** Profile3GPP2::pPrimaryV4DnsAddress
- 8.332.2.26 **USHORT\*** Profile3GPP2::pPriV6DnsAddress
- 8.332.2.27 **BYTE\*** Profile3GPP2::pRATType
- 8.332.2.28 **ULONG\*** Profile3GPP2::pSecondaryV4DnsAddress
- 8.332.2.29 **USHORT\*** Profile3GPP2::pSecV6DnsAddress
- 8.332.2.30 **CHAR\*** Profile3GPP2::pUserId
- 8.332.2.31 **WORD\*** Profile3GPP2::pUserIdSize

### 8.333 ProfileIdentifier Struct Reference

#### Data Fields

- [BYTE profileType](#)
- [BYTE profileIndex](#)

#### 8.333.1 Detailed Description

This structure contains the Profile Identifier Information

- Parameter values default to their data type's maximum unsigned value unless explicitly stated otherwise.

#### Parameters

<i>profileType</i>	<ul style="list-style-type: none"> <li>• Identifies the type of profile 0x00 = 3GPP</li> </ul>
<i>profileIndex</i>	<ul style="list-style-type: none"> <li>• Index of profile whose settings were loaded prior to session parameter negotiation for the current call. If this TLV is not present, data call parameters are based on device default settings for each parameter</li> </ul>

#### 8.333.2 Field Documentation

- 8.333.2.1 **BYTE** ProfileIdentifier::profileIndex
- 8.333.2.2 **BYTE** ProfileIdentifier::profileType

### 8.334 protocolSubtypeElement Struct Reference

#### Data Fields

- [WORD PhysicalLayer](#)

- [WORD ControlMac](#)
- [WORD AccessMac](#)
- [WORD ForwardMac](#)
- [WORD ReverseMac](#)
- [WORD KeyExchange](#)
- [WORD AuthProt](#)
- [WORD EncryptProt](#)
- [WORD SecProt](#)
- [WORD IdleState](#)
- [WORD MultDisc](#)
- [WORD VirtStream](#)

### 8.334.1 Detailed Description

This structure contains Protocol Subtype Elements for Protocol Subtype List

#### Parameters

<i>PhysicalLayer</i>	<ul style="list-style-type: none"> <li>• Specifies Physical Layer Protocol subtype</li> </ul>
<i>ControlMac</i>	<ul style="list-style-type: none"> <li>• Specifies Control Channel MAC Protocol subtype</li> </ul>
<i>AccessMac</i>	<ul style="list-style-type: none"> <li>• Specifies Access Channel MAC Protocol subtype</li> </ul>
<i>ForwardMac</i>	<ul style="list-style-type: none"> <li>• Specifies Forward Traffic Channel MAC Protocol subtype</li> </ul>
<i>ReverseMac</i>	<ul style="list-style-type: none"> <li>• Specifies Reverse Traffic Channel MAC Protocol subtype</li> </ul>
<i>KeyExchange</i>	<ul style="list-style-type: none"> <li>• Specifies Key exchange Protocol subtype</li> </ul>
<i>AuthProt</i>	<ul style="list-style-type: none"> <li>• Specifies Authentication Protocol subtype</li> </ul>
<i>EncryptProt</i>	<ul style="list-style-type: none"> <li>• Specifies Encryption Protocol subtype</li> </ul>
<i>SecProt</i>	<ul style="list-style-type: none"> <li>• Specifies Security Protocol subtype</li> </ul>
<i>IdleState</i>	<ul style="list-style-type: none"> <li>• Specifies Idle state Protocol subtype</li> </ul>
<i>MultDisc</i>	<ul style="list-style-type: none"> <li>• Specifies Generic multimode capability discovery Protocol subtype</li> </ul>

<i>VirtStream</i>	<ul style="list-style-type: none"> <li>• Specifies Generic Virtual Stream Protocol subtype</li> </ul>
-------------------	---

### 8.334.2 Field Documentation

8.334.2.1 WORD protocolSubtypeElement::AccessMac

8.334.2.2 WORD protocolSubtypeElement::AuthProt

8.334.2.3 WORD protocolSubtypeElement::ControlMac

8.334.2.4 WORD protocolSubtypeElement::EncryptProt

8.334.2.5 WORD protocolSubtypeElement::ForwardMac

8.334.2.6 WORD protocolSubtypeElement::IdleState

8.334.2.7 WORD protocolSubtypeElement::KeyExchange

8.334.2.8 WORD protocolSubtypeElement::MultDisc

8.334.2.9 WORD protocolSubtypeElement::PhysicalLayer

8.334.2.10 WORD protocolSubtypeElement::ReverseMac

8.334.2.11 WORD protocolSubtypeElement::SecProt

8.334.2.12 WORD protocolSubtypeElement::VirtStream

## 8.335 PSDetachReq Struct Reference

### Data Fields

- [BYTE](#) \* [pDetachAction](#)

### 8.335.1 Detailed Description

This structure contains information about the SLQSSwiPSDetach request parameters.

#### Parameters

<i>pDetachAction</i> [1- N]	<ul style="list-style-type: none"> <li>• Values <ul style="list-style-type: none"> <li>– 2- Initiates an immediate packet domain detach.</li> </ul> </li> </ul>
--------------------------------	---

### 8.335.2 Field Documentation

8.335.2.1 [BYTE](#)\* PSDetachReq::pDetachAction

## 8.336 qaQmi3Gpp2TimeZone Struct Reference

### Data Fields

- [BYTE leapSeconds](#)
- [BYTE localTimeOffset](#)
- [BYTE daylightSavings](#)

### 8.336.1 Detailed Description

This structure contains the 3GPP2TimeZone parameters

#### Parameters

<i>leapSeconds</i>	<ul style="list-style-type: none"><li>• leap seconds - Number of leap seconds since the start of CDMA system time.</li></ul>
<i>localTimeOffset</i>	<ul style="list-style-type: none"><li>• Local Time Offset - Offset of system time in units of 30 minutes; the value in this field conveys as 8 bit 2's compliment number.</li></ul>
<i>daylightSavings</i>	<ul style="list-style-type: none"><li>• Day Light Savings Indicator<ul style="list-style-type: none"><li>– 0x00 - OFF (daylight savings not in effect)</li><li>– 0x01 - ON (daylight savings in effect)</li></ul></li></ul>

### 8.336.2 Field Documentation

8.336.2.1 **BYTE** qaQmi3Gpp2TimeZone::daylightSavings

8.336.2.2 **BYTE** qaQmi3Gpp2TimeZone::leapSeconds

8.336.2.3 **BYTE** qaQmi3Gpp2TimeZone::localTimeOffset

## 8.337 qaQmiInterfaceInfo Struct Reference

### Data Fields

- [BYTE qaQmiinstanceid](#)
- [eQaQMIService qaQmisvctype](#)
- [ULONG v4sessionId](#)
- [ULONG v6sessionId](#)

### 8.337.1 Detailed Description

Structure used to store the service, interface and session information

## Parameters

<i>qaQmiinstanceid</i>	<ul style="list-style-type: none"> <li>The interface instance ID <ul style="list-style-type: none"> <li>0x00 - PDP instance ID 0</li> <li>0x01 - PDP instance ID 1</li> <li>0x02 - PDP instance ID 2</li> </ul> </li> </ul>
<i>qaQmisvctype</i>	<ul style="list-style-type: none"> <li>The service type information. See <a href="#">eQaQMIService</a> for more information</li> </ul>
<i>v4sessionId</i>	<ul style="list-style-type: none"> <li>IPv4 QMI client session handle</li> </ul>
<i>v6sessionId</i>	<ul style="list-style-type: none"> <li>IPv6 QMI client session handle</li> </ul>

## 8.337.2 Field Documentation

8.337.2.1 BYTE qaQmiInterfaceInfo::qaQmiinstanceid

8.337.2.2 eQaQMIService qaQmiInterfaceInfo::qaQmisvctype

8.337.2.3 ULONG qaQmiInterfaceInfo::v4sessionId

8.337.2.4 ULONG qaQmiInterfaceInfo::v6sessionId

## 8.338 qaQmiServingSystemParam Struct Reference

## Data Fields

- [servSystem](#) ServingSystem
- BYTE roamIndicatorVal
- [dataSrvCapabilities](#) DataSrvCapabilities
- [currentPLMN](#) CurrentPLMN
- WORD SystemID
- WORD NetworkID
- WORD BasestationID
- ULONG BasestationLatitude
- ULONG BasestationLongitude
- [roamIndList](#) RoamingIndicatorList
- BYTE defaultRoamInd
- [qaQmi3Gpp2TimeZone](#) Gpp2TimeZone
- BYTE CDMA\_P\_Rev
- BYTE GppTimeZone
- BYTE GppNetworkDSTAdjustment
- WORD Lac
- ULONG CellID
- BYTE concSvcInfo
- BYTE PRLInd

- [BYTE DTMIInd](#)
- [detailSvcInfo DetailedSvcInfo](#)
- [CDMASysInfoExt CDMASystemInfoExt](#)
- [BYTE hdrPersonality](#)
- [WORD trackAreaCode](#)
- [callBarStatus CallBarStatus](#)

### 8.338.1 Detailed Description

This structure contains the Serving System parameters

- Parameter values default to their data type's maximum unsigned value unless explicitly stated otherwise.

#### Parameters

<i>ServingSystem</i>	<ul style="list-style-type: none"> <li>• Serving System</li> <li>• See <a href="#">servSystem</a> for more information</li> </ul>
<i>roamIndicatorVal</i>	<ul style="list-style-type: none"> <li>• Optional parameter indicating Roaming Indicator value</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0x00 - Roaming</li> <li>– 0x01 - Home</li> <li>– 0x02 - Flashing</li> <li>– 0x03 and above - Operator defined values</li> </ul> </li> </ul>
<i>DataSrv-Capabilities</i>	<ul style="list-style-type: none"> <li>• Optional parameter indicating Data services capability</li> <li>• See <a href="#">dataSrvCapabilities</a> for more information</li> </ul>
<i>CurrentPLMN</i>	<ul style="list-style-type: none"> <li>• Optional parameter indicating Current PLMN</li> <li>• See <a href="#">currentPLMN</a> for more information</li> </ul>
<i>SystemID</i>	<ul style="list-style-type: none"> <li>• Optional parameter indicating System ID</li> </ul>
<i>NetworkID</i>	<ul style="list-style-type: none"> <li>• Optional parameter indicating Network ID</li> </ul>
<i>BaseStationID</i>	<ul style="list-style-type: none"> <li>• Optional parameter indicating Base Station Identification Number</li> </ul>

<i>BaseStation-Latitude</i>	<ul style="list-style-type: none"> <li>Optional parameter indicating Base station latitude in units of 0.25 sec, expressed as a two's complement signed number with positive numbers signifying North latitude</li> </ul>
<i>Basestation-Longitude</i>	<ul style="list-style-type: none"> <li>Optional parameter indicating Base station longitude in units of 0.25 sec, expressed as a Two's complement signed number with positive numbers signifying East longitude</li> </ul>
<i>Roaming-IndicatorList</i>	<ul style="list-style-type: none"> <li>Optional parameter indicating Roaming Indicator List</li> <li>See <a href="#">roamIndList</a> for more information</li> </ul>
<i>defaultRoamInd</i>	<ul style="list-style-type: none"> <li>Optional parameter indicating Default Roaming Indicator</li> <li>Values: <ul style="list-style-type: none"> <li>0x00 - Roaming</li> <li>0x01 - Home</li> </ul> </li> </ul>
<i>Gpp2TimeZone</i>	<ul style="list-style-type: none"> <li>Optional parameter indicating 3GPP2 Time Zone</li> <li>See <a href="#">qaQmi3Gpp2TimeZone</a> for more information</li> </ul>
<i>CDMA_P_Rev</i>	<ul style="list-style-type: none"> <li>Optional parameter indicating CDMA P_Rev in use</li> </ul>
<i>GppTimeZone</i>	<ul style="list-style-type: none"> <li>Optional parameter indicating Offset from Universal time, i.e., difference between local time and Universal time, in increments of 15 min. (signed value).</li> </ul>
<i>GppNetworkDS-TAdjustment</i>	<ul style="list-style-type: none"> <li>Optional parameter indicating 3GPP network daylight saving adjustment</li> <li>Values: <ul style="list-style-type: none"> <li>0x00 - No adjustment for Daylight Saving Time</li> <li>0x01 - 1 hr adjustment for Daylight Saving Time</li> <li>0x02 - 2 hr adjustment for Daylight Saving Time</li> </ul> </li> </ul>
<i>Lac</i>	<ul style="list-style-type: none"> <li>Optional parameter indicating 3GPP Location Area Code</li> </ul>
<i>CellID</i>	<ul style="list-style-type: none"> <li>Optional parameter indicating 3GPP Cell ID</li> </ul>
<i>concSvcInfo</i>	<ul style="list-style-type: none"> <li>Optional parameter indicating 3GPP2 concurrent service Info</li> <li>Values: <ul style="list-style-type: none"> <li>0x00 - Concurrent service not available</li> <li>0x01 - Concurrent service available</li> </ul> </li> </ul>
Generated on Wed Oct 28 2015 00:40:12 for LinuxQMISDK by Doxygen	



<i>PRLInd</i>	<ul style="list-style-type: none"> <li>Optional parameter indicating 3GPP2 PRL Indicator</li> <li>Values: <ul style="list-style-type: none"> <li>0x00 - System not in PRL</li> <li>0x01 - System is in PRL</li> </ul> </li> </ul>
<i>DTMInd</i>	<ul style="list-style-type: none"> <li>Optional parameter indicating Dual Transfer Mode Indication(GSM Only)</li> <li>Values: <ul style="list-style-type: none"> <li>0x00 - DTM not supported</li> <li>0x01 - DTM supported</li> </ul> </li> </ul>
<i>DetailedSvcInfo</i>	<ul style="list-style-type: none"> <li>Optional parameter indicating Detailed service information</li> <li>See <a href="#">detailSvcInfo</a> for more information</li> </ul>
<i>CDMASystem-InfoExt</i>	<ul style="list-style-type: none"> <li>Optional parameter indicating CDMA System Info Ext</li> <li>See <a href="#">CDMASysInfoExt</a> for more information</li> </ul>
<i>hdrPersonality</i>	<ul style="list-style-type: none"> <li>Optional parameter indicating HDR Personality Information</li> <li>Values: <ul style="list-style-type: none"> <li>0x00 - Unknown</li> <li>0x01 - HRPD</li> <li>0x02 - eHRPD</li> </ul> </li> </ul>
<i>trackAreaCode</i>	<ul style="list-style-type: none"> <li>Optional parameter indicating Tracking area code information for LTE</li> </ul>
<i>CallBarStatus</i>	<ul style="list-style-type: none"> <li>Optional parameter indicating Call Barring Status</li> <li>See <a href="#">callBarStatus</a> for more information</li> </ul>

## 8.338.2 Field Documentation

8.338.2.1 **WORD** qaQmiServingSystemParam::BasestationID

8.338.2.2 **ULONG** qaQmiServingSystemParam::BasestationLatitude

8.338.2.3 **ULONG** qaQmiServingSystemParam::BasestationLongitude

8.338.2.4 **callBarStatus** qaQmiServingSystemParam::CallBarStatus

- 8.338.2.5 **BYTE** qaQmiServingSystemParam::CDMA\_P\_Rev
- 8.338.2.6 **CDMASysInfoExt** qaQmiServingSystemParam::CDMASysInfoExt
- 8.338.2.7 **ULONG** qaQmiServingSystemParam::CellID
- 8.338.2.8 **BYTE** qaQmiServingSystemParam::concSvcInfo
- 8.338.2.9 **currentPLMN** qaQmiServingSystemParam::CurrentPLMN
- 8.338.2.10 **dataSrvCapabilities** qaQmiServingSystemParam::DataSrvCapabilities
- 8.338.2.11 **BYTE** qaQmiServingSystemParam::defaultRoamInd
- 8.338.2.12 **detailSvcInfo** qaQmiServingSystemParam::DetailedSvcInfo
- 8.338.2.13 **BYTE** qaQmiServingSystemParam::DTMInd
- 8.338.2.14 **qaQmi3Gpp2TimeZone** qaQmiServingSystemParam::Gpp2TimeZone
- 8.338.2.15 **BYTE** qaQmiServingSystemParam::GppNetworkDSTAdjustment
- 8.338.2.16 **BYTE** qaQmiServingSystemParam::GppTimeZone
- 8.338.2.17 **BYTE** qaQmiServingSystemParam::hdrPersonality
- 8.338.2.18 **WORD** qaQmiServingSystemParam::Lac
- 8.338.2.19 **WORD** qaQmiServingSystemParam::NetworkID
- 8.338.2.20 **BYTE** qaQmiServingSystemParam::PRLInd
- 8.338.2.21 **BYTE** qaQmiServingSystemParam::roamIndicatorVal
- 8.338.2.22 **roamIndList** qaQmiServingSystemParam::RoamingIndicatorList
- 8.338.2.23 **servSystem** qaQmiServingSystemParam::ServingSystem
- 8.338.2.24 **WORD** qaQmiServingSystemParam::SystemID
- 8.338.2.25 **WORD** qaQmiServingSystemParam::trackAreaCode

## 8.339 QmiCbkCatEventStatusReportInd Struct Reference

### Data Fields

- [BYTE event\\_Index](#)
- struct [CatCommonEventTlv CCETlv](#) [11]

### 8.339.1 Field Documentation

- 8.339.1.1 struct [CatCommonEventTlv](#) QmiCbkCatEventStatusReportInd::CCETlv[11]
- 8.339.1.2 **BYTE** QmiCbkCatEventStatusReportInd::event\_Index

## 8.340 QmiCbkLocCradleMountInd Struct Reference

### Data Fields

- [ULONG cradleMountConfigStatus](#)

### 8.340.1 Detailed Description

This structure contains LOC Cradle Mount Config Status

#### Parameters

<i>cradleMount-ConfigStatus</i>	<ul style="list-style-type: none"> <li>• Values           <ul style="list-style-type: none"> <li>– 0 - Request was completed successfully</li> <li>– 1 - Request failed because of a general failure.</li> <li>– 2 - Request failed because it is not supported.</li> <li>– 3 - Request failed because it contained invalid parameters</li> <li>– 4 - Request failed because the engine is busy</li> <li>– 5 - Request failed because the phone is offline</li> <li>– 6 - Request failed because it timed out</li> <li>– 7 - Request failed because an undefined configuration was requested</li> <li>– 8 - engine could not allocate sufficient memory</li> <li>– 9 - Request failed because the maximum number of Geofences are already programmed</li> <li>– 10 -Location service failed because of an XTRA version-based file format check failure</li> </ul> </li> </ul>
---------------------------------	---

### 8.340.2 Field Documentation

8.340.2.1 [ULONG QmiCbkLocCradleMountInd::cradleMountConfigStatus](#)

## 8.341 QmiCbkLocEventTimeSyncInd Struct Reference

### Data Fields

- [ULONG timeSyncRefCounter](#)

### 8.341.1 Detailed Description

This structure contains LOC Event Time Sync Reference COUNTER

#### Parameters

<i>timeSyncRef-Counter</i>	<ul style="list-style-type: none"> <li>• Sent by the location engine when it needs to synchronize location engine and control point (sensor processor) times.</li> </ul>
----------------------------	--

### 8.341.2 Field Documentation

8.341.2.1 ULONG QmiCbkLocEventTimeSyncInd::timeSyncRefCounter

## 8.342 QmiCbkLocInjectSensorDataInd Struct Reference

### Data Fields

- ULONG injectSensorDataStatus
- ULONG \* pOpaqueIdentifier
- BYTE \* pAccelSamplesAccepted
- BYTE \* pGyroSamplesAccepted
- BYTE \* pAccelTempSamplesAccepted
- BYTE \* pGyroTempSamplesAccepted

### 8.342.1 Detailed Description

This structure contains LOC Inject Sensor Data

#### Parameters

<i>injectSensor-DataStatus</i>	<ul style="list-style-type: none"> <li>• Values               <ul style="list-style-type: none"> <li>– 0 - Request was completed successfully</li> <li>– 1 - Request failed because of a general failure.</li> <li>– 2 - Request failed because it is not supported.</li> <li>– 3 - Request failed because it contained invalid parameters</li> <li>– 4 - Request failed because the engine is busy</li> <li>– 5 - Request failed because the phone is offline</li> <li>– 6 - Request failed because it timed out</li> <li>– 7 - Request failed because an undefined configuration was requested</li> <li>– 8 - engine could not allocate sufficient memory</li> <li>– 9 - Request failed because the maximum number of Geofences are already programmed</li> <li>– 10 -Location service failed because of an XTRA version-based file format check failure</li> </ul> </li> </ul>
--------------------------------	---

<i>pOpaque- Identifier</i>	<ul style="list-style-type: none"> <li>• Sent in by the client echoed so the client can relate the indication to the request.</li> </ul>
<i>pAccelSamples- Accepted</i>	<ul style="list-style-type: none"> <li>• Lets the client know how many 3-axis accelerometer samples were accepted.</li> <li>• This field is present only if the accelerometer samples were sent in the request.</li> </ul>
<i>pGyroSamples- Accepted</i>	<ul style="list-style-type: none"> <li>• Lets the client know how many 3-axis gyroscope samples were accepted.</li> <li>• This field is present only if the gyroscope samples were sent in the request.</li> </ul>
<i>pAccelTemp- Samples- Accepted</i>	<ul style="list-style-type: none"> <li>• Lets the client know how many accelerometer temperature samples were accepted.</li> <li>• This field is present only if the accelerometer temperature samples were sent in the request.</li> </ul>
<i>pGyroTemp- Samples- Accepted</i>	<ul style="list-style-type: none"> <li>• Lets the client know how many gyroscope temperature samples were accepted.</li> <li>• This field is present only if the gyroscope temperature samples were sent in the request.</li> </ul>

### 8.342.2 Field Documentation

8.342.2.1 **ULONG** QmiCbkLocInjectSensorDataInd::injectSensorDataStatus

8.342.2.2 **BYTE\*** QmiCbkLocInjectSensorDataInd::pAccelSamplesAccepted

8.342.2.3 **BYTE\*** QmiCbkLocInjectSensorDataInd::pAccelTempSamplesAccepted

8.342.2.4 **BYTE\*** QmiCbkLocInjectSensorDataInd::pGyroSamplesAccepted

8.342.2.5 **BYTE\*** QmiCbkLocInjectSensorDataInd::pGyroTempSamplesAccepted

8.342.2.6 **ULONG\*** QmiCbkLocInjectSensorDataInd::pOpaqueIdentifier

## 8.343 QmiCbkLocInjectTimeInd Struct Reference

### Data Fields

- [ULONG injectTimeSyncStatus](#)

### 8.343.1 Detailed Description

This structure contains LOC Inject Time Sync Data Status

## Parameters

<i>injectTimeSync-Status</i>	<ul style="list-style-type: none"> <li>• Values <ul style="list-style-type: none"> <li>– 0 - Request was completed successfully</li> <li>– 1 - Request failed because of a general failure.</li> <li>– 2 - Request failed because it is not supported.</li> <li>– 3 - Request failed because it contained invalid parameters</li> <li>– 4 - Request failed because the engine is busy</li> <li>– 5 - Request failed because the phone is offline</li> <li>– 6 - Request failed because it timed out</li> <li>– 7 - Request failed because an undefined configuration was requested</li> <li>– 8 - engine could not allocate sufficient memory</li> <li>– 9 - Request failed because the maximum number of Geofences are already programmed</li> <li>– 10 -Location service failed because of an XTRA version-based file format check failure</li> </ul> </li> </ul>
------------------------------	---

## 8.343.2 Field Documentation

## 8.343.2.1 ULONG QmiCbkLocInjectTimeInd::injectTimeSyncStatus

## 8.344 QmiCbkLocPositionReportInd Struct Reference

## Data Fields

- ULONG sessionStatus
- BYTE sessionId
- ULONGLONG \* pLatitude
- ULONGLONG \* pLongitude
- ULONG \* pHorUncCircular
- ULONG \* pHorUncEllipseSemiMinor
- ULONG \* pHorUncEllipseSemiMajor
- ULONG \* pHorUncEllipseOrientAzimuth
- BYTE \* pHorConfidence
- ULONG \* pHorReliability
- ULONG \* pSpeedHorizontal
- ULONG \* pSpeedUnc
- ULONG \* pAltitudeWrtEllipsoid
- ULONG \* pAltitudeWrtMeanSeaLevel
- ULONG \* pVertUnc
- BYTE \* pVertConfidence
- ULONG \* pVertReliability
- ULONG \* pSpeedVertical
- ULONG \* pHeading
- ULONG \* pHeadingUnc
- ULONG \* pMagneticDeviation
- ULONG \* pTechnologyMask
- precisionDilution \* pPrecisionDilution

- ULONGLONG \* pTimestampUtc
- BYTE \* pLeapSeconds
- gpsTime \* pGpsTime
- ULONG \* pTimeUnc
- ULONG \* pTimeSrc
- sensorDataUsage \* pSensorDataUsage
- ULONG \* pFixId
- svUsedforFix \* pSvUsedforFix
- BYTE \* pAltitudeAssumed

### 8.344.1 Detailed Description

This structure contains Event Position Report

#### Parameters

<i>sessionStatus</i>	<ul style="list-style-type: none"> <li>• Values           <ul style="list-style-type: none"> <li>– 0 - Session was successful</li> <li>– 1 - Session is still in progress; further position reports will be generated until either the fix criteria specified by the client are met or the client response timeout occurs.</li> <li>– 2 - Session failed..</li> <li>– 3 - Fix request failed because the session timed out.</li> <li>– 4 - Fix request failed because the session was ended by the user.</li> <li>– 5 - Fix request failed due to bad parameters in the request.</li> <li>– 6 - Fix request failed because the phone is offline.</li> <li>– 7 - Fix request failed because the engine is locked</li> </ul> </li> </ul>
<i>sessionId</i>	<ul style="list-style-type: none"> <li>• ID of the session that was specified in the Start request</li> <li>• Range - 0 to 255</li> </ul>
<i>pLatitude</i>	<ul style="list-style-type: none"> <li>• Type - Floating point</li> <li>• Units - Degrees</li> <li>• Range - -90.0 to 90.0</li> <li>• Positive values indicate northern latitude</li> <li>• Negative values indicate southern latitude</li> </ul>

<i>pLongitude</i>	<ul style="list-style-type: none"> <li>• Type - Floating point</li> <li>• Units - Degrees</li> <li>• Range - -180.0 to 180.0</li> <li>• Positive values indicate eastern latitude</li> <li>• Negative values indicate western latitude</li> </ul>
<i>pHorUncCircular</i>	<ul style="list-style-type: none"> <li>• Horizontal position uncertainty.</li> <li>• Units - Meters</li> </ul>
<i>pHorUncEllipse-SemiMinor</i>	<ul style="list-style-type: none"> <li>• Semi-minor axis of horizontal elliptical uncertainty.</li> <li>• Units - Meters</li> </ul>
<i>pHorUncEllipse-SemiMajor</i>	<ul style="list-style-type: none"> <li>• Semi-major axis of horizontal elliptical uncertainty.</li> <li>• Units: Meters</li> </ul>
<i>pHorUncEllipse-OrientAzimuth</i>	<ul style="list-style-type: none"> <li>• Elliptical horizontal uncertainty azimuth of orientation.</li> <li>• Units - Decimal degrees</li> <li>• Range - 0 to 180</li> </ul>
<i>pHorConfidence</i>	<ul style="list-style-type: none"> <li>• Horizontal uncertainty confidence.</li> <li>• If both elliptical and horizontal uncertainties are specified in this message, the confidence corresponds to the elliptical uncertainty.</li> <li>• Units - Percentage</li> <li>• Range 0-99</li> </ul>
<i>pHorReliability</i>	<ul style="list-style-type: none"> <li>• Values <ul style="list-style-type: none"> <li>– 0 - Location reliability is not set.</li> <li>– 1 - Location reliability is very low; use it at your own risk</li> <li>– 2 - Location reliability is low; little or no cross-checking is possible.</li> <li>– 3 - Location reliability is medium; limited cross-check passed</li> <li>– 4 - Location reliability is high; strong cross-check passed</li> </ul> </li> </ul>



<i>pSpeed-Horizontal</i>	<ul style="list-style-type: none"> <li>• Horizontal speed.</li> <li>• Units - Meters/second</li> </ul>
<i>pSpeedUnc</i>	<ul style="list-style-type: none"> <li>• 3-D Speed uncertainty.</li> <li>• Units - Meters/second.</li> </ul>
<i>pAltitudeWrt-Ellipsoid</i>	<ul style="list-style-type: none"> <li>• Altitude With Respect to Ellipsoid.</li> <li>• Units - Meters</li> <li>• Range -500 to 15883</li> </ul>
<i>pAltitudeWrt-MeanSeaLevel</i>	<ul style="list-style-type: none"> <li>• Altitude With Respect to Sea Level.</li> <li>• Units - Meters</li> </ul>
<i>pVertUnc</i>	<ul style="list-style-type: none"> <li>• Vertical uncertainty.</li> <li>• Units - Meters</li> </ul>
<i>pVertConfidence</i>	<ul style="list-style-type: none"> <li>• Vertical uncertainty confidence.</li> <li>• Units - Percentage</li> <li>• Range 0 to 99</li> </ul>
<i>pVertReliability</i>	<ul style="list-style-type: none"> <li>• Values <ul style="list-style-type: none"> <li>– 0 - Location reliability is not set.</li> <li>– 1 - Location reliability is very low; use it at your own risk.</li> <li>– 2 - Location reliability is low; little or no cross-checking is possible</li> <li>– 3 - Location reliability is medium; limited cross-check passed</li> <li>– 4 - Location reliability is high; strong cross-check passed</li> </ul> </li> </ul>
<i>pSpeedVertical</i>	<ul style="list-style-type: none"> <li>• Vertical speed.</li> <li>• Units - Meters/second</li> </ul>
<i>pHeading</i>	<ul style="list-style-type: none"> <li>• Heading.</li> <li>• Units - Degree</li> <li>• Range 0 to 359.999</li> </ul>

<i>pHeadingUnc</i>	<ul style="list-style-type: none"> <li>• Heading uncertainty.</li> <li>• Units - Degree</li> <li>• Range 0 to 359.999</li> </ul>
<i>pMagnetic-Deviation</i>	<ul style="list-style-type: none"> <li>• Difference between the bearing to true north and the bearing shown on a magnetic compass. The deviation is positive when the magnetic north is east of true north.</li> </ul>
<i>pTechnology-Mask</i>	<ul style="list-style-type: none"> <li>• Values <ul style="list-style-type: none"> <li>– 0x00000001 - Satellites were used to generate the fix</li> <li>– 0x00000002 - Cell towers were used to generate the fix</li> <li>– 0x00000004 - Wi-Fi access points were used to generate the fix</li> <li>– 0x00000008 - Sensors were used to generate the fix</li> <li>– 0x00000010 - Reference Location was used to generate the fix</li> <li>– 0x00000020 - Coarse position injected into the location engine was used to generate the fix</li> <li>– 0x00000040 - AFLT was used to generate the fix</li> <li>– 0x00000080 - GNSS and network-provided measurements were used to generate the fix</li> </ul> </li> </ul>
<i>-pPrecision-Dilution</i>	<ul style="list-style-type: none"> <li>• See <a href="#">precisionDilution</a> for more information</li> </ul>
<i>pTimestampUtc</i>	<ul style="list-style-type: none"> <li>• UTC timestamp</li> <li>• Units - Milliseconds since Jan. 1, 1970</li> </ul>
<i>pLeapSeconds</i>	<ul style="list-style-type: none"> <li>• Leap second information. If leapSeconds is not available, timestampUtc is calculated based on a hard-coded value for leap seconds.</li> <li>• Units - Seconds</li> </ul>
<i>-pGpsTime</i>	<ul style="list-style-type: none"> <li>• See <a href="#">gpsTime</a> for more information</li> </ul>
<i>pTimeUnc</i>	<ul style="list-style-type: none"> <li>• Time uncertainty.</li> <li>• Units - Milliseconds</li> </ul>
<i>pTimeSrc</i>	<ul style="list-style-type: none"> <li>• Values <ul style="list-style-type: none"> <li>– 0 - Invalid time.</li> <li>– 1 - Time is set by the 1X system.</li> <li>– 2 - Time is set by WCDMA/GSM time tagging.</li> <li>– 3 - Time is set by an external injection.</li> </ul> </li> </ul>
	<ul style="list-style-type: none"> <li>– 4 - Time is set after decoding over-the-air GPS navigation data from one GPS satellite.</li> <li>– 5 - Time is set after decoding over-the-air GPS navigation data from multiple satellites.</li> </ul>

<i>-pSensorData-Usage</i>	<ul style="list-style-type: none"> <li>• See <a href="#">sensorDataUsage</a> for more information</li> </ul>
<i>pFixId</i>	<ul style="list-style-type: none"> <li>• Fix count for the session. Starts with 0 and increments by one for each successive position report for a particular session.</li> </ul>
<i>-pSvUsedforFix</i>	<ul style="list-style-type: none"> <li>• See <a href="#">svUsedforFix</a> for more information</li> </ul>
<i>pAltitude-Assumed</i>	<ul style="list-style-type: none"> <li>• Indicates whether altitude is assumed or calculated.</li> </ul>

- Value
  - 0x00 - Altitude is calculated
  - 0x01 - Altitude is assumed

## 8.344.2 Field Documentation

- 8.344.2.1 **BYTE\*** QmiCbkLocPositionReportInd::pAltitudeAssumed
- 8.344.2.2 **ULONG\*** QmiCbkLocPositionReportInd::pAltitudeWrtEllipsoid
- 8.344.2.3 **ULONG\*** QmiCbkLocPositionReportInd::pAltitudeWrtMeanSeaLevel
- 8.344.2.4 **ULONG\*** QmiCbkLocPositionReportInd::pFixId
- 8.344.2.5 **gpsTime\*** QmiCbkLocPositionReportInd::pGpsTime
- 8.344.2.6 **ULONG\*** QmiCbkLocPositionReportInd::pHeading
- 8.344.2.7 **ULONG\*** QmiCbkLocPositionReportInd::pHeadingUnc
- 8.344.2.8 **BYTE\*** QmiCbkLocPositionReportInd::pHorConfidence
- 8.344.2.9 **ULONG\*** QmiCbkLocPositionReportInd::pHorReliability
- 8.344.2.10 **ULONG\*** QmiCbkLocPositionReportInd::pHorUncCircular
- 8.344.2.11 **ULONG\*** QmiCbkLocPositionReportInd::pHorUncEllipseOrientAzimuth
- 8.344.2.12 **ULONG\*** QmiCbkLocPositionReportInd::pHorUncEllipseSemiMajor
- 8.344.2.13 **ULONG\*** QmiCbkLocPositionReportInd::pHorUncEllipseSemiMinor
- 8.344.2.14 **ULONGLONG\*** QmiCbkLocPositionReportInd::pLatitude
- 8.344.2.15 **BYTE\*** QmiCbkLocPositionReportInd::pLeapSeconds
- 8.344.2.16 **ULONGLONG\*** QmiCbkLocPositionReportInd::pLongitude
- 8.344.2.17 **ULONG\*** QmiCbkLocPositionReportInd::pMagneticDeviation

- 8.344.2.18 **precisionDilution\*** `QmiCbkLocPositionReportInd::pPrecisionDilution`
- 8.344.2.19 **sensorDataUsage\*** `QmiCbkLocPositionReportInd::pSensorDataUsage`
- 8.344.2.20 **ULONG\*** `QmiCbkLocPositionReportInd::pSpeedHorizontal`
- 8.344.2.21 **ULONG\*** `QmiCbkLocPositionReportInd::pSpeedUnc`
- 8.344.2.22 **ULONG\*** `QmiCbkLocPositionReportInd::pSpeedVertical`
- 8.344.2.23 **svUsedforFix\*** `QmiCbkLocPositionReportInd::pSvUsedforFix`
- 8.344.2.24 **ULONG\*** `QmiCbkLocPositionReportInd::pTechnologyMask`
- 8.344.2.25 **ULONG\*** `QmiCbkLocPositionReportInd::pTimeSrc`
- 8.344.2.26 **ULONGLONG\*** `QmiCbkLocPositionReportInd::pTimestampUtc`
- 8.344.2.27 **ULONG\*** `QmiCbkLocPositionReportInd::pTimeUnc`
- 8.344.2.28 **BYTE\*** `QmiCbkLocPositionReportInd::pVertConfidence`
- 8.344.2.29 **ULONG\*** `QmiCbkLocPositionReportInd::pVertReliability`
- 8.344.2.30 **ULONG\*** `QmiCbkLocPositionReportInd::pVertUnc`
- 8.344.2.31 **BYTE** `QmiCbkLocPositionReportInd::sessionId`
- 8.344.2.32 **ULONG** `QmiCbkLocPositionReportInd::sessionStatus`

## 8.345 QmiCbkLocSensorStreamingInd Struct Reference

### Data Fields

- [accelAcceptReady](#) \* [pAccelAcceptReady](#)
- [gyroAcceptReady](#) \* [pGyroAcceptReady](#)
- [accelTempAcceptReady](#) \* [pAccelTempAcceptReady](#)
- [gyroTempAcceptReady](#) \* [pGyroTempAcceptReady](#)

### 8.345.1 Detailed Description

This structure contains LOC Event Sensor Streaming Ready Status

#### Parameters

<i>-pAccelAcceptReady</i>	<ul style="list-style-type: none"> <li>• See <a href="#">accelAcceptReady</a> for more information</li> </ul>
<i>-pGyroAcceptReady</i>	<ul style="list-style-type: none"> <li>• See <a href="#">gyroAcceptReady</a> for more information</li> </ul>

<i>-pAccelTempAcceptReady</i>	<ul style="list-style-type: none"> <li>• See <a href="#">accelTempAcceptReady</a> for more information</li> </ul>
<i>-pGyroTempAcceptReady</i>	<ul style="list-style-type: none"> <li>• See <a href="#">gyroTempAcceptReady</a> for more information</li> </ul>

## 8.345.2 Field Documentation

8.345.2.1 `accelAcceptReady*` `QmiCbkLocSensorStreamingInd::pAccelAcceptReady`

8.345.2.2 `accelTempAcceptReady*` `QmiCbkLocSensorStreamingInd::pAccelTempAcceptReady`

8.345.2.3 `gyroAcceptReady*` `QmiCbkLocSensorStreamingInd::pGyroAcceptReady`

8.345.2.4 `gyroTempAcceptReady*` `QmiCbkLocSensorStreamingInd::pGyroTempAcceptReady`

## 8.346 QmiCbkNasLTECphyCalInfo Struct Reference

### Data Fields

- [PhyCaAggScellIndType](#) `sPhyCaAggScellIndType`
- [PhyCaAggScellDIBw](#) `sPhyCaAggScellDIBw`
- [PhyCaAggScellInfo](#) `sPhyCaAggScellInfo`
- [PhyCaAggPcellInfo](#) `sPhyCaAggPcellInfo`
- [PhyCaAggScellIndex](#) `sPhyCaAggScellIndex`

### 8.346.1 Detailed Description

Structure for storing the LTEC PHY CA indication parameters.

#### Parameters

<i>pPhyCaAggScellIndType</i>	<ul style="list-style-type: none"> <li>• See <a href="#">PhyCaAggScellIndType</a> for more information.</li> </ul>
<i>sPhyCaAggScellDIBw</i>	<ul style="list-style-type: none"> <li>• See <a href="#">PhyCaAggScellDIBw</a> for more information.</li> </ul>
<i>sPhyCaAggScellInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">PhyCaAggScellInfo</a> for more information.</li> </ul>
<i>sPhyCaAggPcellInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">PhyCaAggPcellInfo</a> for more information.</li> </ul>
<i>sPhyCaAggScellIndex</i>	<ul style="list-style-type: none"> <li>• See <a href="#">PhyCaAggScellIndex</a> for more information.</li> </ul>

### 8.346.2 Field Documentation

8.346.2.1 **PhyCaAggPcellInfo** QmiCbkNasLTECphyCalInfo::sPhyCaAggPcellInfo

8.346.2.2 **PhyCaAggScellIDIBw** QmiCbkNasLTECphyCalInfo::sPhyCaAggScellIDIBw

8.346.2.3 **PhyCaAggScellIndex** QmiCbkNasLTECphyCalInfo::sPhyCaAggScellIndex

8.346.2.4 **PhyCaAggScellIndType** QmiCbkNasLTECphyCalInfo::sPhyCaAggScellIndType

8.346.2.5 **PhyCaAggScellInfo** QmiCbkNasLTECphyCalInfo::sPhyCaAggScellInfo

## 8.347 QmiCbkSwiOmaDmEventStatusReportInd Struct Reference

### Data Fields

- struct [sessionInfoTlv](#) SITlv

### 8.347.1 Field Documentation

8.347.1.1 struct sessionInfoTlv QmiCbkSwiOmaDmEventStatusReportInd::SITlv

## 8.348 QmiCbkSwiOmaDmEventStatusReportIndExt Struct Reference

### Data Fields

- struct [sessionInfoTlvExt](#) SITlv

### 8.348.1 Field Documentation

8.348.1.1 struct sessionInfoTlvExt QmiCbkSwiOmaDmEventStatusReportIndExt::SITlv

## 8.349 QmiCbkWdsStatisticsIndState Struct Reference

### Data Fields

- [DataULongTlv](#) TxOkConutTlv
- [DataULongTlv](#) RxOkConutTlv
- [DataULongLongTlv](#) TxOkByteCountTlv
- [DataULongLongTlv](#) RxOkByteCountTlv
- [DataULongTlv](#) TxDropConutTlv
- [DataULongTlv](#) RxDropConutTlv

### 8.349.1 Detailed Description

WDS Pkt RM Transfer Statistics data structure for individual session

Parameters

<i>TxOkConutTlv</i>	<ul style="list-style-type: none"> <li>• Tx Ok Packet Tlv Value.</li> </ul>
---------------------	---

<i>RxOkConutTlv</i>	<ul style="list-style-type: none"> <li>Rx Ok Packet Tlv Value.</li> </ul>
<i>TxOkByteCountTlv</i>	<ul style="list-style-type: none"> <li>Tx Ok Byte Count Packet Tlv Value.</li> </ul>
<i>RxOkByteCountTlv</i>	<ul style="list-style-type: none"> <li>Rx Ok Byte Count Packet Tlv Value.</li> </ul>
<i>TxDropConutTlv</i>	<ul style="list-style-type: none"> <li>Tx Drop Count Packet Tlv Value.</li> </ul>
<i>RxDropConutTlv</i>	<ul style="list-style-type: none"> <li>Rx Drop Count Packet Tlv Value.</li> </ul>

### 8.349.2 Field Documentation

8.349.2.1 **DataUlongTlv** QmiCbkWdsStatisticsIndState::RxDropConutTlv

8.349.2.2 **DataUlongLongTlv** QmiCbkWdsStatisticsIndState::RxOkByteCountTlv

8.349.2.3 **DataUlongTlv** QmiCbkWdsStatisticsIndState::RxOkConutTlv

8.349.2.4 **DataUlongTlv** QmiCbkWdsStatisticsIndState::TxDropConutTlv

8.349.2.5 **DataUlongLongTlv** QmiCbkWdsStatisticsIndState::TxOkByteCountTlv

8.349.2.6 **DataUlongTlv** QmiCbkWdsStatisticsIndState::TxOkConutTlv

## 8.350 qmifwinfo\_s Struct Reference

### Data Fields

- union {
  - struct [fwinfo\\_s g](#)
  - struct [slqsfwinfo\\_s s](#)

### 8.350.1 Detailed Description

Top level structure for storing information about firmware images. union of structures depending on device type, MC77xx or MC83xx

#### Parameters

<i>g</i>	- structure for MC83xx devices
<i>s</i>	- structure for devices with SPKG CWE file support

- List of various Firmware Images Supported

Technology Initials Carrier

Region

Network Technology

D3600	S	eGOBI_IMG_CAR_SPRINT	eGOBI_IMG_REG_NA	eGOBI_IMG_TECH_CDMA	
D3600	V	eGOBI_IMG_CAR_VERIZON	eGOBI_IMG_REG_NA	eGOBI_IMG_TECH_CDMA	
D3600	C	eGOBI_IMG_CAR_CHINA_TELECOM	eGOBI_IMG_REG_ASIA	eGOBI_IMG_TECH_CDMA	
D3600	G	eGOBI_IMG_CAR_GENERIC_CDMA	eGOBI_IMG_REG_GLOBAL	eGOBI_IMG_TECH_CDMA	(item for Generic)
D3600	H	eGOBI_IMG_CAR_GENERIC_CDMA	eGOBI_IMG_REG_GLOBAL	eGOBI_IMG_TECH_CDMA	(item for Generic)
D3200	V	eGOBI_IMG_CAR_VODAFONE	eGOBI_IMG_REG_NA	eGOBI_IMG_TECH_UMTS	
D3200	A	eGOBI_IMG_CAR_ATT	eGOBI_IMG_REG_NA	eGOBI_IMG_TECH_UMTS	
D3200	L	eGOBI_IMG_CAR_TMOBILE	eGOBI_IMG_REG_EU	eGOBI_IMG_TECH_UMTS	
D3200	G	eGOBI_IMG_CAR_GENERIC	eGOBI_IMG_REG_GLOBAL	eGOBI_IMG_TECH_UMTS	
D3200	H	eGOBI_IMG_CAR_TELEFONICA	eGOBI_IMG_REG_EU	eGOBI_IMG_TECH_UMTS	
D3200	I	eGOBI_IMG_CAR_TELCOM_ITALIA	eGOBI_IMG_REG_NA	eGOBI_IMG_TECH_UMTS	
D3200	O	eGOBI_IMG_CAR_ORANGE	eGOBI_IMG_REG_NA	eGOBI_IMG_TECH_UMTS	
D3200	U	eGOBI_IMG_CAR_GENERIC	eGOBI_IMG_REG_GLOBAL	eGOBI_IMG_TECH_UMTS	
D3200	R	eGOBI_IMG_CAR_ROGERS	eGOBI_IMG_REG_NA	eGOBI_IMG_TECH_UMTS	
D3600	A	eGOBI_IMG_CAR_AERIS	eGOBI_IMG_REG_NA	eGOBI_IMG_TECH_CDMA	

#### See Also

[fwinfo\\_s](#)  
[slqsfwinfo\\_s](#)

### 8.350.2 Field Documentation

8.350.2.1 `union { ... } qmifwinfo_s::dev`

8.350.2.2 `struct fwinfo_s qmifwinfo_s::g`

8.350.2.3 `struct slqsfwinfo_s qmifwinfo_s::s`

## 8.351 QmiNas3GppNetworkInfo Struct Reference

### Data Fields

- [WORD pMCC](#)
- [WORD pMNC](#)
- [ULONG plnUse](#)
- [ULONG pRoaming](#)
- [ULONG pForbidden](#)
- [ULONG pPreferred](#)
- [CHAR pDescription](#) [255]

#### 8.351.1 Detailed Description

This structure contains the PerformNetworkScan response parameters. This structure will hold the array of the network scan information.

##### Parameters

<i>pMCC</i>	<ul style="list-style-type: none"> <li>• Mobile Country Code</li> </ul>
-------------	---



<i>pMNC</i>	<ul style="list-style-type: none"> <li>• Mobile Networ Code</li> </ul>
<i>pInUse</i>	<ul style="list-style-type: none"> <li>• Is the Network the current serving Network <ul style="list-style-type: none"> <li>– 0 – Unknown</li> <li>– 1 – Current serving network</li> <li>– 2 – Not current serving network, available</li> </ul> </li> </ul>
<i>pRoaming</i>	<ul style="list-style-type: none"> <li>• Home/Roam Status of the Network <ul style="list-style-type: none"> <li>– 0 – Unknown</li> <li>– 1 – Home</li> <li>– 2 – Roam</li> </ul> </li> </ul>
<i>pForbidden</i>	<ul style="list-style-type: none"> <li>• Is the Network in the forbidden network list <ul style="list-style-type: none"> <li>– 0 – Unknown</li> <li>– 1 – Forbidden</li> <li>– 2 – Not Forbidden</li> </ul> </li> </ul>
<i>pPreferred</i>	<ul style="list-style-type: none"> <li>• Is the Network in the Preferred network list <ul style="list-style-type: none"> <li>– 0 – Unknown</li> <li>– 1 – Preferred</li> <li>– 2 – Not Preferred</li> </ul> </li> </ul>
<i>pDescription</i>	<ul style="list-style-type: none"> <li>• Network Name/Description</li> </ul>

## 8.351.2 Field Documentation

8.351.2.1 **CHAR** QmiNas3GppNetworkInfo::pDescription[255]

8.351.2.2 **ULONG** QmiNas3GppNetworkInfo::pForbidden

8.351.2.3 **ULONG** QmiNas3GppNetworkInfo::pInUse

8.351.2.4 **WORD** QmiNas3GppNetworkInfo::pMCC

8.351.2.5 **WORD** QmiNas3GppNetworkInfo::pMNC

8.351.2.6 **ULONG** QmiNas3GppNetworkInfo::pPreferred

8.351.2.7 **ULONG** QmiNas3GppNetworkInfo::pRoaming

## 8.352 QmiNasGetRFBandInfoResp Struct Reference

### Data Fields

- struct qmTlvResult [results](#)
- **BYTE** \* [pInstancesSize](#)
- struct [RFBandInfoElements](#) \* [pRFBandInfoElements](#)

### 8.352.1 Field Documentation

8.352.1.1 **BYTE**\* QmiNasGetRFBandInfoResp::pInstancesSize

8.352.1.2 struct [RFBandInfoElements](#)\* QmiNasGetRFBandInfoResp::pRFBandInfoElements

8.352.1.3 struct qmTlvResult QmiNasGetRFBandInfoResp::results

## 8.353 QmiNasPerformNetworkScanResp Struct Reference

### Data Fields

- struct qmTlvResult [results](#)
- **BYTE** \* [pInstanceSize](#)
- struct [QmiNas3GppNetworkInfo](#) \* [pInstances](#)

### 8.353.1 Field Documentation

8.353.1.1 struct [QmiNas3GppNetworkInfo](#)\* QmiNasPerformNetworkScanResp::pInstances

8.353.1.2 **BYTE**\* QmiNasPerformNetworkScanResp::pInstanceSize

8.353.1.3 struct qmTlvResult QmiNasPerformNetworkScanResp::results

## 8.354 QmiWdsIpAddressInfo Struct Reference

### Data Fields

- **ULONG** \* [pIPAddressV4](#)
- **USHORT** \* [pIPAddressV6](#)
- **BYTE** \* [pIPv6prefixlen](#)

## 8.354.1 Detailed Description

## Parameters

<i>pIPAddressV4</i> [- <i>OUT</i> ]	<ul style="list-style-type: none"> <li>Current IPv4 address</li> <li>default value of 0 if not reported by the device.</li> </ul>
<i>pIPAddressV6</i> [- <i>OUT</i> ]	<ul style="list-style-type: none"> <li>Current IPv6 address</li> </ul> <p>Space for storing the 8 element array of type USHORT for the IPv6 address is allocated by the application.</p> <p>The IP Address is stored in the user supplied buffer as follows:</p> <p>User buffer: [&lt;U0&gt;..&lt;&lt;U7&gt;]</p> <p>IPv6 address from the network: 1234:2A01:.....:5678</p> <p>User buffer contents: U0 corresponds to 1234 U1 corresponds to 2A01 ----- ----- U7 corresponds to 5678</p>
<i>pIPv6prefixlen</i> [- <i>OUT</i> ]	<ul style="list-style-type: none"> <li>IPv6 prefix length in number of bits</li> </ul>

## 8.354.2 Field Documentation

8.354.2.1 **ULONG\*** QmiWdsIpAddressInfo::pIPAddressV48.354.2.2 **USHORT\*** QmiWdsIpAddressInfo::pIPAddressV68.354.2.3 **BYTE\*** QmiWdsIpAddressInfo::pIPv6prefixlen

## 8.355 qmiWdsRunTimeSettings Struct Reference

## Data Fields

- CHAR \*** pProfileName
- ULONG \*** pPDPTType
- CHAR \*** pAPNName
- ULONG \*** pPrimaryDNSV4
- ULONG \*** pSecondaryDNSV4
- struct UMTSQoS \*** pUMTSGrantedQoS
- struct GPRSQoS \*** pGPRSGrantedQoS
- CHAR \*** pUsername
- ULONG \*** pAuthentication
- ULONG \*** pIPAddressV4

- struct [ProfileIdentifier](#) \* [pProfileID](#)
- [ULONG](#) \* [pGWAddressV4](#)
- [ULONG](#) \* [pSubnetMaskV4](#)
- [BYTE](#) \* [pPCSCFAddrPCO](#)
- struct [PCSCFIPv4ServerAddressList](#) \* [pServerAddrList](#)
- struct [PCSCFFQDNAddressList](#) \* [pPCSCFFQDNAddrList](#)
- [USHORT](#) \* [pPrimaryDNSV6](#)
- [USHORT](#) \* [pSecondaryDNSV6](#)
- [ULONG](#) \* [pMtu](#)
- struct [DomainNameList](#) \* [pDomainList](#)
- [BYTE](#) \* [pIPFamilyPreference](#)
- [BYTE](#) \* [pIMCNflag](#)
- [WORD](#) \* [pTechnology](#)
- struct [IPv6AddressInfo](#) \* [pIPv6AddrInfo](#)
- struct [IPv6GWAddressInfo](#) \* [pIPv6GWAddrInfo](#)

### 8.355.1 Detailed Description

This structure contains the [WdsRunTimeSettings](#) Information

- Parameter values default to their data type's maximum unsigned value unless explicitly stated otherwise.

#### Parameters

<i>pProfileName</i>	<ul style="list-style-type: none"> <li>• Profile name</li> </ul> <p>One or more bytes describing the profile. Description may be a user-defined name for the profile.</p> <p>QMI_ERR_ARG_TOO_LONG is returned if profile_name is too long.</p>
<i>pPDPTType</i>	<ul style="list-style-type: none"> <li>• PDP type</li> </ul> <p>– 0x00 – PDP-IP (IPv4)</p>
<i>pAPNName</i>	<ul style="list-style-type: none"> <li>• Access point name</li> </ul> <p>String parameter that is a logical name used to select the GGSN and external packet data network.</p> <p>If value is NULL or omitted, then the subscription default value is requested.</p> <p>QMI_ERR_ARG_TOO_LONG is returned if the APN name is too long.</p>

<i>pPrimaryDNSV4</i>	<ul style="list-style-type: none"> <li>• Primary DNS IPv4 Address</li> </ul>
<i>pSecondaryDNSV4</i>	<ul style="list-style-type: none"> <li>• Secondary DNS IPv4 Address</li> </ul>
<i>pUMTSGrantedQoS</i>	<ul style="list-style-type: none"> <li>• UMTS Granted QoS</li> </ul>
<i>pGPRSGrantedQoS</i>	<ul style="list-style-type: none"> <li>• GPRS Granted QoS</li> </ul>
<i>pUsername</i>	<ul style="list-style-type: none"> <li>• User name used during data network authentication</li> </ul>
<i>pAuthentication</i>	<ul style="list-style-type: none"> <li>• Authentication preference <ul style="list-style-type: none"> <li>– Bit 0 – PAP preference <ul style="list-style-type: none"> <li>* 0 – PAP is never performed</li> <li>* 1 – PAP may be performed</li> </ul> </li> <li>– Bit 1 – CHAP preference <ul style="list-style-type: none"> <li>* 0 – CHAP is never performed</li> <li>* 1 – CHAP may be performed</li> </ul> </li> </ul> </li> </ul>
<i>pIPAddressV4</i>	<ul style="list-style-type: none"> <li>• IPV4 Address assigned to the TE</li> </ul>
<i>pProfileID</i>	<ul style="list-style-type: none"> <li>• Profile Identifier</li> </ul>
<i>pGWAddressV4</i>	<ul style="list-style-type: none"> <li>• IPV4 Gateway Address</li> </ul>
<i>pSubnetMaskV4</i>	<ul style="list-style-type: none"> <li>• IPV4 Subnet Mask</li> </ul>
<i>pPCSCFAddrPCO</i>	<ul style="list-style-type: none"> <li>• PCSCF address using PCO values <ul style="list-style-type: none"> <li>– 1 – (TRUE) implies request PCSCF address using PCO</li> <li>– 0 – (FALSE) implies do not request. This is the default value.</li> </ul> </li> </ul>
<i>pServerAddrList</i>	<ul style="list-style-type: none"> <li>• P-CSCF IPv4 Server Address List</li> </ul>
<i>pPCSCFFQDNAddrList</i>	<ul style="list-style-type: none"> <li>• P-CSCF FQDN Address List</li> </ul>
<i>pPrimaryDNSV6</i>	<ul style="list-style-type: none"> <li>• Primary DNS IPv6 Address</li> </ul>

<i>pSecondaryDN-SV6</i>	<ul style="list-style-type: none"> <li>• Secondary DNS IPv6 Address</li> </ul>
<i>mtu</i>	<ul style="list-style-type: none"> <li>• MTU</li> </ul>
<i>pDomainList</i>	<ul style="list-style-type: none"> <li>• Domain-Name List</li> </ul>
<i>pIPFamily-Preference</i>	<ul style="list-style-type: none"> <li>• IP family <ul style="list-style-type: none"> <li>– 0x04 – IPV4 ADDR</li> <li>– 0x06 – IPV6 ADDR</li> </ul> </li> </ul>
<i>pIMCNflag</i>	<ul style="list-style-type: none"> <li>• IM CN Flag <ul style="list-style-type: none"> <li>– 0x00 – FALSE</li> <li>– 0x01 – TRUE</li> </ul> </li> </ul>
<i>pTechnology</i>	<ul style="list-style-type: none"> <li>• Technology <ul style="list-style-type: none"> <li>– CDMA – 0x8001</li> <li>– UMTS – 0x8004</li> </ul> </li> </ul>
<i>pIPV6Address-Info</i>	<ul style="list-style-type: none"> <li>• IPV6 Address Information</li> </ul>
<i>pIPV6GW-AddressInfo</i>	<ul style="list-style-type: none"> <li>• IPV6 Gateway Address Information</li> </ul>

## 8.355.2 Field Documentation

8.355.2.1 **CHAR\*** qmiWdsRunTimeSettings::pAPNName

8.355.2.2 **ULONG\*** qmiWdsRunTimeSettings::pAuthentication

8.355.2.3 **struct DomainNameList\*** qmiWdsRunTimeSettings::pDomainList

8.355.2.4 **struct GPRSQoS\*** qmiWdsRunTimeSettings::pGPRSGrantedQoS

8.355.2.5 **ULONG\*** qmiWdsRunTimeSettings::pGWAddressV4

8.355.2.6 **BYTE\*** qmiWdsRunTimeSettings::pIMCNflag

8.355.2.7 **ULONG\*** qmiWdsRunTimeSettings::pIPAddressV4

8.355.2.8 **BYTE\*** qmiWdsRunTimeSettings::pIPFamilyPreference

- 8.355.2.9 struct IPV6AddressInfo\* qmiWdsRunTimeSettings::pIPv6AddrInfo
- 8.355.2.10 struct IPV6GWAddressInfo\* qmiWdsRunTimeSettings::pIPv6GWAddrInfo
- 8.355.2.11 ULONG\* qmiWdsRunTimeSettings::pMtu
- 8.355.2.12 BYTE\* qmiWdsRunTimeSettings::pPCSCFAddrPCO
- 8.355.2.13 struct PCSCFFQDNAddressList\* qmiWdsRunTimeSettings::pPCSCFFQDNAddrList
- 8.355.2.14 ULONG\* qmiWdsRunTimeSettings::pPDPTType
- 8.355.2.15 ULONG\* qmiWdsRunTimeSettings::pPrimaryDNSV4
- 8.355.2.16 USHORT\* qmiWdsRunTimeSettings::pPrimaryDNSV6
- 8.355.2.17 struct ProfileIdentifier\* qmiWdsRunTimeSettings::pProfileID
- 8.355.2.18 CHAR\* qmiWdsRunTimeSettings::pProfileName
- 8.355.2.19 ULONG\* qmiWdsRunTimeSettings::pSecondaryDNSV4
- 8.355.2.20 USHORT\* qmiWdsRunTimeSettings::pSecondaryDNSV6
- 8.355.2.21 struct PCSCFIPv4ServerAddressList\* qmiWdsRunTimeSettings::pServerAddrList
- 8.355.2.22 ULONG\* qmiWdsRunTimeSettings::pSubnetMaskV4
- 8.355.2.23 WORD\* qmiWdsRunTimeSettings::pTechnology
- 8.355.2.24 struct UMTSQoS\* qmiWdsRunTimeSettings::pUMTSGrantedQoS
- 8.355.2.25 CHAR\* qmiWdsRunTimeSettings::pUsername

## 8.356 QosClassID Struct Reference

### Data Fields

- [BYTE QCI](#)
- [ULONG gDIBitRate](#)
- [ULONG maxDIBitRate](#)
- [ULONG gUIBitRate](#)
- [ULONG maxUIBitRate](#)

### 8.356.1 Detailed Description

structure contains 3GPP LTE QoS parameters

- Parameter values default to their data type's maximum unsigned value unless explicitly stated otherwise.

## Parameters

<i>QCI</i>	<ul style="list-style-type: none"> <li>QOS specified using the QOS Class Identifier (QOS) values QCI value 0 - Requests the network to assign the appropriate QCI value QCI values 1-4 - Associated with guaranteed bit rates QCI values 5-9 - Associated with non-guaranteed bit rates</li> </ul>
<i>gDlBitRate</i>	<ul style="list-style-type: none"> <li>Guaranteed DL bit rate</li> </ul>
<i>maxDlBitRate</i>	<ul style="list-style-type: none"> <li>maxDlBitRate</li> </ul>
<i>gUlBitRate</i>	<ul style="list-style-type: none"> <li>Guaranteed UL bit rate</li> </ul>
<i>maxUlBitRate</i>	<ul style="list-style-type: none"> <li>Maximum UL bit rate</li> </ul>

## 8.356.2 Field Documentation

8.356.2.1 **ULONG** QosClassID::gDlBitRate8.356.2.2 **ULONG** QosClassID::gUlBitRate8.356.2.3 **ULONG** QosClassID::maxDlBitRate8.356.2.4 **ULONG** QosClassID::maxUlBitRate8.356.2.5 **BYTE** QosClassID::QCI

## 8.357 QosEventInfo Struct Reference

## Data Fields

- ULONG** \* [pDataBearer](#)
- ULONG** \* [pPacketsCountTX](#)
- ULONG** \* [pPacketsCountRX](#)
- ULONGLONG** \* [pTotalBytesTX](#)
- ULONGLONG** \* [pTotalBytesRX](#)

## 8.357.1 Detailed Description

Contains the WDS event information and information about the interface

## Parameters

<i>pQmiInterface-Info</i>	<ul style="list-style-type: none"> <li>See <a href="#">qaQmiInterfaceInfo</a> for more information</li> </ul>
---------------------------	---



<i>pDataBearer</i>	<ul style="list-style-type: none"> <li>• Data bearer technology (NULL if not present) <ul style="list-style-type: none"> <li>– 0x00 - Indicates that this field is ignored</li> <li>– 0x01 - CDMA 1X</li> <li>– 0x02 - EV-DO Rev 0</li> <li>– 0x03 - GPRS</li> <li>– 0x04 - WCDMA</li> <li>– 0x05 - EV-DO Rev A</li> <li>– 0x06 - EDGE</li> <li>– 0x07 - HSDPA and WCDMA</li> <li>– 0x08 - WCDMA and HSUPA</li> <li>– 0x09 - HSDPA and HSUPA</li> <li>– 0x0A - LTE</li> <li>– 0x0B - EV-DO Rev A EHRPD</li> <li>– 0x0C - HSDPA+ and WCDMA</li> <li>– 0x0D - HSDPA+ and HSUPA</li> <li>– 0x0E - DC_HSDPA+ and WCDMA</li> <li>– 0x0F - DC_HSDPA+ and HSUPA</li> <li>– 0x8000 - NULL Bearer</li> <li>– 0xFF - Unknown Technology</li> </ul> </li> </ul>
<i>pDormancy-Status</i>	<ul style="list-style-type: none"> <li>• Dormancy status (NULL if not present) <ul style="list-style-type: none"> <li>– 1 - traffic channel dormant</li> <li>– 2 - traffic channel active</li> </ul> </li> </ul>
<i>pPacketsCount-TX</i>	<ul style="list-style-type: none"> <li>• Packets transmitted without error (NULL if not present)</li> </ul>
<i>pPacketsCount-RX</i>	<ul style="list-style-type: none"> <li>• Packets received without error (NULL if not present)</li> </ul>
<i>pTotalBytesTX</i>	<ul style="list-style-type: none"> <li>• Bytes transmitted without error (NULL if not present)</li> </ul>
<i>pTotalBytesRX</i>	<ul style="list-style-type: none"> <li>• Bytes received without error (NULL if not present)</li> </ul>

## 8.357.2 Field Documentation

### 8.357.2.1 ULONG\* QosEventInfo::pDataBearer

### 8.357.2.2 ULONG\* QosEventInfo::pPacketsCountRX

8.357.2.3 **ULONG\*** QosEventInfo::pPacketsCountTX

8.357.2.4 **ULONGLONG\*** QosEventInfo::pTotalBytesRX

8.357.2.5 **ULONGLONG\*** QosEventInfo::pTotalBytesTX

## 8.358 QosFlowInfo Struct Reference

### Data Fields

- [QosFlowInfoState](#) \* [pQFlowState](#)
- [swiQosFlow](#) \* [pTxQFlowGranted](#)
- [swiQosFlow](#) \* [pRxQFlowGranted](#)
- [swiQosFilter](#) \* [pTxQFilter](#) [[MAX\\_QOS\\_FILTER\\_TLV](#)]
- [swiQosFilter](#) \* [pRxQFilter](#) [[MAX\\_QOS\\_FILTER\\_TLV](#)]
- **BYTE** \* [pBearerID](#)

### 8.358.1 Detailed Description

This structure contains QoS flow info

#### Parameters

<i>pQFlowState</i>	<ul style="list-style-type: none"> <li>• QoS flow state information, please check <a href="#">QosFlowInfoState</a> for more information</li> </ul>
<i>pTxQFlow-Granted</i>	<ul style="list-style-type: none"> <li>• pointer to the Tx Qos flow granted, please check <a href="#">swiQosFlow</a> for more information</li> </ul>
<i>pRxQFlow-Granted</i>	<ul style="list-style-type: none"> <li>• pointer to the Rx Qos flow granted</li> </ul>
<i>pTxQFilter</i>	<ul style="list-style-type: none"> <li>• pointer to the Tx Qos filter</li> </ul>
<i>pRxQFilter</i>	<ul style="list-style-type: none"> <li>• pointer to the Rx Qos flow</li> </ul>
<i>pBearerID</i>	<ul style="list-style-type: none"> <li>• pointer to the bearer ID</li> <li>• Bearer ID or Radio Link Protocol (RLP) ID of the activated flow.</li> <li>• Valid Values - 0 to 16</li> <li>• 0xFF - Invalid value.</li> </ul>

### 8.358.2 Field Documentation

8.358.2.1 **BYTE\*** QosFlowInfo::pBearerID

8.358.2.2 **QosFlowInfoState\*** QosFlowInfo::pQFlowState

8.358.2.3 `swiQosFilter*` `QosFlowInfo::pRxQFilter[MAX_QOS_FILTER_TLV]`

8.358.2.4 `swiQosFlow*` `QosFlowInfo::pRxQFlowGranted`

8.358.2.5 `swiQosFilter*` `QosFlowInfo::pTxQFilter[MAX_QOS_FILTER_TLV]`

8.358.2.6 `swiQosFlow*` `QosFlowInfo::pTxQFlowGranted`

## 8.359 QosFlowInfoState Struct Reference

### Data Fields

- [ULONG id](#)
- [BYTE isNewFlow](#)
- [BYTE state](#)

### 8.359.1 Detailed Description

This structure contains QoS flow state

#### Parameters

<i>id</i>	QoS identifier
<i>isNewFlow</i>	<ul style="list-style-type: none"> <li>• 1 – Newly added flow</li> <li>• 0 – Existing flow</li> </ul>
<i>state</i>	<p>This indicates that the flow that was added/modified/deleted:</p> <ul style="list-style-type: none"> <li>• 0x01 – Flow activated</li> <li>• 0x02 – Flow modified</li> <li>• 0x03 – Flow deleted</li> <li>• 0x04 – Flow suspended</li> <li>• 0x05 – Flow enabled</li> <li>• 0x06 – Flow disabled</li> </ul>

### 8.359.2 Field Documentation

8.359.2.1 **ULONG** `QosFlowInfoState::id`

8.359.2.2 **BYTE** `QosFlowInfoState::isNewFlow`

8.359.2.3 **BYTE** `QosFlowInfoState::state`

## 8.360 QosMap Struct Reference

### Data Fields

- [BYTE dscp](#)
- [ULONG qos\\_id](#)

- [BYTE state](#)

### 8.360.1 Detailed Description

This structure contains the SLQSQoSDumpMap Information

Parameters

<i>dscp</i>	<ul style="list-style-type: none"> <li>• Differential Service Code Point(DSCP) value</li> </ul>
<i>qos_id</i>	<ul style="list-style-type: none"> <li>• QoS identifier</li> </ul>
<i>state</i>	<ul style="list-style-type: none"> <li>• QoS Flow state</li> </ul>

### 8.360.2 Field Documentation

8.360.2.1 **BYTE** QosMap::dscp

8.360.2.2 **ULONG** QosMap::qos\_id

8.360.2.3 **BYTE** QosMap::state

## 8.361 readResult Struct Reference

Data Fields

- [WORD contentLen](#)
- [BYTE content](#) [255+1]

### 8.361.1 Detailed Description

This structure contains the information for write operation.

Parameters

<i>contentLen</i>	<ul style="list-style-type: none"> <li>• Number of sets of content.</li> </ul>
<i>content</i> [ <i>MAX_DESCRIPTION_LENGTH</i> ]	<ul style="list-style-type: none"> <li>• Read content.</li> <li>• The content is the sequence of bytes as read from the card.</li> </ul>

### 8.361.2 Field Documentation

8.361.2.1 **BYTE** readResult::content[255+1]

## 8.361.2.2 WORD readResult::contentLen

## 8.362 readTransparentInfo Struct Reference

### Data Fields

- [WORD offset](#)
- [WORD length](#)

### 8.362.1 Detailed Description

This structure contains the information for read operation.

#### Parameters

<i>offset</i>	<ul style="list-style-type: none"><li>• Offset for the read operation.</li></ul>
<i>length</i>	<ul style="list-style-type: none"><li>• Length of the content to be read.</li><li>• The value 0 is used to read the complete file.</li></ul>

### 8.362.2 Field Documentation

## 8.362.2.1 WORD readTransparentInfo::length

## 8.362.2.2 WORD readTransparentInfo::offset

## 8.363 redirNumInfo Struct Reference

### Data Fields

- [BYTE PI](#)
- [BYTE SI](#)
- [BYTE numType](#)
- [BYTE numPlan](#)
- [BYTE reason](#)
- [BYTE numLen](#)
- [BYTE number](#) [255]

### 8.363.1 Detailed Description

This structure contains Redirecting Number Information

#### Parameters

<i>PI</i>	<ul style="list-style-type: none"><li>• Presentation indicator; refer to [S1, Table 2.7.4.4-1] for valid values.</li></ul>
-----------	--

<i>SI</i>	<ul style="list-style-type: none"> <li>Number of sets of following elements <ul style="list-style-type: none"> <li>Caller Id</li> </ul> </li> </ul>
<i>SI</i>	<ul style="list-style-type: none"> <li>Number screening indicator.</li> <li>Values: <ul style="list-style-type: none"> <li>0x00 - QMI_VOICE_SI_USER_PROVIDED_NOT_SCREENED - Provided user is not screened</li> <li>0x01 - QMI_VOICE_SI_USER_PROVIDED_VERIFIED_PASSED - Provided user passed verification</li> <li>0x02 - QMI_VOICE_SI_USER_PROVIDED_VERIFIED_FAILED - Provided user failed verification</li> <li>0x03 - QMI_VOICE_SI_NETWORK_PROVIDED - Provided network</li> </ul> </li> </ul>
<i>numType</i>	<ul style="list-style-type: none"> <li>Number type.</li> <li>Values: <ul style="list-style-type: none"> <li>0x00 - QMI_VOICE_NUM_TYPE_UNKNOWN - Unknown</li> <li>0x01 - QMI_VOICE_NUM_TYPE_INTERNATIONAL - International</li> <li>0x02 - QMI_VOICE_NUM_TYPE_NATIONAL - National</li> <li>0x03 - QMI_VOICE_NUM_TYPE_NETWORK_SPECIFIC - Network-specific</li> <li>0x04 - QMI_VOICE_NUM_TYPE_SUBSCRIBER - Subscriber</li> <li>0x05 - QMI_VOICE_NUM_TYPE_RESERVED - Reserved</li> <li>0x06 - QMI_VOICE_NUM_TYPE_ABBREVIATED - Abbreviated</li> <li>0x07 - QMI_VOICE_NUM_TYPE_RESERVED_EXTENSION - Reserved extension</li> </ul> </li> </ul>
<i>numPlan</i>	<ul style="list-style-type: none"> <li>Number plan.</li> <li>Values: <ul style="list-style-type: none"> <li>0x00 - QMI_VOICE_NUM_PLAN_UNKNOWN - Unknown</li> <li>0x01 - QMI_VOICE_NUM_PLAN_ISDN - ISDN</li> <li>0x03 - QMI_VOICE_NUM_PLAN_DATA - Data</li> <li>0x04 - QMI_VOICE_NUM_PLAN_TELEX - Telex</li> <li>0x08 - QMI_VOICE_NUM_PLAN_NATIONAL - National</li> <li>0x09 - QMI_VOICE_NUM_PLAN_PRIVATE - Private</li> <li>0x0B - QMI_VOICE_NUM_PLAN_RESERVED_CTS - Reserved cordless telephony system</li> <li>0x0F - QMI_VOICE_NUM_PLAN_RESERVED_EXTENSION - Reserved extension</li> </ul> </li> </ul>
<i>reason</i>	-Redirecting reason; refer to [S1, Table 3.7.5.11-1] for valid values
<i>numLen</i>	
	<ul style="list-style-type: none"> <li>Provides the length of number which follows</li> </ul>
<i>number[255]</i>	<ul style="list-style-type: none"> <li>number of numLen length, NULL terminated.</li> </ul>

### 8.363.2 Field Documentation

8.363.2.1 **BYTE** `redirNumInfo::number[255]`

8.363.2.2 **BYTE** `redirNumInfo::numLen`

8.363.2.3 **BYTE** `redirNumInfo::numPlan`

8.363.2.4 **BYTE** `redirNumInfo::numType`

8.363.2.5 **BYTE** `redirNumInfo::PI`

8.363.2.6 **BYTE** `redirNumInfo::reason`

8.363.2.7 **BYTE** `redirNumInfo::SI`

## 8.364 registerRefresh Struct Reference

### Data Fields

- **BYTE** `registerFlag`
- **BYTE** `voteForInit`
- **WORD** `numFiles`
- **fileInfo** `arrfileInfo [255]`

### 8.364.1 Detailed Description

This structure contains paramaters of refresh Information

## Parameters

<i>registerFlag</i>	<ul style="list-style-type: none"> <li>Flag that indicates whether to register or deregister for refresh indications. Valid values: <ul style="list-style-type: none"> <li>0 - Deregister</li> <li>1 - Register</li> </ul> </li> </ul>
<i>voteForInit</i>	<ul style="list-style-type: none"> <li>Flag that indicates whether to vote for the init when there is a refresh. Valid values: <ul style="list-style-type: none"> <li>0 - Client does not vote for initialization</li> <li>1 - Client votes for initialization</li> </ul> </li> </ul>
<i>numFiles</i>	<ul style="list-style-type: none"> <li>Number of sets of the following elements: <ul style="list-style-type: none"> <li>file_id</li> <li>path_len</li> <li>path</li> </ul> </li> </ul>
<i>arrfileInfo</i>	<ul style="list-style-type: none"> <li>Array of file Information structure.</li> <li>See /ref <a href="#">fileInfo</a> for more information</li> </ul>

## 8.364.2 Field Documentation

8.364.2.1 `fileInfo` `registerRefresh::arrfileInfo[255]`8.364.2.2 `WORD` `registerRefresh::numFiles`8.364.2.3 `BYTE` `registerRefresh::registerFlag`8.364.2.4 `BYTE` `registerRefresh::voteForInit`

## 8.365 remainingRetries Struct Reference

## Data Fields

- `BYTE` [verifyLeft](#)
- `BYTE` [unblockLeft](#)

## 8.365.1 Detailed Description

This structure contains the information about the retries remaining.



## Parameters

<i>verifyLeft</i>	<ul style="list-style-type: none"><li>• Number of remaining attempts to verify the PIN.</li><li>• 0xFF, if unavailable.</li></ul>
<i>unblockLeft</i>	<ul style="list-style-type: none"><li>• Number of remaining attempts to unblock the PIN.</li><li>• 0xFF, if unavailable.</li></ul>

## Note

This value is returned only when the enable/disable operation has failed. This information is not sent for a hidden key PIN type.

## 8.365.2 Field Documentation

8.365.2.1 **BYTE** remainingRetries::unblockLeft

8.365.2.2 **BYTE** remainingRetries::verifyLeft

## 8.366 remotePartyName Struct Reference

## Data Fields

- [BYTE](#) namePI
- [BYTE](#) codingScheme
- [BYTE](#) nameLen
- [BYTE](#) callerName [255]

## 8.366.1 Detailed Description

This structure contains information about the names that are dialed from the device or from which a call is received on the device.

## Parameters

<i>namePI</i>	<ul style="list-style-type: none"> <li>Name presentation indicator. <ul style="list-style-type: none"> <li>0x00 - PRESENTATION_NAME_PRESENTATION_ALLOWED - Allowed presentation</li> <li>0x01 - PRESENTATION_NAME_PRESENTATION_RESTRICTED - Restricted presentation</li> <li>0x02 - PRESENTATION_NAME_UNAVAILABLE - Unavailable presentation</li> <li>0x03 - PRESENTATION_NAME_NAME_PRESENTATION_RESTRICTED - Restricted name presentation</li> <li>0xFF - Not Available</li> </ul> </li> </ul>
<i>codingScheme</i>	<ul style="list-style-type: none"> <li>Refer to Table10 <a href="#">qaGobiApiTableCodingScheme.h</a> for coding schemes</li> <li>0xFF - Not Available</li> </ul>
<i>nameLen</i>	<ul style="list-style-type: none"> <li>Provides the length of name which follow.</li> <li>If zero(0) then no further information exists.</li> </ul>
<i>callerName[MAX_DESCRIPTOR_LENGTH]</i>	<ul style="list-style-type: none"> <li>Name in ASCII, NULL ending.</li> </ul>

## 8.366.2 Field Documentation

8.366.2.1 BYTE remotePartyName::callerName[255]

8.366.2.2 BYTE remotePartyName::codingScheme

8.366.2.3 BYTE remotePartyName::nameLen

8.366.2.4 BYTE remotePartyName::namePI

## 8.367 remotePartyNum Struct Reference

## Data Fields

- BYTE presentationInd
- BYTE numLen
- BYTE remPartyNumber [81]

## 8.367.1 Detailed Description

This structure contains information about the numbers that are dialed from the device or from which a call is received on the device.

## Parameters

<i>presentationInd</i>	<ul style="list-style-type: none"> <li>• Presentation indicator. <ul style="list-style-type: none"> <li>– 0x00 - PRESENTATION_ALLOWED - Allowed presentation</li> <li>– 0x01 - PRESENTATION_RESTRICTED - Restricted presentation</li> <li>– 0x02 - PRESENTATION_NUM_UNAVAILABLE - Unavailable presentation</li> <li>– 0x04 - PRESENTATION_PAYPHONE - Payphone presentation (GSM/UMTS specific)</li> <li>– 0xFF - Not Available</li> </ul> </li> </ul>
<i>numLen</i>	<ul style="list-style-type: none"> <li>• Provides the length of number which follow.</li> <li>• If zero(0) then no further information exists.</li> </ul>
<i>remParty- Number[MAX_ CALL_NO_LEN]</i>	<ul style="list-style-type: none"> <li>• Array of numbers in ASCII, NULL ending.</li> </ul>

## 8.367.2 Field Documentation

8.367.2.1 BYTE remotePartyNum::numLen

8.367.2.2 BYTE remotePartyNum::presentationInd

8.367.2.3 BYTE remotePartyNum::remPartyNumber[81]

## 8.368 ReqFieldsList Struct Reference

## Data Fields

- [BYTE requestFieldsLen](#)
- [BYTE requestFields](#) [256]

## 8.368.1 Detailed Description

This structure contains the Supported Request Fields List Information

## Parameters

<i>requestFields- Len</i>	<ul style="list-style-type: none"> <li>• Number of sets of the request fields.</li> </ul>
<i>requestFields</i>	<ul style="list-style-type: none"> <li>• Describes which optional field IDs are supported in QMI Request.</li> <li>• Array of uint8 is a bitmask where each bit represents a field ID.</li> <li>• Field 0-15 are mandatory, First Bit represents field ID 16,</li> <li>• Starting with the LSB, bit 0 represents Field ID 16, bit 1 represents ID 17.</li> </ul>

### 8.368.2 Field Documentation

8.368.2.1 **BYTE** ReqFieldsList::requestFields[256]

8.368.2.2 **BYTE** ReqFieldsList::requestFieldsLen

## 8.369 RespFieldsList Struct Reference

### Data Fields

- [BYTE responseFieldsLen](#)
- [BYTE responseFields](#) [256]

### 8.369.1 Detailed Description

This structure contains the Supported Response Fields List Information

#### Parameters

<i>responseFieldsLen</i>	<ul style="list-style-type: none"><li>• Number of sets of the response fields.</li></ul>
<i>responseFields</i>	<ul style="list-style-type: none"><li>• Describes which optional field IDs are supported in QMI Response.</li><li>• Format is same as request field.</li></ul>

### 8.369.2 Field Documentation

8.369.2.1 **BYTE** RespFieldsList::responseFields[256]

8.369.2.2 **BYTE** RespFieldsList::responseFieldsLen

## 8.370 RFBandInfoElements Struct Reference

### Data Fields

- [BYTE radiolInterface](#)
- [WORD activeBandClass](#)
- [WORD activeChannel](#)

### 8.370.1 Detailed Description

This structure contains the RFBandInfo response parameters.

## Parameters

<i>radioInterface</i>	<ul style="list-style-type: none"><li>• Radio interface technology<ul style="list-style-type: none"><li>– See <a href="#">Tables</a> for Radio Interface</li></ul></li></ul>
<i>activeBandClass</i>	<ul style="list-style-type: none"><li>• Active Band Class<ul style="list-style-type: none"><li>– See <a href="#">Tables</a> for Band Classes</li></ul></li></ul>
<i>activeChannel</i>	<ul style="list-style-type: none"><li>• Active channel (0 if channel is not relevant to the reported technology)</li></ul>

## 8.370.2 Field Documentation

8.370.2.1 WORD RFBandInfoElements::activeBandClass

8.370.2.2 WORD RFBandInfoElements::activeChannel

8.370.2.3 BYTE RFBandInfoElements::radioInterface

## 8.371 roamIndList Struct Reference

## Data Fields

- [BYTE numInstances](#)
- [BYTE radioInterface](#) [0x0A]
- [BYTE roamIndicator](#) [0x0A]

## 8.371.1 Detailed Description

This structure contains the Roaming Indicator List

- Parameter values default to their data type's maximum unsigned value unless explicitly stated otherwise.

## Parameters

<i>numInstances</i>	<ul style="list-style-type: none"><li>• number of sets of radio interface currently in use and roaming indicator<ul style="list-style-type: none"><li>– defaults to zero</li></ul></li></ul>
---------------------	--

<i>radiolInterface</i>	<ul style="list-style-type: none"> <li>• Radio Interface currently in use</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0x01 - RADIO_IF_CDMA_1X - cdma2000 1X</li> <li>– 0x02 - RADIO_IF_CDMA_1XEVDO - cdma2000 HRPD (1xEV-DO)</li> <li>– 0x03 - RADIO_IF_AMPS - AMPS</li> <li>– 0x04 - RADIO_IF_GSM - GSM</li> <li>– 0x05 - RADIO_IF_UMTS - UMTS</li> <li>– 0x08 - RADIO_IF_LTE - LTE</li> </ul> </li> </ul>
<i>roamIndicator</i>	<ul style="list-style-type: none"> <li>• Roaming Indicator</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0x00 - Roaming</li> <li>– 0x01 - Home</li> </ul> </li> </ul>

## 8.371.2 Field Documentation

8.371.2.1 **BYTE** roamIndList::numInstances

8.371.2.2 **BYTE** roamIndList::radiolInterface[0x0A]

8.371.2.3 **BYTE** roamIndList::roamIndicator[0x0A]

## 8.372 RoamingInfo Struct Reference

### Data Fields

- [BYTE TlvPresent](#)
- [BYTE roaming\\_ind](#)

### 8.372.1 Field Documentation

8.372.1.1 **BYTE** RoamingInfo::roaming\_ind

8.372.1.2 **BYTE** RoamingInfo::TlvPresent

## 8.373 roamTimer Struct Reference

### Data Fields

- [BYTE namID](#)
- [ULONG roamTimerValue](#)

### 8.373.1 Detailed Description

This structure contains information about the Roam Timer.

## Parameters

<i>namID</i>	<ul style="list-style-type: none"> <li>• Index of the NAM(Number Assignment Module) to be configured.</li> <li>• Range 0 to 3.</li> <li>• Some modems support only 1 or 2 NAMs.</li> <li>• 0xFF,if not available.</li> </ul>
<i>roamTimerValue</i>	<ul style="list-style-type: none"> <li>• Time in minutes.</li> <li>• Cumulative air time is slammed.</li> <li>• 0xFFFFFFFF,if not available.</li> </ul>

## 8.373.2 Field Documentation

## 8.373.2.1 BYTE roamTimer::namID

## 8.373.2.2 ULONG roamTimer::roamTimerValue

## 8.374 RSRPThresh Struct Reference

## Data Fields

- [BYTE RSRPThresListLen](#)
- [SHORT \\* pRSRPThresList](#)

## 8.374.1 Detailed Description

This structure contains RSRP threshold related parameters.

## Parameters

<i>RSRPThresListLen</i>	<ul style="list-style-type: none"> <li>• Length of the LTE RSRP threshold list parameter to follow</li> </ul>
<i>pRSRPThresList</i>	<ul style="list-style-type: none"> <li>• Sequence of thresholds delimiting current RSRP event reporting bands</li> <li>• Every time a new RSRP value crosses a specified threshold value, an event report indication message with the new RSRQ value is sent to the requesting control point. For this field             <ul style="list-style-type: none"> <li>– RSRP values are applicable only for LTE</li> <li>– RSRP values are measured in dBm, with a range of -44 dBm to -140 dBm</li> <li>– Each RSRP threshold value is a signed byte value</li> <li>– Maximum number of threshold values is 16</li> <li>– At least one value must be specified</li> </ul> </li> </ul>



### 8.374.2 Field Documentation

8.374.2.1 **SHORT\*** RSRPThresh::pRSRPThresList

8.374.2.2 **BYTE** RSRPThresh::RSRPThresListLen

## 8.375 rsrqInformation Struct Reference

### Data Fields

- [INT8](#) rsrq
- [BYTE](#) radiolf

### 8.375.1 Detailed Description

This structure contains the RSRQ Information

#### Parameters

<i>rsrq</i>	<ul style="list-style-type: none"> <li>• RSRQ value in dB (signed integer value); valid range is -3 to -20 (-3 means -3 dB, -20 means -20 dB)</li> </ul>
<i>radiolf</i>	<ul style="list-style-type: none"> <li>• Radio interface technology of the signal being measured <ul style="list-style-type: none"> <li>– 0x08 - LTE</li> </ul> </li> </ul>

### 8.375.2 Field Documentation

8.375.2.1 **BYTE** rsrqInformation::radiolf

8.375.2.2 **INT8** rsrqInformation::rsrq

## 8.376 RSRQThresh Struct Reference

### Data Fields

- [BYTE](#) RSRQThresListLen
- [INT8 \\*](#) pRSRQThresList

### 8.376.1 Detailed Description

This structure contains RSRQ threshold related parameters.

#### Parameters

<i>RSRQThresListLen</i>	<ul style="list-style-type: none"> <li>• Length of the LTE RSRQ threshold list parameter to follow</li> </ul>
-------------------------	---

<i>pRSRQThresList</i>	<ul style="list-style-type: none"> <li>• Sequence of thresholds delimiting current RSRQ event reporting bands</li> <li>• Every time a new RSRQ value crosses a threshold value, an event report indication message with the new RSRQ value is sent to the requesting control point. For this field <ul style="list-style-type: none"> <li>– RSRQ values are applicable only for LTE</li> <li>– RSRQ values are measured in dBm, with a range of -20 dBm to -3 dBm</li> <li>– Each RSRQ threshold value is a signed byte value</li> <li>– Maximum number of threshold values is 16</li> <li>– At least one value must be specified</li> </ul> </li> </ul>
-----------------------	--

## 8.376.2 Field Documentation

8.376.2.1 INT8\* RSRQThresh::pRSRQThresList

8.376.2.2 BYTE RSRQThresh::RSRQThresListLen

## 8.377 RSSIThresh Struct Reference

### Data Fields

- [BYTE RSSIThresListLen](#)
- [INT8 \\* pRSSIThresList](#)

### 8.377.1 Detailed Description

This structure contains RSSI threshold related parameters.

#### Parameters

<i>RSSIThresListLen</i>	<ul style="list-style-type: none"> <li>• Length of the RSSI threshold list parameter to follow</li> </ul>
<i>pRSSIThresList</i>	<ul style="list-style-type: none"> <li>• RSSI in dBm( signed bytes )</li> <li>• A value of -125 dBm or lower is used to indicate No Signal</li> <li>• RSSI values have the following ranges (in dBm) <ul style="list-style-type: none"> <li>– CDMA is -105 to -21</li> <li>– HDR is -118 to -13</li> <li>– GSM is -111 to -48</li> <li>– WCDMA is -121 to 0</li> <li>– LTE is -120 to 0</li> </ul> </li> <li>• Threshold values specified above are used for all RATs</li> <li>• The maximum number of threshold values is 16, each a signed byte value.</li> </ul>

### 8.377.2 Field Documentation

8.377.2.1 INT8\* RSSIThresh::pRSSIThesList

8.377.2.2 BYTE RSSIThresh::RSSIThesListLen

## 8.378 RXAGCList Struct Reference

### Data Fields

- WORD \* pRXStaticGain
- WORD \* pRXAIG
- WORD \* pRXExpThres
- WORD \* pRXExpSlope
- WORD \* pRXComprThres
- WORD \* pRXComprSlope

### 8.378.1 Detailed Description

This structure contains the SLQSGetAudioPathConfig parameters related to AV\_RXAGCLIST.

#### Parameters

<i>pRXStaticGain</i>	<ul style="list-style-type: none"> <li>• RX pre-compressor static gain</li> </ul>
<i>pRXAIG</i>	<ul style="list-style-type: none"> <li>• RX pre-compressor gain selection flag</li> </ul>
<i>pRXExpThres</i>	<ul style="list-style-type: none"> <li>• RX expansion threshold</li> </ul>
<i>pRXExpSlope</i>	<ul style="list-style-type: none"> <li>• RX expansion slope</li> </ul>
<i>pRXComprThres</i>	<ul style="list-style-type: none"> <li>• RX compression threshold</li> </ul>
<i>pRXComprSlope</i>	<ul style="list-style-type: none"> <li>• RX compression slope</li> </ul>

### 8.378.2 Field Documentation

8.378.2.1 WORD\* RXAGCList::pRXAIG

8.378.2.2 WORD\* RXAGCList::pRXComprSlope

8.378.2.3 WORD\* RXAGCList::pRXComprThres

8.378.2.4 WORD\* RXAGCList::pRXExpSlope

8.378.2.5 WORD\* RXAGCList::pRXExpThres

8.378.2.6 WORD\* RXAGCList::pRXStaticGain

## 8.379 RXAVCList Struct Reference

### Data Fields

- WORD \* pAVRXAVCSens
- WORD \* pAVRXAVCHeadroom

### 8.379.1 Detailed Description

This structure contains the SLQSGetAudioPathConfig parameters related to AV\_RXAVCLIST.

#### Parameters

<i>pAVRXAVC-Sens</i>	<ul style="list-style-type: none"> <li>• AVC variation from nominal sensitivity</li> </ul>
<i>pAVRXAVC-Headroom</i>	<ul style="list-style-type: none"> <li>• AVC headroom</li> </ul>

### 8.379.2 Field Documentation

8.379.2.1 WORD\* RXAVCList::pAVRXAVCHeadroom

8.379.2.2 WORD\* RXAVCList::pAVRXAVCSens

## 8.380 rxInfo Struct Reference

### Data Fields

- BYTE isRadioTuned
- ULONG rxPower
- ULONG ecio
- ULONG rscp
- ULONG rsrp
- ULONG phase

### 8.380.1 Detailed Description

This structure contains the Rx Information.

## Parameters

<i>isRadioTuned</i>	<ul style="list-style-type: none"> <li>Whether Rx is tuned to a channel: <ul style="list-style-type: none"> <li>0x00 - Not tuned</li> <li>0x01 - Tuned</li> <li>0xFF - Not Available</li> </ul> </li> <li>If the radio is tuned, instantaneous values are set for the signal information fields below.</li> <li>If the radio is not tuned, or is delayed or invalid, the values are set depending on each technology.</li> </ul>
<i>rx_pwr</i>	<ul style="list-style-type: none"> <li>Rx power value in 1/10 dbm resolution. <ul style="list-style-type: none"> <li>0xFFFFFFFF - Not Available</li> </ul> </li> </ul>
<i>ecio</i>	<ul style="list-style-type: none"> <li>ECIO in 1/10 dbm; valid for CDMA, HDR, GSM, WCDMA, and LTE. <ul style="list-style-type: none"> <li>0xFFFFFFFF - Not Available</li> </ul> </li> </ul>
<i>rscp</i>	<ul style="list-style-type: none"> <li>Received signal code power in 1/10 dbm.</li> <li>Valid for WCDMA. <ul style="list-style-type: none"> <li>0xFFFFFFFF - Not Available</li> </ul> </li> </ul>
<i>rsrp</i>	<ul style="list-style-type: none"> <li>Current reference signal received power in 1/10 dbm valid for LTE. <ul style="list-style-type: none"> <li>0xFFFFFFFF - Not Available</li> </ul> </li> </ul>
<i>phase</i>	<ul style="list-style-type: none"> <li>Phase in 1/100 degrees; valid for LTE.</li> <li>When the phase is unknown, 0xFFFFFFFF is used.</li> </ul>

## 8.380.2 Field Documentation

8.380.2.1 **ULONG** rxInfo::ecio8.380.2.2 **BYTE** rxInfo::isRadioTuned8.380.2.3 **ULONG** rxInfo::phase8.380.2.4 **ULONG** rxInfo::rscp8.380.2.5 **ULONG** rxInfo::rsrp

8.380.2.6 ULONG rxInfo::rxPower

## 8.381 RXPCMIIRFitr Struct Reference

### Data Fields

- WORD \* pFlag
- WORD \* pStageCnt
- BYTE \* pStage0Val
- BYTE \* pStage1Val
- BYTE \* pStage2Val
- BYTE \* pStage3Val
- BYTE \* pStage4Val

### 8.381.1 Detailed Description

This structure contains the SLQSGetAudioPathConfig parameters related to AV\_RXPCMIIRFLTR.

#### Parameters

<i>pFlag</i>	<ul style="list-style-type: none"> <li>• Flag               <ul style="list-style-type: none"> <li>– 0x0000 - IIR filter disable</li> <li>– 0xffff - IIR filter enable</li> </ul> </li> </ul>
<i>pStageCnt</i>	<ul style="list-style-type: none"> <li>• Stage Count               <ul style="list-style-type: none"> <li>– 0-4</li> </ul> </li> </ul>
<i>pStage0Val</i>	<ul style="list-style-type: none"> <li>• A 20 BYTE sized parameter indicating Stage 0 value               <ul style="list-style-type: none"> <li>– A1</li> <li>– A2</li> <li>– B0</li> <li>– B1</li> <li>– B2</li> </ul> </li> </ul>

<i>pStage1Val</i>	<ul style="list-style-type: none"> <li>• A 20 BYTE sized parameter indicating Stage 1 value <ul style="list-style-type: none"> <li>– A1</li> <li>– A2</li> <li>– B0</li> <li>– B1</li> <li>– B2</li> </ul> </li> </ul>
<i>pStage2Val</i>	<ul style="list-style-type: none"> <li>• A 20 BYTE sized parameter indicating Stage 2 value <ul style="list-style-type: none"> <li>– A1</li> <li>– A2</li> <li>– B0</li> <li>– B1</li> <li>– B2</li> </ul> </li> </ul>
<i>pStage3Val</i>	<ul style="list-style-type: none"> <li>• A 20 BYTE sized parameter indicating Stage 3 value <ul style="list-style-type: none"> <li>– A1</li> <li>– A2</li> <li>– B0</li> <li>– B1</li> <li>– B2</li> </ul> </li> </ul>
<i>pStage4Val</i>	<ul style="list-style-type: none"> <li>• A 20 BYTE sized parameter indicating Stage 4 value <ul style="list-style-type: none"> <li>– A1</li> <li>– A2</li> <li>– B0</li> <li>– B1</li> <li>– B2</li> </ul> </li> </ul>

## 8.381.2 Field Documentation

8.381.2.1 WORD\* RXPCMIIRFitr::pFlag

8.381.2.2 BYTE\* RXPCMIIRFitr::pStage0Val

8.381.2.3 BYTE\* RXPCMIIRFitr::pStage1Val

8.381.2.4 BYTE\* RXPCMIIRFitr::pStage2Val

8.381.2.5 BYTE\* RXPCMIIRFitr::pStage3Val

8.381.2.6 **BYTE\*** `RXPCMIIRFitr::pStage4Val`

8.381.2.7 **WORD\*** `RXPCMIIRFitr::pStageCnt`

## 8.382 `rxSignalStrengthListElement` Struct Reference

### Data Fields

- [SHORT](#) `rxSignalStrength`
- [BYTE](#) `radioIf`

### 8.382.1 Detailed Description

This structure contains the Received Signal Strength Information

#### Parameters

<i>rxSignalStrength</i>	<ul style="list-style-type: none"> <li>• Received signal strength in dBm <ul style="list-style-type: none"> <li>– For CDMA and UMTS, this indicates forward link pilotEc.</li> <li>– For GSM, the received signal strength.</li> <li>– For LTE, this indicates the total received wideband power observed by UE.</li> </ul> </li> </ul>
<i>radioIf</i>	<ul style="list-style-type: none"> <li>• Radio interface technology of the signal being radio_if measured <ul style="list-style-type: none"> <li>– 0x00 - RADIO_IF_NO_SVC - None (no service)</li> <li>– 0x01 - RADIO_IF_CDMA_1X - cdma2000 1X</li> <li>– 0x02 - RADIO_IF_CDMA_1XEVD0 - cdma2000 HRPD (1xEV-DO)</li> <li>– 0x03 - RADIO_IF_AMPS - AMPS</li> <li>– 0x04 - RADIO_IF_GSM - GSM</li> <li>– 0x05 - RADIO_IF_UMTS - UMTS</li> <li>– 0x08 - RADIO_IF_LTE - LTE</li> </ul> </li> </ul>

#### Note

First element of the RSSI list always contains the current Signal strength and Radio Interface.

### 8.382.2 Field Documentation

8.382.2.1 **BYTE** `rxSignalStrengthListElement::radioIf`

8.382.2.2 **SHORT** `rxSignalStrengthListElement::rxSignalStrength`

## 8.383 `sApnExtraParams` Struct Reference

### Data Fields

- [ULONG](#) `apnId`



- [BYTE ambr\\_ul](#)
- [BYTE ambr\\_dl](#)
- [BYTE ambr\\_ul\\_ext](#)
- [BYTE ambr\\_dl\\_ext](#)
- [BYTE ambr\\_ul\\_ext2](#)
- [BYTE ambr\\_dl\\_ext2](#)

### 8.383.1 Detailed Description

This structure contains the Extra Apn Params

#### Parameters

<i>apnId</i>	<ul style="list-style-type: none"> <li>• APN id</li> <li>• ID identifying the APN that the client would like to query the AMBR params</li> </ul>
<i>ambr_ul</i>	<ul style="list-style-type: none"> <li>• APN AMBR uplink</li> <li>• APN AMBR uplink values from 1 kbps to 8640 kbps</li> </ul>
<i>ambr_dl</i>	<ul style="list-style-type: none"> <li>• APN AMBR downlink</li> <li>• APN AMBR downlink values from 1 kbps to 8640 kbps</li> </ul>
<i>ambr_ul_ext</i>	<ul style="list-style-type: none"> <li>• Extended APN AMBR uplink</li> <li>• APN AMBR uplink values from 8700 kbps to 256 Mbps</li> </ul>
<i>ambr_dl_ext</i>	<ul style="list-style-type: none"> <li>• Extended APN AMBR downlink</li> <li>• APN AMBR downlink values from 8700 kbps to 256 Mbps</li> </ul>
<i>ambr_ul_ext2</i>	<ul style="list-style-type: none"> <li>• Second extended APN AMBR uplink</li> <li>• APN AMBR uplink values from 256 Mbps to 65280 Mbps</li> </ul>
<i>ambr_dl_ext2</i>	<ul style="list-style-type: none"> <li>• Second extended APN AMBR downlink</li> <li>• APN AMBR downlink values from 256 Mbps to 65280 Mbps</li> </ul>

### 8.383.2 Field Documentation

8.383.2.1 **BYTE** sApnExtraParams::ambr\_dl

8.383.2.2 **BYTE** sApnExtraParams::ambr\_dl\_ext

8.383.2.3 **BYTE** sApnExtraParams::ambr\_dl\_ext2

8.383.2.4 **BYTE** sApnExtraParams::ambr\_ul

8.383.2.5 **BYTE** sApnExtraParams::ambr\_ul\_ext

8.383.2.6 **BYTE** sApnExtraParams::ambr\_ul\_ext2

8.383.2.7 **ULONG** sApnExtraParams::apnId

## 8.384 satelliteInfo Struct Reference

### Data Fields

- [BYTE](#) svListLen
- [ULONG](#) validMask
- [ULONG](#) system
- [WORD](#) gnssSvid
- [BYTE](#) healthStatus
- [ULONG](#) svStatus
- [BYTE](#) svInfoMask
- [FLOAT](#) elevation
- [FLOAT](#) azimuth
- [FLOAT](#) snr

### 8.384.1 Detailed Description

Contain fields in struct [satelliteInfo](#)

#### Parameters

<i>svListLen</i>	<ul style="list-style-type: none"> <li>• number of sets of the following elements: <ul style="list-style-type: none"> <li>– validMask</li> <li>– system</li> <li>– gnssSvid</li> <li>– healthStatus</li> <li>– svStatus</li> <li>– svInfoMask</li> <li>– elevation</li> <li>– azimuth</li> <li>– snr</li> </ul> </li> </ul>
------------------	---

<i>validMask</i>	<ul style="list-style-type: none"> <li>• Bitmask indicating which of the fields in this TLV are valid. Valid bitmasks: <ul style="list-style-type: none"> <li>– 0x00000001 - VALID_SYSTEM</li> <li>– 0x00000002 - VALID_GNSS_SVID</li> <li>– 0x00000004 - VALID_HEALTH_STATUS</li> <li>– 0x00000008 - VALID_PROCESS_STATUS</li> <li>– 0x00000010 - VALID_SVINFO_MASK</li> <li>– 0x00000020 - VALID_ELEVATION</li> <li>– 0x00000040 - VALID_AZIMUTH</li> <li>– 0x00000080 - VALID_SNR</li> </ul> </li> </ul>
<i>system</i>	<ul style="list-style-type: none"> <li>• Indicates to which constellation this <a href="#">SV</a> belongs. Valid values: <ul style="list-style-type: none"> <li>– eQMI_LOC_SV_SYSTEM_GPS (1) - GPS satellite</li> <li>– eQMI_LOC_SV_SYSTEM_GALILEO (2) - GALILEO satellite</li> <li>– eQMI_LOC_SV_SYSTEM_SBAS (3) - SBAS satellite</li> <li>– eQMI_LOC_SV_SYSTEM_COMPASS (4) - COMPASS satellite</li> <li>– eQMI_LOC_SV_SYSTEM_GLONASS (5) - GLONASS satellite</li> <li>– eQMI_LOC_SV_SYSTEM_BDS (6) - BDS satellite</li> </ul> </li> </ul>
<i>gnssSvId</i>	<ul style="list-style-type: none"> <li>• GNSS <a href="#">SV</a> ID. The GPS and GLONASS SVs can be disambiguated using the system field. Range: <ul style="list-style-type: none"> <li>– FOR GPS: 1 to 32</li> <li>– FOR GLONASS: 1 to 32</li> <li>– FOR SBAS: 120 to 151</li> <li>– for BDS: 201 to 237</li> </ul> </li> </ul>
<i>healthStatus</i>	<ul style="list-style-type: none"> <li>• health status. Range: 0 - 1 <ul style="list-style-type: none"> <li>– 0 - unhealthy</li> <li>– 1 - healthy</li> </ul> </li> </ul>
<i>svStatus</i>	<ul style="list-style-type: none"> <li>• <a href="#">SV</a> process status. Valid values: <ul style="list-style-type: none"> <li>– eQMI_LOC_SV_STATUS_IDLE (1) - <a href="#">SV</a> is not being actively processed</li> <li>– eQMI_LOC_SV_STATUS_SEARCH (2) - The system is searching for this <a href="#">SV</a></li> <li>– eQMI_LOC_SV_STATUS_TRACK (3) - <a href="#">SV</a> is being tracked</li> </ul> </li> </ul>

<i>svInfoMask</i>	<ul style="list-style-type: none"> <li>Indicates whether almanac and ephemeris information is available. Valid bitmasks: <ul style="list-style-type: none"> <li>0x01 - SVINFO_HAS_EPHEMERIS</li> <li>0x02 - SVINFO_HAS_ALMANAC</li> </ul> </li> </ul>
<i>elevation</i>	<ul style="list-style-type: none"> <li>SV elevation angle. <ul style="list-style-type: none"> <li>Units: Degrees</li> <li>Range: 0 to 90</li> </ul> </li> </ul>
<i>azimuth</i>	<ul style="list-style-type: none"> <li>SV azimuth angle. <ul style="list-style-type: none"> <li>Units: Degrees</li> <li>Range: 0 to 360</li> </ul> </li> </ul>
<i>snr</i>	<ul style="list-style-type: none"> <li>SV signal-to-noise ratio <ul style="list-style-type: none"> <li>Units: dB-Hz</li> </ul> </li> </ul>

**Note**

None

**8.384.2 Field Documentation**8.384.2.1 **FLOAT** `satelliteInfo::azimuth`8.384.2.2 **FLOAT** `satelliteInfo::elevation`8.384.2.3 **WORD** `satelliteInfo::gnssSvId`8.384.2.4 **BYTE** `satelliteInfo::healthStatus`8.384.2.5 **FLOAT** `satelliteInfo::snr`8.384.2.6 **BYTE** `satelliteInfo::svInfoMask`8.384.2.7 **BYTE** `satelliteInfo::svListLen`8.384.2.8 **ULONG** `satelliteInfo::svStatus`8.384.2.9 **ULONG** `satelliteInfo::system`8.384.2.10 **ULONG** `satelliteInfo::validMask`**8.385 sensorDataUsage\_s Struct Reference**

## Data Fields

- [ULONG usageMask](#)
- [ULONG aidingIndicatorMask](#)

### 8.385.1 Detailed Description

This structure contains Sensor Data Usage info.

#### Parameters

<i>usageMask</i>	<ul style="list-style-type: none"> <li>• Specifies which sensors were used in calculating the position in the position report.</li> </ul>
------------------	---

- Value
  - 0x00000001 - Accelerometer used
  - 0x00000002 - Gyroscope used

#### Parameters

<i>aidingIndicatorMask</i>	
----------------------------	--

- Specifies which results were aided by sensors.

- Value
  - 0x00000001 - AIDED\_HEADING
  - 0x00000002 - AIDED\_SPEED
  - 0x00000004 - AIDED\_POSITION
  - 0x00000008 - AIDED\_VELOCITY

### 8.385.2 Field Documentation

8.385.2.1 **ULONG** sensorDataUsage\_s::aidingIndicatorMask

8.385.2.2 **ULONG** sensorDataUsage\_s::usageMask

## 8.386 serialNumbersInfo Struct Reference

## Data Fields

- [BYTE esnSize](#)
- [CHAR \\* pESNString](#)
- [BYTE imeiSize](#)
- [CHAR \\* pIMEIString](#)
- [BYTE meidSize](#)
- [CHAR \\* pMEIDString](#)
- [BYTE imeiSvnSize](#)
- [CHAR \\* pImeiSvnString](#)

### 8.386.1 Detailed Description

Returns all the serial numbers assigned to the device. These serial numbers include the ESN (Electronic serial number of the device), the IMEI (International Mobile Equipment Identity) and MEID (Mobile Equipment Identifier).

## Parameters

<i>esnSize</i>	<ul style="list-style-type: none"> <li>The maximum number of characters (including NULL terminator) that the ESN string array can contain</li> </ul>
<i>pESNString[OUT]</i>	<ul style="list-style-type: none"> <li>NULL-terminated ESN string. Empty string is returned when ESN is not supported/programmed</li> </ul>
<i>imeiSize</i>	<ul style="list-style-type: none"> <li>The maximum number of characters (including NULL terminator) that the IMEI string array can contain</li> </ul>
<i>pIMEIString[OUT]</i>	<ul style="list-style-type: none"> <li>NULL terminated IMEI string. Empty string is returned when IMEI is not supported/programmed</li> </ul>
<i>meidSize</i>	<ul style="list-style-type: none"> <li>The maximum number of characters (including NULL terminator) that the MEID string array can contain</li> </ul>
<i>pMEIDString[OUT]</i>	<ul style="list-style-type: none"> <li>NULL-terminated MEID string. Empty string is returned when MEID is not supported/programmed</li> </ul>
<i>imeiSvnSize</i>	<ul style="list-style-type: none"> <li>The maximum number of characters (including NULL terminator) that the IMEI SVN string array can contain.</li> </ul>
<i>pImeiSvnString[OUT]</i>	<ul style="list-style-type: none"> <li>NULL-terminated IMEI SVN string. Empty string is returned when IMEI SVN is not supported/programmed.</li> </ul>

## 8.386.2 Field Documentation

8.386.2.1 BYTE serialNumbersInfo::esnSize

8.386.2.2 BYTE serialNumbersInfo::imeiSize

8.386.2.3 BYTE serialNumbersInfo::imeiSvnSize

8.386.2.4 BYTE serialNumbersInfo::meidSize

8.386.2.5 CHAR\* serialNumbersInfo::pESNString

8.386.2.6 CHAR\* serialNumbersInfo::pIMEIString

8.386.2.7 CHAR\* serialNumbersInfo::pImeiSvnString

8.386.2.8 CHAR\* serialNumbersInfo::pMEIDString

## 8.387 serviceProviderName Struct Reference

### Data Fields

- [BYTE displayCondition](#)
- [BYTE spnLength](#)
- [BYTE spn](#) [255]

### 8.387.1 Detailed Description

This structure contains Service Provider Name as defined in 3GPP TS 31.102 (Section 4.2.12) from multiple sources.

#### Parameters

<i>displayCondition</i>	<ul style="list-style-type: none"><li>• Display condition.</li></ul>
<i>spnLength</i>	<ul style="list-style-type: none"><li>• It provides length of spn.</li></ul>
<i>spn</i>	<ul style="list-style-type: none"><li>• Service provider name string must use: The SMS default 7-bit coded alphabet as defined in 3GPP TS 23.038 with bit 8 set to 9.</li></ul>

### 8.387.2 Field Documentation

8.387.2.1 **BYTE** serviceProviderName::displayCondition

8.387.2.2 **BYTE** serviceProviderName::spn[255]

8.387.2.3 **BYTE** serviceProviderName::spnLength

## 8.388 ServingSystemInfo Struct Reference

### Data Fields

- [BYTE registrationState](#)
- [BYTE csAttachState](#)
- [BYTE psAttachState](#)
- [BYTE selectedNetwork](#)
- [BYTE radiolInterfaceNo](#)
- [BYTE radiolInterfaceList](#) [255]
- [BYTE hdrPersonality](#)

### 8.388.1 Detailed Description

This structure will hold the serving system parameters information

## Parameters

<i>registrationState</i>	- Registration state of the mobile <ul style="list-style-type: none"> <li>• 0 - QMI_NAS_NOT_REGISTERED Not registered;mobile is not currently searching for a new network to provide service</li> <li>• 1 - QMI_NAS_REGISTERED Registered with a network</li> <li>• 2 - QMI_NAS_NOT_REGISTERED_SEARCHING Not registered, but mobile is currently searching for a new network to provide service</li> <li>• 3 - QMI_NAS_REGISTRATION_DENIED Registration denied by the visible network</li> <li>• 4 - QMI_NAS_REGISTRATION_UNKNOWN Registration state is unknown</li> </ul>
<i>csAttachState</i>	- Circuit Switch domain attach state of the mobile <ul style="list-style-type: none"> <li>• 0 - Unknown or not applicable</li> <li>• 1 - Attached</li> <li>• 2 - Detached</li> </ul>
<i>psAttachState</i>	- Packet domain attach state of the mobile <ul style="list-style-type: none"> <li>• 0 - Unknown or not applicable</li> <li>• 1 - Attached</li> <li>• 2 - Detached</li> </ul>
<i>selectedNetwork</i>	- Type of selected radio access network <ul style="list-style-type: none"> <li>• 0x00 - Unknown</li> <li>• 0x01 - 3GPP2 network</li> <li>• 0x02 - 3GPP network</li> </ul>
<i>radioInterfaceNo</i>	- Number of radio interfaces currently in use; this indicates how many radio_if identifiers follow this field
<i>radioInterface-List</i>	- Radio interface currently in use (each is 1 byte) <ul style="list-style-type: none"> <li>• 0x00 - None (no service)</li> <li>• 0x01 - cdma2000 1X</li> <li>• 0x02 - cdma2000 HRPD (1xEV-DO)</li> <li>• 0x03 - AMPS</li> <li>• 0x04 - GSM</li> <li>• 0x05 - UMTS</li> <li>• 0x08 - LTE</li> </ul>
<i>hdrPersonality</i>	- HDR personality information (valid only for EVDO) <ul style="list-style-type: none"> <li>• 0x00 - Unknown</li> <li>• 0x01 - HRPD</li> <li>• 0x02 - eHRPD</li> </ul>



Note: None

## 8.388.2 Field Documentation

8.388.2.1 **BYTE** ServingSystemInfo::csAttachState

8.388.2.2 **BYTE** ServingSystemInfo::hdrPersonality

8.388.2.3 **BYTE** ServingSystemInfo::psAttachState

8.388.2.4 **BYTE** ServingSystemInfo::radiolInterfaceList[255]

8.388.2.5 **BYTE** ServingSystemInfo::radiolInterfaceNo

8.388.2.6 **BYTE** ServingSystemInfo::registrationState

8.388.2.7 **BYTE** ServingSystemInfo::selectedNetwork

## 8.389 servSystem Struct Reference

### Data Fields

- [BYTE regState](#)
- [BYTE csAttachState](#)
- [BYTE psAttachState](#)
- [BYTE selNetwork](#)
- [BYTE numRadioInterfaces](#)
- [BYTE radiolInterface](#) [0x0A]

### 8.389.1 Detailed Description

This structure contains the Serving System parameters

- Parameter values default to their data type's maximum unsigned value unless explicitly stated otherwise.

## Parameters

<i>regState</i>	<ul style="list-style-type: none"> <li>• Registration state - Registration state of the mobile</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0 - Not Registered; mobile is not currently searching for a new network to provide service</li> <li>– 1 - Registered with a network</li> <li>– 2 - Not registered, but mobile is currently searching for a new network to provide service</li> <li>– 3 - Registration denied by visible network</li> <li>– 4 - Registration state is unknown</li> </ul> </li> </ul>
<i>csAttachState</i>	<ul style="list-style-type: none"> <li>• CS Attach State - Circuit-switched domain attach state of the mobile</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0 - Unknown or not applicable</li> <li>– 1 - Attached</li> <li>– 2 - Detached</li> </ul> </li> </ul>
<i>psAttachState</i>	<ul style="list-style-type: none"> <li>• PS Attach State - Packet-switched domain attach state of the mobile</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0 - Unknown or not applicable</li> <li>– 1 - Attached</li> <li>– 2 - Detached</li> </ul> </li> </ul>
<i>selNetwork</i>	<ul style="list-style-type: none"> <li>• Selected Network - Type of selected radio access network</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0 - Unknown</li> <li>– 1 - 3GPP2 network</li> <li>– 2 - 3GPP network</li> </ul> </li> </ul>

<i>numRadio-Interfaces</i>	<ul style="list-style-type: none"> <li>• In Use Radio Interfaces Number <ul style="list-style-type: none"> <li>– Number of radio interfaces currently in use</li> <li>– defaults to zero</li> </ul> </li> </ul>
<i>radioInterface</i>	<ul style="list-style-type: none"> <li>• Radio Interface(s) modem discovered</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0x00 - RADIO_IF_NO_SVC - None(no service)</li> <li>– 0x01 - RADIO_IF_CDMA_1X - cdma2000 1X</li> <li>– 0x02 - RADIO_IF_CDMA_1XEVD0 - cdma2000 HRPD (1xEV-DO)</li> <li>– 0x03 - RADIO_IF_AMPS - AMPS</li> <li>– 0x04 - RADIO_IF_GSM - GSM</li> <li>– 0x05 - RADIO_IF_UMTS - UMTS</li> <li>– 0x08 - RADIO_IF_LTE - LTE</li> </ul> </li> </ul>

## 8.389.2 Field Documentation

8.389.2.1 **BYTE** servSystem::csAttachState

8.389.2.2 **BYTE** servSystem::numRadioInterfaces

8.389.2.3 **BYTE** servSystem::psAttachState

8.389.2.4 **BYTE** servSystem::radioInterface[0x0A]

8.389.2.5 **BYTE** servSystem::regState

8.389.2.6 **BYTE** servSystem::selNetwork

## 8.390 sessionInfo Union Reference

### Data Fields

- struct [omaDmFotaTlv](#) [omaDmFota](#)
- struct [omaDmConfigTlv](#) [omaDmConfig](#)
- struct [omaDmNotificationsTlv](#) [omaDmNotifications](#)

## 8.390.1 Detailed Description

This union [sessionInfo](#) consist of [omaDmFotaTlv](#), [omaDmConfigTlv](#) and [omaDmNotificationsTlv](#), out of which one will be unpacked against pEventFields.

## 8.390.2 Field Documentation

8.390.2.1 **struct** [omaDmConfigTlv](#) sessionInfo::omaDmConfig

8.390.2.2 struct **omaDmFotaTlv** sessionInfo::omaDmFota

8.390.2.3 struct **omaDmNotificationsTlv** sessionInfo::omaDmNotifications

## 8.391 sessionInfoExt Union Reference

### Data Fields

- struct [omaDmFotaTlvExt](#) omaDmFota
- struct [omaDmConfigTlvExt](#) omaDmConfig

### 8.391.1 Detailed Description

This union [sessionInfo](#) consist of [omaDmFotaTlv](#) and [omaDmConfigTlv](#), out of which one will be unpacked against pEventFields.

### 8.391.2 Field Documentation

8.391.2.1 struct **omaDmConfigTlvExt** sessionInfoExt::omaDmConfig

8.391.2.2 struct **omaDmFotaTlvExt** sessionInfoExt::omaDmFota

## 8.392 sessionInfoTlv Struct Reference

### Data Fields

- [BYTE](#) TlvPresent
- [ULONG](#) sessionType
- [sessionInformation](#) sessionInfo

### 8.392.1 Detailed Description

Structure used to store all [sessionInfo\(Union\)](#)TLV Value.

#### Parameters

<i>TlvPresent</i>	- Boolean indicating the presence of the TLV in the QMI response
<i>sessionType</i>	- sessiontype Value

### 8.392.2 Field Documentation

8.392.2.1 **sessionInformation** sessionInfoTlv::sessionInfo

8.392.2.2 **ULONG** sessionInfoTlv::sessionType

8.392.2.3 **BYTE** sessionInfoTlv::TlvPresent

## 8.393 sessionInfoTlvExt Struct Reference

### Data Fields

- [BYTE](#) TlvPresent

- [ULONG sessionType](#)
- [sessionInformationExt sessionInfo](#)

### 8.393.1 Detailed Description

Structure used to store all [sessionInfo\(Union\)](#)TLV Value.

Parameters

<i>TlvPresent</i>	- Boolean indicating the presence of the TLV in the QMI response
<i>sessionType</i>	- sessiontype Value

### 8.393.2 Field Documentation

8.393.2.1 [sessionInformationExt sessionInfoTlvExt::sessionInfo](#)

8.393.2.2 [ULONG sessionInfoTlvExt::sessionType](#)

8.393.2.3 [BYTE sessionInfoTlvExt::TlvPresent](#)

## 8.394 SetAudioPathConfigReq Struct Reference

Data Fields

- [BYTE Profile](#)
- [BYTE \\* pECMode](#)
- [BYTE \\* pNSEnable](#)
- [WORD \\* pTXGain](#)
- [WORD \\* pDTMF\\_TXGain](#)
- [WORD \\* pCodecSTGain](#)
- [TXPCMIIRFiltr \\* pTXPCMIIRFiltr](#)
- [RXPCMIIRFiltr \\* pRXPCMIIRFiltr](#)
- [BYTE \\* pRXAVCAGCSwitch](#)
- [BYTE \\* pTXAVCSwitch](#)
- [RXAGCList \\* pRXAGCList](#)
- [RXAVCList \\* pRXAVCList](#)
- [TXAGCList \\* pTXAGCList](#)

### 8.394.1 Detailed Description

This structure contains the SLQSSetAudioPathConfig request parameters.

Parameters

<i>Profile</i>	[Mandatory] <ul style="list-style-type: none"><li>• Audio Profile<ul style="list-style-type: none"><li>– 0-9</li></ul></li></ul>
----------------	--

<i>pECMode</i>	[Optional] <ul style="list-style-type: none"> <li>• AV_EC             <ul style="list-style-type: none"> <li>– 0 - Echo cancellation off</li> <li>– 1 - Handset echo mode</li> <li>– 2 - Headset mode</li> <li>– 3 - Car kit mode</li> <li>– 4 - Speaker Mode</li> </ul> </li> </ul>
<i>pNSEnable</i>	[Optional] <ul style="list-style-type: none"> <li>• Noise Suppression             <ul style="list-style-type: none"> <li>– 0 - Noise suppression off</li> <li>– 1 - Noise suppression on</li> </ul> </li> </ul>
<i>pTXGain</i>	[Optional] <ul style="list-style-type: none"> <li>• TX Voice volume             <ul style="list-style-type: none"> <li>– 0x0000 - 0xffff</li> </ul> </li> </ul>
<i>pDTMFTXGain</i>	[Optional] <ul style="list-style-type: none"> <li>• AV_DTMFTXG             <ul style="list-style-type: none"> <li>– 0x0000 - 0xffff</li> </ul> </li> </ul>
<i>pCodecSTGain</i>	[Optional] <ul style="list-style-type: none"> <li>• AV_CODEECSTG             <ul style="list-style-type: none"> <li>– 0x0000 - 0xffff</li> </ul> </li> </ul>
<i>pTXPCMIIRFtr</i>	[Optional] <ul style="list-style-type: none"> <li>• See <a href="#">TXPCMIIRFtr</a> for more information</li> </ul>
<i>pRXPCMIIRFtr</i>	[Optional] <ul style="list-style-type: none"> <li>• See <a href="#">RXPCMIIRFtr</a> for more information</li> </ul>
<i>pRXAVCAGC-Switch</i>	[Optional] <ul style="list-style-type: none"> <li>• RX AVC/AGC Switch</li> </ul>
<i>pTXAVCSwitch</i>	[Optional] <ul style="list-style-type: none"> <li>• TX AVC Switch</li> </ul>
<i>pRXAGCList</i>	[Optional] <ul style="list-style-type: none"> <li>• See <a href="#">RXAGCList</a> for more information</li> </ul>
<i>pRXAVCList</i>	[Optional] <ul style="list-style-type: none"> <li>• See <a href="#">RXAVCList</a> for more information</li> </ul>
<i>pTXAGCList</i>	[Optional] <ul style="list-style-type: none"> <li>• See <a href="#">TXAGCList</a> for more information</li> </ul>

## 8.394.2 Field Documentation

- 8.394.2.1 **WORD\*** SetAudioPathConfigReq::pCodecSTGain
- 8.394.2.2 **WORD\*** SetAudioPathConfigReq::pDTMFTXGain
- 8.394.2.3 **BYTE\*** SetAudioPathConfigReq::pECMode
- 8.394.2.4 **BYTE\*** SetAudioPathConfigReq::pNSEnable
- 8.394.2.5 **BYTE** SetAudioPathConfigReq::Profile
- 8.394.2.6 **RXAGCList\*** SetAudioPathConfigReq::pRXAGCList
- 8.394.2.7 **BYTE\*** SetAudioPathConfigReq::pRXAVCAGCSwitch
- 8.394.2.8 **RXAVCList\*** SetAudioPathConfigReq::pRXAVCList
- 8.394.2.9 **RXPCMIIRFitr\*** SetAudioPathConfigReq::pRXPCMIIRFitr
- 8.394.2.10 **TXAGCList\*** SetAudioPathConfigReq::pTXAGCList
- 8.394.2.11 **BYTE\*** SetAudioPathConfigReq::pTXAVCSwitch
- 8.394.2.12 **WORD\*** SetAudioPathConfigReq::pTXGain
- 8.394.2.13 **TXPCMIIRFitr\*** SetAudioPathConfigReq::pTXPCMIIRFitr

## 8.395 SetAudioProfileReq Struct Reference

### Data Fields

- [BYTE Profile](#)
- [BYTE EarMute](#)
- [BYTE MicMute](#)
- [BYTE Generator](#)
- [BYTE Volume](#)

### 8.395.1 Detailed Description

This structure contains the SLQSSetAudioProfile request parameters.

## Parameters

<i>Profile</i>	<ul style="list-style-type: none"> <li>• Audio Profile <ul style="list-style-type: none"> <li>– 0 - Handset</li> <li>– 1 - Headset</li> <li>– 2 - Car Kit</li> <li>– 3 - Speaker phone</li> <li>– 4 - Auxiliary</li> <li>– 5 - TTY</li> <li>– 6 - Auxiliary external PCM</li> <li>– 7 - Primary external PCM</li> <li>– 8 - External slave PCM</li> <li>– 9 - I2S</li> </ul> </li> </ul>
<i>EarMute</i>	<ul style="list-style-type: none"> <li>• Ear Mute Setting <ul style="list-style-type: none"> <li>– 0 - unmuted</li> <li>– 1 - muted</li> </ul> </li> </ul>
<i>MicMute</i>	<ul style="list-style-type: none"> <li>• MIC Mute Setting <ul style="list-style-type: none"> <li>– 0 - unmuted</li> <li>– 1 - muted</li> </ul> </li> </ul>
<i>Generator</i>	<ul style="list-style-type: none"> <li>• Audio Generator <ul style="list-style-type: none"> <li>– 0 - Voice</li> <li>– 1 - Key Beep</li> <li>– 2 - MIDI</li> </ul> </li> </ul>
<i>Volume</i>	<ul style="list-style-type: none"> <li>• Audio Volume Level <ul style="list-style-type: none"> <li>– 0 to 7</li> </ul> </li> </ul>

## 8.395.2 Field Documentation

8.395.2.1 BYTE SetAudioProfileReq::EarMute

8.395.2.2 BYTE SetAudioProfileReq::Generator

8.395.2.3 BYTE SetAudioProfileReq::MicMute



8.395.2.4 **BYTE** SetAudioProfileReq::Profile

8.395.2.5 **BYTE** SetAudioProfileReq::Volume

## 8.396 SetAudioVolTLBConfigReq Struct Reference

### Data Fields

- [BYTE Profile](#)
- [BYTE Generator](#)
- [BYTE Volume](#)
- [BYTE Item](#)
- [WORD VolValue](#)

### 8.396.1 Detailed Description

This structure contains the SLQSSetAudioVolTLBConfig request parameters

#### Parameters

<i>Profile</i>	<ul style="list-style-type: none"><li>• Audio Profile<ul style="list-style-type: none"><li>– 0-9</li></ul></li></ul>
<i>Generator</i>	<ul style="list-style-type: none"><li>• Audio Generator<ul style="list-style-type: none"><li>– 0-2</li></ul></li></ul>
<i>Volume</i>	<ul style="list-style-type: none"><li>• Audio Volume Level<ul style="list-style-type: none"><li>– 0-7</li></ul></li></ul>
<i>Item</i>	<ul style="list-style-type: none"><li>• Item<ul style="list-style-type: none"><li>– 13 - AV_RXVOLDB</li><li>– 14 - AV_DTMFVOLDB</li><li>– 15 - AV_PAD</li></ul></li></ul>

<i>Value</i>	<ul style="list-style-type: none"><li>• Value to be set to the volume table</li></ul>
--------------	---

## 8.396.2 Field Documentation

8.396.2.1 **BYTE** SetAudioVolTLBConfigReq::Generator

8.396.2.2 **BYTE** SetAudioVolTLBConfigReq::Item

8.396.2.3 **BYTE** SetAudioVolTLBConfigReq::Profile

8.396.2.4 **BYTE** SetAudioVolTLBConfigReq::Volume

8.396.2.5 **WORD** SetAudioVolTLBConfigReq::VolValue

## 8.397 SetAudioVolTLBConfigResp Struct Reference

### Data Fields

- [WORD ResCode](#)

### 8.397.1 Detailed Description

This structure contains the SLQSSetAudioVolTLBConfig response parameters.

#### Parameters

<i>ResCode</i>	<ul style="list-style-type: none"><li>• Result of requested item</li></ul>
----------------	--

## 8.397.2 Field Documentation

8.397.2.1 **WORD** SetAudioVolTLBConfigResp::ResCode

## 8.398 setCustomSettingV2 Struct Reference

### Data Fields

- [CHAR cust\\_id](#) [64+1]
- [WORD value\\_length](#)
- [BYTE cust\\_value](#) [8+1]

### 8.398.1 Detailed Description

This structure contains customization settings set to modem

## Parameters

<i>cust_id</i>	<ul style="list-style-type: none"><li>Customization ID (Maximum 64 bytes)</li></ul>
<i>value_length</i>	<ul style="list-style-type: none"><li>length of <i>cust_value</i> field</li></ul>
<i>cust_value</i>	<ul style="list-style-type: none"><li>Customization Setting Value (Maximum 8 bytes)</li></ul>

## 8.398.2 Field Documentation

8.398.2.1 CHAR setCustomSettingV2::cust\_id[64+1]

8.398.2.2 BYTE setCustomSettingV2::cust\_value[8+1]

8.398.2.3 WORD setCustomSettingV2::value\_length

## 8.399 SetIMSSMSConfigReq Struct Reference

## Data Fields

- [BYTE \\* pSMSFormat](#)
- [BYTE \\* pSMSOverIPNwInd](#)
- [BYTE \\* pPhoneCtxtURLen](#)
- [BYTE \\* pPhoneCtxtURI](#)

## 8.399.1 Detailed Description

This structure contains the SLQSSetIMSSMSConfig request parameters.

## Parameters

<i>pSMSFormat</i>	<ul style="list-style-type: none"><li>SMS format<ul style="list-style-type: none"><li>0 - 3GPP</li><li>1 - 3GPP2</li></ul></li></ul>
<i>pSMSOverIPNwInd</i>	<ul style="list-style-type: none"><li>SMS over IP Network Indication Flag<ul style="list-style-type: none"><li>TRUE - Turn on mobile-originated SMS</li><li>FALSE - Turn off mobile-originated SMS</li></ul></li></ul>

<i>pPhoneCtxtURLen</i>	<ul style="list-style-type: none"> <li>• Length of Phone context Universal Resource Identifier to follow</li> </ul>
<i>pPhoneCtxtURI</i>	<ul style="list-style-type: none"> <li>• Phone context universal resource identifier</li> <li>• Length of this string must be specified in pPhoneCtxtURLen parameter</li> </ul>

### 8.399.2 Field Documentation

8.399.2.1 **BYTE\*** SetIMSSMSConfigReq::pPhoneCtxtURI

8.399.2.2 **BYTE\*** SetIMSSMSConfigReq::pPhoneCtxtURLen

8.399.2.3 **BYTE\*** SetIMSSMSConfigReq::pSMSFormat

8.399.2.4 **BYTE\*** SetIMSSMSConfigReq::pSMSOverIPNwInd

## 8.400 SetIMSSMSConfigResp Struct Reference

### Data Fields

- **BYTE \*** [pSettingResp](#)

### 8.400.1 Detailed Description

This structure contains the SLQSSetIMSSMSConfig response parameters.

#### Parameters

<i>pSettingResp</i>	<ul style="list-style-type: none"> <li>• Settings standard response type. A settings specific error code is returned when the standard response error type is QMI_ERR_CAUSE_CODE</li> </ul>
---------------------	---

### 8.400.2 Field Documentation

8.400.2.1 **BYTE\*** SetIMSSMSConfigResp::pSettingResp

## 8.401 SetIMSUserConfigReq Struct Reference

### Data Fields

- **BYTE \*** [pIMSDomainLen](#)
- **BYTE \*** [pIMSDomain](#)

### 8.401.1 Detailed Description

This structure contains the SLQSSetIMSUserConfig request parameters.

## Parameters

<i>pIMSDomainLen</i>	<ul style="list-style-type: none"><li>Length of IMS Domain Name to follow</li></ul>
<i>pIMSDomain</i>	<ul style="list-style-type: none"><li>IMS domain name</li></ul>

## 8.401.2 Field Documentation

8.401.2.1 BYTE\* SetIMSUserConfigReq::pIMSDomain

8.401.2.2 BYTE\* SetIMSUserConfigReq::pIMSDomainLen

## 8.402 SetIMSUserConfigResp Struct Reference

## Data Fields

- BYTE \* pSettingResp

## 8.402.1 Detailed Description

This structure contains the SLQSSetIMSUserConfig response parameters.

## Parameters

<i>pSettingResp</i>	<ul style="list-style-type: none"><li>Settings standard response type. A settings specific error code is returned when the standard response error type is QMI_ERR_CAUSE_CODE</li></ul>
---------------------	---

## 8.402.2 Field Documentation

8.402.2.1 BYTE\* SetIMSUserConfigResp::pSettingResp

## 8.403 SetIMSVoIPConfigReq Struct Reference

## Data Fields

- WORD \* pSessionExpiryTimer
- WORD \* pMinSessionExpiryTimer
- BYTE \* pAmrWbEnable
- BYTE \* pScrAmrEnable
- BYTE \* pScrAmrWbEnable
- BYTE \* pAmrMode
- WORD \* pAmrWBMode
- BYTE \* pAmrOctetAligned
- BYTE \* pAmrWBOctetAligned
- WORD \* pRingingTimer
- WORD \* pRingBackTimer
- WORD \* pRTPRTCPInactTimer

### 8.403.1 Detailed Description

This structure contains the SLQSSetIMSVoIPConfig request parameters.

## Parameters

<i>pSessionExpiry-Timer</i>	<ul style="list-style-type: none"> <li>• Session duration, in seconds</li> </ul>
<i>pMinSession-ExpiryTimer</i>	<ul style="list-style-type: none"> <li>• Minimum allowed value for session expiry timer, in seconds</li> </ul>
<i>pAmrWbEnable</i>	<ul style="list-style-type: none"> <li>• Flag to enable/disable Adaptive Multirate Codec(AMR) WideBand(WB) audio</li> <li>• Values: <ul style="list-style-type: none"> <li>– True - Enable</li> <li>– False - Disable</li> </ul> </li> </ul>
<i>pScrAmrEnable</i>	<ul style="list-style-type: none"> <li>• Flag to enable/disable Source Control Rate(SCR) for AMR NarrowBand (NB)</li> <li>• Values: <ul style="list-style-type: none"> <li>– True - Enable</li> <li>– False - Disable</li> </ul> </li> </ul>
<i>pScrAmrWb-Enable</i>	<ul style="list-style-type: none"> <li>• Flag to enable/disable SCR for AMR WB Audio</li> <li>• Values: <ul style="list-style-type: none"> <li>– True - Enable</li> <li>– False - Disable</li> </ul> </li> </ul>
<i>pAmrMode</i>	<ul style="list-style-type: none"> <li>• BitMask for AMR NB modes allowed</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0x1 - 4.75 kbps</li> <li>– 0x2 - 5.15 kbps</li> <li>– 0x4 - 5.9 kbps</li> <li>– 0x8 - 6.17 kbps</li> <li>– 0x10 - 7.4 kbps</li> <li>– 0x20 - 7.95 kbps</li> <li>– 0x40 - 10.2 kbps</li> <li>– 0x80 - 12.2 kbps</li> </ul> </li> </ul>

<i>pAmrWBMode</i>	<ul style="list-style-type: none"> <li>• BitMask for AMR WB modes allowed</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0x1 - 6.60 kbps</li> <li>– 0x2 - 8.85 kbps</li> <li>– 0x4 - 12.65 kbps</li> <li>– 0x8 - 14.25 kbps</li> <li>– 0x10 - 15.85 kbps</li> <li>– 0x20 - 18.25 kbps</li> <li>– 0x40 - 19.85 kbps</li> <li>– 0x80 - 23.05 kbps</li> <li>– 0x100 - 23.85 kbps</li> </ul> </li> </ul>
<i>pAmrOctet-Aligned</i>	<ul style="list-style-type: none"> <li>• Flag to indicate if the octet is aligned for AMR NB Audio</li> <li>• Values: <ul style="list-style-type: none"> <li>– True - Aligned</li> <li>– False - Not aligned, Bandwidth Efficient mode</li> </ul> </li> </ul>
<i>pAmrWBOctet-Aligned</i>	<ul style="list-style-type: none"> <li>• Flag to indicate if the octet is aligned for AMR WB Audio</li> <li>• Values: <ul style="list-style-type: none"> <li>– True - Aligned</li> <li>– False - Not aligned, Bandwidth Efficient mode</li> </ul> </li> </ul>
<i>pRingingTimer</i>	<ul style="list-style-type: none"> <li>• Duration of ringing timer, in seconds. The ringing timer starts on the ringing event. If the call is not answered within the duration of this timer, the call is disconnected.</li> </ul>
<i>pRingBackTimer</i>	<ul style="list-style-type: none"> <li>• Duration of ringback timer, in seconds. The ringback timer starts on the ringback event. If the call is not answered within the duration of this timer, the call is disconnected.</li> </ul>
<i>pRTPRTCP-InactTimer</i>	<ul style="list-style-type: none"> <li>• Duration of RTP/RTCP inactivity timer, in seconds. If no RTP/RTCP packet is received prior to the expiry of this timer, the call is disconnected.</li> </ul>

## 8.403.2 Field Documentation

8.403.2.1 **BYTE\*** SetIMSVoIPConfigReq::pAmrMode

8.403.2.2 **BYTE\*** SetIMSVoIPConfigReq::pAmrOctetAligned



- 8.403.2.3 **BYTE\*** SetIMSVoIPConfigReq::pAmrWbEnable
- 8.403.2.4 **WORD\*** SetIMSVoIPConfigReq::pAmrWBMode
- 8.403.2.5 **BYTE\*** SetIMSVoIPConfigReq::pAmrWBOctetAligned
- 8.403.2.6 **WORD\*** SetIMSVoIPConfigReq::pMinSessionExpiryTimer
- 8.403.2.7 **WORD\*** SetIMSVoIPConfigReq::pRingBackTimer
- 8.403.2.8 **WORD\*** SetIMSVoIPConfigReq::pRingingTimer
- 8.403.2.9 **WORD\*** SetIMSVoIPConfigReq::pRTPRTCPIInactTimer
- 8.403.2.10 **BYTE\*** SetIMSVoIPConfigReq::pScrAmrEnable
- 8.403.2.11 **BYTE\*** SetIMSVoIPConfigReq::pScrAmrWbEnable
- 8.403.2.12 **WORD\*** SetIMSVoIPConfigReq::pSessionExpiryTimer

## 8.404 SetIMSVoIPConfigResp Struct Reference

### Data Fields

- [BYTE \\* pSettingResp](#)

### 8.404.1 Detailed Description

This structure contains the SLQSSetIMSVoIPConfig response parameters.

#### Parameters

<i>pSettingResp</i>	<ul style="list-style-type: none"><li>• Settings standard response type. A settings specific error code is returned when the standard response error type is QMI_ERR_CAUSE_CODE</li></ul>
---------------------	---

### 8.404.2 Field Documentation

- 8.404.2.1 **BYTE\*** SetIMSVoIPConfigResp::pSettingResp

## 8.405 SetM2MAudioAVCFGReq Struct Reference

### Data Fields

- [BYTE Profile](#)
- [BYTE Device](#)
- [BYTE PIFACEId](#)
- [PCMparams \\* pPCMPParams](#)

### 8.405.1 Detailed Description

This structure contains the SLQSSetM2MAudioAVCFG request parameters.

## Parameters

<i>Profile</i>	<ul style="list-style-type: none"> <li>• Audio Profile <ul style="list-style-type: none"> <li>– 0-5</li> </ul> </li> </ul>
<i>Device</i>	<ul style="list-style-type: none"> <li>• ACDB Device</li> <li>• See <a href="#">qaGobiApiTableSwiAudio.h</a> for more information on ACDB Device</li> </ul>
<i>PIFACEId</i>	<ul style="list-style-type: none"> <li>• Physical Interface</li> <li>• See <a href="#">qaGobiApiTableSwiAudio.h</a> for more information on physical interface</li> </ul>
<i>pPCMPParams</i>	<ul style="list-style-type: none"> <li>• PCM parameters</li> <li>• See <a href="#">PCMparams</a> for more information</li> </ul>

## 8.405.2 Field Documentation

8.405.2.1 BYTE SetM2MAudioAVCFGReq::Device

8.405.2.2 BYTE SetM2MAudioAVCFGReq::PIFACEId

8.405.2.3 PCMparams\* SetM2MAudioAVCFGReq::pPCMPParams

8.405.2.4 BYTE SetM2MAudioAVCFGReq::Profile

## 8.406 SetM2MAudioLPBKReq Struct Reference

## Data Fields

- [BYTE Enable](#)

## 8.406.1 Detailed Description

This structure contains the SLQSSetM2MAudioLPBK request parameters.

## Parameters

<i>Enable</i>	<ul style="list-style-type: none"> <li>• Operation to be performed <ul style="list-style-type: none"> <li>– 0 - stop</li> <li>– 1 - VOCODER loop</li> <li>– 2 - internal codec loop</li> </ul> </li> </ul>
---------------	--

## 8.406.2 Field Documentation

### 8.406.2.1 BYTE SetM2MAudioLPBKReq::Enable

## 8.407 SetM2MAudioProfileReq Struct Reference

### Data Fields

- [BYTE Profile](#)
- [BYTE \\* pEarMute](#)
- [BYTE \\* pMicMute](#)
- [BYTE \\* pGenerator](#)
- [BYTE \\* pVolume](#)
- [BYTE \\* pCwtMute](#)

### 8.407.1 Detailed Description

This structure contains the SLQSSetM2MAudioProfile request parameters.

#### Parameters

<i>Profile</i>	<ul style="list-style-type: none"><li>• Audio Profile Number<ul style="list-style-type: none"><li>– 0-5</li></ul></li></ul>
<i>pEarMute</i>	<ul style="list-style-type: none"><li>• Ear Mute<ul style="list-style-type: none"><li>– 0 - mute</li><li>– 1 - unmute</li></ul></li></ul>
<i>pMicMute</i>	<ul style="list-style-type: none"><li>• Mic Mute<ul style="list-style-type: none"><li>– 0 - mute</li><li>– 1 - unmute</li></ul></li></ul>
<i>pGenerator</i>	<ul style="list-style-type: none"><li>• Generator<ul style="list-style-type: none"><li>– 0 - voice</li></ul></li></ul>
<i>pVolume</i>	<ul style="list-style-type: none"><li>• Set RX Volume level<ul style="list-style-type: none"><li>– 0-5</li></ul></li></ul>

<i>pCwtMute</i>	<ul style="list-style-type: none"> <li>• Call Waiting Tone Mute <ul style="list-style-type: none"> <li>– 0 - Mute</li> <li>– 1 - UnMute</li> </ul> </li> </ul>
-----------------	--

## 8.407.2 Field Documentation

8.407.2.1 **BYTE\*** SetM2MAudioProfileReq::pCwtMute

8.407.2.2 **BYTE\*** SetM2MAudioProfileReq::pEarMute

8.407.2.3 **BYTE\*** SetM2MAudioProfileReq::pGenerator

8.407.2.4 **BYTE\*** SetM2MAudioProfileReq::pMicMute

8.407.2.5 **BYTE** SetM2MAudioProfileReq::Profile

8.407.2.6 **BYTE\*** SetM2MAudioProfileReq::pVolume

## 8.408 SetM2MAudioVolumeReq Struct Reference

### Data Fields

- [BYTE Profile](#)
- [BYTE Generator](#)
- [BYTE Level](#)

### 8.408.1 Detailed Description

This structure contains the SLQSSetM2MAudioProfile request parameters.

#### Parameters

<i>Profile</i>	<ul style="list-style-type: none"> <li>• Audio Profile Number <ul style="list-style-type: none"> <li>– 0-5</li> </ul> </li> </ul>
<i>Generator</i>	<ul style="list-style-type: none"> <li>• Generator <ul style="list-style-type: none"> <li>– 0 - voice</li> </ul> </li> </ul>

<i>Level</i>	<ul style="list-style-type: none"><li>• Audio volume level<ul style="list-style-type: none"><li>– 0-5</li></ul></li></ul>
--------------	---

## 8.408.2 Field Documentation

8.408.2.1 **BYTE** SetM2MAudioVolumeReq::Generator

8.408.2.2 **BYTE** SetM2MAudioVolumeReq::Level

8.408.2.3 **BYTE** SetM2MAudioVolumeReq::Profile

## 8.409 SetM2MAVMMuteReq Struct Reference

### Data Fields

- [BYTE Profile](#)
- [BYTE EarMute](#)
- [BYTE MicMute](#)
- [BYTE \\* pCwtMute](#)

## 8.409.1 Detailed Description

This structure contains the SLQSSetM2MAVMMute request parameters.

### Parameters

<i>Profile</i>	<ul style="list-style-type: none"><li>• Audio Profile Number<ul style="list-style-type: none"><li>– 0-5</li></ul></li></ul>
<i>EarMute</i>	<ul style="list-style-type: none"><li>• Ear Mute<ul style="list-style-type: none"><li>– 0-1</li></ul></li></ul>
<i>MicMute</i>	<ul style="list-style-type: none"><li>• Mic Mute<ul style="list-style-type: none"><li>– 0-1</li></ul></li></ul>
<i>pCwtMute</i>	[ Optional ] <ul style="list-style-type: none"><li>• Call Waiting Tone Mute<ul style="list-style-type: none"><li>– 0-1</li></ul></li></ul>

## 8.409.2 Field Documentation

8.409.2.1 **BYTE** SetM2MAVMuteReq::EarMute

8.409.2.2 **BYTE** SetM2MAVMuteReq::MicMute

8.409.2.3 **BYTE\*** SetM2MAVMuteReq::pCwtMute

8.409.2.4 **BYTE** SetM2MAVMuteReq::Profile

## 8.410 SetM2MSpkrGainReq Struct Reference

### Data Fields

- [BYTE Profile](#)
- [WORD Value](#)

### 8.410.1 Detailed Description

This structure contains the SLQSSetM2MSpkrGain request parameters.

#### Parameters

<i>Profile</i>	<ul style="list-style-type: none"><li>• Audio Profile Number<ul style="list-style-type: none"><li>– 0-5</li></ul></li></ul>
<i>Value</i>	<ul style="list-style-type: none"><li>• RX speakerphone gain<ul style="list-style-type: none"><li>– 0x0 - 0x7fff</li></ul></li></ul>

### 8.410.2 Field Documentation

8.410.2.1 **BYTE** SetM2MSpkrGainReq::Profile

8.410.2.2 **WORD** SetM2MSpkrGainReq::Value

## 8.411 setPINProtection Struct Reference

### Data Fields

- [BYTE pinID](#)
- [BYTE pinOperation](#)
- [BYTE pinLength](#)
- [BYTE pinValue](#) [255]

### 8.411.1 Detailed Description

This structure contains the information about the pin protection parameters that need to be set.

## Parameters

<i>pinID</i>	<ul style="list-style-type: none"> <li>Indicates the PIN ID to be enabled or disabled. <ul style="list-style-type: none"> <li>1 - PIN1 (also called PIN)</li> <li>2 - PIN2</li> <li>3 - Universal PIN</li> <li>4 - Hidden key</li> </ul> </li> </ul>
<i>pinOperation</i>	<ul style="list-style-type: none"> <li>Indicates whether the PIN is enabled or disabled. <ul style="list-style-type: none"> <li>0 - Disable the PIN</li> <li>1 - Enable the PIN</li> </ul> </li> </ul>
<i>pinLength</i>	<ul style="list-style-type: none"> <li>Length of the following elements i.e. pin value.</li> </ul>
<i>pinValue</i> [MAX_DESCRIPTION_LENGTH]	<ul style="list-style-type: none"> <li>PIN value.</li> <li>This value is a sequence of ASCII characters.</li> </ul>

## 8.411.2 Field Documentation

8.411.2.1 BYTE setPINProtection::pinID

8.411.2.2 BYTE setPINProtection::pinLength

8.411.2.3 BYTE setPINProtection::pinOperation

8.411.2.4 BYTE setPINProtection::pinValue[255]

## 8.412 SetRegMgrConfigReq Struct Reference

## Data Fields

- WORD \* pPriCSCFPort
- BYTE \* pCSCFPortNameLen
- BYTE \* pCSCFPortName
- BYTE \* pIMSTestMode

## 8.412.1 Detailed Description

This structure contains the SLQSSetRegMgrConfig request parameters.

## Parameters

<i>pPriCSCFPort</i>	<ul style="list-style-type: none"> <li>Primary call session control function port</li> </ul>
<i>pCSCFPort-NameLen</i>	<ul style="list-style-type: none"> <li>Length of the CSCF <a href="#">Port</a> name parameter to follow</li> </ul>
<i>pCSCFPort-Name</i>	<ul style="list-style-type: none"> <li>Call Session control port, fully qualified domain name</li> <li>Length of this string must be specified in pCSCFPortNameLen parameter</li> </ul>
<i>pIMSTestMode</i>	<ul style="list-style-type: none"> <li>IMS Test mode Enabled. <ul style="list-style-type: none"> <li>TRUE - Enable, no IMS registration</li> <li>FALSE - Disable, IMS registration is initiated</li> </ul> </li> </ul>

## 8.412.2 Field Documentation

8.412.2.1 **BYTE\*** SetRegMgrConfigReq::pCSCFPortName8.412.2.2 **BYTE\*** SetRegMgrConfigReq::pCSCFPortNameLen8.412.2.3 **BYTE\*** SetRegMgrConfigReq::pIMSTestMode8.412.2.4 **WORD\*** SetRegMgrConfigReq::pPriCSCFPort

## 8.413 SetRegMgrConfigResp Struct Reference

## Data Fields

- [BYTE \\*](#) [pSettingResp](#)

## 8.413.1 Detailed Description

This structure contains the SLQSSetRegMgrConfig response parameters.

## Parameters

<i>pSettingResp</i>	<ul style="list-style-type: none"> <li>Settings standard response type. A settings specific error code is returned when the standard response error type is QMI_ERR_CAUSE_CODE</li> </ul>
---------------------	---

## 8.413.2 Field Documentation

8.413.2.1 **BYTE\*** SetRegMgrConfigResp::pSettingResp



## 8.414 `setSignalStrengthInfo` Struct Reference

### Data Fields

- [CDMARSSIThresh](#) \* [pCDMARSSIThresh](#)
- [WORD](#) \* [pCDMARSSIDelta](#)
- [CDMAECIOThresh](#) \* [pCDMAECIOThresh](#)
- [WORD](#) \* [pCDMAECIODelta](#)
- [HRRSSIThresh](#) \* [pHRRSSIThresh](#)
- [WORD](#) \* [pHRRSSIDelta](#)
- [HDRECIOThresh](#) \* [pHDRECIOThresh](#)
- [WORD](#) \* [pHDRECIODelta](#)
- [HRSINRThreshold](#) \* [pHRSINRThresh](#)
- [WORD](#) \* [pHRSINRDelta](#)
- [HDRIOThresh](#) \* [pHDRIOThresh](#)
- [WORD](#) \* [pHDRIODelta](#)
- [GSMRSSIThresh](#) \* [pGSMRSSIThresh](#)
- [WORD](#) \* [pGSMRSSIDelta](#)
- [WCDMARSSIThresh](#) \* [pWCDMARSSIThresh](#)
- [WORD](#) \* [pWCDMARSSIDelta](#)
- [WCDMAECIOThresh](#) \* [pWCDMAECIOThresh](#)
- [WORD](#) \* [pWCDMAECIODelta](#)
- [LTERSSIThresh](#) \* [pLTERSSIThresh](#)
- [WORD](#) \* [pLTERSSIDelta](#)
- [LTESNRThreshold](#) \* [pLTESNRThresh](#)
- [WORD](#) \* [pLTESNRDelta](#)
- [LTERSRQThresh](#) \* [pLTERSRQThresh](#)
- [WORD](#) \* [pLTERSRQDelta](#)
- [LTERSRPThresh](#) \* [pLTERSRPThresh](#)
- [WORD](#) \* [pLTERSRPDelta](#)
- [LTERSigRptConfig](#) \* [pLTERSigRptConfig](#)
- [TDSCDMARSCPTThresh](#) \* [pTDSCDMARSCPTThresh](#)
- [WORD](#) \* [pTDSCDMARSCPDelta](#)
- [TDSCDMARSSIThresh](#) \* [pTDSCDMARSSIThresh](#)
- [ULONG](#) \* [pTDSCDMARSSIDelta](#)
- [TDSCDMAECIOThresh](#) \* [pTDSCDMAECIOThresh](#)
- [ULONG](#) \* [pTDSCDMAECIODelta](#)
- [TDSCDMASINRThresh](#) \* [pTDSCDMASINRThresh](#)
- [ULONG](#) \* [pTDSCDMASINRDelta](#)

### 8.414.1 Detailed Description

This structure contains the Signal Strength reporting thresholds Item information.

#### Parameters

<i>pCDMARSSI- Thresh</i>	<ul style="list-style-type: none"> <li>• CDMA RSSI threshold List</li> <li>• See <a href="#">CDMARSSIThresh</a> for more details</li> </ul>
------------------------------	---

<i>pCDMARSSI-Delta</i>	<ul style="list-style-type: none"> <li>• RSSI delta (in units of 0.1 dBm).</li> <li>• A value of 0 is rejected with a QMI_ERR_INVALID_ARG error.</li> </ul>
<i>pCDMAECIO-Thresh</i>	<ul style="list-style-type: none"> <li>• CDMA ECIO Threshold List</li> <li>• See <a href="#">CDMAECIOThresh</a> for more details</li> </ul>
<i>pCDMAECIO-Delta</i>	<ul style="list-style-type: none"> <li>• ECIO delta (in units of 0.1 dB).</li> <li>• A value of 0 is rejected with a QMI_ERR_INVALID_ARG error.</li> </ul>
<i>pHDRRSSI-Thresh</i>	<ul style="list-style-type: none"> <li>• HDR RSSI Threshold List</li> <li>• See <a href="#">HDRRSSIThresh</a> for more details</li> </ul>
<i>pHDRRSSIDelta</i>	<ul style="list-style-type: none"> <li>• RSSI delta (in units of 0.1 dBm)</li> <li>• A value of 0 is rejected with a QMI_ERR_INVALID_ARG error.</li> </ul>
<i>pHDRECIOThresh</i>	<ul style="list-style-type: none"> <li>• HDR ECIO Threshold List</li> <li>• See <a href="#">HDRECIOThresh</a> for more details</li> </ul>
<i>pHDRECIODelta</i>	<ul style="list-style-type: none"> <li>• ECIO delta (in units of 0.1 dB)</li> <li>• A value of 0 is rejected with a QMI_ERR_INVALID_ARG error.</li> </ul>
<i>pHDRSINR-Thresh</i>	<ul style="list-style-type: none"> <li>• HDR SINR Threshold List</li> <li>• See <a href="#">HDRSINRThreshold</a> for more details</li> </ul>
<i>pHDRSINRDelta</i>	<ul style="list-style-type: none"> <li>• SINR delta (in units of 1 SINR level)</li> <li>• A value of 0 is rejected with a QMI_ERR_INVALID_ARG error.</li> </ul>
<i>pHDRIOThresh</i>	<ul style="list-style-type: none"> <li>• HDR IO Threshold List</li> <li>• See <a href="#">HDRIOThresh</a> for more details</li> </ul>
<i>pHDRIODelta</i>	<ul style="list-style-type: none"> <li>• IO delta (in units of 0.1 dBm)</li> <li>• A value of 0 is rejected with a QMI_ERR_INVALID_ARG error.</li> </ul>

<i>pGSMRSSI- Thresh</i>	<ul style="list-style-type: none"> <li>• GSM RSSI Threshold List</li> <li>• See <a href="#">GSMRSSIThresh</a> for more details</li> </ul>
<i>pGSMRSSIDelta</i>	<ul style="list-style-type: none"> <li>• RSSI delta (in units of 0.1 dBm)</li> <li>• A value of 0 is rejected with a QMI_ERR_INVALID_ARG error.</li> </ul>
<i>pWCDMARSSI- Thresh</i>	<ul style="list-style-type: none"> <li>• WCDMA RSSI Threshold List</li> <li>• See <a href="#">WCDMARSSIThresh</a> for more details</li> </ul>
<i>pWCDMARSSI- Delta</i>	<ul style="list-style-type: none"> <li>• RSSI delta (in units of 0.1 dBm).</li> <li>• A value of 0 is rejected with a QMI_ERR_INVALID_ARG error.</li> </ul>
<i>pWCDMAECIO- Thresh</i>	<ul style="list-style-type: none"> <li>• WCDMA ECIO Threshold List</li> <li>• See <a href="#">WCDMAECIOThresh</a> for more details</li> </ul>
<i>pWCDMAECIO- Delta</i>	<ul style="list-style-type: none"> <li>• ECIO delta (in units of 0.1 dB)</li> <li>• A value of 0 is rejected with a QMI_ERR_INVALID_ARG error.</li> </ul>
<i>pLTERSSI- Thresh</i>	<ul style="list-style-type: none"> <li>• LTE RSSI Threshold List</li> <li>• See <a href="#">LTERSSIThresh</a> for more details</li> </ul>
<i>pLTERSSIDelta</i>	<ul style="list-style-type: none"> <li>• RSSI delta (in units of 0.1 dBm)</li> <li>• A value of 0 is rejected with a QMI_ERR_INVALID_ARG error.</li> </ul>
<i>pLTESNR- Thresh</i>	<ul style="list-style-type: none"> <li>• LTE SNR Threshold List</li> <li>• See <a href="#">LTESNRThreshold</a> for more details</li> </ul>
<i>pLTESNRDelta</i>	<ul style="list-style-type: none"> <li>• SNR delta (in units of 0.1 dBm)</li> <li>• A value of 0 is rejected with a QMI_ERR_INVALID_ARG error.</li> </ul>
<i>pLTERSRQ- Thresh</i>	<ul style="list-style-type: none"> <li>• LTE RSRQ Threshold List</li> <li>• See <a href="#">LTERSRQThresh</a> for more details</li> </ul>

<i>pLTERSQR-Delta</i>	<ul style="list-style-type: none"> <li>• RSRQ delta (in units of 0.1 dBm)</li> <li>• A value of 0 is rejected with a QMI_ERR_INVALID_ARG error.</li> </ul>
<i>pLTERSRP-Thresh</i>	<ul style="list-style-type: none"> <li>• LTE RSRP Threshold List</li> <li>• See <a href="#">LTERSRPThresh</a> for more details</li> </ul>
<i>pLTERSRPDelta</i>	<ul style="list-style-type: none"> <li>• RSRP delta (in units of 0.1 dBm).</li> <li>• A value of 0 is rejected with a QMI_ERR_INVALID_ARG error.</li> </ul>
<i>pLTERSigRpt-Config</i>	<ul style="list-style-type: none"> <li>• LTE Signal Report Config</li> <li>• See <a href="#">LTERSigRptConfig</a> for more details</li> </ul>
<i>pTDSCDMARS-CPThresh</i>	<ul style="list-style-type: none"> <li>• TDSCDMA RSCP Threshold List</li> <li>• See <a href="#">TDSCDMARSCPThresh</a> for more details</li> </ul>
<i>pTDSCDMARS-CPDelta</i>	<ul style="list-style-type: none"> <li>• RSCP delta (in units of 0.1 dBm)</li> <li>• A value of 0 is rejected with a QMI_ERR_INVALID_ARG error.</li> </ul>
<i>pTDSCDMARS-SIThresh</i>	<ul style="list-style-type: none"> <li>• TDSCDMA RSSI Threshold List</li> <li>• See <a href="#">TDSCDMARSSIThresh</a> for more details</li> </ul>
<i>pTDSCDMARS-SIDelta</i>	<ul style="list-style-type: none"> <li>• RSSI delta (in dBm) used by TD-SCDMA.</li> </ul>
<i>pTDSCDMAECI-OTthresh</i>	<ul style="list-style-type: none"> <li>• TDSCDMA ECIO Threshold List</li> <li>• See <a href="#">TDSCDMAECIOTthresh</a> for more details</li> </ul>
<i>pTDSCDMAECI-ODelta</i>	<ul style="list-style-type: none"> <li>• ECIO delta (in dB) used by TD-SCDMA</li> </ul>
<i>pTDSCDMASIN-RThresh</i>	<ul style="list-style-type: none"> <li>• TDSCDMA SINR Threshold List</li> <li>• See <a href="#">TDSCDMASINRThresh</a> for more details</li> </ul>
<i>pTDSCDMASIN-RDelta</i>	<ul style="list-style-type: none"> <li>• SINR delta (in dB) used by TD-SCDMA.</li> </ul>

## 8.414.2 Field Documentation

- 8.414.2.1 `WORD* setSignalStrengthInfo::pCDMAECIODelta`
- 8.414.2.2 `CDMAECIOThresh* setSignalStrengthInfo::pCDMAECIOThresh`
- 8.414.2.3 `WORD* setSignalStrengthInfo::pCDMARSSIDelta`
- 8.414.2.4 `CDMARSSIThresh* setSignalStrengthInfo::pCDMARSSIThresh`
- 8.414.2.5 `WORD* setSignalStrengthInfo::pGSMRSSIDelta`
- 8.414.2.6 `GSMRSSIThresh* setSignalStrengthInfo::pGSMRSSIThresh`
- 8.414.2.7 `WORD* setSignalStrengthInfo::pHDRECIODelta`
- 8.414.2.8 `HDRECIOThresh* setSignalStrengthInfo::pHDRECIOThresh`
- 8.414.2.9 `WORD* setSignalStrengthInfo::pHDRIODelta`
- 8.414.2.10 `HDRIOThresh* setSignalStrengthInfo::pHDRIOThresh`
- 8.414.2.11 `WORD* setSignalStrengthInfo::pHRRSSIDelta`
- 8.414.2.12 `HDRRSSIThresh* setSignalStrengthInfo::pHRRSSIThresh`
- 8.414.2.13 `WORD* setSignalStrengthInfo::pHRSINRDelta`
- 8.414.2.14 `HDRSINRThreshold* setSignalStrengthInfo::pHRSINRThresh`
- 8.414.2.15 `WORD* setSignalStrengthInfo::pLTERSRPDelta`
- 8.414.2.16 `LTERSRPThresh* setSignalStrengthInfo::pLTERSRPThresh`
- 8.414.2.17 `WORD* setSignalStrengthInfo::pLTERSRQDelta`
- 8.414.2.18 `LTERSRQThresh* setSignalStrengthInfo::pLTERSRQThresh`
- 8.414.2.19 `WORD* setSignalStrengthInfo::pLTERSSIDelta`
- 8.414.2.20 `LTERSSIThresh* setSignalStrengthInfo::pLTERSSIThresh`
- 8.414.2.21 `LTESigRptConfig* setSignalStrengthInfo::pLTESigRptConfig`
- 8.414.2.22 `WORD* setSignalStrengthInfo::pLTESNRDelta`
- 8.414.2.23 `LTESNRThreshold* setSignalStrengthInfo::pLTESNRThresh`
- 8.414.2.24 `ULONG* setSignalStrengthInfo::pTDSCDMAECIODelta`
- 8.414.2.25 `TDSCDMAECIOThresh* setSignalStrengthInfo::pTDSCDMAECIOThresh`
- 8.414.2.26 `WORD* setSignalStrengthInfo::pTDSCDMARSCPDelta`
- 8.414.2.27 `TDSCDMARSCPThresh* setSignalStrengthInfo::pTDSCDMARSCPThresh`

- 8.414.2.28 **ULONG\*** `setSignalStrengthInfo::pTDSCDMARSSIDelta`
- 8.414.2.29 **TDSCDMARSSIThresh\*** `setSignalStrengthInfo::pTDSCDMARSSIThresh`
- 8.414.2.30 **ULONG\*** `setSignalStrengthInfo::pTDSCDMASINRDelta`
- 8.414.2.31 **TDSCDMASINRThresh\*** `setSignalStrengthInfo::pTDSCDMASINRThresh`
- 8.414.2.32 **WORD\*** `setSignalStrengthInfo::pWCDMAECIODelta`
- 8.414.2.33 **WCDMAECIOThresh\*** `setSignalStrengthInfo::pWCDMAECIOThresh`
- 8.414.2.34 **WORD\*** `setSignalStrengthInfo::pWCDMARSSIDelta`
- 8.414.2.35 **WCDMARSSIThresh\*** `setSignalStrengthInfo::pWCDMARSSIThresh`

## 8.415 SetSIPConfigReq Struct Reference

### Data Fields

- **WORD** \* `pSIPLocalPort`
- **ULONG** \* `pTimerSIPReg`
- **ULONG** \* `pSubscribeTimer`
- **ULONG** \* `pTimerT1`
- **ULONG** \* `pTimerT2`
- **ULONG** \* `pTimerTf`
- **BYTE** \* `pSigCompEnabled`

### 8.415.1 Detailed Description

This structure contains the SLQSSetSIPConfig request parameters.

#### Parameters

<i>pSIPLocalPort</i>	<ul style="list-style-type: none"> <li>• Primary call session control function SIP port number</li> </ul>
<i>pTimerSIPReg</i>	<ul style="list-style-type: none"> <li>• Initial SIP registration duration from the User equipment, in seconds</li> </ul>
<i>pSubscribeTimer</i>	<ul style="list-style-type: none"> <li>• Duration of the subscription by the UE for IMS registration notifications, in seconds</li> </ul>
<i>pTimerT1</i>	<ul style="list-style-type: none"> <li>• RTT estimate, in milliseconds</li> </ul>
<i>pTimerT2</i>	<ul style="list-style-type: none"> <li>• The maximum retransmit interval for non-invite requests and invite responses, in milliseconds</li> </ul>

<i>pTimerTf</i>	<ul style="list-style-type: none"> <li>• Non-invite transaction timeout timer, in milliseconds</li> </ul>
<i>pSigComp-Enabled</i>	<ul style="list-style-type: none"> <li>• Sig Comp Status <ul style="list-style-type: none"> <li>– TRUE - Enable</li> <li>– FALSE - Disable</li> </ul> </li> </ul>

## 8.415.2 Field Documentation

8.415.2.1 **BYTE\*** SetSIPConfigReq::pSigCompEnabled

8.415.2.2 **WORD\*** SetSIPConfigReq::pSIPLocalPort

8.415.2.3 **ULONG\*** SetSIPConfigReq::pSubscribeTimer

8.415.2.4 **ULONG\*** SetSIPConfigReq::pTimerSIPReg

8.415.2.5 **ULONG\*** SetSIPConfigReq::pTimerT1

8.415.2.6 **ULONG\*** SetSIPConfigReq::pTimerT2

8.415.2.7 **ULONG\*** SetSIPConfigReq::pTimerTf

## 8.416 SetSIPConfigResp Struct Reference

### Data Fields

- [BYTE \\*](#) [pSettingResp](#)

### 8.416.1 Detailed Description

This structure contains the SLQSSetSIPConfig response parameters.

#### Parameters

<i>pSettingResp</i>	<ul style="list-style-type: none"> <li>• Settings standard response type. A settings specific error code is returned when the standard response error type is QMI_ERR_CAUSE_CODE</li> </ul>
---------------------	---

## 8.416.2 Field Documentation

8.416.2.1 **BYTE\*** SetSIPConfigResp::pSettingResp

## 8.417 sGetDeviceSeriesResult Struct Reference

### Data Fields

- enum [eGobiDeviceSeries](#) [eDevice](#)

- [ULONG uResult](#)

### 8.417.1 Detailed Description

This structure contains the Device Series

Parameters

<i>eGobiDeviceSeries</i>	<ul style="list-style-type: none"> <li>• The number of device in the device series</li> </ul>
<i>uResult</i>	-eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

### 8.417.2 Field Documentation

8.417.2.1 enum eGobiDeviceSeries sGetDeviceSeriesResult::eDevice

8.417.2.2 ULONG sGetDeviceSeriesResult::uResult

## 8.418 sidNid Struct Reference

Data Fields

- [WORD nid](#)
- [WORD sid](#)

### 8.418.1 Detailed Description

This structure contains the parameters for SidNid Information

Parameters

<i>nid</i>	<ul style="list-style-type: none"> <li>• Network ID</li> </ul>
<i>sid</i>	<ul style="list-style-type: none"> <li>• System ID</li> </ul>

### 8.418.2 Field Documentation

8.418.2.1 WORD sidNid::nid

8.418.2.2 WORD sidNid::sid

## 8.419 sigInfo Struct Reference

Data Fields

- [RSSIThresh](#) \* [pRSSIThresh](#)
- [ECIOTThresh](#) \* [pECIOTThresh](#)
- [HRSINRThresh](#) \* [pHRSINRThresh](#)



- [LTESNRThresh](#) \* [pLTESNRThresh](#)
- [IOTthresh](#) \* [pIOTthresh](#)
- [RSRQThresh](#) \* [pRSRQThresh](#)
- [RSRPThresh](#) \* [pRSRPThresh](#)
- [LTESigRptCfg](#) \* [pLTESigRptCfg](#)

### 8.419.1 Detailed Description

This structure contains the 3gpp Configuration Item information.

#### Parameters

<i>pRSSIThresh</i>	<ul style="list-style-type: none"> <li>• RSSI threshold List</li> <li>• See <a href="#">RSSIThresh</a> for more details</li> </ul>
<i>pECIOTthresh</i>	<ul style="list-style-type: none"> <li>• ECIO Threshold List</li> <li>• See <a href="#">ECIOTthresh</a> for more details</li> </ul>
<i>pHDRSINR- Thresh</i>	<ul style="list-style-type: none"> <li>• HDR SINR Threshold List</li> <li>• See <a href="#">HDRSINRThresh</a> for more details</li> </ul>
<i>pLTESNR- Thresh</i>	<ul style="list-style-type: none"> <li>• LTE SNR Threshold List</li> <li>• See <a href="#">LTESNRThresh</a> for more details</li> </ul>
<i>pIOTthresh</i>	<ul style="list-style-type: none"> <li>• IO Threshold List</li> <li>• See <a href="#">IOTthresh</a> for more details</li> </ul>
<i>pRSRQThresh</i>	<ul style="list-style-type: none"> <li>• RSRQ Threshold List</li> <li>• See <a href="#">RSRQThresh</a> for more details</li> </ul>
<i>pRSRPThresh</i>	<ul style="list-style-type: none"> <li>• RSRP Threshold List</li> <li>• See <a href="#">RSRPThresh</a> for more details</li> </ul>
<i>pLTESigRptCfg</i>	<ul style="list-style-type: none"> <li>• LTE signal report config</li> <li>• See <a href="#">LTESigRptCfg</a> for more details</li> </ul>

### 8.419.2 Field Documentation

#### 8.419.2.1 ECIOTthresh\* sigInfo::pECIOTthresh

8.419.2.2 HDRSINRThresh\* sigInfo::pHDRSINRThresh

8.419.2.3 IOTresh\* sigInfo::pIOTresh

8.419.2.4 LTESigRptCfg\* sigInfo::pLTESigRptCfg

8.419.2.5 LTESNRThresh\* sigInfo::pLTESNRThresh

8.419.2.6 RSRPThresh\* sigInfo::pRSRPThresh

8.419.2.7 RSRQThresh\* sigInfo::pRSRQThresh

8.419.2.8 RSSIThresh\* sigInfo::pRSSIThresh

## 8.420 signalInfo Struct Reference

### Data Fields

- [BYTE signalType](#)
- [BYTE alertPitch](#)
- [BYTE signal](#)

### 8.420.1 Detailed Description

This structure contains Signal Information

#### Parameters

<i>signalType</i>	<ul style="list-style-type: none"> <li>• Call identifier for the call.</li> </ul>
<i>alertPitch</i>	<ul style="list-style-type: none"> <li>• Signal Information</li> </ul>
<i>signal</i>	<ul style="list-style-type: none"> <li>• Caller ID Information</li> </ul>

### 8.420.2 Field Documentation

8.420.2.1 BYTE sigInfo::alertPitch

8.420.2.2 BYTE sigInfo::signal

8.420.2.3 BYTE sigInfo::signalType

## 8.421 SignalStrengthDataType Struct Reference

### Data Fields

- [BYTE thresholdsSize](#)
- [INT8 thresholds](#) [5]

### 8.421.1 Field Documentation

8.421.1.1 INT8 SignalStrengthDataType::thresholds[5]

8.421.1.2 BYTE SignalStrengthDataType::thresholdsSize

## 8.422 slotInfo Struct Reference

### Data Fields

- [BYTE cardState](#)
- [BYTE upinState](#)
- [BYTE upinRetries](#)
- [BYTE upukRetries](#)
- [BYTE errorState](#)
- [BYTE numApp](#)
- [appStatus AppStatus](#) [10]

### 8.422.1 Detailed Description

This structure contains information about the SLOTS present.

#### Parameters

<i>cardState</i>	<ul style="list-style-type: none"><li>• Indicates the state of the card for each slot.<ul style="list-style-type: none"><li>– 0 - Absent</li><li>– 1 - Present</li><li>– 2 - Error</li></ul></li></ul>
<i>upinState</i>	<ul style="list-style-type: none"><li>• Indicates the state of UPIN.<ul style="list-style-type: none"><li>– 0 - Unknown</li><li>– 1 - Enabled and not verified</li><li>– 2 - Enabled and verified</li><li>– 3 - Disabled</li><li>– 4 - Blocked</li><li>– 5 - Permanently blocked</li><li>– 0xFF - Not Available</li></ul></li></ul>

<i>upinRetries</i>	<ul style="list-style-type: none"> <li>Indicates the number of retries remaining to verify the UPIN.</li> <li>If 0xFF, information not available.</li> </ul>
<i>upukRetries</i>	<ul style="list-style-type: none"> <li>Indicates the number of retries remaining to unblock the UPIN.</li> <li>If 0xFF, information not available.</li> </ul>
<i>errorState</i>	<ul style="list-style-type: none"> <li>Indicates the reason for the card error, and is valid only when the card state is Error <ul style="list-style-type: none"> <li>0 - Unknown</li> <li>1 - Power down</li> <li>2 - Poll error</li> <li>3 - No ATR received</li> <li>4 - Volt mismatch</li> <li>5 - Parity error</li> <li>6 - Unknown; possibly removed</li> <li>7 - Card returned technical problems</li> <li>0xFF - Not Available</li> </ul> </li> <li>Other values are possible and reserved for future use.</li> <li>When an unknown value is received, it is to be handled as "Unknown".</li> </ul>
<i>numApp</i>	<ul style="list-style-type: none"> <li>Indicates the number of applications available on the card.</li> <li>The following block is repeated for each application. i.e. AppStatus.</li> <li>If zero(0) then no AppStatus information exists.</li> </ul>
<i>AppStatus[ MAX_ _NO_OF_APPL- ICATIONS]</i>	<ul style="list-style-type: none"> <li>See <a href="#">appStatus</a> for more information.</li> </ul>

## 8.422.2 Field Documentation

### 8.422.2.1 appStatus slotInfo::AppStatus[10]

### 8.422.2.2 BYTE slotInfo::cardState

### 8.422.2.3 BYTE slotInfo::errorState

### 8.422.2.4 BYTE slotInfo::numApp

### 8.422.2.5 BYTE slotInfo::upinRetries

### 8.422.2.6 BYTE slotInfo::upinState

### 8.422.2.7 BYTE slotInfo::upukRetries

## 8.423 slqsautoconnect Struct Reference

### Data Fields

- [BOOL action](#)
- [BYTE acsetting](#)
- [BYTE acroamsetting](#)

### 8.423.1 Detailed Description

structure contains autoconnect settings parameters

#### Parameters

<i>action</i>	<ul style="list-style-type: none"> <li>• 0 - get autoconnect settings</li> <li>• 1 - set autoconnect settings</li> </ul>
<i>acsetting</i>	<ul style="list-style-type: none"> <li>• Current autoconnect setting:             <ul style="list-style-type: none"> <li>– 0x00 - Autoconnect disabled</li> <li>– 0x01 - Autoconnect enabled</li> <li>– 0x02 - Autoconnect paused (resume on powercycle)</li> </ul> </li> </ul>
<i>acroamsetting</i>	<ul style="list-style-type: none"> <li>• Current autoconnect roaming status             <ul style="list-style-type: none"> <li>– 0x00 - Autoconnect always allowed</li> <li>– 0x01 - Autoconnect while in home service area only</li> </ul> </li> </ul>

### 8.423.2 Field Documentation

8.423.2.1 **BYTE** slqsautoconnect::acroamsetting

8.423.2.2 **BYTE** slqsautoconnect::acsetting

8.423.2.3 **BOOL** slqsautoconnect::action

## 8.424 SLQSDDeleteProfileParams Struct Reference

### Data Fields

- [BYTE profileType](#)
- [BYTE profileIndex](#)

### 8.424.1 Detailed Description

This structure contains the information about the profile to be deleted.

## Parameters

<i>profileType</i>	<ul style="list-style-type: none"> <li>Identifies the type of profile <ul style="list-style-type: none"> <li>0x00 – 3GPP</li> </ul> </li> <li>Note: Deletion of 3GPP2 profiles is not supported.</li> </ul>
<i>profileIndex</i>	<ul style="list-style-type: none"> <li>Index of the configured profile to be deleted <ul style="list-style-type: none"> <li>Value from 1-16, inclusive.</li> </ul> </li> </ul>

## 8.424.2 Field Documentation

8.424.2.1 BYTE SLQSDelProfileParams::profileIndex

8.424.2.2 BYTE SLQSDelProfileParams::profileType

## 8.425 slqsfwinfo\_s Struct Reference

## Data Fields

- CHAR [modelid\\_str](#) [20]
- CHAR [bootversion\\_str](#) [85]
- CHAR [appversion\\_str](#) [85]
- CHAR [sku\\_str](#) [15]
- CHAR [packageid\\_str](#) [85]
- CHAR [carrier\\_str](#) [20]
- CHAR [priversion\\_str](#) [16]
- CHAR [cur\\_carr\\_name](#) [17]
- CHAR [cur\\_carr\\_rev](#) [13]

## 8.425.1 Detailed Description

SPKG CWE firmware image info structure

## Parameters

<i>modelid_str</i>	<ul style="list-style-type: none"> <li>device model identifier string</li> </ul>
<i>bootversion_str</i>	<ul style="list-style-type: none"> <li>firmware boot version string</li> </ul>
<i>appversion_str</i>	<ul style="list-style-type: none"> <li>firmware application version string</li> </ul>

<i>sku_str</i>	<ul style="list-style-type: none"> <li>• SKU(PRI) string</li> </ul>
<i>packageid_str</i>	<ul style="list-style-type: none"> <li>• package identifier string</li> </ul>
<i>carrier_str</i>	<ul style="list-style-type: none"> <li>• carrier string</li> <li>• See <a href="#">qaGobiApiTableCarrierCodes.h</a> for carrier codes</li> </ul>
<i>priversion_str</i>	<ul style="list-style-type: none"> <li>• PRI version string</li> </ul>
<i>cur_carr_name</i>	<ul style="list-style-type: none"> <li>• Current PRI Carrier Name</li> </ul>
<i>cur_carr_rev</i>	<ul style="list-style-type: none"> <li>• Current PRI Carrier Revision</li> </ul>

## 8.425.2 Field Documentation

8.425.2.1 **CHAR** slqsfwinfo\_s::appversion\_str[85]

8.425.2.2 **CHAR** slqsfwinfo\_s::bootversion\_str[85]

8.425.2.3 **CHAR** slqsfwinfo\_s::carrier\_str[20]

8.425.2.4 **CHAR** slqsfwinfo\_s::cur\_carr\_name[17]

8.425.2.5 **CHAR** slqsfwinfo\_s::cur\_carr\_rev[13]

8.425.2.6 **CHAR** slqsfwinfo\_s::modelid\_str[20]

8.425.2.7 **CHAR** slqsfwinfo\_s::packageid\_str[85]

8.425.2.8 **CHAR** slqsfwinfo\_s::priversion\_str[16]

8.425.2.9 **CHAR** slqsfwinfo\_s::sku\_str[15]

## 8.426 SlqsNas3GppNetworkInfo Struct Reference

### Data Fields

- [WORD MCC](#)
- [WORD MNC](#)
- [ULONG InUse](#)
- [ULONG Roaming](#)
- [ULONG Forbidden](#)
- [ULONG Preferred](#)
- [CHAR Description](#) [255]

### 8.426.1 Detailed Description

Contain the 3GPP network information.



## Parameters

<i>MCC</i>	<ul style="list-style-type: none"> <li>• Mobile Country Code</li> </ul>
<i>MNC</i>	<ul style="list-style-type: none"> <li>• Mobile Network Code</li> </ul>
<i>InUse</i>	<ul style="list-style-type: none"> <li>• Is the Network the current serving Network <ul style="list-style-type: none"> <li>– 0 - Unknown</li> <li>– 1 - Current serving network</li> <li>– 2 - Not current serving network, available</li> </ul> </li> </ul>
<i>Roaming</i>	<ul style="list-style-type: none"> <li>• Home/Roam Status of the Network <ul style="list-style-type: none"> <li>– 0 - Unknown</li> <li>– 1 - Home</li> <li>– 2 - Roam</li> </ul> </li> </ul>
<i>Forbidden</i>	<ul style="list-style-type: none"> <li>• Is the Network in the forbidden network list <ul style="list-style-type: none"> <li>– 0 - Unknown</li> <li>– 1 - Forbidden</li> <li>– 2 - Not Forbidden</li> </ul> </li> </ul>
<i>Preferred</i>	<ul style="list-style-type: none"> <li>• Is the Network in the Preferred network list <ul style="list-style-type: none"> <li>– 0 - Unknown</li> <li>– 1 - Preferred</li> <li>– 2 - Not Preferred</li> </ul> </li> </ul>
<i>Description</i>	<ul style="list-style-type: none"> <li>• Network Name/Description</li> <li>• This is a NULL terminated string.</li> </ul>

## 8.426.2 Field Documentation

8.426.2.1 CHAR SlqsNas3GppNetworkInfo::Description[255]

8.426.2.2 ULONG SlqsNas3GppNetworkInfo::Forbidden

8.426.2.3 ULONG SlqsNas3GppNetworkInfo::InUse

8.426.2.4 WORD SIqsNas3GppNetworkInfo::MCC

8.426.2.5 WORD SIqsNas3GppNetworkInfo::MNC

8.426.2.6 ULONG SIqsNas3GppNetworkInfo::Preferred

8.426.2.7 ULONG SIqsNas3GppNetworkInfo::Roaming

## 8.427 SIqsNas3GppNetworkRAT Struct Reference

### Data Fields

- [WORD MCC](#)
- [WORD MNC](#)
- [BYTE RAT](#)

### 8.427.1 Detailed Description

Contain the 3GPP radio access technology information.

#### Parameters

<i>MCC</i>	<ul style="list-style-type: none"><li>• Mobile Country Code</li></ul>
<i>MNC</i>	<ul style="list-style-type: none"><li>• Mobile Network Code</li></ul>
<i>RAT</i>	<ul style="list-style-type: none"><li>• Radio Access Technology<ul style="list-style-type: none"><li>– 0x04 - GERAN</li><li>– 0x05 - UMTS</li></ul></li></ul>

### 8.427.2 Field Documentation

8.427.2.1 WORD SIqsNas3GppNetworkRAT::MCC

8.427.2.2 WORD SIqsNas3GppNetworkRAT::MNC

8.427.2.3 BYTE SIqsNas3GppNetworkRAT::RAT

## 8.428 SIqsNasPcsDigit Struct Reference

### Data Fields

- [WORD MCC](#)
- [WORD MNC](#)
- [BYTE includes\\_pcs\\_digit](#)

### 8.428.1 Detailed Description

Contain the PCS Digit information

## Parameters

<i>MCC</i>	<ul style="list-style-type: none"> <li>• Mobile Country Code</li> </ul>
<i>MNC</i>	<ul style="list-style-type: none"> <li>• Mobile Network Code</li> </ul>
<i>includes_pcs_-digit</i>	<ul style="list-style-type: none"> <li>• this field is use to interpret the length of corresponding MNC reported</li> <li>• 0x01 - MNC is a three-digit value</li> <li>• 0x00 - MNC is a two-digit value</li> </ul>

## 8.428.2 Field Documentation

8.428.2.1 BYTE SlqsNasPcsDigit::includes\_pcs\_digit

8.428.2.2 WORD SlqsNasPcsDigit::MCC

8.428.2.3 WORD SlqsNasPcsDigit::MNC

## 8.429 slqssendasyncsmsparams\_s Struct Reference

## Data Fields

- [ULONG messageFormat](#)
- [ULONG messageSize](#)
- [BYTE \\* pMessage](#)
- [BYTE \\* pForceOnDC](#)
- [BYTE \\* pServiceOption](#)
- [BYTE \\* pFollowOnDC](#)
- [BYTE \\* pLinktimer](#)
- [BYTE \\* pSmsOnlms](#)
- [BYTE \\* pRetryMessage](#)
- [ULONG \\* pRetryMessageld](#)
- [ULONG \\* pUserData](#)

## 8.429.1 Detailed Description

This structure contains SMS parameters

## Parameters

<i>messageFormat</i>	<ul style="list-style-type: none"> <li>• Message format</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0 - CDMA (IS-637B)</li> <li>– 1 - 5 (Reserved)</li> <li>– 6 - GSM/WCDMA PP</li> </ul> </li> </ul>
<i>messageSize</i>	<ul style="list-style-type: none"> <li>• The length of the message contents in bytes</li> </ul>
<i>pMessage</i>	<ul style="list-style-type: none"> <li>• The message contents</li> </ul>
<i>pForceOnDC</i>	<ul style="list-style-type: none"> <li>• Force the message to be sent on the CDMA dedicated channel.</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0x00 - Do not care about the channel on which the message is sent</li> <li>– 0x01 - Request to send the message over the dedicated channel</li> </ul> </li> </ul>
<i>pServiceOption</i>	<ul style="list-style-type: none"> <li>• Service option:</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0x00 - SO_AUTO - AUTO (choose the best service option)</li> <li>– 0x06 - SO_6 - Service option 6</li> <li>– 0x0E - SO_14 - Service option 14</li> </ul> </li> </ul>
<i>pFollowOnDC</i>	<ul style="list-style-type: none"> <li>• Flag to request not to disconnect the CDMA dedicated channel after the send operation is complete.</li> <li>• This TLV can be included if more messages are expected to follow.</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0x01 - FOLLOW_ON_DC_ON - On (don't disconnect after send operation) Any value other than 0x01 is treated as an absence of this TLV.</li> </ul> </li> </ul>

<i>pLinktimer</i>	<ul style="list-style-type: none"> <li>Keeps the GW SMS link open for the specified number of seconds; can be enabled if more messages are expected to follow</li> </ul>
<i>pSmsOnIms</i>	<ul style="list-style-type: none"> <li>Indicates whether the message is to be sent on IMS.</li> <li>Values: <ul style="list-style-type: none"> <li>0x00 - Message is not to be sent on IMS</li> <li>0x01 - Message is to be sent on IMS</li> <li>0x02 to 0xFF - Reserved</li> </ul> </li> </ul>
<i>pRetryMessage</i>	<ul style="list-style-type: none"> <li>Indicates this message is a retry message.</li> <li>Values: <ul style="list-style-type: none"> <li>0x01 - WMS_MESSAGE_IS_A_RETRY - Message is a retry message Note: Any value other than 0x01 in this field is treated as an absence of this TLV.</li> </ul> </li> </ul>
<i>pRetryMessage-Id</i>	<ul style="list-style-type: none"> <li>Message ID to be used in the retry message.</li> <li>The message ID specified here is used instead of the message ID encoded in the raw message.</li> </ul>
<i>pUserData</i>	<ul style="list-style-type: none"> <li>Enables the control point to associate the request with the corresponding indication.</li> <li>The control point might send numerous requests.</li> <li>This TLV will help the control point to identify the request for which the received indication belongs.</li> </ul>

## 8.429.2 Field Documentation

8.429.2.1 **ULONG** `slqssendasyncsmsparams_s::messageFormat`

8.429.2.2 **ULONG** `slqssendasyncsmsparams_s::messageSize`

8.429.2.3 **BYTE\*** `slqssendasyncsmsparams_s::pFollowOnDC`

8.429.2.4 **BYTE\*** `slqssendasyncsmsparams_s::pForceOnDC`

8.429.2.5 **BYTE\*** `slqssendasyncsmsparams_s::pLinktimer`

8.429.2.6 **BYTE\*** `slqssendasyncsmsparams_s::pMessage`

8.429.2.7 **BYTE\*** `slqssendasyncsmsparams_s::pRetryMessage`

8.429.2.8 **ULONG\*** `slqssendasyncsmsparams_s::pRetryMessageId`

8.429.2.9 **BYTE\*** slqssendasyncsmsparams\_s::pServiceOption

8.429.2.10 **BYTE\*** slqssendasyncsmsparams\_s::pSmsOnlms

8.429.2.11 **ULONG\*** slqssendasyncsmsparams\_s::pUserData

## 8.430 slqssendsmsparams\_s Struct Reference

### Data Fields

- [ULONG messageFormat](#)
- [ULONG messageSize](#)
- [BYTE \\* pMessage](#)
- [USHORT messageID](#)
- [ULONG messageFailureCode](#)
- [BYTE \\* pLinktimer](#)

### 8.430.1 Detailed Description

This structure contains SMS parameters

#### Parameters

<i>messageFormat</i>	<ul style="list-style-type: none"> <li>• Message format               <ul style="list-style-type: none"> <li>– 0 - CDMA (IS-637B)</li> <li>– 1 - 5 (Reserved)</li> <li>– 6 - GSM/WCDMA PP</li> </ul> </li> </ul>
<i>messageSize</i>	<ul style="list-style-type: none"> <li>• The length of the message contents in bytes</li> </ul>
<i>pMessage[IN]</i>	<ul style="list-style-type: none"> <li>• The message contents in PDU format contains SMS header and payload message</li> </ul>
<i>pMessageID</i>	<ul style="list-style-type: none"> <li>• message reference ID</li> </ul>
<i>pMessage-FailureCode</i>	<ul style="list-style-type: none"> <li>• message failure code. If cause code is not provided, then value will be 0xFFFFFFFF</li> </ul>
<i>pLinktimer</i>	<ul style="list-style-type: none"> <li>• Keeps the GW SMS link open for the specified number of seconds; can be enabled if more messages are expected to follow</li> </ul>

### 8.430.2 Field Documentation

8.430.2.1 **ULONG** slqssendsmsparams\_s::messageFailureCode

8.430.2.2 **ULONG** slqssendsmsparams\_s::messageFormat

8.430.2.3 **USHORT** slqssendsmsparams\_s::messageID

8.430.2.4 **ULONG** slqssendsmsparams\_s::messageSize

8.430.2.5 **BYTE\*** slqssendsmsparams\_s::pLinktimer

8.430.2.6 **BYTE\*** slqssendsmsparams\_s::pMessage

## 8.431 slqsSessionStateInfo Struct Reference

### Data Fields

- [qaQmiInterfaceInfo](#) \* [pQmiInterfaceInfo](#)
- **ULONG** [reconfiguration\\_required](#)
- **ULONG** [state](#)
- **ULONG** [sessionEndReason](#)

### 8.431.1 Detailed Description

Contains the session state information and information about the interface

#### Parameters

<i>pQmiInterface-Info</i>	<ul style="list-style-type: none"> <li>• See <a href="#">qaQmiInterfaceInfo</a> for more information</li> </ul>
<i>state</i>	<ul style="list-style-type: none"> <li>• Current Link Status               <ul style="list-style-type: none"> <li>– 1 Disconnected</li> <li>– 2 Connected</li> <li>– 3 Suspended (Unsupported)</li> <li>– 4 Authenticating</li> </ul> </li> </ul>
<i>reconfiguration_-required</i>	<ul style="list-style-type: none"> <li>• Indicates if host needs to be reconfigured               <ul style="list-style-type: none"> <li>– 0 No need to reconfigure</li> <li>– 1 Reconfiguration required</li> </ul> </li> </ul>
<i>sessionEnd-Reason</i>	<ul style="list-style-type: none"> <li>• See <a href="#">qaGobiApiTableCallEndReasons.h</a> for Call End Reason</li> </ul>

### 8.431.2 Field Documentation

8.431.2.1 **qaQmiInterfaceInfo**\* slqsSessionStateInfo::pQmiInterfaceInfo

8.431.2.2 **ULONG** slqsSessionStateInfo::reconfiguration\_required

8.431.2.3 **ULONG** slqsSessionStateInfo::sessionEndReason



8.431.2.4    `ULONG` slqsSessionStateInfo::state

## 8.432    slqsSignalStrengthInfo Struct Reference

### Data Fields

- `USHORT` signalStrengthReqMask
- `USHORT` rxSignalStrengthListLen
- `struct rxSignalStrengthListElement` rxSignalStrengthList [18]
- `USHORT` ecioListLen
- `struct ecioListElement` ecioList [18]
- `INT32` lo
- `BYTE` sinr
- `USHORT` errorRateListLen
- `struct errorRateListElement` errorRateList [18]
- `struct rsrqInformation` rsrqInfo
- `SHORT` ltesnr
- `SHORT` ltersrp

### 8.432.1    Detailed Description

This structure contains the Signal Strength Information

## Parameters

<i>rfInfoReqMask</i>	<ul style="list-style-type: none"> <li>Request Mask           <ul style="list-style-type: none"> <li>Request additional signal information for: Bit 0 - RSSI Information bit Valid values are: 0 - Do Not Request Additional Info for RSSI 1 - Request Additional Info for RSSI</li> <li>Bit 1 - ECIO Information bit Valid values are: 0 - Do Not Request Additional Info for ECIO 1 - Request Additional Info for ECIO</li> <li>Bit 2 - IO Information bit Valid values are: 0 - Do Not Request Additional Info for IO 1 - Request Additional Info for IO</li> <li>Bit 3 - SINR Information bit Valid values are: 0 - Do Not Request Additional Info for SINR 1 - Request Additional Info for SINR</li> <li>Bit 4 - ERROR RATE Information bit Valid values are: 0 - Do Not Request Additional Info for Error Rate 1 - Request Additional Info for Error Rate</li> <li>Bit 5 - RSRQ Information bit Valid values are: 0 - Do Not Request Additional Info for RSRQ 1 - Request Additional Info for RSRQ</li> <li>Bit 6 - LTE SNR information bit Valid values are: 0 - Do not request additional information for LTE SNR 1 - Request additional information for LTE SNR</li> <li>Bit 7 - LTE RSRP Information bit Valid values are: 0 - Do not request additional information for LTE RSRP 1 - Request additional information for LTE RSRP</li> </ul> </li> </ul>
<i>rxSignal-StrengthListLen</i>	<ul style="list-style-type: none"> <li>Number of elements in Receive Signal Strength List</li> </ul>
<i>rxSignal-StrengthList</i>	<ul style="list-style-type: none"> <li>See <a href="#">rxSignalStrengthListElement</a> for more information</li> </ul>
<i>ecioListLen</i>	<ul style="list-style-type: none"> <li>Number of elements in ECIO List</li> </ul>

<i>ecioList</i>	<ul style="list-style-type: none"> <li>• See <a href="#">ecioListElement</a> for more information</li> </ul>
<i>lo</i>	<ul style="list-style-type: none"> <li>• Received lo in dBm; IO is only applicable for 1xEV-DO</li> </ul>
<i>sinr</i>	<ul style="list-style-type: none"> <li>• SINR level           <ul style="list-style-type: none"> <li>– SINR is only applicable for 1xEV-DO; valid levels are 0 to 8 where maximum value for 0 - SINR_LEVEL_0 is -9 dB 1 - SINR_LEVEL_1 is -6 dB 2 - SINR_LEVEL_2 is -4.5 dB 3 - SINR_LEVEL_3 is -3 dB 4 - SINR_LEVEL_4 is -2 dB 5 - SINR_LEVEL_5 is +1 dB 6 - SINR_LEVEL_6 is +3 dB 7 - SINR_LEVEL_7 is +6 dB 8 - SINR_LEVEL_8 is +9 dB</li> </ul> </li> </ul>
<i>errorRateListLen</i>	<ul style="list-style-type: none"> <li>• Number of elements in Error Rate List</li> </ul>
<i>errorRateList</i>	<ul style="list-style-type: none"> <li>• See <a href="#">errorRateListElement</a> for more information</li> </ul>
<i>rsrqInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">rsrqInformation</a> for more information</li> </ul>
<i>ltesnr</i>	<ul style="list-style-type: none"> <li>• LTE SNR level as a scaled integer in units of 0.1 dB; e.g., -16 dB has a value of -160 and 24.6 dB has a value of 246. LTE SNR is included only when the current serving system is LTE</li> </ul>
<i>ltersrp</i>	<ul style="list-style-type: none"> <li>• LTE SNR level as a scaled integer in units of 0.1 dB; e.g., -16 dB has a value of -160 and 24.6 dB has a value of 246. LTE SNR is included only when the current serving system is LTE</li> </ul>

## 8.432.2 Field Documentation

8.432.2.1 struct `ecioListElement` `slqsSignalStrengthInfo::ecioList[18]`

8.432.2.2 USHORT `slqsSignalStrengthInfo::ecioListLen`

8.432.2.3 struct `errorRateListElement` `slqsSignalStrengthInfo::errorRateList[18]`

8.432.2.4 USHORT `slqsSignalStrengthInfo::errorRateListLen`

8.432.2.5 INT32 `slqsSignalStrengthInfo::lo`

8.432.2.6 SHORT `slqsSignalStrengthInfo::ltersrp`

8.432.2.7 SHORT `slqsSignalStrengthInfo::ltesnr`

8.432.2.8 struct `rsrqInformation` `slqsSignalStrengthInfo::rsrqInfo`

8.432.2.9 struct rxSignalStrengthListElement slqsSignalStrengthInfo::rxSignalStrengthList[18]

8.432.2.10 USHORT slqsSignalStrengthInfo::rxSignalStrengthListLen

8.432.2.11 USHORT slqsSignalStrengthInfo::signalStrengthReqMask

8.432.2.12 BYTE slqsSignalStrengthInfo::sinr

## 8.433 SLQSSignalStrengthsIndReq Struct Reference

### Data Fields

- [BYTE rxSignalStrengthDelta](#)
- [BYTE ecioDelta](#)
- [BYTE ioDelta](#)
- [BYTE sinrDelta](#)
- [BYTE rsrqDelta](#)
- [BYTE ecioThresholdListLen](#)
- [SHORT ecioThresholdList \[10\]](#)
- [BYTE sinrThresholdListLen](#)
- [BYTE sinrThresholdList \[5\]](#)
- [WORD lteSnrDelta](#)
- [BYTE lteRsrpDelta](#)

### 8.433.1 Detailed Description

Structure for storing the input parameters passed for SLQSSetSignalStrengthsCallback by the user.

#### Parameters

<i>rxSignalStrengthDelta</i>	<ul style="list-style-type: none"> <li>RSSI delta(in dBm) at which an event report indication, including the current RSSI, will be sent to the requesting control point.</li> </ul>
<i>ecioDelta</i>	<ul style="list-style-type: none"> <li>ECIO delta at which an event report indication, ecioDelta including the current ECIO, will be sent to the requesting control point.</li> <li>ECIO delta is an unsigned 1 byte value that increments in negative 0.5 dBm, e.g., ecio_delta of 2 means a change of -1 dBm.</li> </ul>
<i>ioDelta</i>	<ul style="list-style-type: none"> <li>IO delta (in dBm) at which an event report indication, ioDelta including the current IO, will be sent to the requesting control point.</li> </ul>
<i>sinrDelta</i>	<ul style="list-style-type: none"> <li>SINR delta level at which an event report indication, sinrDelta including the current SINR, will be sent to the requesting control point.</li> </ul>

<i>rsrqDelta</i>	<ul style="list-style-type: none"> <li>• RSRQ delta level at which an event report indication, including the current RSRQ, will be sent to the requesting control point.</li> </ul>
<i>ecioThreshold-ListLen</i>	<ul style="list-style-type: none"> <li>• Number of elements in the ECIO threshold list.</li> </ul>
<i>ecioThreshold-List</i>	<ul style="list-style-type: none"> <li>• A sequence of thresholds delimiting Ecio event reporting bands. Every time a new Ecio value crosses a threshold value, an event report indication message with the new ECIO value is sent to the requesting control point. For this field: <ul style="list-style-type: none"> <li>– Maximum number of threshold values is 10</li> <li>– At least one value must be specified.</li> </ul> </li> </ul>
<i>sinrThreshold-ListLen</i>	<ul style="list-style-type: none"> <li>• Number of elements in the SINR threshold list.</li> </ul>
<i>sinrThreshold-List</i>	<ul style="list-style-type: none"> <li>• A sequence of thresholds delimiting SINR event reporting bands. Every time a new S-INR value crosses a threshold value, an event report indication message with the new sinr value is sent to the requesting control point. For this field: <ul style="list-style-type: none"> <li>– Maximum number of threshold values is 5</li> <li>– At least one value must be specified.</li> </ul> </li> </ul>
<i>lteSnrdelta</i>	<ul style="list-style-type: none"> <li>• LTE SNR delta level at which an event report indication, including the current SNR, will be sent to the requesting control point. LTE SNR delta level is an unsigned 2 byte value, representing the delta in units of 0.1 dB, e.g., lte_snr_delta of 3 means a change 0.3dB.</li> </ul>
<i>lteRsrpdelta</i>	<ul style="list-style-type: none"> <li>• -LTE RSRP delta level at which an event report -indication, including the current RSRP, will be sent -to the requesting control point. LTE RSRP delta -level is an unsigned 1 byte value, representing the -delta in dB.</li> </ul>

## Note

None

## 8.433.2 Field Documentation

8.433.2.1 BYTE SLQSSignalStrengthsIndReq::ecioDelta

8.433.2.2 SHORT SLQSSignalStrengthsIndReq::ecioThresholdList[10]

8.433.2.3 BYTE SLQSSignalStrengthsIndReq::ecioThresholdListLen

8.433.2.4 BYTE SLQSSignalStrengthsIndReq::ioDelta

8.433.2.5 BYTE SLQSSignalStrengthsIndReq::lteRsrpDelta

8.433.2.6 WORD SLQSSignalStrengthsIndReq::lteSnrDelta

8.433.2.7 **BYTE** SLQSSignalStrengthsIndReq::rsrqDelta

8.433.2.8 **BYTE** SLQSSignalStrengthsIndReq::rxSignalStrengthDelta

8.433.2.9 **BYTE** SLQSSignalStrengthsIndReq::sinrDelta

8.433.2.10 **BYTE** SLQSSignalStrengthsIndReq::sinrThresholdList[5]

8.433.2.11 **BYTE** SLQSSignalStrengthsIndReq::sinrThresholdListLen

## 8.434 SLQSSignalStrengthsInformation Struct Reference

### Data Fields

- struct [rxSignalStrengthListElement](#) rxSignalStrengthInfo
- struct [ecioListElement](#) ecioInfo
- [ULONG](#) io
- [BYTE](#) sinr
- struct [errorRateListElement](#) errorRateInfo
- struct [rsrqInformation](#) rsrqInfo
- struct [lteSnrinformation](#) lteSnrinfo
- struct [lteRsrpinformation](#) lteRsrpinfo

### 8.434.1 Detailed Description

Structure for Received Signal Strength Information.

#### Parameters

<i>rxSignal- StrengthInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">rxSignalStrengthListElement</a> for more information.</li> </ul>
<i>ecioInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">ecioListElement</a> for more information.</li> </ul>
<i>io</i>	<ul style="list-style-type: none"> <li>• Received IO in dBm; IO is only applicable for 1xEV-DO.</li> </ul>
<i>sinr</i>	<ul style="list-style-type: none"> <li>• SINR level           <ul style="list-style-type: none"> <li>– SINR is only applicable for 1xEV-DO; valid levels are 0 to 8 where maximum value for 0 - SINR_LEVEL_0 is -9 dB 1 - SINR_LEVEL_1 is -6 dB 2 - SINR_LEVEL_2 is -4.5 dB 3 - SINR_LEVEL_3 is -3 dB 4 - SINR_LEVEL_4 is -2 dB 5 - SINR_LEVEL_5 is +1 dB 6 - SINR_LEVEL_6 is +3 dB 7 - SINR_LEVEL_7 is +6 dB 8 - SINR_LEVEL_8 is +9 dB</li> </ul> </li> </ul>

<i>errorRateInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">errorRateListElement</a> for more information.</li> </ul>
<i>rsrqInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">rsrqInformation</a> for more information.</li> </ul>
<i>lteSnrinfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">lteSnrinformation</a> for more information.</li> </ul>
<i>lteRsrpinfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">lteRsrpinformation</a> for more information.</li> </ul>

**Note**

None

**8.434.2 Field Documentation**

8.434.2.1 struct ecioListElement SLQSSignalStrengthsInformation::ecioInfo

8.434.2.2 struct errorRateListElement SLQSSignalStrengthsInformation::errorRateInfo

8.434.2.3 ULONG SLQSSignalStrengthsInformation::io

8.434.2.4 struct lteRsrpinformation SLQSSignalStrengthsInformation::lteRsrpinfo

8.434.2.5 struct lteSnrinformation SLQSSignalStrengthsInformation::lteSnrinfo

8.434.2.6 struct rsrqInformation SLQSSignalStrengthsInformation::rsrqInfo

8.434.2.7 struct rxSignalStrengthListElement SLQSSignalStrengthsInformation::rxSignalStrengthInfo

8.434.2.8 BYTE SLQSSignalStrengthsInformation::sinr

**8.435 slqsWdsEventInfo Struct Reference****Data Fields**

- [qaQmiInterfaceInfo](#) \* [pQmiInterfaceInfo](#)
- ULONG \* [pDormancyStatus](#)
- ULONG \* [pDataBearer](#)
- ULONG \* [pPacketsCountTX](#)
- ULONG \* [pPacketsCountRX](#)
- ULONGLONG \* [pTotalBytesTX](#)
- ULONGLONG \* [pTotalBytesRX](#)

**8.435.1 Detailed Description**

Contains the WDS event information and information about the interface

## Parameters

<i>pQmiInterface-Info</i>	<ul style="list-style-type: none"> <li>• See <a href="#">qaQmiInterfaceInfo</a> for more information</li> </ul>
<i>pDataBearer,-</i>	<p>Data bearer technology (NULL if not present)</p> <ul style="list-style-type: none"> <li>• 0x00 - Indicates that this field is ignored</li> <li>• 0x01 - CDMA 1X</li> <li>• 0x02 - EV-DO Rev 0</li> <li>• 0x03 - GPRS</li> <li>• 0x04 - WCDMA</li> <li>• 0x05 - EV-DO Rev A</li> <li>• 0x06 - EDGE</li> <li>• 0x07 - HSDPA and WCDMA</li> <li>• 0x08 - WCDMA and HSUPA</li> <li>• 0x09 - HSDPA and HSUPA</li> <li>• 0x0A - LTE</li> <li>• 0x0B - EV-DO Rev A EHRPD</li> <li>• 0x0C - HSDPA+ and WCDMA</li> <li>• 0x0D - HSDPA+ and HSUPA</li> <li>• 0x0E - DC_HSDPA+ and WCDMA</li> <li>• 0x0F - DC_HSDPA+ and HSUPA</li> <li>• 0x8000 - NULL Bearer</li> <li>• 0xFF - Unknown Technology</li> </ul>



<i>pDormancy-Status</i>	<ul style="list-style-type: none"> <li>• Dormancy status (NULL if not present) <ul style="list-style-type: none"> <li>– 1 - traffic channel dormant</li> <li>– 2 - traffic channel active</li> </ul> </li> </ul>
<i>pPacketsCount-TX</i>	<ul style="list-style-type: none"> <li>• Packets transmitted without error (NULL if not present)</li> </ul>
<i>pPacketsCount-RX</i>	<ul style="list-style-type: none"> <li>• Packets received without error (NULL if not present)</li> </ul>
<i>pTotalBytesTX</i>	<ul style="list-style-type: none"> <li>• Bytes transmitted without error (NULL if not present)</li> </ul>
<i>pTotalBytesRX</i>	<ul style="list-style-type: none"> <li>• Bytes received without error (NULL if not present)</li> </ul>

### 8.435.2 Field Documentation

8.435.2.1 **ULONG\*** `slqsWdsEventInfo::pDataBearer`

8.435.2.2 **ULONG\*** `slqsWdsEventInfo::pDormancyStatus`

8.435.2.3 **ULONG\*** `slqsWdsEventInfo::pPacketsCountRX`

8.435.2.4 **ULONG\*** `slqsWdsEventInfo::pPacketsCountTX`

8.435.2.5 **qaQmiInterfaceInfo\*** `slqsWdsEventInfo::pQmiInterfaceInfo`

8.435.2.6 **ULONGLONG\*** `slqsWdsEventInfo::pTotalBytesRX`

8.435.2.7 **ULONGLONG\*** `slqsWdsEventInfo::pTotalBytesTX`

## 8.436 SMSAsyncRawSend\_s Struct Reference

### Data Fields

- [WORD](#) `sendStatus`
- [WORD](#) `messageID`
- [WORD](#) `causeCode`
- [BYTE](#) `errorClass`
- [WORD](#) `RPCause`
- [BYTE](#) `TPCause`
- [BYTE](#) `msgDelFailureType`
- [BYTE](#) `msgDelFailureCause`
- [BYTE](#) `alphaIDLen`
- [BYTE \\*](#) `pAlphaID`
- [ULONG](#) `userData`

### 8.436.1 Detailed Description

This structure contains SMS parameters

## Parameters

<i>sendStatus</i>	<ul style="list-style-type: none"> <li>• Send Status</li> <li>• Values: <ul style="list-style-type: none"> <li>– QMI_ERR_NONE – No error in the request</li> <li>– QMI_ERR_CAUSE_CODE - SMS cause code</li> <li>– QMI_ERR_MESSAGE_DELIVERY_FAILURE - Message could not be delivered</li> <li>– QMI_ERR_NO_MEMORY - Device could not allocate memory to formulate a response</li> </ul> </li> </ul>
<i>messageID</i>	<ul style="list-style-type: none"> <li>• Unique ID assigned by WMS for non-retry messages.</li> </ul>
<i>causeCode</i>	<ul style="list-style-type: none"> <li>• WMS cause code</li> </ul>
<i>errorClass</i>	<ul style="list-style-type: none"> <li>• Error Class</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0x00 - ERROR_CLASS_TEMPORARY</li> <li>– 0x01 - ERROR_CLASS_PERMANENT</li> </ul> </li> </ul>
<i>RPCause</i>	<ul style="list-style-type: none"> <li>• GW RP cause</li> </ul>
<i>TPCause</i>	<ul style="list-style-type: none"> <li>• GW TP Cause</li> </ul>
<i>msgDelFailure-Type</i>	<ul style="list-style-type: none"> <li>• Message delivery failure type</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0x00 - WMS_MESSAGE_DELIVERY_FAILURE_TEMPORARY</li> <li>– 0x01 - WMS_MESSAGE_DELIVERY_FAILURE_PERMANENT</li> </ul> </li> </ul>
<i>msgDelFailure-Cause</i>	<ul style="list-style-type: none"> <li>• Message delivery failure cause</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0x00 - WMS_MESSAGE_BLOCKED_DUE_TO_CALL_CONTROL</li> </ul> </li> </ul>

<i>alphaIDLen</i>	<ul style="list-style-type: none"> <li>• Number of sets of the pAlphaID</li> </ul>
<i>pAlphaID</i>	<ul style="list-style-type: none"> <li>• Alpha ID</li> </ul>
<i>userData</i>	<ul style="list-style-type: none"> <li>• Identifies the request associated with this indication.</li> </ul>

## 8.436.2 Field Documentation

8.436.2.1 **BYTE** SMSAsyncRawSend\_s::alphaIDLen

8.436.2.2 **WORD** SMSAsyncRawSend\_s::causeCode

8.436.2.3 **BYTE** SMSAsyncRawSend\_s::errorClass

8.436.2.4 **WORD** SMSAsyncRawSend\_s::messageID

8.436.2.5 **BYTE** SMSAsyncRawSend\_s::msgDelFailureCause

8.436.2.6 **BYTE** SMSAsyncRawSend\_s::msgDelFailureType

8.436.2.7 **BYTE\*** SMSAsyncRawSend\_s::pAlphaID

8.436.2.8 **WORD** SMSAsyncRawSend\_s::RPCause

8.436.2.9 **WORD** SMSAsyncRawSend\_s::sendStatus

8.436.2.10 **BYTE** SMSAsyncRawSend\_s::TPCause

8.436.2.11 **ULONG** SMSAsyncRawSend\_s::userData

## 8.437 SMSCAddress Struct Reference

### Data Fields

- [BYTE length](#)
- [BYTE data](#) [256]

### 8.437.1 Detailed Description

This structure holds SMSC information

#### Parameters

<i>length</i>	<ul style="list-style-type: none"> <li>• Number of sets of following element</li> </ul>
---------------	---

<i>data</i>	<ul style="list-style-type: none"><li>• SMSC address</li></ul>
-------------	--

### 8.437.2 Field Documentation

8.437.2.1 **BYTE** SMSCAddress::data[256]

8.437.2.2 **BYTE** SMSCAddress::length

## 8.438 SMSEtwsMessage Struct Reference

### Data Fields

- [BYTE](#) notificationType
- [WORD](#) length
- [BYTE](#) data [1254]

### 8.438.1 Detailed Description

This structure holds information related earthquake and Tsunami warning system

#### Parameters

<i>notificationType</i>	<ul style="list-style-type: none"><li>• Message mode 0x00 - Primary 0x01 - Secondary GSM 0x02 - Secondary UMTS</li></ul>
<i>length</i>	<ul style="list-style-type: none"><li>• Number of sets of following elements</li></ul>
<i>data</i>	<ul style="list-style-type: none"><li>• Raw message data</li></ul>

### 8.438.2 Field Documentation

8.438.2.1 **BYTE** SMSEtwsMessage::data[1254]

8.438.2.2 **WORD** SMSEtwsMessage::length

8.438.2.3 **BYTE** SMSEtwsMessage::notificationType

## 8.439 SMSEtwsPlmn Struct Reference

### Data Fields

- [WORD](#) mobileCountryCode
- [WORD](#) mobileNetworkCode

### 8.439.1 Detailed Description

This structure holds information related ETWS PLMN

## Parameters

<i>mobileCountry-Code</i>	<ul style="list-style-type: none"> <li>16 bit representation of MCC value range : 0 -999</li> </ul>
<i>mobileNetwork-Code</i>	<ul style="list-style-type: none"> <li>16 bit representation of MNC value range : 0 -999</li> </ul>

## 8.439.2 Field Documentation

8.439.2.1 WORD SMSEtwsPlmn::mobileCountryCode

8.439.2.2 WORD SMSEtwsPlmn::mobileNetworkCode

## 8.440 SMSEventInfo\_s Struct Reference

## Data Fields

- [BYTE smsEventType](#)
- [SMSMTMessageInfo](#) \* [pMTMessageInfo](#)
- [SMSTransferRouteMTMessageInfo](#) \* [pTransferRouteMTMessageInfo](#)
- [SMSMessageModelInfo](#) \* [pMessageModelInfo](#)
- [SMSEtwsMessageInfo](#) \* [pEtwsMessageInfo](#)
- [SMSEtwsPlmnInfo](#) \* [pEtwsPlmnInfo](#)
- [SMSCAddressInfo](#) \* [pSMSCAddressInfo](#)
- [SMSOnIMSInfo](#) \* [pSMSOnIMSInfo](#)

## 8.440.1 Detailed Description

This structure will hold the information related to received SMS events

## Parameters

<i>smsEventType</i>	<ul style="list-style-type: none"> <li>Type of the SMS events that are received. This is a bit map of <a href="#">SMSEventType</a>. Only the parameters (which follows) related to the events received would be filled, and the rest of the parameters would be NULL</li> </ul>
<i>pMTMessage-Info</i>	<ul style="list-style-type: none"> <li>pointer to the <a href="#">SMSMTMessageInfo</a> structure NULL, if this event is not present in the smsEventType parameter</li> </ul>
<i>pTransferRoute-MTMessageInfo</i>	<ul style="list-style-type: none"> <li>pointer to the <a href="#">SMSTransferRouteMTMessageInfo</a> structure . NULL, if this event is not present in the smsEventType parameter</li> </ul>

<i>pMessageMode-Info</i>	<ul style="list-style-type: none"> <li>pointer to the <a href="#">SMSMessageModeInfo</a> structure</li> <li>NULL, if this event is not present in the smsEventType parameter</li> </ul>
<i>pEtwsMessage-Info</i>	<ul style="list-style-type: none"> <li>pointer to the <a href="#">SMSEtwsMessageInfo</a> structure</li> <li>NULL, if this event is not present in the smsEventType parameter</li> </ul>
<i>pEtwsPlmnInfo</i>	<ul style="list-style-type: none"> <li>pointer to the <a href="#">SMSEtwsPlmnInfo</a> structure</li> <li>NULL, if this event is not present in the smsEventType parameter</li> </ul>
<i>pSMSCAddress-Info</i>	<ul style="list-style-type: none"> <li>pointer to the <a href="#">SMSCAddressInfo</a> structure</li> <li>NULL, if this event is not present in the smsEventType parameter</li> </ul>
<i>pSMSOnIMSInfo</i>	<ul style="list-style-type: none"> <li>pointer to the <a href="#">SMSOnIMSInfo</a> structure</li> <li>NULL, if this event is not present in the smsEventType parameter Note: None</li> </ul>

## 8.440.2 Field Documentation

8.440.2.1 [SMSEtwsMessageInfo](#)\* [SMSEventInfo\\_s::pEtwsMessageInfo](#)

8.440.2.2 [SMSEtwsPlmnInfo](#)\* [SMSEventInfo\\_s::pEtwsPlmnInfo](#)

8.440.2.3 [SMSMessageModeInfo](#)\* [SMSEventInfo\\_s::pMessageModeInfo](#)

8.440.2.4 [SMSMTMessageInfo](#)\* [SMSEventInfo\\_s::pMTMessageInfo](#)

8.440.2.5 [SMSCAddressInfo](#)\* [SMSEventInfo\\_s::pSMSCAddressInfo](#)

8.440.2.6 [SMSOnIMSInfo](#)\* [SMSEventInfo\\_s::pSMSOnIMSInfo](#)

8.440.2.7 [SMSTransferRouteMTMessageInfo](#)\* [SMSEventInfo\\_s::pTransferRouteMTMessageInfo](#)

8.440.2.8 BYTE [SMSEventInfo\\_s::smsEventType](#)

## 8.441 smsMaxStorageSizeReq Struct Reference

### Data Fields

- [BYTE](#) [storageType](#)
- [BYTE](#) \* [pMessageMode](#)

### 8.441.1 Detailed Description

This structure contains get store max size resquest parameters



## Parameters

<i>storageType</i>	<ul style="list-style-type: none"> <li>SMS message storage type <ul style="list-style-type: none"> <li>0 - UIM - Invalid in case of CDMA device that does not require SIM</li> <li>1 - NV</li> </ul> </li> </ul>
<i>pMessageMode(optional</i>	parameter) <ul style="list-style-type: none"> <li>0x00 - CDMA, LTE (if network type is CDMA)</li> <li>0x01 - GW, LTE (if network type is UMTS)</li> </ul>

## Note

The Message Mode TLV must be included if the device is capable of supporting more than one protocol

## 8.441.2 Field Documentation

8.441.2.1 **BYTE\*** smsMaxStorageSizeReq::pMessageMode

8.441.2.2 **BYTE** smsMaxStorageSizeReq::storageType

## 8.442 smsMaxStorageSizeResp Struct Reference

## Data Fields

- [ULONG maxStorageSize](#)
- [ULONG freeSlots](#)

## 8.442.1 Detailed Description

This structure contains get store max size response parameters

## Parameters

<i>maxStorageSize</i>	- <ul style="list-style-type: none"> <li>Memory Store Size</li> </ul>
<i>freeSlots</i>	- <ul style="list-style-type: none"> <li>Optional parameter indicating how much Memory is available</li> <li>function <a href="#">SLQSSmsGetMaxStorageSize()</a> returns a default value 0xFFFFFFFF for parameter values if no response is received from the device.</li> </ul>

## 8.442.2 Field Documentation

8.442.2.1 **ULONG** smsMaxStorageSizeResp::freeSlots

8.442.2.2 **ULONG** smsMaxStorageSizeResp::maxStorageSize

## 8.443 SMSMemoryInfo Struct Reference

### Data Fields

- [BYTE storageType](#)
- [BYTE messageMode](#)

#### 8.443.1 Detailed Description

This structure holds information related to memory

##### Parameters

<i>storageType</i>	<ul style="list-style-type: none"><li>• Indicates the type of memory storage 0x00 - STORAGE_TYPE_UIM 0x01 - STORAGE_TYPE_NV</li></ul>
<i>messageMode</i>	<ul style="list-style-type: none"><li>• Indicates the type of memory mode 0x00 - MESSAGE_MODE_CDMA - CDMA 0x01 - MESSAGE_MODE_GW - GW</li></ul>

#### 8.443.2 Field Documentation

8.443.2.1 **BYTE** SMSMemoryInfo::messageMode

8.443.2.2 **BYTE** SMSMemoryInfo::storageType

## 8.444 SMSMessageMode Struct Reference

### Data Fields

- [BYTE messageMode](#)

#### 8.444.1 Detailed Description

This structure holds information related to message mode

##### Parameters

<i>messageMode</i>	<ul style="list-style-type: none"><li>• Message mode 0x00 - CDMA 0x01 - GW</li></ul>
--------------------	--

#### 8.444.2 Field Documentation

8.444.2.1 **BYTE** SMSMessageMode::messageMode

## 8.445 smsMsgprotocolResp Struct Reference

### Data Fields

- [BYTE msgProtocol](#)

### 8.445.1 Detailed Description

This structure contains get message protocol response parameters

#### Parameters

<i>msgProtocol</i>	- <ul style="list-style-type: none"><li>• Message Protocol</li><li>• Values:<ul style="list-style-type: none"><li>– 0x00 - MESSAGE_PROTOCOL_CDMA</li><li>– 0x01 - MESSAGE_PROTOCOL_WCDMA</li></ul></li></ul>
--------------------	--

### 8.445.2 Field Documentation

#### 8.445.2.1 BYTE smsMsgprotocolResp::msgProtocol

## 8.446 SMSMTMessage Struct Reference

#### Data Fields

- [ULONG storageType](#)
- [ULONG messageIndex](#)

### 8.446.1 Detailed Description

This structure holds information related to MT SMS

#### Parameters

<i>storageType</i>	<ul style="list-style-type: none"><li>• SMS message storage type for the new message</li><li>0 - UIM 1 - NV</li></ul>
<i>messageIndex</i>	<ul style="list-style-type: none"><li>• Index of the new message</li></ul>

### 8.446.2 Field Documentation

#### 8.446.2.1 ULONG SMSMTMessage::messageIndex

#### 8.446.2.2 ULONG SMSMTMessage::storageType

## 8.447 SMSOnIMS Struct Reference

#### Data Fields

- [BYTE smsOnIMS](#)

### 8.447.1 Detailed Description

This structure holds information related to message mode

## Parameters

<i>smsOnIMS</i>	<ul style="list-style-type: none"><li>Indicates whether the message is received from IMS 0x00 - Message is not received from IMS 0x01 - Message is received from IMS 0x02-0xFF - Reserved Note: In multiple modem solutions, this TLV may be used to help the client determine with which modem to communicate. This TLV may not be supported on all implementations.</li></ul>
-----------------	---

## 8.447.2 Field Documentation

## 8.447.2.1 BYTE SMSOnIMS::smsOnIMS

## 8.448 smsRouteEntry Struct Reference

## Data Fields

- [BYTE messageType](#)
- [BYTE messageClass](#)
- [BYTE routeStorage](#)
- [BYTE receiptAction](#)

## 8.448.1 Detailed Description

This structure contains SMS route entry details

## Parameters

<i>messageType</i>	- <ul style="list-style-type: none"><li>Message type matching this route</li><li>Values:<ul style="list-style-type: none"><li>0x00 - MESSAGE_TYPE_POINT_TO_POINT</li></ul></li></ul>
<i>messageClass</i>	- <ul style="list-style-type: none"><li>Message Class</li><li>Values:<ul style="list-style-type: none"><li>0x00 - MESSAGE_CLASS_0</li><li>0x01 - MESSAGE_CLASS_1</li><li>0x02 - MESSAGE_CLASS_2</li><li>0x03 - MESSAGE_CLASS_3</li><li>0x04 - MESSAGE_CLASS_NONE</li><li>0x05 - MESSAGE_CLASS_CDMA</li></ul></li></ul>

<i>routeStorage</i>	- <ul style="list-style-type: none"> <li>• If the receiptAction is store where to store the message</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0x00 - STORAGE_TYPE_UIM</li> <li>– 0x01 - STORAGE_TYPE_NV</li> <li>– 0xFF - STORAGE_TYPE_NONE</li> </ul> </li> </ul>
<i>receiptAction</i>	- <ul style="list-style-type: none"> <li>• Action to be taken on receipt of a message matching the specified type and class for this route</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0x00 - DISCARD (discarded without notification)</li> <li>– 0x01 - STORE AND NOTIFY (stored and notified to the registered clients)</li> <li>– 0x02 - TRANSFER ONLY (transferred to the client, client expected to send the ACK)</li> <li>– 0x03 - TRANSFER AND ACK (transferred to the client, device expected to send the ACK)</li> </ul> </li> </ul>

## 8.448.2 Field Documentation

8.448.2.1 **BYTE** smsRouteEntry::messageClass

8.448.2.2 **BYTE** smsRouteEntry::messageType

8.448.2.3 **BYTE** smsRouteEntry::receiptAction

8.448.2.4 **BYTE** smsRouteEntry::routeStorage

## 8.449 smsSetRoutesReq Struct Reference

### Data Fields

- [WORD](#) numOfRoutes
- [smsRouteEntry](#) routeList [0x0A]
- **BYTE** \* pTransferStatusReport

### 8.449.1 Detailed Description

This structure contains SMS route request parameters

## Parameters

<i>numOfRoutes</i>	- <ul style="list-style-type: none"> <li>Number of sets of the following element</li> </ul>
<i>routeList</i>	- <ul style="list-style-type: none"> <li>Array containing the set of <a href="#">smsRouteEntry</a></li> </ul>
<i>pTransferStatus-Report</i>	- <ul style="list-style-type: none"> <li>0x01 - Status reoprt are transferred to the client (optional)</li> </ul>

## 8.449.2 Field Documentation

8.449.2.1 WORD smsSetRoutesReq::numOfRoutes

8.449.2.2 BYTE\* smsSetRoutesReq::pTransferStatusReport

8.449.2.3 smsRouteEntry smsSetRoutesReq::routeList[0x0A]

## 8.450 SMSTransferRouteMTMessage Struct Reference

## Data Fields

- [BYTE](#) *ackIndicator*
- [ULONG](#) *transactionID*
- [BYTE](#) *format*
- [WORD](#) *length*
- [BYTE](#) *data* [256]

## 8.450.1 Detailed Description

This structure holds information related to transfer route MT SMS

## Parameters

<i>ackIndicator</i>	<ul style="list-style-type: none"> <li>Parameter to indicate if ACK must be sent by the control point 0x00 - Send ACK 0x01 - Do not send ACK</li> </ul>
<i>transactionID</i>	<ul style="list-style-type: none"> <li>Transaction ID of the message</li> </ul>
<i>format</i>	<ul style="list-style-type: none"> <li>Message format 0x00 - CDMA 0x02 - 0x05 - Reserved 0x06 - GW_PP 0x07 - GW_BC</li> </ul>
<i>length</i>	<ul style="list-style-type: none"> <li>Length of the raw message. This length should not exceed the maximum WMS payload length of 256 bytes</li> </ul>

<i>data</i>	<ul style="list-style-type: none"> <li>• Raw message data</li> </ul>
-------------	--

## 8.450.2 Field Documentation

8.450.2.1 **BYTE** SMSTransferRouteMTMessage::ackIndicator

8.450.2.2 **BYTE** SMSTransferRouteMTMessage::data[256]

8.450.2.3 **BYTE** SMSTransferRouteMTMessage::format

8.450.2.4 **WORD** SMSTransferRouteMTMessage::length

8.450.2.5 **ULONG** SMSTransferRouteMTMessage::transactionID

## 8.451 sQosFlowStat Struct Reference

### Data Fields

- [ULONG bearerId](#)
- [ULONG tx\\_pkt](#)
- [ULONG tx\\_pkt\\_drp](#)
- [ULONGLONG tx\\_bytes](#)
- [ULONGLONG tx\\_bytes\\_drp](#)

### 8.451.1 Detailed Description

This structure contains the Data statistic per QoS flow

#### Parameters

<i>bearerId</i>	<ul style="list-style-type: none"> <li>• Bearer ID</li> </ul>
<i>tx_pkt</i>	<ul style="list-style-type: none"> <li>• number of sent packets for the QoS flow ID</li> </ul>
<i>tx_pkt_drp</i>	<ul style="list-style-type: none"> <li>• number of dropped(TX) packets for the QoS flow ID</li> </ul>
<i>tx_bytes</i>	<ul style="list-style-type: none"> <li>• number of sent bytes for the QoS flow ID</li> </ul>
<i>tx_bytes_drp</i>	<ul style="list-style-type: none"> <li>• number of dropped(TX) bytes for the QoS flow ID</li> </ul>

## 8.451.2 Field Documentation

8.451.2.1 **ULONG** sQosFlowStat::bearerId



8.451.2.2 **ULONGLONG** sQosFlowStat::tx\_bytes

8.451.2.3 **ULONGLONG** sQosFlowStat::tx\_bytes\_drp

8.451.2.4 **ULONG** sQosFlowStat::tx\_pkt

8.451.2.5 **ULONG** sQosFlowStat::tx\_pkt\_drp

## 8.452 sQosStat Struct Reference

### Data Fields

- [ULONG](#) apnId
- [ULONG](#) total\_tx\_pkt
- [ULONG](#) total\_tx\_pkt\_drp
- [ULONG](#) total\_rx\_pkt
- [ULONGLONG](#) total\_tx\_bytes
- [ULONGLONG](#) total\_tx\_bytes\_drp
- [ULONGLONG](#) total\_rx\_bytes
- [ULONG](#) numQosFlow
- [sQosFlowStat](#) qosFlow [(10)]

### 8.452.1 Detailed Description

This structure contains the Data statistic per QoS flow

#### Parameters

<i>apnId</i>	<ul style="list-style-type: none"> <li>• APN id</li> <li>• ID identifying the connected APN that the client would like to query the data statistic for</li> </ul>
<i>total_tx_pkt</i>	<ul style="list-style-type: none"> <li>• sum of all packets sent</li> </ul>
<i>total_tx_pkt_drp</i>	<ul style="list-style-type: none"> <li>• sum of all(TX) packets dropped</li> </ul>
<i>total_rx_pkt</i>	<ul style="list-style-type: none"> <li>• sum of all packets received</li> </ul>
<i>total_tx_bytes</i>	<ul style="list-style-type: none"> <li>• sum of all bytes sent</li> </ul>
<i>total_tx_bytes_drp</i>	<ul style="list-style-type: none"> <li>• sum of all(TX) bytes dropped</li> </ul>
<i>total_rx_bytes</i>	<ul style="list-style-type: none"> <li>• number of received bytes for the QoS flow ID</li> </ul>

<i>numQosFlow</i>	<ul style="list-style-type: none"> <li>• pointer to number of QoS flow Stat</li> </ul>
<i>qosFlow[MAX_QOS_SPEC_PER_APN]</i>	<ul style="list-style-type: none"> <li>• Data statistic per QoS flow</li> <li>• See <a href="#">sQosFlowStat</a> for more information</li> <li>• See <a href="#">MAX_QOS_SPEC_PER_APN</a> for more information</li> </ul>

## 8.452.2 Field Documentation

8.452.2.1 **ULONG** sQosStat::apnId

8.452.2.2 **ULONG** sQosStat::numQosFlow

8.452.2.3 **sQosFlowStat** sQosStat::qosFlow[(10)]

8.452.2.4 **ULONGLONG** sQosStat::total\_rx\_bytes

8.452.2.5 **ULONG** sQosStat::total\_rx\_pkt

8.452.2.6 **ULONGLONG** sQosStat::total\_tx\_bytes

8.452.2.7 **ULONGLONG** sQosStat::total\_tx\_bytes\_drp

8.452.2.8 **ULONG** sQosStat::total\_tx\_pkt

8.452.2.9 **ULONG** sQosStat::total\_tx\_pkt\_drp

## 8.453 SrvStatusInfo Struct Reference

### Data Fields

- [BYTE](#) srvStatus
- [BYTE](#) isPrefDataPath

### 8.453.1 Detailed Description

Structure for storing the service status information for CDMA and HDR networks.

## Parameters

<i>srvStatus</i>	<ul style="list-style-type: none"> <li>• Service status of the system. <ul style="list-style-type: none"> <li>– 0x00 - No service</li> <li>– 0x01 - Limited service</li> <li>– 0x02 - Service</li> <li>– 0x03 - Limited regional service</li> <li>– 0x04 - Power save</li> <li>– 0xFF - Not Available</li> </ul> </li> </ul>
<i>isPrefDataPath</i>	<ul style="list-style-type: none"> <li>• Whether the RAT is the preferred data path. <ul style="list-style-type: none"> <li>– 0x00 - Not preferred</li> <li>– 0x01 - Preferred</li> <li>– 0xFF - Not Available</li> </ul> </li> </ul>

## 8.453.2 Field Documentation

## 8.453.2.1 BYTE SrvStatusInfo::isPrefDataPath

## 8.453.2.2 BYTE SrvStatusInfo::srvStatus

## 8.454 ssdatasession\_params Struct Reference

## Data Fields

- [BOOL action](#)
- [BYTE instanceId](#)
- [ULONG \\* pTechnology](#)
- [ULONG \\* pProfileId3GPP](#)
- [ULONG \\* pProfileId3GPP2](#)
- [ULONG sessionId](#)
- [ULONG failureReason](#)
- [ULONG failureReasonv4](#)
- [ULONG failureReasonv6](#)
- [ULONG rcv4](#)
- [ULONG rcv6](#)
- [ULONG v4sessionId](#)
- [ULONG v6sessionId](#)
- [BYTE ipfamily](#)
- [ULONG \\* pAuthentication](#)
- [CHAR \\* pUsername](#)
- [CHAR \\* pPassword](#)
- [ULONG verbFailReasonType](#)
- [ULONG verbFailReason](#)

### 8.454.1 Detailed Description

This structure contains the start/stop data session params Information

## Parameters

<i>action</i>	<ul style="list-style-type: none"> <li>• 1 - Start Session</li> <li>• 0 - Stop Session</li> </ul>
<i>pTechnology</i>	<ul style="list-style-type: none"> <li>• Indicates the technology preference (optional) <ul style="list-style-type: none"> <li>– 1 - UMTS</li> <li>– 2 - CDMA</li> <li>– 3 - eMBMS</li> <li>– 4 - Modem Link Label. Modem Link is an interface for transferring data between entities on AP and modem.</li> </ul> </li> </ul>
<i>pProfileId3GPP</i>	<ul style="list-style-type: none"> <li>• configured 3GPP profile identifier</li> </ul>
<i>pProfileId3GPP2</i>	<ul style="list-style-type: none"> <li>• configured 3GPP2 profile identifier</li> </ul>
<i>sessionId</i> [IN\OUT]	<ul style="list-style-type: none"> <li>• [IN] - Passed session ID when stopping the data session</li> <li>• [OUT] - Assigned session ID when starting a data session</li> </ul>
<i>failureReason</i>	<ul style="list-style-type: none"> <li>• Reason data session failed to be established</li> <li>• See <a href="#">qaGobiApiTableCallEndReasons.h</a> for Call End Reason</li> </ul>
<i>failureReasonv4</i>	<ul style="list-style-type: none"> <li>• Reason v4 data session failed to be established</li> <li>• See <a href="#">qaGobiApiTableCallEndReasons.h</a> for Call End Reason</li> </ul>
<i>failureReasonv6</i>	<ul style="list-style-type: none"> <li>• Reason v6 data session failed to be established</li> <li>• See <a href="#">qaGobiApiTableCallEndReasons.h</a> for Call End Reason</li> </ul>
<i>rc4</i>	<ul style="list-style-type: none"> <li>• v4 result code</li> <li>• See <a href="#">qmerrno.h</a></li> </ul>
<i>rc6</i>	<ul style="list-style-type: none"> <li>• v6 result code</li> <li>• See <a href="#">qmerrno.h</a></li> </ul>

<i>v4sessionId</i>	<ul style="list-style-type: none"> <li>• Do not modify - used for internal management of data sessions</li> <li>• Non zero value indicates that a session is active</li> </ul>
<i>v6sessionId</i>	<ul style="list-style-type: none"> <li>• Do not modify - used for internal management of data sessions</li> <li>• Non zero value indicates that a session is active</li> </ul>
<i>ipfamily</i>	<ul style="list-style-type: none"> <li>• 4 for an IPv4 data session</li> <li>• 6 for an IPv6 data session</li> <li>• 7 for an IPv4v6 data session</li> </ul>
<i>pAuthentication</i>	<ul style="list-style-type: none"> <li>• Authentication type, it can be PAP or CHAP</li> </ul>
<i>pUsername</i>	<ul style="list-style-type: none"> <li>• username for authentication process</li> </ul>
<i>pPassword</i>	<ul style="list-style-type: none"> <li>• password for authentication process</li> </ul>
<i>verbFailReason- Type</i>	<ul style="list-style-type: none"> <li>• Parameter describing type of verbose failure reason</li> <li>• See <a href="#">qaGobiApiTableCallEndReasons.h</a> for Call End Reason Type</li> </ul>
<i>verbFailReason</i>	<ul style="list-style-type: none"> <li>• Verbose reason explaining why call failed. Depends on verbFailReasonType parameter</li> <li>• See <a href="#">qaGobiApiTableCallEndReasons.h</a> for Call End Reason</li> </ul>

## 8.454.2 Field Documentation

8.454.2.1 **BOOL** ssdatasession\_params::action

8.454.2.2 **ULONG** ssdatasession\_params::failureReason

8.454.2.3 **ULONG** ssdatasession\_params::failureReasonv4

8.454.2.4 **ULONG** ssdatasession\_params::failureReasonv6

8.454.2.5 **BYTE** ssdatasession\_params::instanceId

8.454.2.6 **BYTE** ssdatasession\_params::ipfamily

8.454.2.7 **ULONG\*** ssdatasession\_params::pAuthentication

8.454.2.8 **CHAR\*** ssdatasession\_params::pPassword

- 8.454.2.9 **ULONG\*** ssdatasession\_params::pProfileId3GPP
- 8.454.2.10 **ULONG\*** ssdatasession\_params::pProfileId3GPP2
- 8.454.2.11 **ULONG\*** ssdatasession\_params::pTechnology
- 8.454.2.12 **CHAR\*** ssdatasession\_params::pUsername
- 8.454.2.13 **ULONG** ssdatasession\_params::rcv4
- 8.454.2.14 **ULONG** ssdatasession\_params::rcv6
- 8.454.2.15 **ULONG** ssdatasession\_params::sessionId
- 8.454.2.16 **ULONG** ssdatasession\_params::v4sessionId
- 8.454.2.17 **ULONG** ssdatasession\_params::v6sessionId
- 8.454.2.18 **ULONG** ssdatasession\_params::verbFailReason
- 8.454.2.19 **ULONG** ssdatasession\_params::verbFailReasonType

## 8.455 SupportedMsgList Struct Reference

### Data Fields

- [WORD supportedMsgLen](#)
- [BYTE supportedMsgs](#) [256]

### 8.455.1 Detailed Description

This structure contains the Supported Messages List Information

#### Parameters

<i>supportedMsgLen</i>	<ul style="list-style-type: none"><li>• Number of sets of the supported messages</li></ul>
<i>supportedMsgs</i>	<ul style="list-style-type: none"><li>• Array of uint8 is a bitmask where each bit represents a message ID.</li><li>• Starting with the LSB, bit 0 represents message ID 0, bit 1 represents message ID 1.</li></ul>

### 8.455.2 Field Documentation

- 8.455.2.1 **WORD** SupportedMsgList::supportedMsgLen
- 8.455.2.2 **BYTE** SupportedMsgList::supportedMsgs[256]

## 8.456 SUPSInfo Struct Reference

## Data Fields

- [BYTE svcType](#)
- [BYTE isModByCC](#)

### 8.456.1 Detailed Description

This structure contains information about the Supplementary Services.

#### Parameters

<i>svcType</i>	<ul style="list-style-type: none"> <li>• Service type. <ul style="list-style-type: none"> <li>– 0x01 - SERVICE_TYPE_ACTIVATE - Activate</li> <li>– 0x02 - SERVICE_TYPE_DEACTIVATE - Deactivate</li> <li>– 0x03 - SERVICE_TYPE_REGISTER - Register</li> <li>– 0x04 - SERVICE_TYPE_ERASE - Erase</li> <li>– 0x05 - SERVICE_TYPE_INTERROGATE - Interrogate</li> <li>– 0x06 - SERVICE_TYPE_REGISTER_PASSWORD - Register password</li> <li>– 0x07 - SERVICE_TYPE_USSD - USSD</li> </ul> </li> </ul>
<i>isModByCC</i>	<ul style="list-style-type: none"> <li>• Indicates whether the supplementary service data is modified by the card (SIM/USIM) as part of the call control: <ul style="list-style-type: none"> <li>– 0 - False</li> <li>– 1 - True</li> </ul> </li> </ul>

### 8.456.2 Field Documentation

#### 8.456.2.1 [BYTE SUPSInfo::isModByCC](#)

#### 8.456.2.2 [BYTE SUPSInfo::svcType](#)

## 8.457 SV Struct Reference

## Data Fields

- [WORD id](#)
- [ULONG system](#)
- [BYTE mask](#)

### 8.457.1 Detailed Description

This structure contains the Delete [SV](#) Info



## Parameters

<i>id</i>	<ul style="list-style-type: none"> <li>• <a href="#">SV</a> ID of the satellite whose data is to be deleted</li> <li>• Range: <ul style="list-style-type: none"> <li>– For GPS: 1 to 32</li> <li>– For SBAS: 33 to 64</li> <li>– For GLONASS: 65 to 96</li> </ul> </li> </ul>
<i>system</i>	<ul style="list-style-type: none"> <li>• Indicates to which constellation this <a href="#">SV</a> belongs</li> <li>• Valid values: <ul style="list-style-type: none"> <li>– eQMI_LOC_SV_SYSTEM_GPS (1) - GPS satellite</li> <li>– eQMI_LOC_SV_SYSTEM_GALILEO (2) - GALILEO satellite</li> <li>– eQMI_LOC_SV_SYSTEM_SBAS (3) - SBAS satellite</li> <li>– eQMI_LOC_SV_SYSTEM_COMPASS (4) - COMPASS satellite</li> <li>– eQMI_LOC_SV_SYSTEM_GLONASS (5) - GLONASS satellite</li> <li>– eQMI_LOC_SV_SYSTEM_BDS (6) - BDS satellite</li> </ul> </li> </ul>
<i>mask</i>	<ul style="list-style-type: none"> <li>• Indicates if the ephemeris or almanac for a satellite is to be deleted</li> <li>• Valid values: <ul style="list-style-type: none"> <li>– 0x01 - DELETE_EPHEMERIS</li> <li>– 0x02 - DELETE_ALMANAC</li> </ul> </li> </ul>

## 8.457.2 Field Documentation

## 8.457.2.1 WORD SV::id

## 8.457.2.2 BYTE SV::mask

## 8.457.2.3 ULONG SV::system

## 8.458 SVInfo Struct Reference

## Data Fields

- [BYTE](#) len
- [SV](#) \* pSV

## 8.458.1 Detailed Description

This structure contains the elements of Delete [SV](#) Info

## Parameters

<i>len</i>	<ul style="list-style-type: none"> <li>Number of sets of the following elements in struct <a href="#">SV</a>: <ul style="list-style-type: none"> <li>gnssSvId</li> <li>system</li> <li>deleteSvInfoMask</li> </ul> </li> </ul>
<i>pSV</i>	<ul style="list-style-type: none"> <li>Pointer to struct <a href="#">SV</a>. See <a href="#">SV</a> for more information</li> </ul>

## 8.458.2 Field Documentation

## 8.458.2.1 BYTE SVInfo::len

## 8.458.2.2 SV\* SVInfo::pSV

## 8.459 svUsedforFix\_s Struct Reference

## Data Fields

- [BYTE gnssSvUsedList\\_len](#)
- [WORD gnssSvUsedList \[255\]](#)

## 8.459.1 Detailed Description

This structure contains SVs Used to Calculate the Fix.

## Parameters

<i>gnssSvUsedList_len</i>	<ul style="list-style-type: none"> <li>Number of sets of gnssSvUsedList</li> </ul>
<i>pGnssSvUsedList</i>	<ul style="list-style-type: none"> <li>Entry in the list contains the <a href="#">SV</a> ID of a satellite used for calculating this position report.</li> <li>Following information is associated with each <a href="#">SV</a> ID: <ul style="list-style-type: none"> <li>GPS - 1 to 32</li> <li>SBAS - 33 to 64</li> <li>GLONASS - 65 to 96</li> <li>QZSS - 193 to 197</li> <li>BDS - 201 to 237</li> </ul> </li> </ul>

## 8.459.2 Field Documentation

## 8.459.2.1 WORD svUsedforFix\_s::gnssSvUsedList[255]

8.459.2.2 BYTE svUsedforFix\_s::gnssSvUsedList\_len

## 8.460 SWI\_STRUCT\_CarrierImage Struct Reference

### Data Fields

- [ULONG m\\_nCarrierId](#)
- [ULONG m\\_nFolderId](#)
- [ULONG m\\_nStorage](#)
- [BYTE m\\_FwImageId](#) [16]
- [BYTE m\\_FwBuildId](#) [32]
- [BYTE m\\_PriImageId](#) [16]
- [BYTE m\\_PriBuildId](#) [32]

### 8.460.1 Detailed Description

This structure contains the Carrier Image parameters.

#### Parameters

<i>m_nCarrierId</i>	<ul style="list-style-type: none"> <li>• Unique numeric carrier ID indicating the carrier that the following images belong to</li> </ul>
<i>m_nFolderId</i>	<ul style="list-style-type: none"> <li>• Unique numeric folder ID indicating the folder where the images should reside on the host storage.</li> </ul>
<i>m_nStorage</i>	<ul style="list-style-type: none"> <li>• Information of storage type</li> <li>• Values <ul style="list-style-type: none"> <li>– 0 - Device</li> <li>– 1 - Host</li> </ul> </li> </ul>
<i>m_FwImageId</i>	<ul style="list-style-type: none"> <li>• Firmware image ID</li> </ul>
<i>m_FwBuildId</i>	<ul style="list-style-type: none"> <li>• Firmware build ID</li> </ul>
<i>m_PriImageId</i>	<ul style="list-style-type: none"> <li>• PRI image ID</li> </ul>
<i>m_PriBuildId</i>	<ul style="list-style-type: none"> <li>• PRI build ID</li> </ul>

### 8.460.2 Field Documentation

8.460.2.1 BYTE SWI\_STRUCT\_CarrierImage::m\_FwBuildId[32]

8.460.2.2 **BYTE** SWI\_STRUCT\_CarrierImage::m\_FwImageld[16]

8.460.2.3 **ULONG** SWI\_STRUCT\_CarrierImage::m\_nCarrierId

8.460.2.4 **ULONG** SWI\_STRUCT\_CarrierImage::m\_nFolderId

8.460.2.5 **ULONG** SWI\_STRUCT\_CarrierImage::m\_nStorage

8.460.2.6 **BYTE** SWI\_STRUCT\_CarrierImage::m\_PriBuildId[32]

8.460.2.7 **BYTE** SWI\_STRUCT\_CarrierImage::m\_PrImageld[16]

## 8.461 swiModemStatusResp Struct Reference

### Data Fields

- [CommInfo](#) [commonInfo](#)
- [LTEInfo](#) \* [pLTEInfo](#)

### 8.461.1 Detailed Description

Structure for storing the SLQS Nas Swi Modem Status response parameters.

#### Parameters

<i>commonInfo</i>	(mandatory) <ul style="list-style-type: none"> <li>• See <a href="#">CommInfo</a> for more information</li> </ul>
<i>pLTEInfo</i>	(optional) <ul style="list-style-type: none"> <li>• See <a href="#">LTEInfo</a> for more information</li> </ul>

### 8.461.2 Field Documentation

8.461.2.1 **CommInfo** swiModemStatusResp::commonInfo

8.461.2.2 **LTEInfo\*** swiModemStatusResp::pLTEInfo

## 8.462 SwiOTAMsg\_s Struct Reference

### Data Fields

- [ULONG](#) type
- [WORD](#) data\_len
- [BYTE](#) data [2048]
- [LteNasReleaseInfo](#) \* [pLteNasRelInfo](#)
- [ULONGLONG](#) \* [pTime](#)

### 8.462.1 Detailed Description

This structure contains OTA message

## Parameters

<i>type</i>	<ul style="list-style-type: none"> <li>• message type <ul style="list-style-type: none"> <li>– 0 - LTE ESM uplink</li> <li>– 1 - LTE ESM downlink</li> <li>– 2 - LTE EMM uplink</li> <li>– 3 - LTE EMM downlink</li> <li>– 4 - GSM/UMTS uplink</li> <li>– 5 - GSM/UMTS downlink</li> </ul> </li> </ul>
<i>data_len</i>	<ul style="list-style-type: none"> <li>• OTA Message Content Length</li> </ul>
<i>data</i>	<ul style="list-style-type: none"> <li>• OTA Message Content</li> </ul>
<i>pLteNasRelInfo</i>	<ul style="list-style-type: none"> <li>• LTE NAS Release Info</li> <li>• see <a href="#">LteNasReleaseInfo</a> for details</li> </ul>
<i>pTime</i>	<ul style="list-style-type: none"> <li>• Seconds in local time since Jan. 6th 1980 00:00:00 UTC</li> </ul>

## 8.462.2 Field Documentation

8.462.2.1 BYTE SwiOTAMsg\_s::data[2048]

8.462.2.2 WORD SwiOTAMsg\_s::data\_len

8.462.2.3 LteNasReleaseInfo\* SwiOTAMsg\_s::pLteNasRelInfo

8.462.2.4 ULONGLONG\* SwiOTAMsg\_s::pTime

8.462.2.5 ULONG SwiOTAMsg\_s::type

## 8.463 swiPDPRuntimeSettingsReq Struct Reference

## Data Fields

- [BYTE contextId](#)
- [BYTE contextType](#)

## 8.463.1 Detailed Description

This structure contains the PDP Runtime Settings Request parameters.

## Parameters

<i>contextId</i>	<ul style="list-style-type: none"> <li>Context Identifier</li> </ul>
<i>v4sessionId</i>	<ul style="list-style-type: none"> <li>The v4 session ID for which the runtime settings are to be retrieved</li> <li>provide a NULL pointer if not applicable</li> </ul>
<i>v6sessionId</i>	<ul style="list-style-type: none"> <li>The v6 session ID for which the runtime settings are to be retrieved</li> <li>provide a NULL pointer if not applicable</li> </ul>

## 8.463.2 Field Documentation

8.463.2.1 BYTE swiPDPRuntimeSettingsReq::contextId

8.463.2.2 BYTE swiPDPRuntimeSettingsReq::contextType

## 8.464 swiPDPRuntimeSettingsResp Struct Reference

## Data Fields

- BYTE \* pContextId
- BYTE \* pBearerId
- CHAR \* pAPNName
- ULONG \* pIPv4Address
- ULONG \* pIPv4GWAddress
- ULONG \* pPrDNSIPv4Address
- ULONG \* pSeDNSIPv4Address
- struct IPV6AddressInfo \* pIPv6Address
- struct IPV6AddressInfo \* pIPv6GWAddress
- WORD \* pPrDNSIPv6Address
- WORD \* pSeDNSIPv6Address
- ULONG \* pPrPCSCFIPv4Address
- ULONG \* pSePCSCFIPv4Address
- WORD \* pPrPCSCFIPv6Address
- WORD \* pSePCSCFIPv6Address

## 8.464.1 Detailed Description

This structure contains the response parameters retrieved by the API SLQSWdsSwiPDPRuntimeSettings

## Parameters

<i>pContextId</i>	(optional) <ul style="list-style-type: none"> <li>Context Identifier <ul style="list-style-type: none"> <li>0xFF - Not Available</li> </ul> </li> </ul>
<i>pBearerId</i>	(optional) <ul style="list-style-type: none"> <li>Bearer Identity</li> <li>An EPS bearer identity uniquely identifies an EPS bearer for one UE accessing via E-UTRAN. The EPS Bearer Identity is allocated by the MME. <ul style="list-style-type: none"> <li>0xFF - Not Available</li> </ul> </li> </ul>
<i>pAPNName</i>	(optional) <ul style="list-style-type: none"> <li>APN name associated with the context id <ul style="list-style-type: none"> <li>NULL terminated by default.</li> </ul> </li> </ul>
<i>pIPv4Address</i>	(optional) <ul style="list-style-type: none"> <li>IPv4 Address <ul style="list-style-type: none"> <li>0xFFFF - Not Available</li> </ul> </li> </ul>
<i>pIPv4GW-Address</i>	(optional) <ul style="list-style-type: none"> <li>IPv4 Gateway Address <ul style="list-style-type: none"> <li>0xFFFF - Not Available</li> </ul> </li> </ul>
<i>pPrDNSIPv4-Address</i>	(optional) <ul style="list-style-type: none"> <li>Primary DNS IPv4 Address <ul style="list-style-type: none"> <li>0xFFFF - Not Available</li> </ul> </li> </ul>
<i>pSeDNSIPv4-Address</i>	(optional) <ul style="list-style-type: none"> <li>Secondary DNS IPv4 Address <ul style="list-style-type: none"> <li>0xFFFF - Not Available</li> </ul> </li> </ul>
<i>pIPv6Address</i>	(optional) <ul style="list-style-type: none"> <li>IPv6 Address</li> <li>See <a href="#">IPV6AddressInfo</a> for more information</li> </ul>

<i>pIPv6GW-Address</i>	(optional) <ul style="list-style-type: none"> <li>IPv6 Gateway Address</li> <li>See <a href="#">IPv6AddressInfo</a> for more information</li> </ul>
<i>pPrDNSIPv6-Address</i>	(optional) <ul style="list-style-type: none"> <li>Primary IPv6 DNS Address(in network byte order)</li> <li>This is an 8-element array of 16-bit numbers, each of which is in big-endian format</li> </ul>
<i>pSeDNSIPv6-Address</i>	(optional) <ul style="list-style-type: none"> <li>Secondary IPv6 DNS Address(in network byte order)</li> <li>This is an 8-element array of 16-bit numbers, each of which is in big-endian format</li> </ul>
<i>pPrPCSCFIPv4-Address</i>	(optional) <ul style="list-style-type: none"> <li>Primary PCSCF IPv4 Address</li> </ul>
<i>pSePCSCFIPv4-Address</i>	(optional) <ul style="list-style-type: none"> <li>Secondary PCSCF IPv4 Address</li> </ul>
<i>pPrPCSCFIPv6-Address</i>	(optional) <ul style="list-style-type: none"> <li>Primary PCSCF IPv6 Address</li> <li>This is an 8-element array of 16-bit numbers, each of which is in big-endian format</li> </ul>
<i>pSePCSCFIPv6-Address</i>	(optional) <ul style="list-style-type: none"> <li>Secondary PCSCF IPv6 Address</li> <li>This is an 8-element array of 16-bit numbers, each of which is in big-endian format</li> </ul>

**Note**

Parameters which are mentioned as NULL will be ignored.

**8.464.2 Field Documentation**

**8.464.2.1 CHAR\*** swiPDPRuntimeSettingsResp::pAPNName

**8.464.2.2 BYTE\*** swiPDPRuntimeSettingsResp::pBearerId

**8.464.2.3 BYTE\*** swiPDPRuntimeSettingsResp::pContextId

**8.464.2.4 ULONG\*** swiPDPRuntimeSettingsResp::pIPv4Address

**8.464.2.5 ULONG\*** swiPDPRuntimeSettingsResp::pIPv4GWAddress

**8.464.2.6 struct IPV6AddressInfo\*** swiPDPRuntimeSettingsResp::pIPv6Address

**8.464.2.7 struct IPV6AddressInfo\*** swiPDPRuntimeSettingsResp::pIPv6GWAddress

**8.464.2.8 ULONG\*** swiPDPRuntimeSettingsResp::pPrDNSIPv4Address



- 8.464.2.9    **WORD\*** swiPDPRuntimeSettingsResp::pPrDNSIPv6Address
- 8.464.2.10    **ULONG\*** swiPDPRuntimeSettingsResp::pPrPCSCFIPv4Address
- 8.464.2.11    **WORD\*** swiPDPRuntimeSettingsResp::pPrPCSCFIPv6Address
- 8.464.2.12    **ULONG\*** swiPDPRuntimeSettingsResp::pSeDNSIPv4Address
- 8.464.2.13    **WORD\*** swiPDPRuntimeSettingsResp::pSeDNSIPv6Address
- 8.464.2.14    **ULONG\*** swiPDPRuntimeSettingsResp::pSePCSCFIPv4Address
- 8.464.2.15    **WORD\*** swiPDPRuntimeSettingsResp::pSePCSCFIPv6Address

## 8.465    swiQosFilter Struct Reference

### Data Fields

- [BYTE index](#)
- [BYTE version](#)
- [IPv4Addr \\* pIPv4SrcAddr](#)
- [IPv4Addr \\* pIPv4DstAddr](#)
- [BYTE \\* pNxtHdrProto](#)
- [Tos \\* pTos](#)
- [IPv6Addr \\* pIPv6SrcAddr](#)
- [IPv6Addr \\* pIPv6DstAddr](#)
- [IPv6TrafCls \\* pIPv6TrafCls](#)
- [ULONG \\* pIPv6Label](#)
- [Port \\* pTCPSrcPort](#)
- [Port \\* pTCPDstPort](#)
- [Port \\* pUDPSrcPort](#)
- [Port \\* pUDPDstPort](#)
- [ULONG \\* pEspSpi](#)
- [WORD \\* pPrecedence](#)
- [WORD \\* pld](#)
- [Port \\* pTranSrcPort](#)
- [Port \\* pTranDstPort](#)

### 8.465.1    Detailed Description

This structure contains the QoS Filter Request

## Parameters

<i>index</i>	IP filter index Integer that uniquely identifies each filter instance This TLV must be present in the request
<i>version</i>	IP filter version Identifies whether the filter is associated with IPv4 or IPv6; value specified also implies that only TLVs defined for that IP version, i.e., TLVs with IPv4 or IPv6 in the name, can be specified <ul style="list-style-type: none"> <li>• 0x04 – IPv4</li> <li>• 0x06 – Ipv6</li> </ul>
<i>pIPv4SrcAddr</i>	IPv4 filter soruce address See <a href="#">IPv4Addr</a> for more information <ul style="list-style-type: none"> <li>• Implemented only for unsolicited indication</li> </ul>
<i>pIPv4DstAddr</i>	IPv4 filter destination address See <a href="#">IPv4Addr</a> for more information <ul style="list-style-type: none"> <li>• Implemented only for unsolicited indication</li> </ul>
<i>pNxtHdrProto</i>	IP filter next header protocol This TLV must be present if any non-IP filter TLV(s) are provided If this field is specified, only IP packets belonging to specified higher layer protocol are considered when filtering The following protocols may be specified: <ul style="list-style-type: none"> <li>• 0x01 = ICMP</li> <li>• 0x06 = TCP</li> <li>• 0x11 = UDP</li> <li>• 0x32 = ESP Note: The next header protocol field will be set to 0xFD (TCP &amp; UDP) if a TFT is received specifying a source or destination port number, but IP next header type is not specified.</li> </ul>
<i>pTos</i>	IPv4 filter type of service See <a href="#">Tos</a> for more information
<i>pIPv6SrcAddr</i>	IPv6 filter soruce address See <a href="#">IPv6Addr</a> for more information <ul style="list-style-type: none"> <li>• Implemented only for unsolicited indication</li> </ul>
<i>pIPv6DstAddr</i>	IPv6 filter destination address See <a href="#">IPv6Addr</a> for more information <ul style="list-style-type: none"> <li>• Implemented only for unsolicited indication</li> </ul>
<i>pIPv6TrafCls</i>	IPv6 filter traffic class See <a href="#">IPv6TrafCls</a> for more information
<i>pIPv6Label</i>	IPv6 flow label Packet matches the IPv6 flow label filter if: ( *pIPv6Label == flow label in the IPv6 header) <ul style="list-style-type: none"> <li>• Implemented only for unsolicited indication</li> </ul>
<i>pTCPSrcPort</i>	TCP filter source port filter See <a href="#">Port</a> for more information <ul style="list-style-type: none"> <li>• Implemented only for unsolicited indication</li> </ul>

<i>pTCPDstPort</i>	TCP filter destination port filter See <a href="#">Port</a> for more information <ul style="list-style-type: none"> <li>Implemented only for unsolicited indication</li> </ul>
<i>pUDPSrcPort</i>	UDP filter source port filter See <a href="#">Port</a> for more information <ul style="list-style-type: none"> <li>Implemented only for unsolicited indication</li> </ul>
<i>pUDPDstPort</i>	UDP filter destination port filter See <a href="#">Port</a> for more information <ul style="list-style-type: none"> <li>Implemented only for unsolicited indication</li> </ul>
<i>pEspSpi</i>	ESP filter security policy index Security policy index to uniquely identify each IP flow for filtering encrypted packets for encapsulating security payload <ul style="list-style-type: none"> <li>Implemented only for unsolicited indication</li> </ul>
<i>pPrecedence</i>	Filter Precedence Specifies the order in which filters are applied; lower numerical value has higher precedence Note: This TLV only applies to network-initiated QoS; QoS requests containing this TLV from control points will be ignored
<i>pId</i>	Filter ID Unique identifier for each filter; filter ID is assigned by the modem Note: This TLV only applies to network-initiated QoS; QoS requests containing this TLV from control points will be ignored
<i>pTranSrcPort</i>	Transport protocol filter source port See <a href="#">Port</a> for more information <ul style="list-style-type: none"> <li>Implemented only for unsolicited indication</li> </ul>
<i>pUDPDstPort</i>	Transport protocol filter destination port See <a href="#">Port</a> for more information <ul style="list-style-type: none"> <li>Implemented only for unsolicited indication</li> </ul>

## 8.465.2 Field Documentation

8.465.2.1 **BYTE** swiQosFilter::index

8.465.2.2 **ULONG\*** swiQosFilter::pEspSpi

8.465.2.3 **WORD\*** swiQosFilter::pId

8.465.2.4 **IPv4Addr\*** swiQosFilter::pIPv4DstAddr

8.465.2.5 **IPv4Addr\*** swiQosFilter::pIPv4SrcAddr

8.465.2.6 **IPv6Addr\*** swiQosFilter::pIPv6DstAddr

8.465.2.7 **ULONG\*** swiQosFilter::pIPv6Label

8.465.2.8 **IPv6Addr\*** swiQosFilter::pIPv6SrcAddr

8.465.2.9 **IPv6TrafCls\*** swiQosFilter::pIPv6TrafCls

8.465.2.10 **BYTE\*** swiQosFilter::pNextHdrProto

8.465.2.11 **WORD\*** swiQosFilter::pPrecedence

8.465.2.12 **Port\*** swiQosFilter::pTCPDstPort

8.465.2.13 **Port\*** swiQosFilter::pTCPSrcPort

8.465.2.14 **Tos\*** swiQosFilter::pTos

8.465.2.15 **Port\*** swiQosFilter::pTranDstPort

8.465.2.16 **Port\*** swiQosFilter::pTranSrcPort

8.465.2.17 **Port\*** swiQosFilter::pUDPDstPort

8.465.2.18 **Port\*** swiQosFilter::pUDPSrcPort

8.465.2.19 **BYTE** swiQosFilter::version

## 8.466 swiQosFlow Struct Reference

### Data Fields

- [BYTE](#) *index*
- [WORD](#) \* [pProfileId3GPP2](#)
- [BYTE](#) \* [p3GPP2Pri](#)
- [BYTE](#) \* [pTrafficClass](#)
- [dataRate](#) \* [pDataRate](#)
- [tokenBucket](#) \* [pTokenBucket](#)
- [ULONG](#) \* [pLatency](#)
- [ULONG](#) \* [pJitter](#)
- [pktErrRate](#) \* [pPktErrRate](#)
- [ULONG](#) \* [pMinPolicedPktSz](#)
- [ULONG](#) \* [pMaxAllowedPktSz](#)
- [WORD](#) \* [p3GPPResResidualBER](#)
- [BYTE](#) \* [p3GPPTraHdlPri](#)
- [BYTE](#) \* [p3GPPImCn](#)
- [BYTE](#) \* [p3GPPSigInd](#)
- [BYTE](#) \* [pLteQci](#)

### 8.466.1 Detailed Description

This structure contains the QoS Flow Request

#### Parameters

<i>index</i>	<ul style="list-style-type: none"><li>• IP flow index</li><li>• Integer that uniquely identifies each flow instance</li><li>• Unique index must be assigned by the control point to every flow_spec instance</li></ul>
--------------	--

<i>pProfileId3GPP2</i>	<ul style="list-style-type: none"> <li>• IP flow 3GPP2 profile ID</li> <li>• A profile ID is shorthand for a defined set of QoS flow parameters specified by the network; to be present while requesting QoS for a CDMA device</li> </ul>
<i>p3GPP2Pri</i>	<ul style="list-style-type: none"> <li>• IP flow 3GPP2 flow priority</li> <li>• Flow priority used by the network in case of contention between flows with same QoS; this parameter applies for CDMA devices</li> </ul>
<i>pTrafficClass</i>	<ul style="list-style-type: none"> <li>• IP flow traffic class</li> <li>• Integer that designates the requested traffic class:</li> <li>• 0 – Conversational</li> <li>• 1 – Streaming</li> <li>• 2 – Interactive</li> <li>• 3 – Background</li> </ul>
<i>pDataRate</i>	<ul style="list-style-type: none"> <li>• IP flow data rate min max</li> <li>• See <a href="#">dataRate</a> for more information</li> </ul>
<i>pTokenBucket</i>	<ul style="list-style-type: none"> <li>• IP flow data rate token bucket</li> <li>• See <a href="#">tokenBucket</a> for more information</li> </ul>
<i>pLatency</i>	<ul style="list-style-type: none"> <li>• IP flow latency</li> <li>• Maximum delay (in milliseconds) that can be tolerated by an IP packet during transfer through the wireless link</li> </ul>
<i>pJitter</i>	<ul style="list-style-type: none"> <li>• IP flow jitter</li> <li>• Difference between the maximum and minimum latency (in milliseconds) that can be tolerated by an IP packet during the transfer through the wireless link</li> </ul>
<i>pPktErrRate</i>	<ul style="list-style-type: none"> <li>• IP flow packet error rate</li> <li>• See <a href="#">pktErrRate</a> for more information</li> </ul>
<i>pMinPolicedPkt-Sz</i>	<ul style="list-style-type: none"> <li>• IP flow minimum policed packet size</li> <li>• Integer that defines the minimum packet size (in bytes) that will be policed for QoS guarantees; any IP packets that are smaller than the minimum specified policed size may not receive requested QoS</li> </ul>

<i>pMaxAllowed-PktSz</i>	<ul style="list-style-type: none"> <li>• IP flow maximum allowed packet size</li> <li>• Integer that defines the maximum packet size (in bytes) allowed in the IP flow; any IP packets greater in size than the maximum allowed packet size are not queued for transmission</li> </ul>
<i>p3GPPRes-ResidualBER</i>	<ul style="list-style-type: none"> <li>• IP flow 3GPP residual bit error rate</li> <li>• residual_bit_error_rate</li> <li>• 0 = <math>5 \times 10^{-2}</math> residual BER</li> <li>• 1 = <math>1 \times 10^{-2}</math> residual BER</li> <li>• 2 = <math>5 \times 10^{-3}</math> residual BER</li> <li>• 3 = <math>4 \times 10^{-3}</math> residual BER</li> <li>• 4 = <math>1 \times 10^{-3}</math> residual BER</li> <li>• 5 = <math>1 \times 10^{-4}</math> residual BER</li> <li>• 6 = <math>1 \times 10^{-5}</math> residual BER</li> <li>• 7 = <math>1 \times 10^{-6}</math> residual BER</li> <li>• 8 = <math>6 \times 10^{-8}</math> residual BER</li> <li>• Integer that indicates the undetected BER for each IP flow in the delivered packets; Applies only to 3GPP networks</li> </ul>
<i>p3GPPTraHdlPri</i>	<ul style="list-style-type: none"> <li>• 3GPP traffic handling priority</li> <li>• 0 – Relative traffic handling priority 1</li> <li>• 1 – Relative traffic handling priority 2</li> <li>• 2 – Relative traffic handling priority 3</li> <li>• Defines the relative priority of the flow; applies only to 3GPP networks</li> </ul>
<i>p3GPPImCn</i>	<ul style="list-style-type: none"> <li>• IP flow 3GPP IM CN flag</li> <li>• IM CN subsystem signaling flag:</li> <li>• 0x00 – FALSE</li> <li>• 0x01 – TRUE</li> <li>• This parameter applies only to 3GPP networks</li> </ul>

<i>p3GPPSigInd</i>	<ul style="list-style-type: none"> <li>• IP flow 3GPP signaling indication</li> <li>• 0x00 – FALSE</li> <li>• 0x01 – TRUE</li> <li>• This parameter applies only to 3GPP networks</li> </ul>
<i>pLteQci</i>	<ul style="list-style-type: none"> <li>• LTE QoS Class Identifier</li> <li>• QoS Class Identifier(QCI) is a required parameter to request QoS in LTE</li> <li>• QCI values: <ul style="list-style-type: none"> <li>– QCI value 0 requests the network to assign the appropriate QCI value</li> <li>– QCI values 1-4 are associated with guaranteed bitrates</li> <li>– QCI values 5-9 are associated with nonguaranteed bitrates, so the values specified as guaranteed and maximum bitrates are ignored</li> </ul> </li> </ul>

## 8.466.2 Field Documentation

8.466.2.1 **BYTE** swiQosFlow::index

8.466.2.2 **BYTE\*** swiQosFlow::p3GPP2Pri

8.466.2.3 **BYTE\*** swiQosFlow::p3GPPIImCn

8.466.2.4 **WORD\*** swiQosFlow::p3GPPResResidualBER

8.466.2.5 **BYTE\*** swiQosFlow::p3GPPSigInd

8.466.2.6 **BYTE\*** swiQosFlow::p3GPPTraHdlPri

8.466.2.7 **dataRate\*** swiQosFlow::pDataRate

8.466.2.8 **ULONG\*** swiQosFlow::pJitter

8.466.2.9 **ULONG\*** swiQosFlow::pLatency

8.466.2.10 **BYTE\*** swiQosFlow::pLteQci

8.466.2.11 **ULONG\*** swiQosFlow::pMaxAllowedPktSz

8.466.2.12 **ULONG\*** swiQosFlow::pMinPolicedPktSz

8.466.2.13 **pktErrRate\*** swiQosFlow::pPktErrRate

8.466.2.14 **WORD\*** swiQosFlow::pProfileId3GPP2

8.466.2.15 **tokenBucket\*** swiQosFlow::pTokenBucket

8.466.2.16 **BYTE\*** swiQosFlow::pTrafficClass

## 8.467 swiQosGranted Struct Reference

### Data Fields

- [swiQosFlow](#) \* [pTxFlow](#)
- [swiQosFlow](#) \* [pRxFlow](#)

### 8.467.1 Detailed Description

This structure contains the QoS granted flow

#### Parameters

<i>pTxFlow</i>	See <a href="#">swiQosFlow</a> for more information
<i>pRxFlow</i>	See <a href="#">swiQosFlow</a> for more information

### 8.467.2 Field Documentation

8.467.2.1 [swiQosFlow](#)\* [swiQosGranted::pRxFlow](#)

8.467.2.2 [swiQosFlow](#)\* [swiQosGranted::pTxFlow](#)

## 8.468 swiQosIds Struct Reference

### Data Fields

- [BYTE](#) *sz*
- [ULONG](#) \* *plds*

### 8.468.1 Detailed Description

This structure contains the QoS Response parameters.

#### Parameters

<i>sz</i>	Number of QoS identifiers
<i>plds</i>	Identifier for the QoS flow requested; number of QoS identifiers present will be equal to number of QoS specs requested in the QoS Request message

### 8.468.2 Field Documentation

8.468.2.1 [ULONG](#)\* [swiQosIds::plds](#)

8.468.2.2 [BYTE](#) [swiQosIds::sz](#)

## 8.469 swiQosModifyReq Struct Reference

### Data Fields

- [ULONG](#) *id*
- [swiQosFlow](#) \* [pTxFlow](#)
- [swiQosFlow](#) \* [pRxFlow](#)
- [swiQosFilter](#) \* [pTxFilter](#)



- [swiQosFilter](#) \* [pRxFilter](#)

### 8.469.1 Detailed Description

This structure contains the QoS Request parameters.

#### Parameters

<i>id</i>	Identifier for the QoS flow/instance that has been negotiated and that needs to be modified The QoS_identifier is used to reference the actual flow/filter specifications that are in effect as a result of the negotiation triggered by QMI_QOS_REQUEST_QOS_REQ
<i>pTxFlow</i>	See <a href="#">swiQosFlow</a> for more information
<i>pRxFlow</i>	See <a href="#">swiQosFlow</a> for more information
<i>pTxFilter</i>	See <a href="#">swiQosFilter</a> for more information
<i>pRxFilter</i>	See <a href="#">swiQosFilter</a> for more information

### 8.469.2 Field Documentation

8.469.2.1 **ULONG** swiQosModifyReq::id

8.469.2.2 **swiQosFilter\*** swiQosModifyReq::pRxFilter

8.469.2.3 **swiQosFlow\*** swiQosModifyReq::pRxFlow

8.469.2.4 **swiQosFilter\*** swiQosModifyReq::pTxFilter

8.469.2.5 **swiQosFlow\*** swiQosModifyReq::pTxFlow

## 8.470 swiQosReq Struct Reference

### Data Fields

- [BYTE](#) index
- [swiQosFlow](#) \* [pTxFlow](#)
- [swiQosFlow](#) \* [pRxFlow](#)
- [swiQosFilter](#) \* [pTxFilter](#)
- [swiQosFilter](#) \* [pRxFilter](#)

### 8.470.1 Detailed Description

This structure contains the QoS Request parameters.

#### Parameters

<i>index</i>	<ul style="list-style-type: none"> <li>• An integer that uniquely identifies each QoS spec included in the QoS request message</li> </ul>
<i>pTxFlow</i>	<ul style="list-style-type: none"> <li>• See <a href="#">swiQosFlow</a> for more information</li> </ul>

<i>pRxFlow</i>	<ul style="list-style-type: none"> <li>• See <a href="#">swiQosFlow</a> for more information</li> </ul>
<i>pTxFilter</i>	<ul style="list-style-type: none"> <li>• See <a href="#">swiQosFilter</a> for more information</li> </ul>
<i>pRxFilter</i>	<ul style="list-style-type: none"> <li>• See <a href="#">swiQosFilter</a> for more information</li> </ul>

## 8.470.2 Field Documentation

### 8.470.2.1 BYTE swiQosReq::index

### 8.470.2.2 swiQosFilter\* swiQosReq::pRxFilter

### 8.470.2.3 swiQosFlow\* swiQosReq::pRxFlow

### 8.470.2.4 swiQosFilter\* swiQosReq::pTxFilter

### 8.470.2.5 swiQosFlow\* swiQosReq::pTxFlow

## 8.471 swiRMTrasferStaticsReq Struct Reference

### Data Fields

- [BYTE bResetStatistics](#)
- [ULONG ulMask](#)

### 8.471.1 Detailed Description

RM Transfer Statistics Structure

#### Parameters

<i>bResetStatistics</i>	<ul style="list-style-type: none"> <li>• Reset Statistics</li> <li>• Values: <ul style="list-style-type: none"> <li>• 0 - Not Reset</li> <li>• Other - Reset</li> </ul> </li> </ul>
-------------------------	---

<i>ulMask</i>	<ul style="list-style-type: none"><li>• Enable/Disable RM Transfer Statistics Indiscation Mask</li><li>• Bit 0: Tx Packet Ok</li><li>• Bit 1: Rx Packet Ok</li><li>• Bit 2: Tx Bytes Ok</li><li>• Bit 3: Rx Bytes Ok</li><li>• Bit 4: Tx Packets Dropped</li><li>• Bit 5: Rx Packets Dropped</li><li>• Value: -0 - Disable -1 - Enable</li></ul>
---------------	--

### 8.471.2 Field Documentation

8.471.2.1 **BYTE** swiRMTrasnferStaticsReq::bResetStatistics

8.471.2.2 **ULONG** swiRMTrasnferStaticsReq::ulMask

## 8.472 sysInfoCommon Struct Reference

### Data Fields

- [BYTE](#) *srvDomainValid*
- [BYTE](#) *srvDomain*
- [BYTE](#) *srvCapabilityValid*
- [BYTE](#) *srvCapability*
- [BYTE](#) *roamStatusValid*
- [BYTE](#) *roamStatus*
- [BYTE](#) *isSysForbiddenValid*
- [BYTE](#) *isSysForbidden*

### 8.472.1 Detailed Description

Structure for storing the System Information common to CDMA, HDR, GSM, WCDMA and LTE networks.

#### Parameters

<i>srvDomainValid</i>	<ul style="list-style-type: none"><li>• Indicates whether the service domain is valid.<ul style="list-style-type: none"><li>– 0x00 - Invalid</li><li>– 0x01 - Valid</li><li>– 0xFF - Not Available</li></ul></li></ul>
-----------------------	--

<i>srvDomain</i>	<ul style="list-style-type: none"> <li>• Service domain registered on the system. <ul style="list-style-type: none"> <li>– 0x00 - No service</li> <li>– 0x01 - Circuit-switched only</li> <li>– 0x02 - Packet-switched only</li> <li>– 0x03 - Circuit-switched and packet-switched</li> <li>– 0x04 - Camped</li> <li>– 0xFF - Not Available</li> </ul> </li> </ul>
<i>srvCapability-Valid</i>	<ul style="list-style-type: none"> <li>• Indicates whether the service capability is valid. <ul style="list-style-type: none"> <li>– 0x00 - Invalid</li> <li>– 0x01 - Valid</li> <li>– 0xFF - Not Available</li> </ul> </li> </ul>
<i>srvCapability</i>	<ul style="list-style-type: none"> <li>• Current system's service capability. <ul style="list-style-type: none"> <li>– 0x00 - No service</li> <li>– 0x01 - Circuit-switched only</li> <li>– 0x02 - Packet-switched only</li> <li>– 0x03 - Circuit-switched and packet-switched</li> <li>– 0x04 - Camped</li> <li>– 0xFF - Not Available</li> </ul> </li> </ul>
<i>roamStatusValid</i>	<ul style="list-style-type: none"> <li>• Indicates whether the roaming status is valid. <ul style="list-style-type: none"> <li>– 0x00 - Invalid</li> <li>– 0x01 - Valid</li> <li>– 0xFF - Not Available</li> </ul> </li> </ul>
<i>roamStatus</i>	<ul style="list-style-type: none"> <li>• Current roaming status. <ul style="list-style-type: none"> <li>– 0x00 - Off</li> <li>– 0x01 - On</li> <li>– 0x02 - Blinking</li> <li>– 0x03 - Out of the neighborhood</li> <li>– 0x04 - Out of the building</li> <li>– 0x05 - Preferred system</li> <li>– 0x06 - Available system</li> <li>– 0x07 - Alliance partner</li> <li>– 0x08 - Premium partner</li> <li>– 0x09 - Full service</li> <li>– 0x0A - Partial service</li> </ul> </li> </ul>
	<ul style="list-style-type: none"> <li>– 0x0B - Banner is on</li> <li>– 0x0C - Banner is off</li> <li>– 0x0D to 0x3F - Reserved for Standard Enhanced Roaming Indicator Numbers</li> <li>– 0x40 to 0x7F - Reserved for Non-Standard Enhanced Roaming Indicator Num-</li> </ul>

<i>isSysForbidden-Valid</i>	<ul style="list-style-type: none"> <li>Indicates whether the forbidden system is valid. <ul style="list-style-type: none"> <li>0x00 - Invalid</li> <li>0x01 - Valid</li> <li>0xFF - Not Available</li> </ul> </li> </ul>
<i>isSysForbidden</i>	<ul style="list-style-type: none"> <li>Whether the system is forbidden. <ul style="list-style-type: none"> <li>0x00 - Not forbidden</li> <li>0x01 - Forbidden</li> <li>0xFF - Not Available</li> </ul> </li> </ul>

## 8.472.2 Field Documentation

8.472.2.1 **BYTE** sysInfoCommon::isSysForbidden

8.472.2.2 **BYTE** sysInfoCommon::isSysForbiddenValid

8.472.2.3 **BYTE** sysInfoCommon::roamStatus

8.472.2.4 **BYTE** sysInfoCommon::roamStatusValid

8.472.2.5 **BYTE** sysInfoCommon::srvCapability

8.472.2.6 **BYTE** sysInfoCommon::srvCapabilityValid

8.472.2.7 **BYTE** sysInfoCommon::srvDomain

8.472.2.8 **BYTE** sysInfoCommon::srvDomainValid

## 8.473 TDSCDMAECIOThresh Struct Reference

### Data Fields

- [BYTE TDSCDMAECIOThreshListLen](#)
- [ULONG \\* pTDSCDMAECIOThreshList](#)

### 8.473.1 Detailed Description

This structure contains TDSCDMA ECIO threshold related parameters.

#### Parameters

<i>TDSCDMAECIOThreshListLen</i>	<ul style="list-style-type: none"> <li>Length of the TDSCDMA ECIO threshold list parameter to follow</li> </ul>
---------------------------------	---

<i>pTDSCDMAECIOThreshList</i>	<ul style="list-style-type: none"> <li>• Array of ECIO thresholds (in dB) used by TD-SCDMA</li> <li>• Maximum of 32 values.</li> </ul>
-------------------------------	--

### 8.473.2 Field Documentation

8.473.2.1 **ULONG\*** TDSCDMAECIOThresh::pTDSCDMAECIOThreshList

8.473.2.2 **BYTE** TDSCDMAECIOThresh::TDSCDMAECIOThreshListLen

## 8.474 TDSCDMARSCPThresh Struct Reference

### Data Fields

- [BYTE](#) TDSCDMARSCPThreshListLen
- [WORD](#) \* pTDSCDMARSCPThreshList

### 8.474.1 Detailed Description

This structure contains TDSCDMA RSCP threshold related parameters.

#### Parameters

<i>TDSCDMARSCPThreshListLen</i>	<ul style="list-style-type: none"> <li>• Length of the TDSCDMA RSCP threshold list parameter to follow</li> </ul>
<i>pTDSCDMARSCPThreshList</i>	<ul style="list-style-type: none"> <li>• Array of RSCP thresholds (in units of 0.1 dBm)</li> <li>• Maximum of 32 values</li> <li>• Range for RSCP values: -120 to -25 (in dBm).</li> </ul>

### 8.474.2 Field Documentation

8.474.2.1 **WORD\*** TDSCDMARSCPThresh::pTDSCDMARSCPThreshList

8.474.2.2 **BYTE** TDSCDMARSCPThresh::TDSCDMARSCPThreshListLen

## 8.475 TDSCDMARSSIThresh Struct Reference

### Data Fields

- [BYTE](#) TDSCDMARSSIThreshListLen
- [ULONG](#) \* pTDSCDMARSSIThreshList

### 8.475.1 Detailed Description

This structure contains TDSCDMA RSSI threshold related parameters.

## Parameters

<i>TDSCDMARSSIThreshListLen</i>	<ul style="list-style-type: none"> <li>Length of the TDSCDMA RSSI threshold list parameter to follow</li> </ul>
<i>pTDSCDMARSSIThreshList</i>	<ul style="list-style-type: none"> <li>Array of RSSI thresholds (in dBm) used by TD-SCDMA</li> <li>Maximum of 32 values.</li> </ul>

## 8.475.2 Field Documentation

8.475.2.1 `ULONG*` TDSCDMARSSIThresh::pTDSCDMARSSIThreshList8.475.2.2 `BYTE` TDSCDMARSSIThresh::TDSCDMARSSIThreshListLen

## 8.476 TDSCDMASigInfoExt Struct Reference

## Data Fields

- `FLOAT` rssi
- `FLOAT` rscp
- `FLOAT` ecio
- `FLOAT` sinr

## 8.476.1 Detailed Description

This structure contains the TDSCDMA Signal Strength Info Extended

## Parameters

<i>rssi</i>	<ul style="list-style-type: none"> <li>Measured RSSI in dB</li> </ul>
<i>rscp[Optional]</i>	<ul style="list-style-type: none"> <li>Measured RSCP in dBm</li> </ul>
<i>ecio[Optional]</i>	<ul style="list-style-type: none"> <li>Measured ECIO in dBm.</li> </ul>
<i>sinr[Optional]</i>	<ul style="list-style-type: none"> <li>Measured SINR in dB. -15 dB is sent to clients if the actual SINR is less than -15 dB</li> </ul>

## 8.476.2 Field Documentation

8.476.2.1 `FLOAT` TDSCDMASigInfoExt::ecio8.476.2.2 `FLOAT` TDSCDMASigInfoExt::rscp8.476.2.3 `FLOAT` TDSCDMASigInfoExt::rssi

## 8.476.2.4 FLOAT TDSCDMASigInfoExt::sinr

## 8.477 TDSCDMASINRThresh Struct Reference

## Data Fields

- [BYTE TDSCDMASINRThreshListLen](#)
- [ULONG \\* pTDSCDMASINRThreshList](#)

## 8.477.1 Detailed Description

This structure contains TDSCDMA SINR threshold related parameters.

## Parameters

<i>TDSCDMASIN-RThreshListLen</i>	<ul style="list-style-type: none"> <li>• Length of the TDSCDMA SINR threshold list parameter to follow</li> </ul>
<i>pTDSCDMASIN-RThreshList</i>	<ul style="list-style-type: none"> <li>• Array of SINR thresholds (in dB) used by TD-SCDMA</li> <li>• Maximum of 32 values</li> </ul>

## 8.477.2 Field Documentation

## 8.477.2.1 ULONG\* TDSCDMASINRThresh::pTDSCDMASINRThreshList

## 8.477.2.2 BYTE TDSCDMASINRThresh::TDSCDMASINRThreshListLen

## 8.478 TFTIDParams Struct Reference

## Data Fields

- [BYTE filterId](#)
- [BYTE eValid](#)
- [BYTE ipVersion](#)
- [WORD \\* pSourceIP](#)
- [BYTE sourceIPMask](#)
- [BYTE nextHeader](#)
- [WORD destPortRangeStart](#)
- [WORD destPortRangeEnd](#)
- [WORD srcPortRangeStart](#)
- [WORD srcPortRangeEnd](#)
- [ULONG IPSECSPi](#)
- [WORD tosMask](#)
- [ULONG flowLabel](#)

## 8.478.1 Detailed Description

structure contains traffic flow template parameters

- Parameter values default to their data type's maximum unsigned value unless explicitly stated otherwise.



## Parameters

<i>filterId</i>	<ul style="list-style-type: none"> <li>Filter identifier</li> </ul>
<i>eValid</i>	<ul style="list-style-type: none"> <li>Evaluation precedence index</li> </ul>
<i>pVersion</i>	<ul style="list-style-type: none"> <li>IP version number <ul style="list-style-type: none"> <li>4 - IPv4</li> <li>6 - IPv6</li> </ul> </li> </ul>
<i>sourceIP</i>	<ul style="list-style-type: none"> <li>Source IP address <ul style="list-style-type: none"> <li>IPv4 - Fill the first 4 bytes</li> <li>IPv6 - Fill all the 16 bytes</li> </ul> </li> </ul>
<i>sourceIPMask</i>	<ul style="list-style-type: none"> <li>Mask value for the source address</li> </ul>
<i>nextHeader</i>	<ul style="list-style-type: none"> <li>Next header/protocol value</li> </ul>
<i>destPortRange-Start</i>	<ul style="list-style-type: none"> <li>Start value of the destination port range</li> </ul>
<i>destPortRange-End</i>	<ul style="list-style-type: none"> <li>End value of the destination port range</li> </ul>
<i>srcPortRange-Start</i>	<ul style="list-style-type: none"> <li>Start value of the source port range</li> </ul>
<i>srcPortRange-End</i>	<ul style="list-style-type: none"> <li>End value of the source port range</li> </ul>
<i>IPSECSPi</i>	<ul style="list-style-type: none"> <li>IPSEC security parameter index</li> </ul>
<i>tosMask</i>	<ul style="list-style-type: none"> <li>TOS mask (Traffic class for IPv6)</li> </ul>
<i>flowLabel</i>	<ul style="list-style-type: none"> <li>Flow label</li> </ul>

## 8.478.2 Field Documentation

- 8.478.2.1 WORD TFTIDParams::destPortRangeEnd
- 8.478.2.2 WORD TFTIDParams::destPortRangeStart
- 8.478.2.3 BYTE TFTIDParams::eValid
- 8.478.2.4 BYTE TFTIDParams::filterId
- 8.478.2.5 ULONG TFTIDParams::flowLabel
- 8.478.2.6 ULONG TFTIDParams::IPSECSPi
- 8.478.2.7 BYTE TFTIDParams::ipVersion
- 8.478.2.8 BYTE TFTIDParams::nextHeader
- 8.478.2.9 WORD\* TFTIDParams::pSourceIP
- 8.478.2.10 BYTE TFTIDParams::sourceIPMask
- 8.478.2.11 WORD TFTIDParams::srcPortRangeEnd
- 8.478.2.12 WORD TFTIDParams::srcPortRangeStart
- 8.478.2.13 WORD TFTIDParams::tosMask

## 8.479 tokenBucket Struct Reference

### Data Fields

- [ULONG peakRate](#)
- [ULONG tokenRate](#)
- [ULONG bucketSz](#)

### 8.479.1 Detailed Description

This structure contains the TP flow data rate token bucket

#### Parameters

<i>peakRate</i>	Maximum rate at which data can be transmitted when the token bucket is full (bits per second)
<i>tokenRate</i>	Rate at which tokens will be put in the token bucket (bits per second); a token is required to be present in the bucket to send a byte of data
<i>bucketSz</i>	Maximum number of tokens that can be accumulated at any instance (bytes); controls the size of the burst that is allowed at any given time

### 8.479.2 Field Documentation

- 8.479.2.1 ULONG tokenBucket::bucketSz
- 8.479.2.2 ULONG tokenBucket::peakRate
- 8.479.2.3 ULONG tokenBucket::tokenRate

## 8.480 Tos Struct Reference

### Data Fields

- [BYTE val](#)
- [BYTE mask](#)

### 8.480.1 Detailed Description

This structure contains the IPv4 filter type of service

#### Parameters

<i>val</i>	Type of service value
<i>mask</i>	Packet matches the TOS filter if: (IPv4_filter_tos_val and IPv4_filter_tos_mask) == (TOS value in the IP packet & IPv4_filter_tos_mask) Example: <ul style="list-style-type: none"> <li>• IPv4_filter_tos_val = 00101000</li> <li>• IPv4_filter_tos_mask = 11111100 The filter will compare only the first 6 bits in the IPv4_filter_type_of_service with the first 6 bits in the TOS field of the IP packet. The first 6 bits in the TOS field of the IP packet must be 001010 to match the filter. The last 2 bits can be anything since they are ignored by filtering.</li> </ul>

### 8.480.2 Field Documentation

#### 8.480.2.1 BYTE Tos::mask

#### 8.480.2.2 BYTE Tos::val

## 8.481 TransferStatInd Struct Reference

### Data Fields

- [BYTE StatsPeriod](#)
- [ULONG StatsMask](#)

### 8.481.1 Detailed Description

This structure contains Transfer Statistics Indicator

#### Parameters

<i>StatsPeriod</i>	<ul style="list-style-type: none"> <li>• Period between transfer statistics reports               <ul style="list-style-type: none"> <li>– 0 - Do not report</li> <li>– Other - Period between reports (seconds)</li> </ul> </li> </ul>
--------------------	---

<i>StatsMask</i>	<ul style="list-style-type: none"> <li>Requested statistic bit mask. Each bit set causes the corresponding optional TLV to be sent in the indication. All unlisted bits are reserved for future use and must be set to zero. <ul style="list-style-type: none"> <li>0x00000040 - Tx bytes OK</li> <li>0x00000080 - Rx bytes OK</li> </ul> </li> </ul>
------------------	---

## 8.481.2 Field Documentation

8.481.2.1 **ULONG** TransferStatInd::StatsMask

8.481.2.2 **BYTE** TransferStatInd::StatsPeriod

## 8.482 TransferStatsDataType Struct Reference

### Data Fields

- [BYTE interval](#)

## 8.482.1 Field Documentation

8.482.1.1 **BYTE** TransferStatsDataType::interval

## 8.483 TrStatInd Struct Reference

### Data Fields

- [BYTE statsPeriod](#)
- [ULONG statsMask](#)

## 8.483.1 Detailed Description

This structure contains the information about the Transfer Statistics Indicator parameters.

### Parameters

<i>statsPeriod</i>	<ul style="list-style-type: none"> <li>Period between transfer statistics reports. <ul style="list-style-type: none"> <li>0 - Do not report</li> <li>Other - Period between reports (seconds)</li> </ul> </li> </ul>
--------------------	--

<i>statsMask</i>	<ul style="list-style-type: none"> <li>Requested statistic bit mask. <ul style="list-style-type: none"> <li>0x00000001 - Tx packets OK</li> <li>0x00000002 - Rx packets OK</li> <li>0x00000004 - Tx packet errors</li> <li>0x00000008 - Rx packet errors</li> <li>0x00000010 - Tx overflows</li> <li>0x00000020 - Rx overflows</li> <li>0x00000040 - Tx bytes OK</li> <li>0x00000080 - Rx bytes OK</li> </ul> </li> <li>Each bit set causes the corresponding optional information to be sent in SLQSWds-EventReportCallBack.</li> <li>All unlisted bits are reserved for future use and must be set to zero.</li> </ul>
------------------	--

## 8.483.2 Field Documentation

### 8.483.2.1 ULONG TrStatInd::statsMask

### 8.483.2.2 BYTE TrStatInd::statsPeriod

## 8.484 trueIMSI Struct Reference

### Data Fields

- BYTE [mccT](#) [3]
- WORD [imsiT1112](#)
- BYTE [imsiTS1](#) [7]
- BYTE [imsiTS2](#) [3]
- BYTE [imsiTaddrNum](#)

### 8.484.1 Detailed Description

This structure contains the parameters for True IMSI Information

#### Parameters

<i>mccT</i>	<ul style="list-style-type: none"> <li>ASCII character representation of MCC_T</li> </ul>
<i>imsiT1112</i>	<ul style="list-style-type: none"> <li>ASCII character representation of IMSI_T_11_12 value <ul style="list-style-type: none"> <li>0xFFFF - Not Available</li> </ul> </li> </ul>

<i>imsiTS1</i>	<ul style="list-style-type: none"> <li>• ASCII character representation of IMSI_T_S1 value</li> </ul>
<i>imsiTS2</i>	<ul style="list-style-type: none"> <li>• ASCII character representation of IMSI_T_S2 value</li> </ul>
<i>imsiTaddrNum</i>	<ul style="list-style-type: none"> <li>• Value of IMSI_T_ADDR_NUM <ul style="list-style-type: none"> <li>– 0xFF - Not Available</li> </ul> </li> </ul>

## 8.484.2 Field Documentation

8.484.2.1 WORD trueIMSI::imsiT1112

8.484.2.2 BYTE trueIMSI::imsiTaddrNum

8.484.2.3 BYTE trueIMSI::imsiTS1[7]

8.484.2.4 BYTE trueIMSI::imsiTS2[3]

8.484.2.5 BYTE trueIMSI::mccT[3]

## 8.485 TXAGCList Struct Reference

### Data Fields

- WORD \* [pTXStaticGain](#)
- WORD \* [pTXAIG](#)
- WORD \* [pTXExpThres](#)
- WORD \* [pTXExpSlope](#)
- WORD \* [pTXComprThres](#)
- WORD \* [pTXComprSlope](#)

### 8.485.1 Detailed Description

This structure contains the SLQSGetAudioPathConfig parameters related to AV\_TXAGCLIST.

#### Parameters

<i>pTXStaticGain</i>	<ul style="list-style-type: none"> <li>• TX pre-compressor static gain</li> </ul>
<i>pTXAIG</i>	<ul style="list-style-type: none"> <li>• TX pre-compressor gain selection flag</li> </ul>
<i>pTXExpThres</i>	<ul style="list-style-type: none"> <li>• TX expansion threshold</li> </ul>

<i>pTXExpSlope</i>	<ul style="list-style-type: none"><li>• TX expansion slope</li></ul>
<i>pTXComprThres</i>	<ul style="list-style-type: none"><li>• TX compression threshold</li></ul>
<i>pTXComprSlope</i>	<ul style="list-style-type: none"><li>• TX compression slope</li></ul>

## 8.485.2 Field Documentation

8.485.2.1 WORD\* TXAGCList::pTXAIG

8.485.2.2 WORD\* TXAGCList::pTXComprSlope

8.485.2.3 WORD\* TXAGCList::pTXComprThres

8.485.2.4 WORD\* TXAGCList::pTXExpSlope

8.485.2.5 WORD\* TXAGCList::pTXExpThres

8.485.2.6 WORD\* TXAGCList::pTXStaticGain

## 8.486 txInfo Struct Reference

### Data Fields

- [BYTE isInTraffic](#)
- [ULONG txPower](#)

### 8.486.1 Detailed Description

This structure contains the Tx Information.

#### Parameters

<i>isInTraffic</i>	<ul style="list-style-type: none"><li>• Whether the device is in traffic.</li><li>• The txPower field is only meaningful when in the device is in traffic.</li><li>• If it is not in traffic, txPower is invalid.<ul style="list-style-type: none"><li>– 0xFF - Not Available</li></ul></li></ul>
--------------------	---

<i>txPower</i>	<ul style="list-style-type: none"><li>• Tx power value in 1/10 dbm.<ul style="list-style-type: none"><li>– 0xFFFFFFFF - Not Available</li></ul></li></ul>
----------------	---

## 8.486.2 Field Documentation

8.486.2.1 **BYTE** txInfo::isInTraffic

8.486.2.2 **ULONG** txInfo::txPower

## 8.487 TXPCMIIRFiltr Struct Reference

### Data Fields

- **WORD** \* pFlag
- **WORD** \* pStageCnt
- **BYTE** \* pStage0Val
- **BYTE** \* pStage1Val
- **BYTE** \* pStage2Val
- **BYTE** \* pStage3Val
- **BYTE** \* pStage4Val

### 8.487.1 Detailed Description

This structure contains the SLQSGetAudioPathConfig parameters related to AV\_TXPCMIIRFLTR.



## Parameters

<i>pFlag</i>	<ul style="list-style-type: none"><li>• Flag<ul style="list-style-type: none"><li>– 0x0000 - IIR filter disable</li><li>– 0xffff - IIR filter enable</li></ul></li></ul>
<i>pStageCnt</i>	<ul style="list-style-type: none"><li>• Stage Count<ul style="list-style-type: none"><li>– 0-4</li></ul></li></ul>
<i>pStage0Val</i>	<ul style="list-style-type: none"><li>• A 20 BYTE sized parameter indicating Stage 0 value<ul style="list-style-type: none"><li>– A1</li><li>– A2</li><li>– B0</li><li>– B1</li><li>– B2</li></ul></li></ul>
<i>pStage1Val</i>	<ul style="list-style-type: none"><li>• A 20 BYTE sized parameter indicating Stage 1 value<ul style="list-style-type: none"><li>– A1</li><li>– A2</li><li>– B0</li><li>– B1</li><li>– B2</li></ul></li></ul>
<i>pStage2Val</i>	<ul style="list-style-type: none"><li>• A 20 BYTE sized parameter indicating Stage 2 value<ul style="list-style-type: none"><li>– A1</li><li>– A2</li><li>– B0</li><li>– B1</li><li>– B2</li></ul></li></ul>

<i>pStage3Val</i>	<ul style="list-style-type: none"> <li>• A 20 BYTE sized parameter indicating Stage 3 value <ul style="list-style-type: none"> <li>– A1</li> <li>– A2</li> <li>– B0</li> <li>– B1</li> <li>– B2</li> </ul> </li> </ul>
<i>pStage4Val</i>	<ul style="list-style-type: none"> <li>• A 20 BYTE sized parameter indicating Stage 4 value <ul style="list-style-type: none"> <li>– A1</li> <li>– A2</li> <li>– B0</li> <li>– B1</li> <li>– B2</li> </ul> </li> </ul>

## 8.487.2 Field Documentation

8.487.2.1 WORD\* TXPCMIIRFitr::pFlag

8.487.2.2 BYTE\* TXPCMIIRFitr::pStage0Val

8.487.2.3 BYTE\* TXPCMIIRFitr::pStage1Val

8.487.2.4 BYTE\* TXPCMIIRFitr::pStage2Val

8.487.2.5 BYTE\* TXPCMIIRFitr::pStage3Val

8.487.2.6 BYTE\* TXPCMIIRFitr::pStage4Val

8.487.2.7 WORD\* TXPCMIIRFitr::pStageCnt

## 8.488 UIMAuthenticateReq Struct Reference

### Data Fields

- [UIMSessionInformation sessionInfo](#)
- [authenticationData authData](#)
- [ULONG \\* pIndicationToken](#)

### 8.488.1 Detailed Description

This structure contains information of the request parameters associated with a Authenticate API.

## Parameters

<a href="#">sessionInfo</a>	<ul style="list-style-type: none"><li>• See <a href="#">UIMSessionInformation</a> for more information.</li></ul>
<i>authData</i>	<ul style="list-style-type: none"><li>• See <a href="#">authenticationData</a> for more information.</li></ul>
<i>pIndication-Token(optional)</i>	<ul style="list-style-type: none"><li>• Response in Indication.</li><li>• When this TLV is present, it indicates that the result must be provided in a subsequent indication.</li></ul>

## Note

Using NULL for the pointers would make sure that the parameter is not added to the request.

## 8.488.2 Field Documentation

8.488.2.1 [authenticationData](#) UIMAuthenticateReq::authData

8.488.2.2 [ULONG\\*](#) UIMAuthenticateReq::pIndicationToken

8.488.2.3 [UIMSessionInformation](#) UIMAuthenticateReq::sessionInfo

## 8.489 UIMAuthenticateResp Struct Reference

## Data Fields

- [cardResult](#) \* [pCardResult](#)
- [authenticateResult](#) \* [pAuthenticateResult](#)
- [ULONG](#) \* [pIndicationToken](#)

## 8.489.1 Detailed Description

This structure contains information of the response parameters associated with a Authenticate API.

## Parameters

<i>pCard-Result(optional)</i>	<ul style="list-style-type: none"><li>• See <a href="#">cardResult</a> for more information.</li></ul>
<i>pAuthenticate-Result(optional)</i>	<ul style="list-style-type: none"><li>• See <a href="#">authenticateResult</a> for more information.</li></ul>
<i>pIndication-Token(optional)</i>	<ul style="list-style-type: none"><li>• Response in Indication.</li><li>• When this TLV is present, it indicates that the result must be provided in a subsequent indication.</li></ul>

## Note

Using NULL for the pointers would make sure that the parameter is not returned.

## 8.489.2 Field Documentation

8.489.2.1 **authenticateResult\*** `UIMAuthenticateResp::pAuthenticateResult`

8.489.2.2 **cardResult\*** `UIMAuthenticateResp::pCardResult`

8.489.2.3 **ULONG\*** `UIMAuthenticateResp::pIndicationToken`

## 8.490 UIMChangePinReq Struct Reference

### Data Fields

- [UIMSessionInformation sessionInfo](#)
- [changeUIMPIN changePIN](#)
- [BYTE \\*](#) `pKeyReferenceID`
- [ULONG \\*](#) `pIndicationToken`

### 8.490.1 Detailed Description

This structure contains information of the request parameters associated with a Change PIN API.

#### Parameters

<a href="#"><i>sessionInfo</i></a>	<ul style="list-style-type: none"> <li>• See <a href="#">UIMSessionInformation</a> for more information.</li> </ul>
<i>changePIN</i>	<ul style="list-style-type: none"> <li>• See <a href="#">changeUIMPIN</a> for more information.</li> </ul>
<i>pKeyReferenceID(optional)</i>	<ul style="list-style-type: none"> <li>• Indicates the PIN key reference ID.</li> <li>• Indicates the PIN key reference ID. Valid values are from 1 to 8, respectively, for application 1 to application 8.</li> <li>• This TLV is used only for PIN1 and PIN2 and is ignored in all other cases.</li> </ul>
<i>pIndicationToken(optional)</i>	<ul style="list-style-type: none"> <li>• Response in Indication.</li> <li>• When this TLV is present, it indicates that the result must be provided in a subsequent indication.</li> </ul>

## Note

Using NULL for the pointers would make sure that the parameter is not added to the request.

## 8.490.2 Field Documentation

8.490.2.1 **changeUIMPIN** `UIMChangePinReq::changePIN`

8.490.2.2 **ULONG\*** UIMChangePinReq::pIndicationToken

8.490.2.3 **BYTE\*** UIMChangePinReq::pKeyReferenceID

8.490.2.4 **UIMSessionInformation** UIMChangePinReq::sessionInfo

## 8.491 UIMDepersonalizationReq Struct Reference

### Data Fields

- [depersonalizationInformation](#) [depersonalisationInfo](#)

### 8.491.1 Detailed Description

This structure contains information of the request parameters associated with a Depersonalization API.

#### Parameters

<i>depersonalisation-Info</i>	<ul style="list-style-type: none"><li>• See <a href="#">depersonalizationInformation</a> for more information.</li></ul>
-------------------------------	--

### 8.491.2 Field Documentation

8.491.2.1 **depersonalizationInformation** UIMDepersonalizationReq::depersonalisationInfo

## 8.492 UIMDepersonalizationResp Struct Reference

### Data Fields

- [remainingRetries](#) \* [pRemainingRetries](#)

### 8.492.1 Detailed Description

This structure contains information of the response parameters associated with a Depersonalization API.

#### Parameters

<i>pRemainingRetries(optional)</i>	<ul style="list-style-type: none"><li>• See <a href="#">remainingRetries</a> for more information.</li></ul>
------------------------------------	--

#### Note

Using NULL for the pointers would make sure that the parameter is not returned.

### 8.492.2 Field Documentation

8.492.2.1 **remainingRetries\*** UIMDepersonalizationResp::pRemainingRetries

## 8.493 UIMEventRegisterReqResp Struct Reference

## Data Fields

- [ULONG eventMask](#)

### 8.493.1 Detailed Description

This structure contains information of the request parameters associated with a SLQSUIMEventRegister.

#### Parameters

<i>eventMask(-Mandatory)</i>	<ul style="list-style-type: none"> <li>• Bitmask of the events that were successfully enabled. This result can be different from the mask used in the request when notifications are not supported. Additional bits are reserved for future use. <ul style="list-style-type: none"> <li>– Bit 0 - Card status</li> <li>– Bit 1 - SAP connection</li> </ul> </li> </ul>
------------------------------	--

### 8.493.2 Field Documentation

#### 8.493.2.1 ULONG UIMEventRegisterReqResp::eventMask

## 8.494 UIMGetCardStatusResp Struct Reference

## Data Fields

- [cardStatus](#) \* [pCardStatus](#)
- [hotSwapStatus](#) \* [pHotSwapStatus](#)

### 8.494.1 Detailed Description

This structure contains information of the response parameters associated with a Get Card Status API.

#### Parameters

<i>pCard-Status(optional)</i>	<ul style="list-style-type: none"> <li>• See <a href="#">cardStatus</a> for more information.</li> </ul>
<i>pHotSwap-Status(optional)</i>	<ul style="list-style-type: none"> <li>• See <a href="#">hotSwapStatus</a> for more information.</li> </ul>

#### Note

Using NULL for the pointers would make sure that the parameter is not returned.

### 8.494.2 Field Documentation

#### 8.494.2.1 cardStatus\* UIMGetCardStatusResp::pCardStatus

#### 8.494.2.2 hotSwapStatus\* UIMGetCardStatusResp::pHotSwapStatus

## 8.495 UIMGetFileAttributesReq Struct Reference

### Data Fields

- [UIMSessionInformation](#) sessionInfo
- [fileInfo](#) fileIndex
- [ULONG](#) \* pIndicationToken

### 8.495.1 Detailed Description

This structure contains information of the request parameters associated with a Get File Attributes API.

#### Parameters

<a href="#">sessionInfo</a>	<ul style="list-style-type: none"><li>• See <a href="#">UIMSessionInformation</a> for more information.</li></ul>
<a href="#">fileIndex</a>	<ul style="list-style-type: none"><li>• See <a href="#">fileInfo</a> for more information.</li></ul>
<a href="#">pIndication-Token(optional)</a>	<ul style="list-style-type: none"><li>• Response in Indication.</li><li>• When this TLV is present, it indicates that the result must be provided in a subsequent indication.</li></ul>

#### Note

Using NULL for the pointers would make sure that the parameter is not added to the request.

### 8.495.2 Field Documentation

8.495.2.1 [fileInfo](#) UIMGetFileAttributesReq::fileIndex

8.495.2.2 [ULONG](#)\* UIMGetFileAttributesReq::pIndicationToken

8.495.2.3 [UIMSessionInformation](#) UIMGetFileAttributesReq::sessionInfo

## 8.496 UIMGetFileAttributesResp Struct Reference

### Data Fields

- [cardResult](#) \* pCardResult
- [fileAttributes](#) \* pFileAttributes
- [ULONG](#) \* pIndicationToken

### 8.496.1 Detailed Description

This structure contains information of the response parameters associated with a Get File Attributes API.

## Parameters

<i>pCard-Result(optional)</i>	<ul style="list-style-type: none"> <li>• See <a href="#">cardResult</a> for more information.</li> </ul>
<i>pFile-Attributes(optional)</i>	<ul style="list-style-type: none"> <li>• See <a href="#">fileAttributes</a> for more information.</li> </ul>
<i>pIndication-Token(optional)</i>	<ul style="list-style-type: none"> <li>• Response in Indication.</li> <li>• When this TLV is present, it indicates that the result must be provided in a subsequent indication.</li> </ul>

## Note

Using NULL for the pointers would make sure that the parameter is not returned.

## 8.496.2 Field Documentation

8.496.2.1 **cardResult\*** `UIMGetFileAttributesResp::pCardResult`

8.496.2.2 **fileAttributes\*** `UIMGetFileAttributesResp::pFileAttributes`

8.496.2.3 **ULONG\*** `UIMGetFileAttributesResp::pIndicationToken`

## 8.497 UIMPinResp Struct Reference

## Data Fields

- [remainingRetries](#) \* `pRemainingRetries`
- [encryptedPIN1](#) \* `pEncryptedPIN1`
- [ULONG](#) \* `pIndicationToken`

## 8.497.1 Detailed Description

This structure contains information of the response parameters associated with a set of PIN related API's.

## Parameters

<i>pRemaining-Retries(optional)</i>	<ul style="list-style-type: none"> <li>• See <a href="#">remainingRetries</a> for more information.</li> </ul>
<i>pEncryptedPIN1(optional)</i>	<ul style="list-style-type: none"> <li>• See <a href="#">encryptedPIN1</a> for more information.</li> </ul>
<i>pIndication-Token(optional)</i>	<ul style="list-style-type: none"> <li>• Response in Indication.</li> <li>• When this TLV is present, it indicates that the result is provided in a subsequent indication.</li> <li>• 0xFFFFFFFF, if unavailable</li> </ul>



## Note

Using NULL for the pointers would make sure that the parameter is not returned.

## 8.497.2 Field Documentation

8.497.2.1 `encryptedPIN1*` `UIMPinResp::pEncryptedPIN1`

8.497.2.2 `ULONG*` `UIMPinResp::pIndicationToken`

8.497.2.3 `remainingRetries*` `UIMPinResp::pRemainingRetries`

## 8.498 UIMPowerDownReq Struct Reference

### Data Fields

- [BYTE slot](#)

### 8.498.1 Detailed Description

This structure contains information of the request parameters associated with a Power Down.

#### Parameters

<i>slot</i>	<ul style="list-style-type: none"><li>• Indicates the slot to be used.<ul style="list-style-type: none"><li>– 1 - Slot 1</li><li>– 2 - Slot 2</li></ul></li></ul>
-------------	---

## 8.498.2 Field Documentation

8.498.2.1 `BYTE` `UIMPowerDownReq::slot`

## 8.499 UIMPowerUpReq Struct Reference

### Data Fields

- [BYTE slot](#)
- [BYTE \\* pIgnoreHotSwapSwitch](#)

### 8.499.1 Detailed Description

This structure contains information of the request parameters associated with a Power Down.

## Parameters

<i>slot</i>	<ul style="list-style-type: none"> <li>Indicates the slot to be used. <ul style="list-style-type: none"> <li>1 - Slot 1</li> <li>2 - Slot 2</li> </ul> </li> </ul>
<i>plgnoreHot-Swap-Switch(optional)</i>	<ul style="list-style-type: none"> <li>Hot-swap switch status. <ul style="list-style-type: none"> <li>0 - Checks the hot-swap switch status</li> <li>1 - Ignores the hot-swap switch status</li> </ul> </li> </ul>

## 8.499.2 Field Documentation

8.499.2.1 **BYTE\*** `UIMPowerUpReq::plgnoreHotSwapSwitch`8.499.2.2 **BYTE** `UIMPowerUpReq::slot`

## 8.500 UIMReadTransparentReq Struct Reference

## Data Fields

- [UIMSessionInformation sessionInfo](#)
- [fileInfo fileIndex](#)
- [readTransparentInfo readTransparent](#)
- ULONG \*** `pIndicationToken`
- BYTE \*** `pEncryptData`

## 8.500.1 Detailed Description

This structure contains information of the request parameters associated with a Read Transparent API.

## Parameters

<a href="#">sessionInfo</a>	<ul style="list-style-type: none"> <li>See <a href="#">UIMSessionInformation</a> for more information.</li> </ul>
<i>fileIndex</i>	<ul style="list-style-type: none"> <li>See <a href="#">fileInfo</a> for more information.</li> </ul>
<i>readTransparent</i>	<ul style="list-style-type: none"> <li>See <a href="#">readTransparentInfo</a> for more information.</li> </ul>
<i>pIndication-Token(optional)</i>	<ul style="list-style-type: none"> <li>Response in Indication.</li> <li>When this TLV is present, it indicates that the result must be provided in a subsequent indication.</li> </ul>

<i>pEncrypt-Data(optional)</i>	<ul style="list-style-type: none"> <li>• Encrypt Data.</li> <li>• Indicates whether the data read from the card is to be encrypted.</li> </ul>
--------------------------------	--

**Note**

Using NULL for the pointers would make sure that the parameter is not added to the request.

**8.500.2 Field Documentation**

8.500.2.1 **fileInfo** UIMReadTransparentReq::fileIndex

8.500.2.2 **BYTE\*** UIMReadTransparentReq::pEncryptData

8.500.2.3 **ULONG\*** UIMReadTransparentReq::pIndicationToken

8.500.2.4 **readTransparentInfo** UIMReadTransparentReq::readTransparent

8.500.2.5 **UIMSessionInformation** UIMReadTransparentReq::sessionInfo

**8.501 UIMReadTransparentResp Struct Reference****Data Fields**

- [cardResult](#) \* [pCardResult](#)
- [readResult](#) \* [pReadResult](#)
- [ULONG](#) \* [pIndicationToken](#)
- [BYTE](#) \* [pEncryptedData](#)

**8.501.1 Detailed Description**

This structure contains information of the response parameters associated with a Read Transparent API.

**Parameters**

<i>pCardResult</i>	<ul style="list-style-type: none"> <li>• See <a href="#">cardResult</a> for more information.</li> </ul>
<i>pReadResult</i>	<ul style="list-style-type: none"> <li>• See <a href="#">readResult</a> for more information.</li> </ul>
<i>pIndication-Token(optional)</i>	<ul style="list-style-type: none"> <li>• Response in Indication.</li> <li>• When this TLV is present, it indicates that the result must be provided in a subsequent indication.</li> </ul>
<i>pEncrypted-Data(optional)</i>	<ul style="list-style-type: none"> <li>• Encrypted Data.</li> <li>• Indicates whether the data from the card passed in read_result is encrypted.</li> </ul>

## Note

Using NULL for the pointers would make sure that the parameter is not added to the request.

## 8.501.2 Field Documentation

8.501.2.1 **cardResult\*** `UIMReadTransparentResp::pCardResult`

8.501.2.2 **BYTE\*** `UIMReadTransparentResp::pEncryptedData`

8.501.2.3 **ULONG\*** `UIMReadTransparentResp::pIndicationToken`

8.501.2.4 **readResult\*** `UIMReadTransparentResp::pReadResult`

## 8.502 UIMRefreshCompleteReq Struct Reference

### Data Fields

- [UIMSessionInformation sessionInfo](#)
- [BYTE refreshComplete](#)

### 8.502.1 Detailed Description

This structure contains information of the request parameters associated with a SLQSUIMRefreshComplete.

#### Parameters

<a href="#">sessionInfo(-Mandatory)</a>	<ul style="list-style-type: none"> <li>• See <a href="#">UIMSessionInformation</a> for more information.</li> </ul>
<a href="#">refresh-Complete(-Mandatory)</a>	<ul style="list-style-type: none"> <li>• Indicates whether the refresh was successful. Valid values: <ul style="list-style-type: none"> <li>– 0 - Refresh was not completed successfully</li> <li>– 1 - Refresh was completed successfully</li> </ul> </li> </ul>

## 8.502.2 Field Documentation

8.502.2.1 **BYTE** `UIMRefreshCompleteReq::refreshComplete`

8.502.2.2 **UIMSessionInformation** `UIMRefreshCompleteReq::sessionInfo`

## 8.503 UIMRefreshEvent Struct Reference

### Data Fields

- [BYTE stage](#)
- [BYTE mode](#)
- [BYTE sessionType](#)
- [BYTE aidLength](#)
- [BYTE aid](#) [255]
- [WORD numOfFiles](#)

- [fileInfo](#) [arrfileInfo](#) [255]

### 8.503.1 Detailed Description

This structure contains information of parameters associated with the Refresh Event.

#### Parameters

<i>stage</i>	<ul style="list-style-type: none"> <li>• Indicates the stage of the Refresh procedure. <ul style="list-style-type: none"> <li>– 0 - Waiting for OK to refresh</li> <li>– 1 - Refresh started</li> <li>– 2 - Refresh ended successfully</li> <li>– 3 - Refresh failed</li> </ul> </li> </ul>
<i>mode</i>	<ul style="list-style-type: none"> <li>• Indicates the Refresh mode. <ul style="list-style-type: none"> <li>– 0 - Reset</li> <li>– 1 - Init</li> <li>– 2 - Init and FCN</li> <li>– 3 - FCN</li> <li>– 4 - Init and Full FCN</li> <li>– 5 - Application reset</li> <li>– 6 - 3G session reset</li> </ul> </li> </ul>
<i>sessionType</i>	<ul style="list-style-type: none"> <li>• Indicates the session type. <ul style="list-style-type: none"> <li>– 0 - Primary GW provisioning</li> <li>– 1 - Primary 1X provisioning</li> <li>– 2 - Secondary GW provisioning</li> <li>– 3 - Secondary 1X provisioning</li> <li>– 4 - Nonprovisioning on slot 1</li> <li>– 5 - Nonprovisioning on slot 2</li> <li>– 6 - Card on slot 1</li> <li>– 7 - Card on slot 2</li> <li>– 8 - Logical channel on slot 1</li> <li>– 9 - Logical channel on slot 2</li> </ul> </li> </ul>

<i>aidLength</i>	<ul style="list-style-type: none"> <li>Number of sets of the following elements <ul style="list-style-type: none"> <li>Application Identifier</li> </ul> </li> </ul>
<i>aid</i>	<ul style="list-style-type: none"> <li>Application identifier value or channel ID. This value is required for non-provisioning and for logical channel session types. It is ignored in all other cases</li> </ul>
<i>numFiles</i>	<ul style="list-style-type: none"> <li>Number of sets of the following elements: <ul style="list-style-type: none"> <li>file_id</li> <li>path_len</li> <li>path</li> </ul> </li> </ul>
<i>arrfileInfo</i>	<ul style="list-style-type: none"> <li>Array of file Information struct</li> </ul>

## 8.503.2 Field Documentation

8.503.2.1 **BYTE** UIMRefreshEvent::aid[255]

8.503.2.2 **BYTE** UIMRefreshEvent::aidLength

8.503.2.3 **fileInfo** UIMRefreshEvent::arrfileInfo[255]

8.503.2.4 **BYTE** UIMRefreshEvent::mode

8.503.2.5 **WORD** UIMRefreshEvent::numOfFiles

8.503.2.6 **BYTE** UIMRefreshEvent::sessionType

8.503.2.7 **BYTE** UIMRefreshEvent::stage

## 8.504 UIMRefreshGetLastEventReq Struct Reference

### Data Fields

- [UIMSessionInformation sessionInfo](#)

### 8.504.1 Detailed Description

This structure contains information of the request parameters associated with a SLQSUIMRefreshGetLastEvent.

#### Parameters

<i><a href="#">sessionInfo</a>(-Mandatory)</i>	<ul style="list-style-type: none"> <li>See <a href="#">UIMSessionInformation</a> for more information.</li> </ul>
--	---

## 8.504.2 Field Documentation

### 8.504.2.1 UIMSessionInformation UIMRefreshGetLastEventReq::sessionInfo

## 8.505 UIMRefreshGetLastEventResp Struct Reference

### Data Fields

- [UIMRefreshEvent](#) \* [pRefreshEvent](#)

### 8.505.1 Detailed Description

This structure contains information of the response parameters associated with a SLQSUIMRefreshGetLastEvent.

#### Parameters

<i>refreshEvent(- Optional)</i>	<ul style="list-style-type: none"><li>• See <a href="#">UIMRefreshEvent</a> for more information.</li></ul>
-------------------------------------	---

## 8.505.2 Field Documentation

### 8.505.2.1 UIMRefreshEvent\* UIMRefreshGetLastEventResp::pRefreshEvent

## 8.506 UIMRefreshOKReq Struct Reference

### Data Fields

- [UIMSessionInformation](#) [sessionInfo](#)
- [BYTE](#) [OKtoRefresh](#)

### 8.506.1 Detailed Description

This structure contains Parameters of the Session Information

#### Parameters

<i><a href="#">sessionInfo</a></i>	<ul style="list-style-type: none"><li>• Session Information</li><li>• See <a href="#">UIMSessionInformation</a> for more information</li></ul>
<i><a href="#">OKtoRefresh</a></i>	<ul style="list-style-type: none"><li>• Indicates whether a refresh is OK. Valid values:<ul style="list-style-type: none"><li>– 0 - Not OK to refresh</li><li>– 1 - OK to refresh</li></ul></li></ul>

## 8.506.2 Field Documentation

### 8.506.2.1 BYTE UIMRefreshOKReq::OKtoRefresh

8.506.2.2 UIMSessionInformation UIMRefreshOKReq::sessionInfo

## 8.507 UIMRefreshRegisterReq Struct Reference

### Data Fields

- [UIMSessionInformation sessionInfo](#)
- [registerRefresh regRefresh](#)

### 8.507.1 Detailed Description

This structure contains information of the request parameters associated with a Refresh Register.

#### Parameters

<a href="#">sessionInfo</a>	<ul style="list-style-type: none"><li>• Session Information params</li><li>• See <a href="#">UIMSessionInformation</a> for more information</li></ul>
<a href="#">regRefresh</a>	<ul style="list-style-type: none"><li>• Register Refresh parameters</li><li>• See <a href="#">registerRefresh</a> for more information</li></ul>

### 8.507.2 Field Documentation

8.507.2.1 registerRefresh UIMRefreshRegisterReq::regRefresh

8.507.2.2 UIMSessionInformation UIMRefreshRegisterReq::sessionInfo

## 8.508 UIMSessionInformation Struct Reference

### Data Fields

- [BYTE sessionType](#)
- [BYTE aidLength](#)
- [BYTE aid](#) [255]

### 8.508.1 Detailed Description

This structure contains the Session Information.



## Parameters

<i>sessionType</i>	<ul style="list-style-type: none"> <li>Indicates the session type. <ul style="list-style-type: none"> <li>0 - Primary GW provisioning</li> <li>1 - Primary 1X provisioning</li> <li>2 - Secondary GW provisioning</li> <li>3 - Secondary 1X provisioning</li> <li>4 - Non-provisioning on slot 1</li> <li>5 - Non-provisioning on slot 2</li> <li>6 - Card on slot 1</li> <li>7 - Card on slot 2</li> <li>8 - Logical channel on slot 1</li> <li>9 - Logical channel on slot 2</li> </ul> </li> </ul>
<i>aidLength</i>	<ul style="list-style-type: none"> <li>Length of the following elements i.e. Application Identifier.</li> </ul>
<i>aid</i>	<ul style="list-style-type: none"> <li>Application identifier value or channel ID.</li> <li>This value is required for non-provisioning and for logical channel session types. It is ignored in all other cases.</li> </ul>

## 8.508.2 Field Documentation

8.508.2.1 BYTE UIMSessionInformation::aid[255]

8.508.2.2 BYTE UIMSessionInformation::aidLength

8.508.2.3 BYTE UIMSessionInformation::sessionType

## 8.509 UIMSetPinProtectionReq Struct Reference

## Data Fields

- [UIMSessionInformation sessionInfo](#)
- [setPINProtection pinProtection](#)
- [BYTE \\* pKeyReferenceID](#)
- [ULONG \\* pIndicationToken](#)

## 8.509.1 Detailed Description

This structure contains information of the request parameters associated with a set pin protection API.

## Parameters

<i><a href="#">sessionInfo</a></i>	<ul style="list-style-type: none"> <li>• See <a href="#">UIMSessionInformation</a> for more information.</li> </ul>
<i><a href="#">pinProtection</a></i>	<ul style="list-style-type: none"> <li>• See <a href="#">setPINProtection</a> for more information.</li> </ul>
<i><a href="#">pKeyReferenceID(optional)</a></i>	<ul style="list-style-type: none"> <li>• Indicates the PIN key reference ID.</li> <li>• Indicates the PIN key reference ID. Valid values are from 1 to 8, respectively, for application 1 to application 8.</li> <li>• This TLV is used only for PIN1 and PIN2 and is ignored in all other cases.</li> </ul>
<i><a href="#">pIndication-Token(optional)</a></i>	<ul style="list-style-type: none"> <li>• Response in Indication.</li> <li>• When this TLV is present, it indicates that the result must be provided in a subsequent indication.</li> </ul>

## Note

Using NULL for the pointers would make sure that the parameter is not added to the request.

## 8.509.2 Field Documentation

8.509.2.1 **ULONG\*** [UIMSetPinProtectionReq::pIndicationToken](#)

8.509.2.2 **setPINProtection** [UIMSetPinProtectionReq::pinProtection](#)

8.509.2.3 **BYTE\*** [UIMSetPinProtectionReq::pKeyReferenceID](#)

8.509.2.4 **UIMSessionInformation** [UIMSetPinProtectionReq::sessionInfo](#)

## 8.510 UIMStatusChangeInfo Struct Reference

## Data Fields

- [cardStatus](#) [statusChange](#)

## 8.510.1 Detailed Description

Structure consist of cardstatus params

## Parameters

<i><a href="#">statusChange</a></i>	<ul style="list-style-type: none"> <li>• See <a href="#">cardStatus</a> for more information</li> </ul>
-------------------------------------	---

## 8.510.2 Field Documentation

8.510.2.1 cardStatus UIMStatusChangeInfo::statusChange

## 8.511 UIMUnlockPinReq Struct Reference

### Data Fields

- [UIMSessionInformation sessionInfo](#)
- [unlockUIMPIN unlockPIN](#)
- [BYTE \\* pKeyReferenceID](#)
- [ULONG \\* pIndicationToken](#)

### 8.511.1 Detailed Description

This structure contains information of the request parameters associated with a Unblock PIN API.

#### Parameters

<a href="#">sessionInfo</a>	<ul style="list-style-type: none"> <li>• See <a href="#">UIMSessionInformation</a> for more information.</li> </ul>
<a href="#">unlockPIN</a>	<ul style="list-style-type: none"> <li>• See <a href="#">unlockUIMPIN</a> for more information.</li> </ul>
<a href="#">pKeyReferenceID(optional)</a>	<ul style="list-style-type: none"> <li>• Indicates the PIN key reference ID.</li> <li>• Indicates the PIN key reference ID. Valid values are from 1 to 8, respectively, for application 1 to application 8.</li> <li>• This TLV is used only for PIN1 and PIN2 and is ignored in all other cases.</li> </ul>
<a href="#">pIndicationToken(optional)</a>	<ul style="list-style-type: none"> <li>• Response in Indication.</li> <li>• When this TLV is present, it indicates that the result must be provided in a subsequent indication.</li> </ul>

#### Note

Using NULL for the pointers would make sure that the parameter is not added to the request.

### 8.511.2 Field Documentation

8.511.2.1 **ULONG\*** UIMUnlockPinReq::pIndicationToken

8.511.2.2 **BYTE\*** UIMUnlockPinReq::pKeyReferenceID

8.511.2.3 **UIMSessionInformation** UIMUnlockPinReq::sessionInfo

8.511.2.4 **unlockUIMPIN** UIMUnlockPinReq::unlockPIN

## 8.512 UIMVerifyPinReq Struct Reference

## Data Fields

- [UIMSessionInformation sessionInfo](#)
- [verifyUIMPIN verifyPIN](#)
- [encryptedPIN1 \\* pEncryptedPIN1](#)
- [BYTE \\* pKeyReferenceID](#)
- [ULONG \\* pIndicationToken](#)

### 8.512.1 Detailed Description

This structure contains information of the request parameters associated with a verify PIN API.

#### Parameters

<a href="#">sessionInfo</a>	<ul style="list-style-type: none"> <li>• See <a href="#">UIMSessionInformation</a> for more information.</li> </ul>
<a href="#">verifyPIN</a>	<ul style="list-style-type: none"> <li>• See <a href="#">verifyUIMPIN</a> for more information.</li> </ul>
<a href="#">pEncryptedPIN1(optional)</a>	<ul style="list-style-type: none"> <li>• See <a href="#">encryptedPIN1</a> for more information.</li> </ul>
<a href="#">pKeyReferenceID(optional)</a>	<ul style="list-style-type: none"> <li>• Indicates the PIN key reference ID.</li> <li>• Indicates the PIN key reference ID. Valid values are from 1 to 8, respectively, for application 1 to application 8.</li> <li>• This TLV is used only for PIN1 and PIN2 and is ignored in all other cases.</li> </ul>
<a href="#">pIndicationToken(optional)</a>	<ul style="list-style-type: none"> <li>• Response in Indication.</li> <li>• When this TLV is present, it indicates that the result must be provided in a subsequent indication.</li> </ul>

#### Note

Using NULL for the pointers would make sure that the parameter is not added to the request.

### 8.512.2 Field Documentation

8.512.2.1 [encryptedPIN1 \\* UIMVerifyPinReq::pEncryptedPIN1](#)

8.512.2.2 [ULONG \\* UIMVerifyPinReq::pIndicationToken](#)

8.512.2.3 [BYTE \\* UIMVerifyPinReq::pKeyReferenceID](#)

8.512.2.4 [UIMSessionInformation UIMVerifyPinReq::sessionInfo](#)

8.512.2.5 [verifyUIMPIN UIMVerifyPinReq::verifyPIN](#)

## 8.513 UMTSInfo Struct Reference

## Data Fields

- [WORD cellID](#)
- [BYTE plmn](#) [3]
- [WORD lac](#)
- [WORD uarfcn](#)
- [WORD psc](#)
- [SHORT rscp](#)
- [SHORT ecio](#)
- [BYTE umtsInst](#)
- [UMTSInstInfo UMTSInstInfo](#) [255]
- [BYTE geranInst](#)
- [geranInstInfo GeranInstInfo](#) [255]

## 8.513.1 Detailed Description

This structure contains information about the UMTS Network.

## Parameters

<i>cellID</i>	<ul style="list-style-type: none"> <li>• Cell ID.</li> <li>• 0xFFFFFFFF indicates cell ID information is not present.</li> </ul>
<i>plmn</i> [ <i>PLMN_LENGTH</i> ]	<ul style="list-style-type: none"> <li>• MCC/MNC information coded as octet 3, 4, and 5.</li> <li>• This field is ignored when nmrCellID is not present.</li> </ul>
<i>lac</i>	<ul style="list-style-type: none"> <li>• Location area code.</li> <li>• This field is ignored when nmrCellID is not present. <ul style="list-style-type: none"> <li>– 0xFFFF - Not Available</li> </ul> </li> </ul>
<i>uarfcn</i>	<ul style="list-style-type: none"> <li>• UTRA absolute RF channel number. <ul style="list-style-type: none"> <li>– 0xFFFF - Not Available</li> </ul> </li> </ul>
<i>psc</i>	<ul style="list-style-type: none"> <li>• Primary scrambling code. <ul style="list-style-type: none"> <li>– 0xFFFF - Not Available</li> </ul> </li> </ul>
<i>rscp</i>	<ul style="list-style-type: none"> <li>• Received signal code power. <ul style="list-style-type: none"> <li>– 0xFFFF - Not Available</li> </ul> </li> </ul>

<i>ecio</i>	<ul style="list-style-type: none"> <li>• ECIO(Signal-to-Interference-ratio). <ul style="list-style-type: none"> <li>– 0xFFFF - Not Available</li> </ul> </li> </ul>
<i>umtsInst</i>	<ul style="list-style-type: none"> <li>• Provides the number of set of UMTS info instances.</li> <li>• If 0(zero), then no information follows it.</li> </ul>
<i>UMTSInstInfo[M-AX_DESCRIPTOR_LENGTH]</i>	<ul style="list-style-type: none"> <li>• See <a href="#">UMTSInstInfo</a> for more information.</li> </ul>
<i>geranInst</i>	<ul style="list-style-type: none"> <li>• Provides the number of set of GERAN info instances.</li> <li>• If 0(zero), then no information follows it.</li> </ul>
<i>GeranInstInfo[M-AX_DESCRIPTOR_LENGTH]</i>	<ul style="list-style-type: none"> <li>• See <a href="#">geranInstInfo</a> for more information.</li> </ul>

## 8.513.2 Field Documentation

8.513.2.1 WORD UMTSInfo::cellID

8.513.2.2 SHORT UMTSInfo::ecio

8.513.2.3 BYTE UMTSInfo::geranInst

8.513.2.4 [geranInstInfo](#) UMTSInfo::GeranInstInfo[255]

8.513.2.5 WORD UMTSInfo::lac

8.513.2.6 BYTE UMTSInfo::plmn[3]

8.513.2.7 WORD UMTSInfo::psc

8.513.2.8 SHORT UMTSInfo::rscp

8.513.2.9 WORD UMTSInfo::uarfcn

8.513.2.10 BYTE UMTSInfo::umtsInst

8.513.2.11 [UMTSInstInfo](#) UMTSInfo::UMTSInstInfo[255]

## 8.514 UMTSInstInfo Struct Reference

### Data Fields

- [WORD](#) [umtsUarfcn](#)
- [WORD](#) [umtsPsc](#)
- [SHORT](#) [umtsRscp](#)
- [SHORT](#) [umtsEcio](#)

### 8.514.1 Detailed Description

This structure contains information about the UMTS Instances in UMTS Network.

#### Parameters

<i>umtsUarfcn</i>	<ul style="list-style-type: none"> <li>• UTRA absolute RF channel number.</li> </ul>
<i>umtsPsc</i>	<ul style="list-style-type: none"> <li>• Primary scrambling code.</li> </ul>
<i>umtsRscp</i>	<ul style="list-style-type: none"> <li>• Received signal code power.</li> </ul>
<i>umtsEcio</i>	<ul style="list-style-type: none"> <li>• ECIO(Signal-to-Interference-ratio).</li> </ul>

### 8.514.2 Field Documentation

8.514.2.1 **SHORT** UMTSinstInfo::umtsEcio

8.514.2.2 **WORD** UMTSinstInfo::umtsPsc

8.514.2.3 **SHORT** UMTSinstInfo::umtsRscp

8.514.2.4 **WORD** UMTSinstInfo::umtsUarfcn

## 8.515 umtsLTENbrCell Struct Reference

### Data Fields

- [WORD](#) earfcn
- [WORD](#) pci
- [ULONG](#) rsrp
- [ULONG](#) rsrq
- [SHORT](#) srxlev
- [BYTE](#) cellsTDD

### 8.515.1 Detailed Description

This structure contains information about the UMTS LTE neighbour Cell.

#### Parameters

<i>earfcn</i>	<ul style="list-style-type: none"> <li>• E-UTRA absolute RF channel number of the detected cell.</li> </ul>
---------------	---

<i>pci</i>	<ul style="list-style-type: none"> <li>Physical cell ID of the detected cell.</li> <li>Range is defined in 3GPP TS 36.211</li> </ul>
<i>rsrp</i>	<ul style="list-style-type: none"> <li>Current received signal strength indication (in dBm) of the detected cell.</li> </ul>
<i>rsrq</i>	<ul style="list-style-type: none"> <li>Current reference signal received quality (in dB) of the detected cell.</li> </ul>
<i>srxlev</i>	<ul style="list-style-type: none"> <li>Cell selection Rx level (Srxlev) value of the detected cell in linear scale.</li> <li>This field is only valid when wcdma_rrc_state is not NAS_WCDMA_RRC_STATE_CELL_FACH or NAS_WCDMA_RRC_STATE_CELL_DCH.</li> </ul>
<i>cellsTDD</i>	<ul style="list-style-type: none"> <li>TRUE if the cell is TDD; FALSE if the cell is FDD.</li> </ul>

## 8.515.2 Field Documentation

8.515.2.1 **BYTE** umtsLTENbrCell::cellsTDD

8.515.2.2 **WORD** umtsLTENbrCell::earfcn

8.515.2.3 **WORD** umtsLTENbrCell::pci

8.515.2.4 **ULONG** umtsLTENbrCell::rsrp

8.515.2.5 **ULONG** umtsLTENbrCell::rsrq

8.515.2.6 **SHORT** umtsLTENbrCell::srxlev

## 8.516 UMTSMinQoS Struct Reference

### Data Fields

- [BYTE](#) trafficClass
- [ULONG](#) maxUplinkBitrate
- [ULONG](#) maxDownlinkBitrate
- [ULONG](#) grntUplinkBitrate
- [ULONG](#) grntDownlinkBitrate
- [BYTE](#) qosDeliveryOrder
- [ULONG](#) maxSDUSize
- [BYTE](#) sduErrorRatio
- [BYTE](#) resBerRatio
- [BYTE](#) deliveryErrSDU
- [ULONG](#) transferDelay
- [ULONG](#) trafficPriority



### 8.516.1 Detailed Description

This structure contains the UMTS Quality Of Service Information

## Parameters

<i>trafficClass</i>	<ul style="list-style-type: none"> <li>• 0x00 - Subscribed</li> <li>• 0x01 - Conversational</li> <li>• 0x02 - Streaming</li> <li>• 0x03 - Interactive</li> <li>• 0x04 - Background</li> </ul>
<i>maxUplinkBitrate</i>	<ul style="list-style-type: none"> <li>• Maximum uplink bit rate in bits/sec</li> </ul>
<i>maxDownlink-Bitrate</i>	<ul style="list-style-type: none"> <li>• Maximum downlink bit rate in bits/sec</li> </ul>
<i>grntUplinkBitrate</i>	<ul style="list-style-type: none"> <li>• Guaranteed uplink bit rate in bits/sec</li> </ul>
<i>grntDownlink-Bitrate</i>	<ul style="list-style-type: none"> <li>• Guaranteed downlink bit rate in bits/sec</li> </ul>
<i>qosDelivery-Order</i>	<ul style="list-style-type: none"> <li>- Qos delivery order</li> <li>• 0x00 - Subscribe</li> <li>• 0x01 - Delivery order on</li> <li>• 0x02 - Delivery order off</li> </ul>
<i>maxSDUSize</i>	<ul style="list-style-type: none"> <li>• Maximum SDU size</li> </ul>
<i>sduErrorRatio</i>	<ul style="list-style-type: none"> <li>- SDU error ratio</li> <li>• Target value for fraction of SDUs lost or detected as erroneous.</li> <li>• 0x00 - Subscribe</li> <li>• 0x01 - <math>1 \times 10^{(-2)}</math></li> <li>• 0x02 - <math>7 \times 10^{(-3)}</math></li> <li>• 0x03 - <math>1 \times 10^{(-3)}</math></li> <li>• 0x04 - <math>1 \times 10^{(-4)}</math></li> <li>• 0x05 - <math>1 \times 10^{(-5)}</math></li> <li>• 0x06 - <math>1 \times 10^{(-6)}</math></li> <li>• 0x07 - <math>1 \times 10^{(-1)}</math></li> </ul>

<i>resBerRatio</i>	<p>- Residual bit error ratio</p> <ul style="list-style-type: none"> <li>• Target value for undetected bit error ratio in the delivered SDUs.</li> <li>• 0x00 - Subscribe</li> <li>• 0x01 - <math>5 \cdot 10^{-2}</math></li> <li>• 0x02 - <math>1 \cdot 10^{-2}</math></li> <li>• 0x03 - <math>5 \cdot 10^{-3}</math></li> <li>• 0x04 - <math>4 \cdot 10^{-3}</math></li> <li>• 0x05 - <math>1 \cdot 10^{-3}</math></li> <li>• 0x06 - <math>1 \cdot 10^{-4}</math></li> <li>• 0x07 - <math>1 \cdot 10^{-5}</math></li> <li>• 0x08 - <math>1 \cdot 10^{-6}</math></li> <li>• 0x09 - <math>1 \cdot 10^{-8}</math></li> </ul>
<i>deliveryErrSDU</i>	<p>- delivery of erroneous SDUs</p> <ul style="list-style-type: none"> <li>• Indicates whether SDUs detected as erroneous shall be delivered or not.</li> <li>• 0x00 - Subscribe</li> <li>• 0x01 - <math>5 \cdot 10^{-2}</math></li> <li>• 0x02 - <math>1 \cdot 10^{-2}</math></li> <li>• 0x03 - <math>5 \cdot 10^{-3}</math></li> <li>• 0x04 - <math>4 \cdot 10^{-3}</math></li> <li>• 0x05 - <math>1 \cdot 10^{-3}</math></li> <li>• 0x06 - <math>1 \cdot 10^{-4}</math></li> <li>• 0x07 - <math>1 \cdot 10^{-5}</math></li> <li>• 0x08 - <math>1 \cdot 10^{-6}</math></li> <li>• 0x09 - <math>1 \cdot 10^{-8}</math></li> </ul>

<i>transferDelay</i>	- Transfer delay (ms) <ul style="list-style-type: none"> <li>Indicates the targeted time between a request to transfer an SDU at one SAP to its delivery at the other SAP in milliseconds.</li> </ul>
<i>trafficPriority</i>	- Transfer handling priority <ul style="list-style-type: none"> <li>Specifies the relative importance for handling of SDUs that belong to the UMTS bearer, compared to the SDUs of other bearers.</li> </ul>

## 8.516.2 Field Documentation

8.516.2.1 **BYTE** UMTSMinQoS::deliveryErrSDU

8.516.2.2 **ULONG** UMTSMinQoS::grntDownlinkBitrate

8.516.2.3 **ULONG** UMTSMinQoS::grntUplinkBitrate

8.516.2.4 **ULONG** UMTSMinQoS::maxDownlinkBitrate

8.516.2.5 **ULONG** UMTSMinQoS::maxSDUSize

8.516.2.6 **ULONG** UMTSMinQoS::maxUplinkBitrate

8.516.2.7 **BYTE** UMTSMinQoS::qosDeliveryOrder

8.516.2.8 **BYTE** UMTSMinQoS::resBerRatio

8.516.2.9 **BYTE** UMTSMinQoS::sduErrorRatio

8.516.2.10 **BYTE** UMTSMinQoS::trafficClass

8.516.2.11 **ULONG** UMTSMinQoS::trafficPriority

8.516.2.12 **ULONG** UMTSMinQoS::transferDelay

## 8.517 UMTSQoS Struct Reference

### Data Fields

- [BYTE trafficClass](#)
- [ULONG maxUplinkBitrate](#)
- [ULONG maxDownlinkBitrate](#)
- [ULONG grntUplinkBitrate](#)
- [ULONG grntDownlinkBitrate](#)
- [BYTE qosDeliveryOrder](#)
- [ULONG maxSDUSize](#)
- [BYTE sduErrorRatio](#)
- [BYTE resBerRatio](#)
- [BYTE deliveryErrSDU](#)
- [ULONG transferDelay](#)
- [ULONG trafficPriority](#)

### 8.517.1 Detailed Description

This structure contains the UMTS Quality Of Service Information

- Parameter values default to their data type's maximum unsigned value unless explicitly stated otherwise.

## Parameters

<i>trafficClass</i>	<ul style="list-style-type: none"> <li>• 0x00 - Subscribed</li> <li>• 0x01 - Conversational</li> <li>• 0x02 - Streaming</li> <li>• 0x03 - Interactive</li> <li>• 0x04 - Background</li> </ul>
<i>maxUplinkBitrate</i>	<ul style="list-style-type: none"> <li>• Maximum uplink bit rate in bits/sec</li> </ul>
<i>maxDownlink-Bitrate</i>	<ul style="list-style-type: none"> <li>• Maximum downlink bit rate in bits/sec</li> </ul>
<i>grntUplinkBitrate</i>	<ul style="list-style-type: none"> <li>• Guaranteed uplink bit rate in bits/sec</li> </ul>
<i>grntDownlink-Bitrate</i>	<ul style="list-style-type: none"> <li>• Guranteed downlink bit rate in bits/sec</li> </ul>
<i>qosDelivery-Order</i>	<ul style="list-style-type: none"> <li>- Qos delivery order</li> <li>• 0x00 - Subscribe</li> <li>• 0x01 - delivery order on</li> <li>• 0x02 - delivery order off</li> </ul>
<i>maxSDUSize</i>	<ul style="list-style-type: none"> <li>• Maximum SDU size</li> </ul>
<i>sduErrorRatio</i>	<ul style="list-style-type: none"> <li>- SDU error ratio</li> <li>• Target value for fraction of SDUs lost or detected as erroneous.</li> <li>• 0x00 - Subscribe</li> <li>• 0x01 - <math>1 \cdot 10^{(-2)}</math></li> <li>• 0x02 - <math>7 \cdot 10^{(-3)}</math></li> <li>• 0x03 - <math>1 \cdot 10^{(-3)}</math></li> <li>• 0x04 - <math>1 \cdot 10^{(-4)}</math></li> <li>• 0x05 - <math>1 \cdot 10^{(-5)}</math></li> <li>• 0x06 - <math>1 \cdot 10^{(-6)}</math></li> <li>• 0x07 - <math>1 \cdot 10^{(-1)}</math></li> </ul>
<i>resBerRatio</i>	<ul style="list-style-type: none"> <li>- Residual bit error ratio</li> <li>• Target value for undetected bit error ratio in in the delivered SDUs.</li> <li>• 0x00 - Subscribe</li> <li>• 0x01 - <math>5 \cdot 10^{(-2)}</math></li> <li>• 0x02 - <math>1 \cdot 10^{(-2)}</math></li> <li>• 0x03 - <math>5 \cdot 10^{(-3)}</math></li> <li>• 0x04 - <math>4 \cdot 10^{(-3)}</math></li> </ul>

<i>deliveryErrSDU</i>	- Delivery of erroneous SDUs <ul style="list-style-type: none"> <li>Indicates whether SDUs detected as erroneous shall be delivered or not.</li> <li>0x00 - Subscribe</li> <li>0x01 - <math>5 \cdot 10^{-2}</math></li> <li>0x02 - <math>1 \cdot 10^{-2}</math></li> <li>0x03 - <math>5 \cdot 10^{-3}</math></li> <li>0x04 - <math>4 \cdot 10^{-3}</math></li> <li>0x05 - <math>1 \cdot 10^{-3}</math></li> <li>0x06 - <math>1 \cdot 10^{-4}</math></li> <li>0x07 - <math>1 \cdot 10^{-5}</math></li> <li>0x08 - <math>1 \cdot 10^{-6}</math></li> <li>0x09 - <math>1 \cdot 10^{-8}</math></li> </ul>
<i>transferDelay</i>	- Transfer delay (ms) <ul style="list-style-type: none"> <li>Indicates the targeted time between a request to transfer an SDU at one SAP to its delivery at the other SAP in milliseconds.</li> </ul>
<i>trafficPriority</i>	- Transfer handling priority <ul style="list-style-type: none"> <li>Specifies the relative importance for handling of SDUs that belong to the UMTS bearer, compared to the SDUs of other bearers.</li> </ul>

## 8.517.2 Field Documentation

8.517.2.1 **BYTE** UMTSQoS::deliveryErrSDU

8.517.2.2 **ULONG** UMTSQoS::grntDownlinkBitrate

8.517.2.3 **ULONG** UMTSQoS::grntUplinkBitrate

8.517.2.4 **ULONG** UMTSQoS::maxDownlinkBitrate

8.517.2.5 **ULONG** UMTSQoS::maxSDUSize

8.517.2.6 **ULONG** UMTSQoS::maxUplinkBitrate

8.517.2.7 **BYTE** UMTSQoS::qosDeliveryOrder

8.517.2.8 **BYTE** UMTSQoS::resBerRatio

8.517.2.9 **BYTE** UMTSQoS::sduErrorRatio

8.517.2.10 **BYTE** UMTSQoS::trafficClass

8.517.2.11 **ULONG** UMTSQoS::trafficPriority

8.517.2.12 **ULONG** UMTSQoS::transferDelay

## 8.518 UMTSReqQoSSigInd Struct Reference

### Data Fields

- struct [UMTSQoS](#) [UMTSReqQoS](#)
- [BYTE](#) [SigInd](#)

### 8.518.1 Detailed Description

structure contains UMTS requested QoS with Signaling Indication flag

- Parameter values default to their data type's maximum unsigned value unless explicitly stated otherwise.

#### Parameters

<i>UMTSReqQoS</i>	<ul style="list-style-type: none"><li>• Contains the UMTS Quality Of Service Information</li></ul>
<i>SigInd</i>	<ul style="list-style-type: none"><li>- Signaling Indication flag</li><li>• TRUE - Signaling indication ON</li><li>• FALSE - Signaling indication OFF</li></ul>

### 8.518.2 Field Documentation

8.518.2.1 [BYTE](#) [UMTSReqQoSSigInd::SigInd](#)

8.518.2.2 struct [UMTSQoS](#) [UMTSReqQoSSigInd::UMTSReqQoS](#)

## 8.519 unblockUIMPIN Struct Reference

### Data Fields

- [BYTE](#) [pinID](#)
- [BYTE](#) [pukLen](#)
- [BYTE](#) [pukVal](#) [255]
- [BYTE](#) [newPINLen](#)
- [BYTE](#) [newPINVal](#) [255]

### 8.519.1 Detailed Description

This structure contains the information about the unblock pin parameters.



## Parameters

<i>pinID</i>	<ul style="list-style-type: none"> <li>Indicates the PIN ID to be changed. <ul style="list-style-type: none"> <li>1 - PIN1 (also called PIN)</li> <li>2 - PIN2</li> <li>3 - Universal PIN</li> </ul> </li> </ul>
<i>pukLen</i>	<ul style="list-style-type: none"> <li>Length of the following elements i.e. puk value.</li> </ul>
<i>pukVal</i> [MAX_P-UK_LENGTH]	<ul style="list-style-type: none"> <li>PIN Unlock Key value.</li> <li>This value is a sequence of ASCII characters.</li> </ul>
<i>pinLen</i>	<ul style="list-style-type: none"> <li>Length of the following elements i.e. new pin value.</li> </ul>
<i>pinVal</i> [MAX_DESCRIPTION_LENGTH]	<ul style="list-style-type: none"> <li>New PIN value.</li> <li>This value is a sequence of ASCII characters.</li> </ul>

## 8.519.2 Field Documentation

8.519.2.1 BYTE unblockUIMPIN::newPINLen

8.519.2.2 BYTE unblockUIMPIN::newPINVal[255]

8.519.2.3 BYTE unblockUIMPIN::pinID

8.519.2.4 BYTE unblockUIMPIN::pukLen

8.519.2.5 BYTE unblockUIMPIN::pukVal[255]

## 8.520 UniversalTime Struct Reference

## Data Fields

- WORD year
- BYTE month
- BYTE day
- BYTE hour
- BYTE minute
- BYTE second
- BYTE dayOfWeek

## 8.520.1 Detailed Description

This structure contains the parameters for Universal Time Information.

## Parameters

<i>year</i>	<ul style="list-style-type: none"><li>• Year.</li></ul>
<i>month</i>	<ul style="list-style-type: none"><li>• Month.<ul style="list-style-type: none"><li>– 1 is January and 12 is December.</li></ul></li></ul>
<i>day</i>	<ul style="list-style-type: none"><li>• Day.<ul style="list-style-type: none"><li>– Range 1 to 31.</li></ul></li></ul>
<i>hour</i>	<ul style="list-style-type: none"><li>• Hour.<ul style="list-style-type: none"><li>– Range 0 to 59.</li></ul></li></ul>
<i>minute</i>	<ul style="list-style-type: none"><li>• Minute.<ul style="list-style-type: none"><li>– Range 0 to 59.</li></ul></li></ul>
<i>second</i>	<ul style="list-style-type: none"><li>• Second.<ul style="list-style-type: none"><li>– Range 0 to 59.</li></ul></li></ul>
<i>dayOfWeek</i>	<ul style="list-style-type: none"><li>• Day of the Week.<ul style="list-style-type: none"><li>– 0 is Monday and 6 is Sunday.</li></ul></li></ul>

## 8.520.2 Field Documentation

### 8.520.2.1 BYTE UniversalTime::day

### 8.520.2.2 BYTE UniversalTime::dayOfWeek

### 8.520.2.3 BYTE UniversalTime::hour

### 8.520.2.4 BYTE UniversalTime::minute

### 8.520.2.5 BYTE UniversalTime::month

### 8.520.2.6 BYTE UniversalTime::second

### 8.520.2.7 WORD UniversalTime::year

## 8.521 USBCompConfig Struct Reference

### Data Fields

- [BYTE](#) \* [pUSBComp](#)

### 8.521.1 Detailed Description

This structure is used to store USB composition information

#### Parameters

<i>pUSBComp[IN]</i>	<ul style="list-style-type: none"> <li>• Current USB Composition</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0..5 - Reserved (non-QMI)</li> <li>– 6 - DM NMEA AT QMI</li> <li>– 7 - DM NMEA AT QMI1 QMI2 QMI3</li> <li>– 8 - DM NMEA AT MBIM</li> <li>– 9 - MBIM</li> <li>– 10 - NMEA MBIM</li> <li>– 11 - DM MBIM</li> <li>– 12 - DM NMEA MBIM</li> <li>– 13-22 are combined compositions. One is for Win8 MBIM interfaces, another is for legacy QMI interfaces</li> <li>– 13 - 6 for QMI, 8 for MBIM</li> <li>– 14 - 6 for QMI, 9 for MBIM</li> <li>– 15 - 6 for QMI, 10 for MBIM</li> <li>– 16 - 6 for QMI, 11 for MBIM</li> <li>– 17 - 6 for QMI, 12 for MBIM</li> <li>– 18 - 7 for QMI, 8 for MBIM</li> <li>– 19 - 7 for QMI, 9 for MBIM</li> <li>– 20 - 7 for QMI, 10 for MBIM</li> <li>– 21 - 7 for QMI, 11 for MBIM</li> <li>– 22 - 7 for QMI, 12 for MBIM</li> </ul> </li> </ul>
---------------------	--

### 8.521.2 Field Documentation

#### 8.521.2.1 [BYTE](#)\* [USBCompConfig::pUSBComp](#)

## 8.522 USBCompParams Struct Reference

### Data Fields

- [BYTE](#) \* [pUSBComp](#)
- [BYTE](#) \* [pNumSupUSBComps](#)

- [BYTE](#) \* [pSupUSBComps](#)

### 8.522.1 Detailed Description

This structure is used to store retrieved USB Composition

#### Parameters

<i>pUSBComp[OUT]</i>	<ul style="list-style-type: none"> <li>• Current USB Composition</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0..5 - Reserved (non-QMI)</li> <li>– 6 - DM NMEA AT QMI</li> <li>– 7 - DM NMEA AT QMI1 QMI2 QMI3</li> <li>– 8 - DM NMEA AT MBIM</li> <li>– 9 - MBIM</li> <li>– 10 - NMEA MBIM</li> <li>– 11 - DM MBIM</li> <li>– 12 - DM NMEA MBIM</li> <li>13-22 are combined compositions. One is for Win8 MBIM interfaces, another is for legacy QMI interfaces</li> <li>– 13 - 6 for QMI, 8 for MBIM</li> <li>– 14 - 6 for QMI, 9 for MBIM</li> <li>– 15 - 6 for QMI, 10 for MBIM</li> <li>– 16 - 6 for QMI, 11 for MBIM</li> <li>– 17 - 6 for QMI, 12 for MBIM</li> <li>– 18 - 7 for QMI, 8 for MBIM</li> <li>– 19 - 7 for QMI, 9 for MBIM</li> <li>– 20 - 7 for QMI, 10 for MBIM</li> <li>– 21 - 7 for QMI, 11 for MBIM</li> <li>– 22 - 7 for QMI, 12 for MBIM</li> </ul> </li> </ul>
----------------------	--

<i>pNumSupUSBComps[OUT]</i>	<ul style="list-style-type: none"> <li>• Number of supported USB compositions in the parameter to follow</li> <li>• Range - 0-255</li> </ul>
<i>pSupUSBComps[OUT]</i>	<ul style="list-style-type: none"> <li>• List of supported USB compositions( 1 Byte each - Max 255 )</li> <li>• Total length is defined by pNumSupUSBComps parameter</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0..5 - Reserved (non-QMI)</li> <li>– 6 - DM NMEA AT QMI</li> <li>– 7 - DM NMEA AT QMI1 QMI2 QMI3</li> <li>– 8 - DM NMEA AT MBIM</li> <li>– 9 - MBIM</li> <li>– 10 - NMEA MBIM</li> <li>– 11 - DM MBIM</li> <li>– 12 - DM NMEA MBIM</li> <li>13-22 are combined compositions. One is for Win8 MBIM interfaces, another is for legacy QMI interfaces</li> <li>– 13 - 6 for QMI, 8 for MBIM</li> <li>– 14 - 6 for QMI, 9 for MBIM</li> <li>– 15 - 6 for QMI, 10 for MBIM</li> <li>– 16 - 6 for QMI, 11 for MBIM</li> <li>– 17 - 6 for QMI, 12 for MBIM</li> <li>– 18 - 7 for QMI, 8 for MBIM</li> <li>– 19 - 7 for QMI, 9 for MBIM</li> <li>– 20 - 7 for QMI, 10 for MBIM</li> <li>– 21 - 7 for QMI, 11 for MBIM</li> <li>– 22 - 7 for QMI, 12 for MBIM</li> </ul> </li> </ul>

## 8.522.2 Field Documentation

8.522.2.1 **BYTE\*** USBCompParams::pNumSupUSBComps

8.522.2.2 **BYTE\*** USBCompParams::pSupUSBComps

8.522.2.3 **BYTE\*** USBCompParams::pUSBComp

## 8.523 USSDNoWaitIndicationInfo Struct Reference

### Data Fields

- **BYTE \*** pError
- **BYTE \*** pFailureCause
- struct USSInfo \* pUSSDData
- **alphaIDInfo \*** pAlphaIdentifier

### 8.523.1 Detailed Description

Contains the parameters passed for USSDNoWaitIndicationCallback by the device.

#### Parameters

<i>pError</i>	<ul style="list-style-type: none"> <li>Type of Error (if any)</li> </ul>
<i>pFailureCause</i>	<ul style="list-style-type: none"> <li>Supplementary services failure cause</li> </ul>
<i>pUSSDData</i>	<ul style="list-style-type: none"> <li>USS Data from Network.</li> <li>See <a href="#">USSInfo</a> for more details.</li> </ul>

### 8.523.2 Field Documentation

8.523.2.1 `alphaIDInfo*` USSDNoWaitIndicationInfo::pAlphaIdentifier

8.523.2.2 `BYTE*` USSDNoWaitIndicationInfo::pError

8.523.2.3 `BYTE*` USSDNoWaitIndicationInfo::pFailureCause

8.523.2.4 `struct USSInfo*` USSDNoWaitIndicationInfo::pUSSDData

## 8.524 USSDRespFNetwork Struct Reference

### Data Fields

- `char *` [pTypeCode](#)
- `char *` [pRespData](#)

### 8.524.1 Detailed Description

This structure contains the response from the network

#### Parameters

<i>pTypeCode</i>	"0" USSD-Notify – text in pRespData "1" USSD-Request – text in pRespData "2" Session terminated by network "3" other local client (eg, SIM Toolkit) has responded "4" Operation not supported "5" Network timeout
<i>pRespData</i>	<ul style="list-style-type: none"> <li>points to a message string received from the network</li> </ul>

### 8.524.2 Field Documentation

8.524.2.1 `char*` USSDRespFNetwork::pRespData

8.524.2.2 `char*` USSDRespFNetwork::pTypeCode

## 8.525 USSInfo Struct Reference

### Data Fields

- [BYTE](#) [ussDCS](#)
- [BYTE](#) [ussLen](#)
- [BYTE](#) [ussData](#) [182]

### 8.525.1 Detailed Description

This structure contains USS Information

#### Parameters

<i>ussDCS</i>	<ul style="list-style-type: none"><li>• 1 - ASCII coding scheme</li><li>• 2 - 8-BIT coding scheme</li><li>• 3 - UCS2</li></ul>
<i>ussLen</i>	<ul style="list-style-type: none"><li>• Range 1 to 182</li></ul>
<i>ussData</i>	<ul style="list-style-type: none"><li>• Data encoded as per the DCS</li></ul>

### 8.525.2 Field Documentation

8.525.2.1 [BYTE](#) [USSInfo::ussData](#)[182]

8.525.2.2 [BYTE](#) [USSInfo::ussDCS](#)

8.525.2.3 [BYTE](#) [USSInfo::ussLen](#)

## 8.526 USSResp Struct Reference

### Data Fields

- [WORD](#) \* [pfailureCause](#)
- [alphaIDInfo](#) \* [pAlphaIDInfo](#)
- [struct](#) [USSInfo](#) \* [pUSSDInfo](#)
- [BYTE](#) \* [pCcResultType](#)
- [BYTE](#) \* [pCallId](#)
- [ccSUPSType](#) \* [pCCSuppsType](#)

### 8.526.1 Field Documentation

8.526.1.1 [alphaIDInfo](#)\* [USSResp::pAlphaIDInfo](#)

8.526.1.2 [BYTE](#)\* [USSResp::pCallId](#)

8.526.1.3 **BYTE\*** USSResp::pCcResultType

8.526.1.4 **ccSUPSType\*** USSResp::pCCSuppsType

8.526.1.5 **WORD\*** USSResp::pfailureCause

8.526.1.6 **struct USSInfo\*** USSResp::pUSSDInfo

## 8.527 UUSInfo Struct Reference

### Data Fields

- [BYTE UUSType](#)
- [BYTE USSDCs](#)
- [BYTE USSDataLen](#)
- [BYTE USSData](#) [255]

### 8.527.1 Detailed Description

This structure contains User to User Signaling Service Information.

#### Parameters

<i>UUSType</i>	<ul style="list-style-type: none"><li>• UUS type values are:<ul style="list-style-type: none"><li>– 0x00 - UUS_DATA</li><li>– 0x01 - UUS_TYPE1_IMPLICIT</li><li>– 0x02 - UUS_TYPE1_REQUIRED</li><li>– 0x03 - UUS_TYPE1_NOT_REQUIRED</li><li>– 0x04 - UUS_TYPE2_REQUIRED</li><li>– 0x05 - UUS_TYPE2_NOT_REQUIRED</li><li>– 0x06 - UUS_TYPE3_REQUIRED</li><li>– 0x07 - UUS_TYPE3_NOT_REQUIRED</li><li>– 0xFF - Not Available</li></ul></li></ul>
----------------	--



<i>UUSDcs</i>	<ul style="list-style-type: none"> <li>• UUS data coding scheme values are: <ul style="list-style-type: none"> <li>– 0x01 - UUS_DCS_USP</li> <li>– 0x02 - UUS_DCS_OHLP</li> <li>– 0x03 - UUS_DCS_X244</li> <li>– 0x04 - UUS_DCS_SMCF</li> <li>– 0x05 - UUS_DCS_IA5</li> <li>– 0x06 - UUS_DCS_RV12RD</li> <li>– 0x07 - UUS_DCS_Q931UNCCM</li> <li>– 0xFF - Not Available</li> </ul> </li> </ul>
<i>UUSDatalen</i>	<ul style="list-style-type: none"> <li>• Number of sets of the following elements. <ul style="list-style-type: none"> <li>– UUSData</li> </ul> </li> <li>• If zero(0) then no further information exists.</li> </ul>
<i>UUSData[MAX_DESCRIPTION_LENGTH]</i>	<ul style="list-style-type: none"> <li>• UUS data encoded as per coding scheme</li> </ul>

## 8.527.2 Field Documentation

8.527.2.1 **BYTE** UUSInfo::UUSData[255]

8.527.2.2 **BYTE** UUSInfo::UUSDatalen

8.527.2.3 **BYTE** UUSInfo::UUSDcs

8.527.2.4 **BYTE** UUSInfo::UUSType

## 8.528 verifyUIMPIN Struct Reference

### Data Fields

- [BYTE](#) pinID
- [BYTE](#) pinLen
- [BYTE](#) pinVal [255]

### 8.528.1 Detailed Description

This structure contains the information about the pin parameters that need to be verified.

## Parameters

<i>pinID</i>	<ul style="list-style-type: none"> <li>Indicates the PIN ID to be verified. <ul style="list-style-type: none"> <li>1 - PIN1 (also called PIN)</li> <li>2 - PIN2</li> <li>3 - Universal PIN</li> <li>4 - Hidden key</li> </ul> </li> </ul>
<i>pinLen</i>	<ul style="list-style-type: none"> <li>Length of the following elements i.e. pin value.</li> </ul>
<i>pinVal</i> [MAX_DESCRIPTION_LENGTH]	<ul style="list-style-type: none"> <li>PIN value.</li> <li>This value is a sequence of ASCII characters.</li> </ul>

## 8.528.2 Field Documentation

8.528.2.1 BYTE verifyUIMPIN::pinID

8.528.2.2 BYTE verifyUIMPIN::pinLen

8.528.2.3 BYTE verifyUIMPIN::pinVal[255]

## 8.529 voiceALSSelectLineInfo Struct Reference

## Data Fields

- [BYTE lineValue](#)

## 8.529.1 Detailed Description

This structure contains ALS Select Line Information Parameters.

## Parameters

<i>lineValue</i>	<ul style="list-style-type: none"> <li>ALS Line Value. <ul style="list-style-type: none"> <li>0x00 - ALS_LINE1 - Line 1 (default)</li> <li>0x01 - ALS_LINE2 - Line 2</li> </ul> </li> </ul>
------------------	---

## 8.529.2 Field Documentation

8.529.2.1 BYTE voiceALSSelectLineInfo::lineValue

## 8.530 voiceALSSetLineSwitchInfo Struct Reference

## Data Fields

- [BYTE switchOption](#)

### 8.530.1 Detailed Description

This structure contains ALS Set Line Switching Information Parameters.

#### Parameters

<i>switchOption</i>	<ul style="list-style-type: none"><li>• Switch Option.<ul style="list-style-type: none"><li>– 0x00 - VOICE_LINE_SWITCHING_NOT_ALLOWED - Line switching is not allowed</li><li>– 0x01 - VOICE_LINE_SWITCHING_ALLOWED - Line switching is allowed</li></ul></li></ul>
---------------------	---

### 8.530.2 Field Documentation

#### 8.530.2.1 BYTE voiceALSSetLineSwitchInfo::switchOption

## 8.531 voiceAnswerCall Struct Reference

## Data Fields

- [BYTE \\* pCallId](#)

### 8.531.1 Detailed Description

Contains the parameters passed for SLQSVoiceAnswerCall.

#### Parameters

<i>pCallId[IN/OUT]</i>	<ul style="list-style-type: none"><li>• Unique call identifier for the call that must be answered.</li></ul>
------------------------	--

### 8.531.2 Field Documentation

#### 8.531.2.1 BYTE\* voiceAnswerCall::pCallId

## 8.532 voiceBindSubscriptionInfo Struct Reference

## Data Fields

- [BYTE subType](#)

### 8.532.1 Detailed Description

This structure contains Bind Subscription Information Parameters.

## Parameters

<i>subsType</i>	<ul style="list-style-type: none"> <li>Subscription Type. <ul style="list-style-type: none"> <li>0x00 - VOICE_SUBS_TYPE_PRIMARY - Primary</li> <li>0x01 - VOICE_SUBS_TYPE_SECONDARY - Secondary</li> </ul> </li> </ul>
-----------------	--

## 8.532.2 Field Documentation

## 8.532.2.1 BYTE voiceBindSubscriptionInfo::subsType

## 8.533 voiceBurstDTMFInfo Struct Reference

## Data Fields

- [burstDTMFInfo](#) [BurstDTMFInfo](#)
- [DTMFLengths](#) \* [pBurstDTMFLengths](#)

## 8.533.1 Detailed Description

This structure contains parameters of burst Dual-Tone Multifrequency (DTMF)

## Parameters

<i>BurstDTMFInfo</i>	<ul style="list-style-type: none"> <li>Burst DTMF Information <ul style="list-style-type: none"> <li>See <a href="#">burstDTMFInfo</a> for more information</li> </ul> </li> </ul>
<i>pBurstDTMF- Lengths</i>	[optional] <ul style="list-style-type: none"> <li>DTMF Lengths <ul style="list-style-type: none"> <li>See <a href="#">DTMFLengths</a> for more information</li> </ul> </li> </ul>

## 8.533.2 Field Documentation

## 8.533.2.1 burstDTMFInfo voiceBurstDTMFInfo::BurstDTMFInfo

## 8.533.2.2 DTMFLengths\* voiceBurstDTMFInfo::pBurstDTMFLengths

## 8.534 voiceCallInfoReq Struct Reference

## Data Fields

- [BYTE](#) [callID](#)

## 8.534.1 Detailed Description

This structure contains information of the request parameters associated with a call.

## Parameters

<i>callID</i>	<ul style="list-style-type: none"> <li>• Call identifier for the call queried for information.</li> </ul>
---------------	---

## 8.534.2 Field Documentation

## 8.534.2.1 BYTE voiceCallInfoReq::callID

## 8.535 voiceCallInfoResp Struct Reference

## Data Fields

- [callInfo](#) \* [pCallInfo](#)
- [remotePartyNum](#) \* [pRemotePartyNum](#)
- [WORD](#) \* [pSrvOpt](#)
- [BYTE](#) \* [pVoicePrivacy](#)
- [BYTE](#) \* [pOTASPStatus](#)
- [remotePartyName](#) \* [pRemotePartyName](#)
- [UUSInfo](#) \* [pUUSInfo](#)
- [BYTE](#) \* [pAlertType](#)
- [alphaIDInfo](#) \* [pAlphaIDInfo](#)
- [connectNumInfo](#) \* [pConnectNumInfo](#)
- [diagInfo](#) \* [pDiagInfo](#)
- [ULONG](#) \* [pAlertingPattern](#)

## 8.535.1 Detailed Description

This structure contains information of the response parameters associated with a call.

## Parameters

<i>pCallInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">callInfo</a> for more information.</li> </ul>
<i>pRemoteParty-Num</i>	<ul style="list-style-type: none"> <li>• See <a href="#">remotePartyNum</a> for more information.</li> </ul>
<i>pSrvOpt</i>	<ul style="list-style-type: none"> <li>• Service option</li> <li>• Applicable only for 3GPP2 devices.</li> <li>• See Table8 <a href="#">qaGobiApiTableServiceOptions.h</a> for standard service option number assignments.</li> </ul>

<i>pVoicePrivacy</i>	<ul style="list-style-type: none"> <li>• Voice Privacy.</li> <li>• Applicable only for 3GPP2 devices.</li> <li>• Values. <ul style="list-style-type: none"> <li>– 0x00 - VOICE_PRIVACY_STANDARD - Standard privacy</li> <li>– 0x01 - VOICE_PRIVACY_ENHANCED - Enhanced privacy</li> <li>– 0xFF - Not Available</li> </ul> </li> </ul>
<i>pOTASPStatus</i>	<ul style="list-style-type: none"> <li>• OTASP status for the OTASP call.</li> <li>• Applicable only for 3GPP2 devices. <ul style="list-style-type: none"> <li>– 0x00 - OTASP_STATUS_SPL_UNLOCKED - SPL unlocked; only for user-initiated OTASP</li> <li>– 0x01 - OTASP_STATUS_SPRC_RETRIES_EXCEEDED - SPC retries exceeded; only for user-initiated OTASP</li> <li>– 0x02 - OTASP_STATUS_AKEY_EXCHANGED - A-key exchanged; only for user-initiated OTASP</li> <li>– 0x03 - OTASP_STATUS_SSD_UPDATED - SSD updated; for both user-initiated OTASP and network-initiated OTASP (OTAPA)</li> <li>– 0x04 - OTASP_STATUS_NAM_DOWNLOADED - NAM downloaded; only for user-initiated OTASP</li> <li>– 0x05 - OTASP_STATUS_MDN_DOWNLOADED - MDN downloaded; only for user-initiated OTASP</li> <li>– 0x06 - OTASP_STATUS_IMSI_DOWNLOADED - IMSI downloaded; only for user-initiated OTASP</li> <li>– 0x07 - OTASP_STATUS_PRL_DOWNLOADED - PRL downloaded; only for user-initiated OTASP</li> <li>– 0x08 - OTASP_STATUS_COMMITTED - Commit successful; only for user-initiated OTASP</li> <li>– 0x09 - OTASP_STATUS_OTAPA_STARTED - OTAPA started; only for network-initiated OTASP (OTAPA)</li> <li>– 0x0A - OTASP_STATUS_OTAPA_STOPPED - OTAPA stopped; only for network-initiated OTASP (OTAPA)</li> <li>– 0x0B - OTASP_STATUS_OTAPA_ABORTED - OTAPA aborted; only for network-initiated OTASP (OTAPA)</li> <li>– 0x0C - OTASP_STATUS_OTAPA_COMMITTED - OTAPA committed; only for network-initiated OTASP (OTAPA)</li> <li>– 0xFF - Not Available</li> </ul> </li> </ul>
<i>pRemotePartyName</i>	<ul style="list-style-type: none"> <li>• Applicable only for 3GPP devices.</li> <li>• See <a href="#">remotePartyName</a> for more information.</li> </ul>
<i>pUUSInfo</i>	<ul style="list-style-type: none"> <li>• Applicable only for 3GPP devices.</li> <li>• See <a href="#">UUSInfo</a> for more information.</li> </ul>
<i>pAlertType</i>	<p>Generated on Wed Oct 28 2015 00:40:12 for LinuxQMISDK by Doxygen</p> <ul style="list-style-type: none"> <li>• Alerting type.</li> <li>• Applicable only for 3GPP devices.</li> </ul>

## 8.535.2 Field Documentation

- 8.535.2.1 **ULONG\*** voiceCallInfoResp::pAlertingPattern
- 8.535.2.2 **BYTE\*** voiceCallInfoResp::pAlertType
- 8.535.2.3 **alphaDInfo\*** voiceCallInfoResp::pAlphaDInfo
- 8.535.2.4 **callInfo\*** voiceCallInfoResp::pCallInfo
- 8.535.2.5 **connectNumInfo\*** voiceCallInfoResp::pConnectNumInfo
- 8.535.2.6 **diagInfo\*** voiceCallInfoResp::pDiagInfo
- 8.535.2.7 **BYTE\*** voiceCallInfoResp::pOTASPStatus
- 8.535.2.8 **remotePartyName\*** voiceCallInfoResp::pRemotePartyName
- 8.535.2.9 **remotePartyNum\*** voiceCallInfoResp::pRemotePartyNum
- 8.535.2.10 **WORD\*** voiceCallInfoResp::pSrvOpt
- 8.535.2.11 **UUSInfo\*** voiceCallInfoResp::pUUSInfo
- 8.535.2.12 **BYTE\*** voiceCallInfoResp::pVoicePrivacy

## 8.536 voiceCallRequestParams Struct Reference

### Data Fields

- [BYTE](#) callNumber [81]
- [BYTE \\*](#) pCallType
- [BYTE \\*](#) pCLIRType
- [UUSInfo \\*](#) pUUSInfo
- [CUGInfo \\*](#) pCUGInfo
- [BYTE \\*](#) pEmergencyCategory
- [calledPartySubAdd \\*](#) pCallPartySubAdd
- [ULONG \\*](#) pSvcType

### 8.536.1 Detailed Description

This structure contains Voice Call Request Parameters

#### Parameters

<i>callNumber[81]</i>	<ul style="list-style-type: none"><li>• Number to be dialed in ASCII string, NULL terminated.</li><li>• Length Range [1 to 81]</li></ul>
-----------------------	--

<i>pCallType</i>	<ul style="list-style-type: none"> <li>the type of call to be dialed. CALL_TYPE_VOICE is automatically selected if this parameter is not provided. When CALL_TYPE_NON_STD_OTASP is selected, the call is sent as a nonstandard OTASP call regardless of the digit string Call type values are: <ul style="list-style-type: none"> <li>0x00 - CALL_TYPE_VOICE - Voice (automatic selection)</li> <li>0x01 - CALL_TYPE_VOICE_FORCED - Avoid modem call classification</li> <li>0x08 - CALL_TYPE_NON_STD_OTASP - Nonstandard OTASP*</li> <li>0x09 - CALL_TYPE_EMERGENCY - Emergency</li> </ul> </li> </ul>
<i>pCLIRType</i>	<ul style="list-style-type: none"> <li>CLIR type values are: <ul style="list-style-type: none"> <li>0x01 - CLIR_SUPPRESSION - Suppression</li> <li>0x02 - CLIR_INVOCATION - Invocation</li> </ul> </li> </ul>
<i>pUUSInfo</i>	<ul style="list-style-type: none"> <li>Pointer to structure of <a href="#">UUSInfo</a> <ul style="list-style-type: none"> <li>See <a href="#">UUSInfo</a> for more information</li> </ul> </li> </ul>
<i>pCUGInfo</i>	<ul style="list-style-type: none"> <li>Pointer to structure of <a href="#">CUGInfo</a> <ul style="list-style-type: none"> <li>See <a href="#">CUGInfo</a> for more information</li> </ul> </li> </ul>
<i>pEmergency-Category</i>	<ul style="list-style-type: none"> <li>Bit mask of emergency number categories. This is only applicable when the call type is set to Emergency. <ul style="list-style-type: none"> <li>Bit 0 - VOICE_EMER_CAT_POLICE_BIT - Police</li> <li>Bit 1 - VOICE_EMER_CAT_AMBULANCE_BIT - Ambulance</li> <li>Bit 2 - VOICE_EMER_CAT_FIRE_BRIGADE_BIT - Fire brigade</li> <li>Bit 3 - VOICE_EMER_CAT_MARINE_GUARD_BIT - Marine guard</li> <li>Bit 4 - VOICE_EMER_CAT_MOUNTAIN_RESCUE_BIT - Mountain rescue</li> <li>Bit 5 - VOICE_EMER_CAT_MANUAL_ECALL_BIT - Manual emergency call</li> <li>Bit 6 - VOICE_EMER_CAT_AUTO_ECALL_BIT - Automatic emergency call</li> <li>Bit 7 - VOICE_EMER_CAT_SPARE_BIT - Spare bit</li> </ul> </li> </ul>
<i>pCallPartySub-Add</i>	<ul style="list-style-type: none"> <li>Pointer to structure of <a href="#">calledPartySubAdd</a> <ul style="list-style-type: none"> <li>See <a href="#">calledPartySubAdd</a> for more information</li> </ul> </li> </ul>
<i>pSvcType</i>	<ul style="list-style-type: none"> <li>Service Type. <ul style="list-style-type: none"> <li>0x01 - VOICE_DIAL_CALL_SRV_TYPE_AUTOMATIC - Automatic</li> <li>0x02 - VOICE_DIAL_CALL_SRV_TYPE_GSM - GSM</li> <li>0x03 - VOICE_DIAL_CALL_SRV_TYPE_WCDMA - WCDMA</li> <li>0x04 - VOICE_DIAL_CALL_SRV_TYPE_CDMA_AUTOMATIC - CDMA automatic</li> <li>0x05 - VOICE_DIAL_CALL_SRV_TYPE_GSM_WCDMA - GSM or WCDMA</li> <li>0x06 - VOICE_DIAL_CALL_SRV_TYPE_LTE -LTE</li> </ul> </li> </ul>



## 8.536.2 Field Documentation

8.536.2.1 **BYTE** voiceCallRequestParams::callNumber[81]

8.536.2.2 **calledPartySubAdd\*** voiceCallRequestParams::pCallPartySubAdd

8.536.2.3 **BYTE\*** voiceCallRequestParams::pCallType

8.536.2.4 **BYTE\*** voiceCallRequestParams::pCLIRType

8.536.2.5 **CUGInfo\*** voiceCallRequestParams::pCUGInfo

8.536.2.6 **BYTE\*** voiceCallRequestParams::pEmergencyCategory

8.536.2.7 **ULONG\*** voiceCallRequestParams::pSvcType

8.536.2.8 **UUSInfo\*** voiceCallRequestParams::pUUSInfo

## 8.537 voiceCallResponseParams Struct Reference

### Data Fields

- **BYTE \*** [pCallID](#)
- [alphaIDInfo](#) \* [pAlphaIDInfo](#)
- **BYTE \*** [pCCResultType](#)
- [ccSUPSType](#) \* [pCCSUPSType](#)

### 8.537.1 Detailed Description

This structure contains Voice Call Response Parameters

#### Parameters

<i>pCallID</i>	<ul style="list-style-type: none"> <li>• Unique call identifier for the dialed call</li> </ul>
<i>pAlphaIDInfo</i>	<ul style="list-style-type: none"> <li>• Pointer to structure of <a href="#">alphaIDInfo</a> <ul style="list-style-type: none"> <li>– See <a href="#">alphaIDInfo</a> for more information</li> </ul> </li> </ul>
<i>pCCResultType</i>	<ul style="list-style-type: none"> <li>• Call Control Result Type. <ul style="list-style-type: none"> <li>– 0x00 - CC_RESULT_TYPE_VOICE - Voice</li> <li>– 0x01 - CC_RESULT_TYPE_SUPS - Supplementary service</li> <li>– 0x02 - CC_RESULT_TYPE_USSD - Unstructured supplementary service</li> </ul> </li> </ul>

<i>pCCSUPSType</i>	<ul style="list-style-type: none"> <li>• Pointer to structure of <a href="#">ccSUPSType</a></li> <li>• Data is present when pCCResultType is present and is other than Voice. <ul style="list-style-type: none"> <li>– See <a href="#">ccSUPSType</a> for more information</li> </ul> </li> </ul>
--------------------	---

## 8.537.2 Field Documentation

8.537.2.1 **alphaIDInfo\*** `voiceCallResponseParams::pAlphaIDInfo`

8.537.2.2 **BYTE\*** `voiceCallResponseParams::pCallID`

8.537.2.3 **BYTE\*** `voiceCallResponseParams::pCCResultType`

8.537.2.4 **ccSUPSType\*** `voiceCallResponseParams::pCCSUPSType`

## 8.538 voiceContDTMFInfo Struct Reference

### Data Fields

- [BYTE \\*](#) `pCallID`
- [BYTE](#) `DTMFdigit`

### 8.538.1 Detailed Description

This structure contains parameters of continuous DTMF

#### Parameters

<i>pCallID[IN/OUT]</i>	<ul style="list-style-type: none"> <li>• Call ID associated with call on which the DTMF information has to be sent. Start continuous DTMF request is sent to the current active/alerting call when pCallId is set to 0xFF.</li> <li>• This is IN/OUT parameter, value passed by user will be packed in request and value received from the device would be returned to the user.</li> <li>• If the call ID value received is 0, no value has been returned by the device</li> </ul>
<i>DTMFdigit</i>	<ul style="list-style-type: none"> <li>• DTMF digit in ASCII.</li> </ul>

## 8.538.2 Field Documentation

8.538.2.1 **BYTE** `voiceContDTMFInfo::DTMFdigit`

8.538.2.2 **BYTE\*** `voiceContDTMFInfo::pCallID`

## 8.539 voiceDTMFEventInfo Struct Reference

## Data Fields

- [DTMFInfo](#) DTMFInformation
- [BYTE](#) \* pOnLength
- [BYTE](#) \* pOffLength

## 8.539.1 Detailed Description

This structure contains the parameters passed for SLQSVoiceSetDTMFEventCallBack by the device.

## Parameters

<i>DTMF- Information(mandatory)</i>	• See <a href="#">DTMFInfo</a> for more information.
<i>pOn- Length(optional)</i>	<ul style="list-style-type: none"> <li>• DTMF Pulse Width <ul style="list-style-type: none"> <li>– 0x00 - DTMF_ONLENGTH_95MS - 95 ms</li> <li>– 0x01 - DTMF_ONLENGTH_150MS - 150 ms</li> <li>– 0x02 - DTMF_ONLENGTH_200MS - 200 ms</li> <li>– 0x03 - DTMF_ONLENGTH_250MS - 250 ms</li> <li>– 0x04 - DTMF_ONLENGTH_300MS - 300 ms</li> <li>– 0x05 - DTMF_ONLENGTH_350MS - 350 ms</li> <li>– 0x06 - DTMF_ONLENGTH_SMS - SMS Tx special pulse width</li> </ul> </li> </ul>
<i>pOff- Length(optional)</i>	<ul style="list-style-type: none"> <li>• DTMF Interdigit Interval <ul style="list-style-type: none"> <li>– 0x00 - DTMF_OFFLENGTH_60MS - 60 ms</li> <li>– 0x01 - DTMF_OFFLENGTH_100MS - 100 ms</li> <li>– 0x02 - DTMF_OFFLENGTH_150MS - 150 ms</li> <li>– 0x03 - DTMF_OFFLENGTH_200MS - 200 ms</li> </ul> </li> </ul>

## Note

None

## 8.539.2 Field Documentation

8.539.2.1 [DTMFInfo](#) voiceDTMFEventInfo::DTMFInformation

8.539.2.2 [BYTE](#)\* voiceDTMFEventInfo::pOffLength

8.539.2.3 [BYTE](#)\* voiceDTMFEventInfo::pOnLength

## 8.540 voiceFlashInfo Struct Reference

## Data Fields

- [BYTE](#) \* pCallID

- [BYTE \\* pFlashPayLd](#)
- [BYTE \\* pFlashType](#)

### 8.540.1 Detailed Description

This structure contains the flash information associated with a call.

#### Parameters

<i>pCallID[IN/OUT]</i>	<ul style="list-style-type: none"> <li>• Unique call identifier associated with the current call.</li> </ul>
<i>pFlashPayLd[IN](optional)</i>	<ul style="list-style-type: none"> <li>• Payload in ASCII to be sent in Flash.</li> <li>• Variable Length, NULL terminated.</li> </ul>
<i>pFlashType[IN](optional)</i>	<ul style="list-style-type: none"> <li>• Flash type. <ul style="list-style-type: none"> <li>– 0 - Simple Flash (default)</li> <li>– 1 - Activate answer hold</li> <li>– 2 - Deactivate answer hold</li> </ul> </li> </ul>

### 8.540.2 Field Documentation

8.540.2.1 [BYTE\\* voiceFlashInfo::pCallID](#)

8.540.2.2 [BYTE\\* voiceFlashInfo::pFlashPayLd](#)

8.540.2.3 [BYTE\\* voiceFlashInfo::pFlashType](#)

## 8.541 voiceGetAllCallInfo Struct Reference

### Data Fields

- [arrCallInfo \\* pArrCallInfo](#)
- [arrRemotePartyNum \\* pArrRemotePartyNum](#)
- [arrRemotePartyName \\* pArrRemotePartyName](#)
- [arrAlertingType \\* pArrAlertingType](#)
- [arrUUSInfo \\* pArrUUSInfo](#)
- [arrSvcOption \\* pArrSvcOption](#)
- [BYTE \\* pOTASPSStatus](#)
- [BYTE \\* pVoicePrivacy](#)
- [arrCallEndReason \\* pArrCallEndReason](#)
- [arrAlphaID \\* pArrAlphaID](#)
- [arrConnectPartyNum \\* pArrConnectPartyNum](#)
- [arrDiagInfo \\* pArrDiagInfo](#)
- [arrCalledPartyNum \\* pArrCalledPartyNum](#)
- [arrRedirPartyNum \\* pArrRedirPartyNum](#)
- [arrAlertingPattern \\* pArrAlertingPattern](#)

### 8.541.1 Detailed Description

This structure contains information about the response parameters with all the calls originating or terminating from a particular device.

## Parameters

<i>pArrCallInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">arrCallInfo</a> for more information.</li> </ul>
<i>pArrRemotePartyNum</i>	<ul style="list-style-type: none"> <li>• See <a href="#">arrRemotePartyNum</a> for more information.</li> </ul>
<i>pArrRemotePartyName</i>	<ul style="list-style-type: none"> <li>• See <a href="#">arrRemotePartyName</a> for more information.</li> </ul>
<i>pArrAlertingType</i>	<ul style="list-style-type: none"> <li>• See <a href="#">arrAlertingType</a> for more information.</li> </ul>
<i>pArrUUSInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">arrUUSInfo</a> for more information.</li> </ul>
<i>pArrSvcOption</i>	<ul style="list-style-type: none"> <li>• See <a href="#">arrSvcOption</a> for more information.</li> </ul>
<i>pOTASPStatus</i>	<ul style="list-style-type: none"> <li>• OTASP status for the OTASP call.</li> <li>• Applicable only for 3GPP2 devices. <ul style="list-style-type: none"> <li>– 0x00 - OTASP_STATUS_SPL_UNLOCKED - SPL unlocked; only for user-initiated OTASP</li> <li>– 0x01 - OTASP_STATUS_SPRC_RETRIES_EXCEEDED - SPC retries exceeded; only for user-initiated OTASP</li> <li>– 0x02 - OTASP_STATUS_AKEY_EXCHANGED - A-key exchanged; only for user-initiated OTASP</li> <li>– 0x03 - OTASP_STATUS_SSD_UPDATED - SSD updated; for both user-initiated OTASP and network-initiated OTASP (OTAPA)</li> <li>– 0x04 - OTASP_STATUS_NAM_DOWNLOADED - NAM downloaded; only for user-initiated OTASP</li> <li>– 0x05 - OTASP_STATUS_MDN_DOWNLOADED - MDN downloaded; only for user-initiated OTASP</li> <li>– 0x06 - OTASP_STATUS_IMSI_DOWNLOADED - IMSI downloaded; only for user-initiated OTASP</li> <li>– 0x07 - OTASP_STATUS_PRL_DOWNLOADED - PRL downloaded; only for user-initiated OTASP</li> <li>– 0x08 - OTASP_STATUS_COMMITTED - Commit successful; only for user-initiated OTASP</li> <li>– 0x09 - OTASP_STATUS_OTAPA_STARTED - OTAPA started; only for network-initiated OTASP (OTAPA)</li> <li>– 0x0A - OTASP_STATUS_OTAPA_STOPPED - OTAPA stopped; only for network-initiated OTASP (OTAPA)</li> <li>– 0x0B - OTASP_STATUS_OTAPA_ABORTED - OTAPA aborted; only for network-initiated OTASP (OTAPA)</li> <li>– 0x0C - OTASP_STATUS_OTAPA_COMMITTED - OTAPA committed; only for network-initiated OTASP (OTAPA)</li> <li>– 0xFF - Not Available</li> </ul> </li> </ul>
<i>pVoicePrivacy</i>	<p>Generated on Wed Oct 28 2015 00:40:12 for LinuxQMI SDK by Doxygen</p> <ul style="list-style-type: none"> <li>• Voice Privacy.</li> <li>• Values.</li> </ul>

<i>pArrCallEndReason</i>	<ul style="list-style-type: none"> <li>• See <a href="#">arrCallEndReason</a> for more information.</li> </ul>
<i>pArrAlphaID</i>	<ul style="list-style-type: none"> <li>• See <a href="#">arrAlphaID</a> for more information.</li> </ul>
<i>pArrConnectPartyNum</i>	<ul style="list-style-type: none"> <li>• See <a href="#">arrConnectPartyNum</a> for more information.</li> </ul>
<i>pArrDiagInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">arrDiagInfo</a> for more information.</li> </ul>
<i>pArrCalledPartyNum</i>	<ul style="list-style-type: none"> <li>• See <a href="#">arrCalledPartyNum</a> for more information.</li> </ul>
<i>pArrRedirPartyNum</i>	<ul style="list-style-type: none"> <li>• See <a href="#">arrRedirPartyNum</a> for more information.</li> </ul>
<i>pArrAlertingPattern</i>	<ul style="list-style-type: none"> <li>• See <a href="#">arrAlertingPattern</a> for more information.</li> </ul>

## 8.541.2 Field Documentation

8.541.2.1 **arrAlertingPattern\*** voiceGetAllCallInfo::pArrAlertingPattern

8.541.2.2 **arrAlertingType\*** voiceGetAllCallInfo::pArrAlertingType

8.541.2.3 **arrAlphaID\*** voiceGetAllCallInfo::pArrAlphaID

8.541.2.4 **arrCalledPartyNum\*** voiceGetAllCallInfo::pArrCalledPartyNum

8.541.2.5 **arrCallEndReason\*** voiceGetAllCallInfo::pArrCallEndReason

8.541.2.6 **arrCallInfo\*** voiceGetAllCallInfo::pArrCallInfo

8.541.2.7 **arrConnectPartyNum\*** voiceGetAllCallInfo::pArrConnectPartyNum

8.541.2.8 **arrDiagInfo\*** voiceGetAllCallInfo::pArrDiagInfo

8.541.2.9 **arrRedirPartyNum\*** voiceGetAllCallInfo::pArrRedirPartyNum

8.541.2.10 **arrRemotePartyName\*** voiceGetAllCallInfo::pArrRemotePartyName

8.541.2.11 **arrRemotePartyNum\*** voiceGetAllCallInfo::pArrRemotePartyNum

8.541.2.12 **arrSvcOption\*** voiceGetAllCallInfo::pArrSvcOption

8.541.2.13 **arrUUSInfo\*** voiceGetAllCallInfo::pArrUUSInfo

8.541.2.14 **BYTE\*** voiceGetAllCallInfo::pOTASPStatus

8.541.2.15 **BYTE\*** voiceGetAllCallInfo::pVoicePrivacy

## 8.542 voiceGetCallBarringReq Struct Reference

### Data Fields

- [BYTE reason](#)
- [BYTE \\* pSvcClass](#)

### 8.542.1 Detailed Description

This structure contains Voice Get Call Barring Request Parameters

#### Parameters

<i>reason</i>	<ul style="list-style-type: none"> <li>• Call Barring Reason</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0x07 - QMI_VOICE_REASON_BARR_ALLOUTGOING - All outgoing</li> <li>– 0x08 - QMI_VOICE_REASON_BARR_OUTGOINGINT - Outgoing internal</li> <li>– 0x09 - QMI_VOICE_REASON_BARR_OUTGOINGINTEXTOHOM - Outgoing external to home</li> <li>– 0x0A - QMI_VOICE_REASON_BARR_ALLINCOMING - All incoming</li> <li>– 0x0B - QMI_VOICE_REASON_BARR_INCOMINGROAMING - Roaming incoming</li> <li>– 0x0C - QMI_VOICE_REASON_BARR_ALLBARRING - All calls are barred</li> <li>– 0x0D - QMI_VOICE_REASON_BARR_ALLOUTGOINGBARRING - All outgoing calls are barred</li> <li>– 0x0E - QMI_VOICE_REASON_BARR_ALLINCOMINGBARRING - All incoming calls are barred</li> </ul> </li> </ul>
---------------	--



<i>pSvcClass</i> [IN/OUT]	<ul style="list-style-type: none"> <li>• Service class is a combination (sum) of information class constants (optional)</li> <li>• See <a href="#">qaGobiApiTableSupServiceInfoClasses.h</a> for service classes.</li> <li>• Service Class is set to 0 if call waiting is not active for any of the information classes.</li> <li>• 0xFF,if Not Available</li> </ul>
---------------------------	--

### 8.542.2 Field Documentation

8.542.2.1 **BYTE\*** voiceGetCallBarringReq::pSvcClass

8.542.2.2 **BYTE** voiceGetCallBarringReq::reason

## 8.543 voiceGetCallBarringResp Struct Reference

### Data Fields

- **BYTE \*** pSvcClass
- **WORD \*** pFailCause
- **alphaIDInfo \*** pAlphaIDInfo
- **BYTE \*** pCCResType
- **BYTE \*** pCallID
- **ccSUPSType \*** pCCSUPSType

### 8.543.1 Detailed Description

This structure contains Voice Get Call Barring Response Parameters

#### Parameters

<i>pSvcClass</i> [IN/OUT]	<ul style="list-style-type: none"> <li>• Service class is a combination (sum) of information class constants (optional)</li> <li>• See <a href="#">qaGobiApiTableSupServiceInfoClasses.h</a> for service classes.</li> <li>• Service Class is set to 0 if call waiting is not active for any of the information classes.</li> <li>• 0xFF,if Not Available</li> </ul>
<i>pFailCause</i>	<ul style="list-style-type: none"> <li>• Supplementary services failure cause (optional)</li> <li>• see <a href="#">qaGobiApiTableVoiceCallEndReasons.h</a> for more information.</li> <li>• 0xFFFF,if Not Available</li> </ul>

<i>pAlphaIDInfo</i>	<ul style="list-style-type: none"> <li>• Pointer to structure of <a href="#">alphaIDInfo</a> (optional) <ul style="list-style-type: none"> <li>– See <a href="#">alphaIDInfo</a> for more information</li> </ul> </li> </ul>
<i>pCCResType</i>	<ul style="list-style-type: none"> <li>• Call Control Result Type (optional) <ul style="list-style-type: none"> <li>– 0x00 - CC_RESULT_TYPE_VOICE - Voice</li> <li>– 0x01 - CC_RESULT_TYPE_SUPS - Supplementary service</li> <li>– 0x02 - CC_RESULT_TYPE_USSD - Unstructured supplementary service</li> <li>– 0xFF - Not Available</li> </ul> </li> </ul>
<i>pCallID</i>	<ul style="list-style-type: none"> <li>• Call ID of the voice call that resulted from call control. (optional)</li> <li>• It is present when pCCResType is present and is Voice.</li> <li>• If zero(0) then invalid.</li> </ul>
<i>pCCSUPSType</i>	<ul style="list-style-type: none"> <li>• Supplementary service data that resulted from call control (optional)</li> <li>• Data is present when pCCResType is present and is other than Voice. <ul style="list-style-type: none"> <li>– See <a href="#">ccSUPSType</a> for more information</li> </ul> </li> </ul>

**Note**

Using NULL for the pointers would make sure that the parameter is not returned or has default value.

**8.543.2 Field Documentation**

8.543.2.1 **alphaIDInfo\*** voiceGetCallBarringResp::pAlphaIDInfo

8.543.2.2 **BYTE\*** voiceGetCallBarringResp::pCallID

8.543.2.3 **BYTE\*** voiceGetCallBarringResp::pCCResType

8.543.2.4 **ccSUPSType\*** voiceGetCallBarringResp::pCCSUPSType

8.543.2.5 **WORD\*** voiceGetCallBarringResp::pFailCause

8.543.2.6 **BYTE\*** voiceGetCallBarringResp::pSvcClass

**8.544 voiceGetCallFWReq Struct Reference****Data Fields**

- [BYTE Reason](#)
- [BYTE \\* pSvcClass](#)

### 8.544.1 Detailed Description

This structure contains Voice Get Call Forwarding Status Request Parameters

## Parameters

<i>Reason</i>	<ul style="list-style-type: none"> <li>• Call Forwarding Reason</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0x01 - QMI_VOICE_REASON_FWDREASON_UNCONDITIONAL - Unconditional call forwarding</li> <li>– 0x02 - QMI_VOICE_REASON_FWDREASON_MOBILEBUSY - Forward when the mobile is busy</li> <li>– 0x03 - QMI_VOICE_REASON_FWDREASON_NOREPLY - Forward when there is no reply</li> <li>– 0x04 - QMI_VOICE_REASON_FWDREASON_UNREACHABLE - Forward when the call is unreachable</li> <li>– 0x05 - QMI_VOICE_REASON_FWDREASON_ALLFORWARDING - All forwarding</li> <li>– 0x06 - QMI_VOICE_REASON_FWDREASON_ALLCONDITIONAL - All conditional forwarding</li> </ul> </li> </ul>
<i>pSvcClass</i>	<ul style="list-style-type: none"> <li>• Service Class is a combination (sum) of information class constants</li> <li>• See <a href="#">qaGobiApiTableSupServiceInfoClasses.h</a> for service classes.</li> </ul>

## 8.544.2 Field Documentation

8.544.2.1 **BYTE\*** voiceGetCallFWReq::pSvcClass8.544.2.2 **BYTE** voiceGetCallFWReq::Reason

## 8.545 voiceGetCallFWResp Struct Reference

## Data Fields

- [getCallFWInfo](#) \* [pGetCallFWInfo](#)
- [WORD](#) \* [pFailCause](#)
- [alphaIDInfo](#) \* [pAlphaIDInfo](#)
- [BYTE](#) \* [pCCResType](#)
- [BYTE](#) \* [pCallID](#)
- [ccSUPSType](#) \* [pCCSUPSType](#)
- [getCallFWExtInfo](#) \* [pGetCallFWExtInfo](#)

## 8.545.1 Detailed Description

This structure contains Voice Get Call Forwarding Status Response Parameters

## Parameters

<i>pGetCallFWInfo</i>	<ul style="list-style-type: none"> <li>• Pointer to structure of <a href="#">getCallFWInfo</a> (optional) <ul style="list-style-type: none"> <li>– See <a href="#">getCallFWInfo</a> for more information</li> </ul> </li> </ul>
<i>pFailCause</i>	<ul style="list-style-type: none"> <li>• Supplementary services failure cause (optional)</li> <li>• see <a href="#">qaGobiApiTableVoiceCallEndReasons.h</a> for more information.</li> <li>• 0xFFFF, if Not Available</li> </ul>
<i>pAlphaIDInfo</i>	<ul style="list-style-type: none"> <li>• Pointer to structure of <a href="#">alphaIDInfo</a> <ul style="list-style-type: none"> <li>– See <a href="#">alphaIDInfo</a> for more information</li> </ul> </li> </ul>
<i>pCCResType</i>	<ul style="list-style-type: none"> <li>• Call Control Result Type (optional) <ul style="list-style-type: none"> <li>– 0x00 - CC_RESULT_TYPE_VOICE - Voice</li> <li>– 0x01 - CC_RESULT_TYPE_SUPS - Supplementary service</li> <li>– 0x02 - CC_RESULT_TYPE_USSD - Unstructured supplementary service</li> <li>– 0xFF - Not Available</li> </ul> </li> </ul>
<i>pCallID</i>	<ul style="list-style-type: none"> <li>• Call ID of the voice call that resulted from call control. (optional)</li> <li>• It is present when pCCResType is present and is Voice.</li> <li>• If zero(0) then invalid.</li> </ul>
<i>pCCSUPSType</i>	<ul style="list-style-type: none"> <li>• Supplementary service data that resulted from call control (optional)</li> <li>• Data is present when pCCResType is present and is other than Voice. <ul style="list-style-type: none"> <li>– See <a href="#">ccSUPSType</a> for more information</li> </ul> </li> </ul>
<i>pGetCallFWExt-Info</i>	<ul style="list-style-type: none"> <li>• Pointer to structure of <a href="#">getCallFWExtInfo</a> (optional) <ul style="list-style-type: none"> <li>– See <a href="#">getCallFWExtInfo</a> for more information</li> </ul> </li> </ul>

**Note**

Using NULL for the pointers would make sure that the parameter is not returned or has default value.

**8.545.2 Field Documentation****8.545.2.1 alphaIDInfo\* voiceGetCallFWResp::pAlphaIDInfo**

8.545.2.2 **BYTE\*** voiceGetCallFWResp::pCallID

8.545.2.3 **BYTE\*** voiceGetCallFWResp::pCCResType

8.545.2.4 **ccSUPSType\*** voiceGetCallFWResp::pCCSUPSType

8.545.2.5 **WORD\*** voiceGetCallFWResp::pFailCause

8.545.2.6 **getCallFWExtInfo\*** voiceGetCallFWResp::pGetCallFWExtInfo

8.545.2.7 **getCallFWInfo\*** voiceGetCallFWResp::pGetCallFWInfo

## 8.546 voiceGetCallWaitInfo Struct Reference

### Data Fields

- **BYTE \*** pSvcClass
- **WORD \*** pFailCause
- **alphaIDInfo \*** pAlphaIDInfo
- **BYTE \*** pCCResType
- **BYTE \*** pCallID
- **ccSUPSType \*** pCCSUPSType

### 8.546.1 Detailed Description

This structure contains Voice Get Call Waiting Response Parameters

#### Parameters

<i>pSvcClass</i> [ <i>IN/O-UT</i> ]	<ul style="list-style-type: none"> <li>• Service class is a combination (sum) of information class constants (optional)</li> <li>• See <a href="#">qaGobiApiTableSupServiceInfoClasses.h</a> for service classes.</li> <li>• Service Class is set to 0 if call waiting is not active for any of the information classes.</li> <li>• 0xFF,if Not Available</li> </ul>
<i>pFailCause</i>	<ul style="list-style-type: none"> <li>• Supplementary services failure cause (optional)</li> <li>• see <a href="#">qaGobiApiTableVoiceCallEndReasons.h</a> for more information.</li> <li>• 0xFFFF,if Not Available</li> </ul>
<i>pAlphaIDInfo</i>	<ul style="list-style-type: none"> <li>• Pointer to structure of <a href="#">alphaIDInfo</a> (optional) <ul style="list-style-type: none"> <li>– See <a href="#">alphaIDInfo</a> for more information</li> </ul> </li> </ul>

<i>pCCResType</i>	<ul style="list-style-type: none"> <li>• Call Control Result Type (optional) <ul style="list-style-type: none"> <li>– 0x00 - CC_RESULT_TYPE_VOICE - Voice</li> <li>– 0x01 - CC_RESULT_TYPE_SUPS - Supplementary service</li> <li>– 0x02 - CC_RESULT_TYPE_USSD - Unstructured supplementary service</li> <li>– 0xFF - Not Available</li> </ul> </li> </ul>
<i>pCallID</i>	<ul style="list-style-type: none"> <li>• Call ID of the voice call that resulted from call control. (optional)</li> <li>• It is present when pCCResType is present and is Voice.</li> <li>• If zero(0) then invalid.</li> </ul>
<i>pCCSUPSType</i>	<ul style="list-style-type: none"> <li>• Supplementary service data that resulted from call control (optional)</li> <li>• Data is present when pCCResType is present and is other than Voice. <ul style="list-style-type: none"> <li>– See <a href="#">ccSUPSType</a> for more information</li> </ul> </li> </ul>

**Note**

Using NULL for the pointers would make sure that the parameter is not returned or has default value.

**8.546.2 Field Documentation**

8.546.2.1 **alphaIDInfo\*** voiceGetCallWaitInfo::pAlphaIDInfo

8.546.2.2 **BYTE\*** voiceGetCallWaitInfo::pCallID

8.546.2.3 **BYTE\*** voiceGetCallWaitInfo::pCCResType

8.546.2.4 **ccSUPSType\*** voiceGetCallWaitInfo::pCCSUPSType

8.546.2.5 **WORD\*** voiceGetCallWaitInfo::pFailCause

8.546.2.6 **BYTE\*** voiceGetCallWaitInfo::pSvcClass

**8.547 voiceGetCLIPResp Struct Reference****Data Fields**

- [CLIPResp](#) \* [pCLIPResp](#)
- [WORD](#) \* [pFailCause](#)
- [alphaIDInfo](#) \* [pAlphaIDInfo](#)
- [BYTE](#) \* [pCCResType](#)
- [BYTE](#) \* [pCallID](#)
- [ccSUPSType](#) \* [pCCSUPSType](#)

### 8.547.1 Detailed Description

This structure contains Voice Get Calling Line Identification Presentation(CLIP) Response Parameters



## Parameters

<i>pCLIPResp</i>	<ul style="list-style-type: none"> <li>• Pointer to structure of <a href="#">CLIPResp</a> (optional) <ul style="list-style-type: none"> <li>– See <a href="#">CLIPResp</a> for more information</li> </ul> </li> </ul>
<i>pFailCause</i>	<ul style="list-style-type: none"> <li>• Supplementary services failure cause (optional)</li> <li>• see <a href="#">qaGobiApiTableVoiceCallEndReasons.h</a> for more information.</li> <li>• 0xFFFF,if Not Available</li> </ul>
<i>pAlphaIDInfo</i>	<ul style="list-style-type: none"> <li>• Pointer to structure of <a href="#">alphaIDInfo</a> (optional) <ul style="list-style-type: none"> <li>– See <a href="#">alphaIDInfo</a> for more information</li> </ul> </li> </ul>
<i>pCCResType</i>	<ul style="list-style-type: none"> <li>• Call Control Result Type (optional) <ul style="list-style-type: none"> <li>– 0x00 - CC_RESULT_TYPE_VOICE - Voice</li> <li>– 0x01 - CC_RESULT_TYPE_SUPS - Supplementary service</li> <li>– 0x02 - CC_RESULT_TYPE_USSD - Unstructured supplementary service</li> <li>– 0xFF - Not Available</li> </ul> </li> </ul>
<i>pCallID</i>	<ul style="list-style-type: none"> <li>• Call ID of the voice call that resulted from call control. (optional)</li> <li>• It is present when pCCResType is present and is Voice.</li> <li>• If zero(0) then invalid.</li> </ul>
<i>pCCSUPSType</i>	<ul style="list-style-type: none"> <li>• Supplementary service data that resulted from call control (optional)</li> <li>• Data is present when pCCResultType is present and is other than Voice. <ul style="list-style-type: none"> <li>– See <a href="#">ccSUPSType</a> for more information</li> </ul> </li> </ul>

## Note

Using NULL for the pointers would make sure that the parameter is not returned or has default value.

## 8.547.2 Field Documentation

8.547.2.1 [alphaIDInfo\\*](#) [voiceGetCLIPResp::pAlphaIDInfo](#)

8.547.2.2 [BYTE\\*](#) [voiceGetCLIPResp::pCallID](#)

8.547.2.3 [BYTE\\*](#) [voiceGetCLIPResp::pCCResType](#)

8.547.2.4 **ccSUPSType\*** voiceGetCLIPResp::pCCSUPSType

8.547.2.5 **CLIPResp\*** voiceGetCLIPResp::pCLIPResp

8.547.2.6 **WORD\*** voiceGetCLIPResp::pFailCause

## 8.548 voiceGetCLIRResp Struct Reference

### Data Fields

- [CLIRResp](#) \* [pCLIRResp](#)
- [WORD](#) \* [pFailCause](#)
- [alphaIDInfo](#) \* [pAlphaIDInfo](#)
- [BYTE](#) \* [pCCResType](#)
- [BYTE](#) \* [pCallID](#)
- [ccSUPSType](#) \* [pCCSUPSType](#)

### 8.548.1 Detailed Description

This structure contains Voice Get Calling Line Identification Restriction (CLIR) Response Parameters

#### Parameters

<i>pCLIRResp</i>	<ul style="list-style-type: none"> <li>• Pointer to structure of <a href="#">CLIRResp</a> (optional) <ul style="list-style-type: none"> <li>– See <a href="#">CLIRResp</a> for more information</li> </ul> </li> </ul>
<i>pFailCause</i>	<ul style="list-style-type: none"> <li>• Supplementary services failure cause (optional)</li> <li>• see <a href="#">qaGobiApiTableVoiceCallEndReasons.h</a> for more information.</li> <li>• 0xFFFF,if Not Available</li> </ul>
<i>pAlphaIDInfo</i>	<ul style="list-style-type: none"> <li>• Pointer to structure of <a href="#">alphaIDInfo</a> (optional) <ul style="list-style-type: none"> <li>– See <a href="#">alphaIDInfo</a> for more information</li> </ul> </li> </ul>
<i>pCCResType</i>	<ul style="list-style-type: none"> <li>• Call Control Result Type (optional) <ul style="list-style-type: none"> <li>– 0x00 - CC_RESULT_TYPE_VOICE - Voice</li> <li>– 0x01 - CC_RESULT_TYPE_SUPS - Supplementary service</li> <li>– 0x02 - CC_RESULT_TYPE_USSD - Unstructured supplementary service</li> <li>– 0xFF - Not Available</li> </ul> </li> </ul>

<i>pCallID</i>	<ul style="list-style-type: none"> <li>• Call ID of the voice call that resulted from call control. (optional)</li> <li>• It is present when pCCResType is present and is Voice.</li> <li>• If zero(0) then invalid.</li> </ul>
<i>pCCSUPSType</i>	<ul style="list-style-type: none"> <li>• Supplementary service data that resulted from call control (optional)</li> <li>• Data is present when pCCResultType is present and is other than Voice. <ul style="list-style-type: none"> <li>– See <a href="#">ccSUPSType</a> for more information</li> </ul> </li> </ul>

**Note**

Using NULL for the pointers would make sure that the parameter is not returned or has default value.

**8.548.2 Field Documentation**

8.548.2.1 **alphaIDInfo\*** voiceGetCLIRResp::pAlphaIDInfo

8.548.2.2 **BYTE\*** voiceGetCLIRResp::pCallID

8.548.2.3 **BYTE\*** voiceGetCLIRResp::pCCResType

8.548.2.4 **ccSUPSType\*** voiceGetCLIRResp::pCCSUPSType

8.548.2.5 **CLIRResp\*** voiceGetCLIRResp::pCLIRResp

8.548.2.6 **WORD\*** voiceGetCLIRResp::pFailCause

**8.549 voiceGetCNAPResp Struct Reference****Data Fields**

- [CNAPResp](#) \* [pCNAPResp](#)
- [WORD](#) \* [pFailCause](#)
- [alphaIDInfo](#) \* [pAlphaIDInfo](#)
- [BYTE](#) \* [pCCResType](#)
- [BYTE](#) \* [pCallID](#)
- [ccSUPSType](#) \* [pCCSUPSType](#)

**8.549.1 Detailed Description**

This structure contains Voice Get Calling Name Presentation(CNAP) Response Parameters

## Parameters

<i>pCNAPResp</i>	<ul style="list-style-type: none"> <li>• Pointer to structure of <a href="#">CNAPResp</a> (optional) <ul style="list-style-type: none"> <li>– See <a href="#">CNAPResp</a> for more information</li> </ul> </li> </ul>
<i>pFailCause</i>	<ul style="list-style-type: none"> <li>• Supplementary services failure cause (optional)</li> <li>• see <a href="#">qaGobiApiTableVoiceCallEndReasons.h</a> for more information.</li> <li>• 0xFFFF,if Not Available</li> </ul>
<i>pAlphaIDInfo</i>	<ul style="list-style-type: none"> <li>• Pointer to structure of <a href="#">alphaIDInfo</a> (optional) <ul style="list-style-type: none"> <li>– See <a href="#">alphaIDInfo</a> for more information</li> </ul> </li> </ul>
<i>pCCResType</i>	<ul style="list-style-type: none"> <li>• Call Control Result Type (optional) <ul style="list-style-type: none"> <li>– 0x00 - CC_RESULT_TYPE_VOICE - Voice</li> <li>– 0x01 - CC_RESULT_TYPE_SUPS - Supplementary service</li> <li>– 0x02 - CC_RESULT_TYPE_USSD - Unstructured supplementary service</li> <li>– 0xFF - Not Available</li> </ul> </li> </ul>
<i>pCallID</i>	<ul style="list-style-type: none"> <li>• Call ID of the voice call that resulted from call control. (optional)</li> <li>• It is present when pCCResType is present and is Voice.</li> <li>• If zero(0) then invalid.</li> </ul>
<i>pCCSUPSType</i>	<ul style="list-style-type: none"> <li>• Supplementary service data that resulted from call control (optional)</li> <li>• Data is present when pCCResultType is present and is other than Voice. <ul style="list-style-type: none"> <li>– See <a href="#">ccSUPSType</a> for more information</li> </ul> </li> </ul>

## Note

Using NULL for the pointers would make sure that the parameter is not returned or has default value.

## 8.549.2 Field Documentation

8.549.2.1 [alphaIDInfo\\*](#) [voiceGetCNAPResp::pAlphaIDInfo](#)

8.549.2.2 [BYTE\\*](#) [voiceGetCNAPResp::pCallID](#)

8.549.2.3 [BYTE\\*](#) [voiceGetCNAPResp::pCCResType](#)

8.549.2.4 **ccSUPSType\*** voiceGetCNAPResp::pCCSUPSType

8.549.2.5 **CNAPResp\*** voiceGetCNAPResp::pCNAPResp

8.549.2.6 **WORD\*** voiceGetCNAPResp::pFailCause

## 8.550 voiceGetCOLPResp Struct Reference

### Data Fields

- [COLPResp](#) \* [pCOLPResp](#)
- [WORD](#) \* [pFailCause](#)
- [alphaIDInfo](#) \* [pAlphaIDInfo](#)
- [BYTE](#) \* [pCCResType](#)
- [BYTE](#) \* [pCallID](#)
- [ccSUPSType](#) \* [pCCSUPSType](#)

### 8.550.1 Detailed Description

This structure contains Voice Get Connected Line Identification Presentation(COLP) Response Parameters

#### Parameters

<i>pCOLPResp</i>	<ul style="list-style-type: none"> <li>• Pointer to structure of <a href="#">COLPResp</a> (optional) <ul style="list-style-type: none"> <li>– See <a href="#">COLPResp</a> for more information</li> </ul> </li> </ul>
<i>pFailCause</i>	<ul style="list-style-type: none"> <li>• Supplementary services failure cause (optional)</li> <li>• see <a href="#">qaGobiApiTableVoiceCallEndReasons.h</a> for more information.</li> <li>• 0xFFFF,if Not Available</li> </ul>
<i>pAlphaIDInfo</i>	<ul style="list-style-type: none"> <li>• Pointer to structure of <a href="#">alphaIDInfo</a> (optional) <ul style="list-style-type: none"> <li>– See <a href="#">alphaIDInfo</a> for more information</li> </ul> </li> </ul>
<i>pCCResType</i>	<ul style="list-style-type: none"> <li>• Call Control Result Type (optional) <ul style="list-style-type: none"> <li>– 0x00 - CC_RESULT_TYPE_VOICE - Voice</li> <li>– 0x01 - CC_RESULT_TYPE_SUPS - Supplementary service</li> <li>– 0x02 - CC_RESULT_TYPE_USSD - Unstructured supplementary service</li> <li>– 0xFF - Not Available</li> </ul> </li> </ul>

<i>pCallID</i>	<ul style="list-style-type: none"> <li>• Call ID of the voice call that resulted from call control. (optional)</li> <li>• It is present when pCCResType is present and is Voice.</li> <li>• If zero(0) then invalid.</li> </ul>
<i>pCCSUPSType</i>	<ul style="list-style-type: none"> <li>• Supplementary service data that resulted from call control (optional)</li> <li>• Data is present when pCCResultType is present and is other than Voice. <ul style="list-style-type: none"> <li>– See <a href="#">ccSUPSType</a> for more information</li> </ul> </li> </ul>

**Note**

Using NULL for the pointers would make sure that the parameter is not returned or has default value.

**8.550.2 Field Documentation**

8.550.2.1 **alphaIDInfo\*** voiceGetCOLPResp::pAlphaIDInfo

8.550.2.2 **BYTE\*** voiceGetCOLPResp::pCallID

8.550.2.3 **BYTE\*** voiceGetCOLPResp::pCCResType

8.550.2.4 **ccSUPSType\*** voiceGetCOLPResp::pCCSUPSType

8.550.2.5 **COLPResp\*** voiceGetCOLPResp::pCOLPResp

8.550.2.6 **WORD\*** voiceGetCOLPResp::pFailCause

**8.551 voiceGetCOLRResp Struct Reference****Data Fields**

- [COLRResp](#) \* [pCOLRResp](#)
- [WORD](#) \* [pFailCause](#)
- [alphaIDInfo](#) \* [pAlphaIDInfo](#)
- [BYTE](#) \* [pCCResType](#)
- [BYTE](#) \* [pCallID](#)
- [ccSUPSType](#) \* [pCCSUPSType](#)

**8.551.1 Detailed Description**

This structure contains Voice Get Connected Line Identification Restriction(COLR) Response Parameters

## Parameters

<i>pCOLRResp</i>	<ul style="list-style-type: none"> <li>• Pointer to structure of <a href="#">COLRResp</a> (optional) <ul style="list-style-type: none"> <li>– See <a href="#">COLRResp</a> for more information</li> </ul> </li> </ul>
<i>pFailCause</i>	<ul style="list-style-type: none"> <li>• Supplementary services failure cause (optional)</li> <li>• see <a href="#">qaGobiApiTableVoiceCallEndReasons.h</a> for more information.</li> <li>• 0xFFFF,if Not Available</li> </ul>
<i>pAlphaIDInfo</i>	<ul style="list-style-type: none"> <li>• Pointer to structure of <a href="#">alphaIDInfo</a> (optional) <ul style="list-style-type: none"> <li>– See <a href="#">alphaIDInfo</a> for more information</li> </ul> </li> </ul>
<i>pCCResType</i>	<ul style="list-style-type: none"> <li>• Call Control Result Type (optional) <ul style="list-style-type: none"> <li>– 0x00 - CC_RESULT_TYPE_VOICE - Voice</li> <li>– 0x01 - CC_RESULT_TYPE_SUPS - Supplementary service</li> <li>– 0x02 - CC_RESULT_TYPE_USSD - Unstructured supplementary service</li> <li>– 0xFF - Not Available</li> </ul> </li> </ul>
<i>pCallID</i>	<ul style="list-style-type: none"> <li>• Call ID of the voice call that resulted from call control. (optional)</li> <li>• It is present when pCCResType is present and is Voice.</li> <li>• If zero(0) then invalid.</li> </ul>
<i>pCCSUPSType</i>	<ul style="list-style-type: none"> <li>• Supplementary service data that resulted from call control (optional)</li> <li>• Data is present when pCCResultType is present and is other than Voice. <ul style="list-style-type: none"> <li>– See <a href="#">ccSUPSType</a> for more information</li> </ul> </li> </ul>

## Note

Using NULL for the pointers would make sure that the parameter is not returned or has default value.

## 8.551.2 Field Documentation

8.551.2.1 [alphaIDInfo\\*](#) [voiceGetCOLRResp::pAlphaIDInfo](#)

8.551.2.2 [BYTE\\*](#) [voiceGetCOLRResp::pCallID](#)

8.551.2.3 [BYTE\\*](#) [voiceGetCOLRResp::pCCResType](#)

8.551.2.4 **ccSUPSType\*** voiceGetCOLRResp::pCCSUPSType

8.551.2.5 **COLRResp\*** voiceGetCOLRResp::pCOLRResp

8.551.2.6 **WORD\*** voiceGetCOLRResp::pFailCause

## 8.552 voiceGetConfigReq Struct Reference

### Data Fields

- [BYTE \\*](#) [pAutoAnswer](#)
- [BYTE \\*](#) [pAirTimer](#)
- [BYTE \\*](#) [pRoamTimer](#)
- [BYTE \\*](#) [pTTYMode](#)
- [BYTE \\*](#) [pPrefVoiceSO](#)
- [BYTE \\*](#) [pAMRStatus](#)
- [BYTE \\*](#) [pPrefVoicePrivacy](#)
- [BYTE \\*](#) [pNamID](#)
- [BYTE \\*](#) [pVoiceDomainPref](#)

### 8.552.1 Detailed Description

This structure contains Voice Get Configuration Request Parameters

#### Parameters

<i>pAuto-Answer(optional)</i>	<ul style="list-style-type: none"> <li>• Indicator to retrieve the Auto Answer Information. <ul style="list-style-type: none"> <li>– 0x01 - Include auto answer information</li> </ul> </li> </ul>
<i>pAir-Timer(optional)</i>	<ul style="list-style-type: none"> <li>• Indicator to retrieve the Air Timer Information. <ul style="list-style-type: none"> <li>– 0x01 - Include air calls timer count information</li> </ul> </li> <li>• Currently Not Supported.</li> </ul>
<i>pRoam-Timer(optional)</i>	<ul style="list-style-type: none"> <li>• Indicator to retrieve the Roam Timer Information. <ul style="list-style-type: none"> <li>– 0x01 - Include roam calls timer information</li> </ul> </li> <li>• Currently Not Supported.</li> </ul>
<i>pTTY-Mode(optional)</i>	<ul style="list-style-type: none"> <li>• Indicator to retrieve the TTY Mode Information. <ul style="list-style-type: none"> <li>– 0x01 - Include TTY configuration status information</li> </ul> </li> </ul>



<i>pPrefVoiceSO(optional)</i>	<ul style="list-style-type: none"> <li>Indicator to retrieve the Preferred Voice SO Information. <ul style="list-style-type: none"> <li>0x01 - Include preferred voice configuration status information</li> </ul> </li> <li>Currently Not Supported.</li> </ul>
<i>pAMR-Status(optional)</i>	<ul style="list-style-type: none"> <li>Indicator to retrieve the AMR Status Information. <ul style="list-style-type: none"> <li>0x01 - Include AMR status information</li> </ul> </li> </ul>
<i>pPrefVoice-Privacy(optional)</i>	<ul style="list-style-type: none"> <li>Indicator to retrieve the Preferred Voice Privacy Information. <ul style="list-style-type: none"> <li>0x01 - Include preferred voice privacy status information</li> </ul> </li> </ul>
<i>pNamID(optional)</i>	<ul style="list-style-type: none"> <li>Index of the Number Assignment Module Index (CDMA subscription) to be configured</li> <li>Range: 0 to 3.</li> <li>Some modems support only 1 or 2 NAMs.</li> <li>The NAM Index is valid only when the request contains at least one of Air Timer, Roam Timer, and Preferred Voice SO.</li> <li>If no nam_id value is specified in the request, the default value is 0.</li> </ul>
<i>pVoiceDomain-Pref(optional)</i>	<ul style="list-style-type: none"> <li>Indicator to retrieve the Preferred Voice <a href="#">Domain</a> Information. <ul style="list-style-type: none"> <li>0x01 - Include voice domain preference information</li> </ul> </li> </ul>

**Note**

Using NULL for the pointers would make sure that the parameter is not returned.

**8.552.2 Field Documentation**

**8.552.2.1** **BYTE\*** voiceGetConfigReq::pAirTimer

**8.552.2.2** **BYTE\*** voiceGetConfigReq::pAMRStatus

**8.552.2.3** **BYTE\*** voiceGetConfigReq::pAutoAnswer

**8.552.2.4** **BYTE\*** voiceGetConfigReq::pNamID

**8.552.2.5** **BYTE\*** voiceGetConfigReq::pPrefVoicePrivacy

**8.552.2.6** **BYTE\*** voiceGetConfigReq::pPrefVoiceSO

**8.552.2.7** **BYTE\*** voiceGetConfigReq::pRoamTimer

8.552.2.8 **BYTE\*** `voiceGetConfigReq::pTTYMode`

8.552.2.9 **BYTE\*** `voiceGetConfigReq::pVoiceDomainPref`

## 8.553 `voiceGetConfigResp` Struct Reference

### Data Fields

- **BYTE \*** `pAutoAnswerStat`
- **airTimer \*** `pAirTimerCnt`
- **roamTimer \*** `pRoamTimerCnt`
- **BYTE \*** `pCurrTTYMode`
- **prefVoiceSO \*** `pCurPrefVoiceSO`
- **curAMRConfig \*** `pCurAMRConfig`
- **BYTE \*** `pCurVoicePrivacyPref`
- **BYTE \*** `pCurVoiceDomainPref`

### 8.553.1 Detailed Description

This structure contains Voice Get Configuration Response Parameters.

#### Parameters

<i>pAutoAnswer-Stat(optional)</i>	<ul style="list-style-type: none"> <li>• Auto Answer Status</li> <li>• Value returned is read from NV_AUTO_ANSWER_I. <ul style="list-style-type: none"> <li>– 0x00 - Disabled</li> <li>– 0x01 - Enabled</li> <li>– 0xFF - Not Available</li> </ul> </li> </ul>
<i>pAirTimer-Cnt(optional)</i>	<ul style="list-style-type: none"> <li>• Air Timer Count</li> <li>• Value returned is read from NV_AIR_CNT_I.</li> <li>• See <a href="#">airTimer</a> for more information</li> </ul>
<i>pRoamTimer-Cnt(optional)</i>	<ul style="list-style-type: none"> <li>• Roam Timer Count</li> <li>• Value returned is read from NV_ROAM_CNT_I.</li> <li>• See <a href="#">roamTimer</a> for more information</li> </ul>

<i>pCurrTTY-Mode(optional)</i>	<ul style="list-style-type: none"> <li>• Current TTY Mode</li> <li>• Value returned is read from NV_TTY_I. <ul style="list-style-type: none"> <li>– 0x00 - TTY_MODE_FULL - Full</li> <li>– 0x01 - TTY_MODE_VCO - Voice carry over</li> <li>– 0x02 - TTY_MODE_HCO - Hearing carry over</li> <li>– 0x03 - TTY_MODE_OFF - Off</li> <li>– 0xFF - Not Available</li> </ul> </li> </ul>
<i>pCurPrefVoiceSO(optional)</i>	<ul style="list-style-type: none"> <li>• Current Preferred Voice SO</li> <li>• Value returned is read from NV_PREF_VOICE_SO_I.</li> <li>• See <a href="#">prefVoiceSO</a> for more information</li> </ul>
<i>pCurAMR-Config(optional)</i>	<ul style="list-style-type: none"> <li>• Current Adaptive Multi-Rate Configuration.</li> <li>• Values returned are read from NV_GSM_ARM_CALL_CONFIG_I and NV_UMTS_A-MR_CODEEC_PREFERENCE_CONFIG_I.</li> <li>• See <a href="#">curAMRConfig</a> for more information</li> </ul>
<i>pCurVoice-Privacy-Pref(optional)</i>	<ul style="list-style-type: none"> <li>• Current Voice Privacy Preference</li> <li>• Value returned is read from NV_VOICE_PRIV_I. <ul style="list-style-type: none"> <li>– 0x00 - Standard privacy</li> <li>– 0x01 - Enhanced privacy</li> <li>– 0xFF - Not Available</li> </ul> </li> </ul>
<i>pCurVoice-Domain-Pref(optional)</i>	<ul style="list-style-type: none"> <li>• Current Voice <a href="#">Domain</a> Preference. <ul style="list-style-type: none"> <li>– 0x00 - Circuit-switched (CS) only</li> <li>– 0x01 - Packet-switched (PS) only</li> <li>– 0x02 - CS is preferred; PS is secondary</li> <li>– 0x03 - PS is preferred; CS is secondary</li> <li>– 0xFF - Not Available</li> </ul> </li> </ul>

**Note**

Using NULL for the pointers would make sure that the parameter is not returned or has default value.

**8.553.2 Field Documentation****8.553.2.1 airTimer\* voiceGetConfigResp::pAirTimerCnt**

- 8.553.2.2 **BYTE\*** voiceGetConfigResp::pAutoAnswerStat
- 8.553.2.3 **curAMRConfig\*** voiceGetConfigResp::pCurAMRConfig
- 8.553.2.4 **prefVoiceSO\*** voiceGetConfigResp::pCurPrefVoiceSO
- 8.553.2.5 **BYTE\*** voiceGetConfigResp::pCurrTTYMode
- 8.553.2.6 **BYTE\*** voiceGetConfigResp::pCurVoiceDomainPref
- 8.553.2.7 **BYTE\*** voiceGetConfigResp::pCurVoicePrivacyPref
- 8.553.2.8 **roamTimer\*** voiceGetConfigResp::pRoamTimerCnt

## 8.554 voicelndicationRegisterInfo Struct Reference

### Data Fields

- **BYTE \*** [pRegDTMFEvents](#)
- **BYTE \*** [pRegVoicePrivacyEvents](#)
- **BYTE \*** [pSuppsNotifEvents](#)

### 8.554.1 Detailed Description

This structure contains parameters of Indication Register Information

#### Parameters

<i>pRegDTMF-Events(optional)</i>	<ul style="list-style-type: none"> <li>• Registration Indication For DTMF Events.</li> <li>• When this registration is enabled, the device learns of DTMF events via the QMI_VOICE_DTMF_IND indication. <ul style="list-style-type: none"> <li>– 0x00 - Disable</li> <li>– 0x01 - Enable</li> </ul> </li> </ul>
<i>pRegVoice-Privacy-Events(optional)</i>	<ul style="list-style-type: none"> <li>• Registration Indication For Voice Privacy Events.</li> <li>• When this registration is enabled, the device learns of DTMF events via the QMI_VOICE_PRIVACY_IND indication. <ul style="list-style-type: none"> <li>– 0x00 - Disable</li> <li>– 0x01 - Enable</li> </ul> </li> </ul>

<i>pSuppsNotif-Events(optional)</i>	<ul style="list-style-type: none"> <li>• Registration Indication For Supplementary Service Notification Events.</li> <li>• When this registration is enabled, the device learns of DTMF events via the QMI_VOICE_SUPS_NOTIFICATION_IND indication. <ul style="list-style-type: none"> <li>– 0x00 - Disable</li> <li>– 0x01 - Enable</li> </ul> </li> </ul>
-------------------------------------	--

**Note**

One of the optional parameter is mandatory to be present in the request.

**8.554.2 Field Documentation**

8.554.2.1 **BYTE\*** voiceIndicationRegisterInfo::pRegDTMFEvents

8.554.2.2 **BYTE\*** voiceIndicationRegisterInfo::pRegVoicePrivacyEvents

8.554.2.3 **BYTE\*** voiceIndicationRegisterInfo::pSuppsNotifEvents

**8.555 voiceInfoRec Struct Reference****Data Fields**

- [BYTE](#) callID
- [signalInfo](#) \* [pSignalInfo](#)
- [callerIDInfo](#) \* [pCallerIDInfo](#)
- [BYTE](#) \* [pDispInfo](#)
- [BYTE](#) \* [pExtDispInfo](#)
- [BYTE](#) \* [pCallerNameInfo](#)
- [BYTE](#) \* [pCallWaitInd](#)
- [connectNumInfo](#) \* [pConnectNumInfo](#)
- [connectNumInfo](#) \* [pCallingPartyInfo](#)
- [calledPartyInfo](#) \* [pCalledPartyInfo](#)
- [redirNumInfo](#) \* [pRedirNumInfo](#)
- [BYTE](#) \* [pCLIRCause](#)
- [NSSAudioCtrl](#) \* [pNSSAudioCtrl](#)
- [BYTE](#) \* [pNSSRelease](#)
- [lineCtrlInfo](#) \* [pLineCtrlInfo](#)
- [extDispRecInfo](#) \* [pExtDispRecInfo](#)

**8.555.1 Detailed Description**

This structure contains Voice record Information

## Parameters

<i>callID</i>	[Mandatory] <ul style="list-style-type: none"> <li>• Call identifier for the call.</li> </ul>
<i>pSignalInfo</i> [-Optional]	<ul style="list-style-type: none"> <li>• Signal Information</li> <li>• See <a href="#">signalInfo</a> for more information</li> </ul>
<i>pCallerIDInfo</i> [-Optional]	<ul style="list-style-type: none"> <li>• Caller ID Information</li> <li>• See <a href="#">callerIDInfo</a> for more information</li> </ul>
<i>pDispInfo</i> [-Optional]	<ul style="list-style-type: none"> <li>• Display Information</li> </ul>
<i>pExtDispInfo</i> [-Optional]	<ul style="list-style-type: none"> <li>• Extended Display Information</li> </ul>
<i>pCallerNameInfo</i> [-Optional]	<ul style="list-style-type: none"> <li>• Caller Name Information</li> </ul>
<i>pCallWaitInd</i> [-Optional]	<ul style="list-style-type: none"> <li>• Call Waiting Indicator</li> </ul>
<i>pConnectNumInfo</i> [-Optional]	<ul style="list-style-type: none"> <li>• Connected Number Information</li> <li>• see <a href="#">connectNumInfo</a> for more information</li> </ul>
<i>pCallingPartyInfo</i> [-Optional]	<ul style="list-style-type: none"> <li>• Calling Party Number Information</li> <li>• This structure is having exactly same elements as <a href="#">connectNumInfo</a></li> <li>• see <a href="#">connectNumInfo</a> for more information</li> </ul>
<i>pCalledPartyInfo</i> [-Optional]	<ul style="list-style-type: none"> <li>• Called Party Number Information</li> <li>• see <a href="#">calledPartyInfo</a> for more information</li> </ul>
<i>pRedirNumInfo</i> [-Optional]	<ul style="list-style-type: none"> <li>• Redirecting Number Information</li> <li>• see <a href="#">redirNumInfo</a> for more information</li> </ul>
<i>pCLIRCause</i> [-Optional]	<ul style="list-style-type: none"> <li>• National Supplementary Services - CLIR</li> <li>• see <a href="#">NSSAudioCtrl</a> for more information</li> </ul>

<i>pNSSAudioCtrl</i> [- Optional]	<ul style="list-style-type: none"> <li>National Supplementary Services - Audio Control</li> </ul>
<i>pNSSRelease</i> [- Optional]	<ul style="list-style-type: none"> <li>National Supplementary Services - Release</li> </ul>
<i>pLineCtrlInfo</i> [- Optional]	<ul style="list-style-type: none"> <li>Line Control Information</li> <li>see <a href="#">lineCtrlInfo</a> for more information</li> </ul>
<i>pExtDispRecInfo</i> [Optional]	<ul style="list-style-type: none"> <li>Extended Display Record Information</li> <li>see <a href="#">extDispRecInfo</a> for more information</li> </ul>

## 8.555.2 Field Documentation

- 8.555.2.1 **BYTE** voiceInfoRec::callID
- 8.555.2.2 **calledPartyInfo\*** voiceInfoRec::pCalledPartyInfo
- 8.555.2.3 **callerIDInfo\*** voiceInfoRec::pCallerIDInfo
- 8.555.2.4 **BYTE\*** voiceInfoRec::pCallerNameInfo
- 8.555.2.5 **connectNumInfo\*** voiceInfoRec::pCallingPartyInfo
- 8.555.2.6 **BYTE\*** voiceInfoRec::pCallWaitInd
- 8.555.2.7 **BYTE\*** voiceInfoRec::pCLIRCause
- 8.555.2.8 **connectNumInfo\*** voiceInfoRec::pConnectNumInfo
- 8.555.2.9 **BYTE\*** voiceInfoRec::pDispInfo
- 8.555.2.10 **BYTE\*** voiceInfoRec::pExtDispInfo
- 8.555.2.11 **extDispRecInfo\*** voiceInfoRec::pExtDispRecInfo
- 8.555.2.12 **lineCtrlInfo\*** voiceInfoRec::pLineCtrlInfo
- 8.555.2.13 **NSSAudioCtrl\*** voiceInfoRec::pNSSAudioCtrl
- 8.555.2.14 **BYTE\*** voiceInfoRec::pNSSRelease
- 8.555.2.15 **redirNumInfo\*** voiceInfoRec::pRedirNumInfo
- 8.555.2.16 **signalInfo\*** voiceInfoRec::pSignalInfo

## 8.556 voiceManageCallsReq Struct Reference

## Data Fields

- [BYTE SUPSType](#)
- [BYTE \\* pCallID](#)

### 8.556.1 Detailed Description

This structure contains Manage Calls Information.

#### Parameters

<i>SUPSType</i>	<ul style="list-style-type: none"> <li>• Supplementary service type during the call.             <ul style="list-style-type: none"> <li>– 0x01 - SUPS_TYPE_RELEASE_HELD_OR_WAITING                 <ul style="list-style-type: none"> <li>* Release is held or waiting</li> </ul> </li> <li>– 0x02 - SUPS_TYPE_RELEASE_ACTIVE_ACCEPT_HELD_OR_WAITING                 <ul style="list-style-type: none"> <li>* Release is active and accepting held or waiting</li> </ul> </li> <li>– 0x03 - SUPS_TYPE_HOLD_ACTIVE_ACCEPT_WAITING_OR_HELD                 <ul style="list-style-type: none"> <li>* Hold is active and accepting waiting or held</li> </ul> </li> <li>– 0x04 - SUPS_TYPE_HOLD_ALL_EXCEPT_SPECIFIED_CALL                 <ul style="list-style-type: none"> <li>* Hold all calls except a specified one</li> </ul> </li> <li>– 0x05 - SUPS_TYPE_MAKE_CONFERENCE_CALL                 <ul style="list-style-type: none"> <li>* Make a conference call</li> </ul> </li> <li>– 0x06 - SUPS_TYPE_EXPLICIT_CALL_TRANSFER                 <ul style="list-style-type: none"> <li>* Explicit call transfer</li> </ul> </li> <li>– 0x07 - SUPS_TYPE_CCBS_ACTIVATION                 <ul style="list-style-type: none"> <li>* Activate completion of calls to busy subscriber</li> </ul> </li> <li>– 0x08 - SUPS_TYPE_END_ALL_CALLS                 <ul style="list-style-type: none"> <li>* End all calls</li> </ul> </li> <li>– 0x09 - SUPS_TYPE_RELEASE_SPECIFIED_CALL                 <ul style="list-style-type: none"> <li>* Release a specified call</li> </ul> </li> </ul> </li> </ul>
-----------------	--



<i>pCallID[Optional]</i>	<ul style="list-style-type: none"><li>• Applicable only for SUPSType 0x04, 0x07, and 0x09</li></ul>
--------------------------	---

## 8.556.2 Field Documentation

8.556.2.1 **BYTE\*** voiceManageCallsReq::pCallID

8.556.2.2 **BYTE** voiceManageCallsReq::SUPSType

## 8.557 voiceManageCallsResp Struct Reference

### Data Fields

- **WORD\*** pFailCause

### 8.557.1 Detailed Description

This structure contains Failure cause Information. Populated when API Fails.

#### Parameters

<i>pFailCause</i>	<ul style="list-style-type: none"><li>• Supplementary service failure causes (optional, supply NULL if not required).</li><li>• See Table8 <a href="#">qaGobiApiTableVoiceCallEndReasons.h</a> for supplementary services failure cause<ul style="list-style-type: none"><li>– 0xFFFF is the value when the information is not received from device</li></ul></li></ul>
-------------------	---

## 8.557.2 Field Documentation

8.557.2.1 **WORD\*** voiceManageCallsResp::pFailCause

## 8.558 voiceOrigUSSDNoWaitInfo Struct Reference

### Data Fields

- struct [USSInfo](#) USSInformation

### 8.558.1 Detailed Description

This structure contains Orig USSD No Wait Information Parameters.

#### Parameters

<i>USSInformation</i>	<ul style="list-style-type: none"><li>• See <a href="#">USSInfo</a> for more information.</li></ul>
-----------------------	---

## 8.558.2 Field Documentation

8.558.2.1 struct USSInfo voiceOrigUSSDNoWaitInfo::USSInformation

## 8.559 voiceOTASPStatusInfo Struct Reference

### Data Fields

- [BYTE callID](#)
- [BYTE OTASPStatus](#)

### 8.559.1 Detailed Description

This structure consist of OTASP or OTAPA event params

#### Parameters

<i>callID</i>	<ul style="list-style-type: none"> <li>• Call identifier for the call.</li> </ul>
<i>OTASPStatus</i>	<ul style="list-style-type: none"> <li>• OTASP status for the OTASP call. Values:             <ul style="list-style-type: none"> <li>– 0x00 - OTASP_STATUS_SPL_UNLOCKED.SPL unlocked; only for user-initiated OTASP</li> <li>– 0x01 - OTASP_STATUS_SPRC_RETRIES_EXCEEDED. SPC retries exceeded; only for user-initiated OTASP</li> <li>– 0x02 - OTASP_STATUS_AKEY_EXCHANGED.A-key exchanged; only for user-initiated OTASP</li> <li>– 0x03 - OTASP_STATUS_SSD_UPDATED. SSD updated; for both user-initiated OTASP and network-initiated OTASP (OTAPA)</li> <li>– 0x04 - OTASP_STATUS_NAM_DOWNLOADED - NAM downloaded; only for user-initiated OTASP</li> <li>– 0x05 - OTASP_STATUS_MDN_DOWNLOADED - MDN downloaded; only for user-initiated OTASP</li> <li>– 0x06 - OTASP_STATUS_IMSI_DOWNLOADED - IMSI downloaded; only for user-initiated OTASP</li> <li>– 0x07 - OTASP_STATUS_PRL_DOWNLOADED - PRL downloaded; only for user-initiated OTASP</li> <li>– 0x08 - OTASP_STATUS_COMMITTED - Commit successful; only for user-initiated OTASP</li> <li>– 0x09 - OTASP_STATUS_OTAPA_STARTED - OTAPA started; only for network-initiated OTASP(OTAPA)</li> <li>– 0x0A - OTASP_STATUS_OTAPA_STOPPED - OTAPA stopped; only for network-initiated OTASP(OTAPA)</li> <li>– 0x0B - OTASP_STATUS_OTAPA_ABORTED - OTAPA aborted; only for network-initiated OTASP(OTAPA)</li> <li>– 0x0C - OTASP_STATUS_OTAPA_COMMITTED - OTAPA committed; only for network-initiated OTASP(OTAPA)</li> </ul> </li> </ul>

## 8.559.2 Field Documentation

8.559.2.1 **BYTE** voiceOTASPStatusInfo::callID

8.559.2.2 **BYTE** voiceOTASPStatusInfo::OTASPStatus

## 8.560 voicePrivacyInfo Struct Reference

### Data Fields

- [BYTE callID](#)
- [BYTE voicePrivacy](#)

### 8.560.1 Detailed Description

Contains the parameters passed for SLQSVoiceSetPrivacyChangeCallBack by the device.

#### Parameters

<i>callID</i>	<ul style="list-style-type: none"><li>• Unique identifier of the call for which the voice privacy is applicable. (mandatory)</li></ul>
<i>voicePrivacy</i>	<ul style="list-style-type: none"><li>• Voice Privacy (mandatory)<ul style="list-style-type: none"><li>– 0x00 - VOICE_PRIVACY_STANDARD - Standard privacy</li><li>– 0x01 - VOICE_PRIVACY_ENHANCED - Enhanced privacy</li></ul></li></ul>

#### Note

None

## 8.560.2 Field Documentation

8.560.2.1 **BYTE** voicePrivacyInfo::callID

8.560.2.2 **BYTE** voicePrivacyInfo::voicePrivacy

## 8.561 voiceSetAllCallStatusCbkInfo Struct Reference

### Data Fields

- [arrCallInfo arrCallInfomation](#)
- [arrRemotePartyNum](#) \* [pArrRemotePartyNum](#)
- [arrRemotePartyName](#) \* [pArrRemotePartyName](#)
- [arrAlertingType](#) \* [pArrAlertingType](#)
- [arrSvcOption](#) \* [pArrSvcOption](#)
- [arrCallEndReason](#) \* [pArrCallEndReason](#)
- [arrAlphaID](#) \* [pArrAlphaID](#)
- [arrConnectPartyNum](#) \* [pArrConnectPartyNum](#)
- [arrDiagInfo](#) \* [pArrDiagInfo](#)
- [arrCalledPartyNum](#) \* [pArrCalledPartyNum](#)

- [arrRedirPartyNum](#) \* [pArrRedirPartyNum](#)
- [arrAlertingPattern](#) \* [pArrAlertingPattern](#)

### 8.561.1 Detailed Description

This structure contains VoiceCall Information parameters. [arrCallInfomation](#) will be populated in case of change in the call information. Other paramters are optional therefore are populated based on device and technology type being used.

#### Parameters

<i>arrCallInfomation</i>	[mandatory] <ul style="list-style-type: none"> <li>• Array of Call Information This must be populated if Indication is received See <a href="#">arrCallInfo</a> for more information.               <ul style="list-style-type: none"> <li>– Applicable for both "3GPP/3GPP2"</li> </ul> </li> </ul>
<i>pArrRemote-PartyNum</i>	[optional] <ul style="list-style-type: none"> <li>• Array of Remote Party Name.( NULL when not present) See <a href="#">arrRemotePartyNum</a> for more information.               <ul style="list-style-type: none"> <li>– Applicable only for "3GPP/3GPP2"</li> </ul> </li> </ul>
<i>pArrRemote-PartyName</i>	[optional] <ul style="list-style-type: none"> <li>• Array of Alerting Type.( NULL when not present) See <a href="#">arrRemotePartyName</a> for more information.               <ul style="list-style-type: none"> <li>– Applicable only for "3GPP"</li> </ul> </li> </ul>
<i>pArrAlertingType</i>	[optional] <ul style="list-style-type: none"> <li>• Array of Alerting Type( NULL when not present) See <a href="#">arrAlertingType</a> for more information.               <ul style="list-style-type: none"> <li>– Applicable only for "3GPP"</li> </ul> </li> </ul>
<i>pArrSvcOption</i>	[optional] <ul style="list-style-type: none"> <li>• Array of Service Option.(NULL when not present) See <a href="#">arrSvcOption</a> for more information.               <ul style="list-style-type: none"> <li>– Applicable only for "3GPP"</li> </ul> </li> </ul>
<i>pArrCallEnd-Reason</i>	[optional] <ul style="list-style-type: none"> <li>• Array of Call End Reason.( NULL when not present) See <a href="#">arrCallEndReason</a> for more information.               <ul style="list-style-type: none"> <li>– Applicable only for "3GPP"</li> </ul> </li> </ul>

<i>pArrAlphaID</i>	[optional] <ul style="list-style-type: none"> <li>• Array of Alpha Identifier( NULL when not present) See <a href="#">arrAlphaID</a> for more information.</li> <li>– Applicable only for "3GPP"</li> </ul>
<i>pArrConnect-PartyNum</i>	[optional] <ul style="list-style-type: none"> <li>• Array of Connected Party Number.( NULL when not present) See <a href="#">arrConnectPartyNum</a> for more information.</li> <li>– Applicable for both "3GPP/3GPP2"</li> </ul>
<i>pArrDiagInfo</i>	[optional] <ul style="list-style-type: none"> <li>• Array of Diagnostic Information.( NULL when not present) See <a href="#">arrDiagInfo</a> for more information.</li> <li>– Applicable only for "3GPP"</li> </ul>
<i>pArrCalledParty-Num</i>	[optional] <ul style="list-style-type: none"> <li>• Array of Called Party Number.( NULL when not present) See <a href="#">arrCalledPartyNum</a> for more information.</li> <li>– Applicable only for "3GPP"</li> </ul>
<i>pArrRedirParty-Num</i>	[optional] <ul style="list-style-type: none"> <li>• Array of Redirecting Party Number.( NULL when not present) See <a href="#">arrRedirPartyNum</a> for more information.</li> <li>– Applicable only for "3GPP"</li> </ul>
<i>pArrAlerting-Pattern</i>	[optional] <ul style="list-style-type: none"> <li>• Array of Alerting Pattern.( NULL when not present) See <a href="#">arrAlertingPattern</a> for more information.</li> <li>– Applicable only for "3GPP"</li> </ul>

**Note**

Optional paramters would be NULL, if not received from the device.

**8.561.2 Field Documentation**

8.561.2.1 **arrCallInfo** voiceSetAllCallStatusCbkInfo::arrCallInfomation

8.561.2.2 **arrAlertingPattern\*** voiceSetAllCallStatusCbkInfo::pArrAlertingPattern

8.561.2.3 **arrAlertingType\*** voiceSetAllCallStatusCbkInfo::pArrAlertingType

8.561.2.4 **arrAlphaID\*** voiceSetAllCallStatusCbkInfo::pArrAlphaID

8.561.2.5 **arrCalledPartyNum\*** voiceSetAllCallStatusCbkInfo::pArrCalledPartyNum

8.561.2.6 **arrCallEndReason\*** voiceSetAllCallStatusCbklInfo::pArrCallEndReason

8.561.2.7 **arrConnectPartyNum\*** voiceSetAllCallStatusCbklInfo::pArrConnectPartyNum

8.561.2.8 **arrDiagInfo\*** voiceSetAllCallStatusCbklInfo::pArrDiagInfo

8.561.2.9 **arrRedirPartyNum\*** voiceSetAllCallStatusCbklInfo::pArrRedirPartyNum

8.561.2.10 **arrRemotePartyName\*** voiceSetAllCallStatusCbklInfo::pArrRemotePartyName

8.561.2.11 **arrRemotePartyNum\*** voiceSetAllCallStatusCbklInfo::pArrRemotePartyNum

8.561.2.12 **arrSvcOption\*** voiceSetAllCallStatusCbklInfo::pArrSvcOption

## 8.562 voiceSetCallBarringPwdInfo Struct Reference

### Data Fields

- [BYTE Reason](#)
- [BYTE oldPasswd](#) [4]
- [BYTE newPasswd](#) [4]
- [BYTE newPasswdAgain](#) [4]

### 8.562.1 Detailed Description

This structure contains Voice Set Call Barring Password Request Parameters

#### Parameters

<i>Reason</i>	<ul style="list-style-type: none"> <li>• Call Barring Reason</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0x07 - QMI_VOICE_REASON_BARR_ALLOUTGOING - All outgoing</li> <li>– 0x08 - QMI_VOICE_REASON_BARR_OUTGOINGINT - Outgoing internal</li> <li>– 0x09 - QMI_VOICE_REASON_BARR_OUTGOINGINTEXTOHOME - Outgoing external to home</li> <li>– 0x0A - QMI_VOICE_REASON_BARR_ALLINCOMING - All incoming</li> <li>– 0x0B - QMI_VOICE_REASON_BARR_INCOMINGROAMING - Roaming incoming</li> <li>– 0x0C - QMI_VOICE_REASON_BARR_ALLBARRING - All calls are barred</li> <li>– 0x0D - QMI_VOICE_REASON_BARR_ALLOUTGOINGBARRING - All outgoing calls are barred</li> <li>– 0x0E - QMI_VOICE_REASON_BARR_ALLINCOMINGBARRING - All incoming calls are barred</li> </ul> </li> </ul>
---------------	---

<i>oldPasswd</i> [PASSWORD_LENGTH]	<ul style="list-style-type: none"> <li>• Old password. <ul style="list-style-type: none"> <li>– Password consists of 4 ASCII digits.</li> <li>– Range: 0000 to 9999.</li> </ul> </li> </ul>
<i>newPasswd</i> [PASSWORD_LENGTH]	<ul style="list-style-type: none"> <li>• New password. <ul style="list-style-type: none"> <li>– Password consists of 4 ASCII digits.</li> <li>– Range: 0000 to 9999.</li> </ul> </li> </ul>
<i>newPasswdAgain</i> [PASSWORD_LENGTH]	<ul style="list-style-type: none"> <li>• New password Again. <ul style="list-style-type: none"> <li>– Password consists of 4 ASCII digits.</li> <li>– Range: 0000 to 9999.</li> </ul> </li> </ul>

## 8.562.2 Field Documentation

8.562.2.1 BYTE voiceSetCallBarringPwdInfo::newPasswd[4]

8.562.2.2 BYTE voiceSetCallBarringPwdInfo::newPasswdAgain[4]

8.562.2.3 BYTE voiceSetCallBarringPwdInfo::oldPasswd[4]

8.562.2.4 BYTE voiceSetCallBarringPwdInfo::Reason

## 8.563 voiceSetCallBarringPwdResp Struct Reference

### Data Fields

- WORD \* pFailCause
- alphaIDInfo \* pAlphaIDInfo
- BYTE \* pCCResType
- BYTE \* pCallID
- ccSUPSType \* pCCSUPSType

### 8.563.1 Detailed Description

This structure contains Voice Set Call Barring Password Response Parameters

## Parameters

<i>pFailCause</i>	<ul style="list-style-type: none"> <li>• Supplementary services failure cause (optional)</li> <li>• see <a href="#">qaGobiApiTableVoiceCallEndReasons.h</a> for more information.</li> <li>• 0xFFFF,if Not Available</li> </ul>
<i>pAlphaIDInfo</i>	<ul style="list-style-type: none"> <li>• Pointer to structure of <a href="#">alphaIDInfo</a> (optional) <ul style="list-style-type: none"> <li>– See <a href="#">alphaIDInfo</a> for more information</li> </ul> </li> </ul>
<i>pCCResType</i>	<ul style="list-style-type: none"> <li>• Call Control Result Type (optional) <ul style="list-style-type: none"> <li>– 0x00 - CC_RESULT_TYPE_VOICE - Voice</li> <li>– 0x01 - CC_RESULT_TYPE_SUPS - Supplementary service</li> <li>– 0x02 - CC_RESULT_TYPE USSD - Unstructured supplementary service</li> <li>– 0xFF - Not Available</li> </ul> </li> </ul>
<i>pCallID</i>	<ul style="list-style-type: none"> <li>• Call ID of the voice call that resulted from call control. (optional)</li> <li>• It is present when pCCResType is present and is Voice.</li> <li>• If zero(0) then invalid.</li> </ul>
<i>pCCSUPSType</i>	<ul style="list-style-type: none"> <li>• Supplementary service data that resulted from call control (optional)</li> <li>• Data is present when pCCResType is present and is other than Voice. <ul style="list-style-type: none"> <li>– See <a href="#">ccSUPSType</a> for more information</li> </ul> </li> </ul>

## Note

Using NULL for the pointers would make sure that the parameter is not returned or has default value.

## 8.563.2 Field Documentation

8.563.2.1 **alphaIDInfo\*** voiceSetCallBarringPwdResp::pAlphaIDInfo

8.563.2.2 **BYTE\*** voiceSetCallBarringPwdResp::pCallID

8.563.2.3 **BYTE\*** voiceSetCallBarringPwdResp::pCCResType

8.563.2.4 **ccSUPSType\*** voiceSetCallBarringPwdResp::pCCSUPSType

8.563.2.5 **WORD\*** voiceSetCallBarringPwdResp::pFailCause

## 8.564 voiceSetConfigReq Struct Reference



## Data Fields

- [BYTE](#) \* [pAutoAnswer](#)
- [airTimer](#) \* [pAirTimerConfig](#)
- [roamTimer](#) \* [pRoamTimerConfig](#)
- [BYTE](#) \* [pTTYMode](#)
- [prefVoiceSO](#) \* [pPrefVoiceSO](#)
- [BYTE](#) \* [pPrefVoiceDomain](#)

## 8.564.1 Detailed Description

This structure contains information about the Set Configuration Request Parameters.

## Parameters

<i>pAutoAnswer</i>	<ul style="list-style-type: none"> <li>• Value specified is written to NV_AUTO_ANSWER_I. (optional)</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0x00 - Disable</li> <li>– 0x01 - Enable</li> </ul> </li> </ul>
<i>pAirTimerConfig</i>	<ul style="list-style-type: none"> <li>• Value specified is written to NV_AIR_CNT_I. (optional)</li> <li>• See <a href="#">airTimer</a> for more information</li> </ul>
<i>pRoamTimerConfig</i>	<ul style="list-style-type: none"> <li>• Value specified is written to NV_ROAM_CNT_I. (optional)</li> <li>• See <a href="#">roamTimer</a> for more information</li> </ul>
<i>pTTYMode</i>	<ul style="list-style-type: none"> <li>• Value specified is written to NV_TTY_I. (optional)</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0x00 - TTY_MODE_FULL - Full</li> <li>– 0x01 - TTY_MODE_VCO - Voice carry over</li> <li>– 0x02 - TTY_MODE_HCO - Hearing carry over</li> <li>– 0x03 - TTY_MODE_OFF - Off</li> </ul> </li> </ul>

<i>pPrefVoiceSO</i>	<ul style="list-style-type: none"> <li>• Value specified is written to NV_PREF_VOICE_SO_I. (optional)</li> <li>• See <a href="#">prefVoiceSO</a> for more information</li> </ul>
<i>pPrefVoice-Domain</i>	<ul style="list-style-type: none"> <li>• Preferred Voice-Domain. (optional)</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0x00 - VOICE_DOMAIN_PREF_CS_ONLY - Circuit-switched (CS) only</li> <li>– 0x01 - VOICE_DOMAIN_PREF_PS_ONLY - Packet-switched (PS) only</li> <li>– 0x02 - VOICE_DOMAIN_PREF_CS_PREF - CS is preferred, PS is secondary</li> <li>– 0x03 - VOICE_DOMAIN_PREF_PS_PREF - PS is preferred, CS is secondary</li> </ul> </li> </ul>

#### Note

One of the optional parameters must be present in the request.

### 8.564.2 Field Documentation

8.564.2.1 **airTimer\*** voiceSetConfigReq::pAirTimerConfig

8.564.2.2 **BYTE\*** voiceSetConfigReq::pAutoAnswer

8.564.2.3 **BYTE\*** voiceSetConfigReq::pPrefVoiceDomain

8.564.2.4 **prefVoiceSO\*** voiceSetConfigReq::pPrefVoiceSO

8.564.2.5 **roamTimer\*** voiceSetConfigReq::pRoamTimerConfig

8.564.2.6 **BYTE\*** voiceSetConfigReq::pTTYMode

### 8.565 voiceSetConfigResp Struct Reference

#### Data Fields

- **BYTE \*** [pAutoAnsStatus](#)
- **BYTE \*** [pAirTimerStatus](#)
- **BYTE \*** [pRoamTimerStatus](#)
- **BYTE \*** [pTTYConfigStatus](#)
- **BYTE \*** [pPrefVoiceSOStatus](#)
- **BYTE \*** [pVoiceDomainPrefStatus](#)

#### 8.565.1 Detailed Description

This structure contains information about the Set Configuration Response Parameters.

## Parameters

<i>pAutoAnsStatus</i>	<ul style="list-style-type: none"><li>• Auto Answer Status. (optional)</li><li>• Values:<ul style="list-style-type: none"><li>– 0x00 - Information was written successfully</li><li>– 0x01 - Information write failed</li><li>– 0xFF - Not Available.</li></ul></li></ul>
<i>pAirTimerStatus</i>	<ul style="list-style-type: none"><li>• Air Timer Status. (optional)</li><li>• Values:<ul style="list-style-type: none"><li>– 0x00 - Information was written successfully</li><li>– 0x01 - Information write failed</li><li>– 0xFF - Not Available.</li></ul></li></ul>
<i>pRoamTimer- Status</i>	<ul style="list-style-type: none"><li>• Roam Timer Status. (optional)</li><li>• Values:<ul style="list-style-type: none"><li>– 0x00 - Information was written successfully</li><li>– 0x01 - Information write failed</li><li>– 0xFF - Not Available.</li></ul></li></ul>
<i>pTTYConfig- Status</i>	<ul style="list-style-type: none"><li>• TTY Config Status. (optional)</li><li>• Values:<ul style="list-style-type: none"><li>– 0x00 - Information was written successfully</li><li>– 0x01 - Information write failed</li><li>– 0xFF - Not Available.</li></ul></li></ul>

<i>pPrefVoiceSO-Status</i>	<ul style="list-style-type: none"> <li>• Preferred Voice SO Status. (optional)</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0x00 - Information was written successfully</li> <li>– 0x01 - Information write failed</li> <li>– 0xFF - Not Available.</li> </ul> </li> </ul>
<i>pVoiceDomain-PrefStatus</i>	<ul style="list-style-type: none"> <li>• Voice-Domain Preference Status. (optional)</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0x00 - Information was written successfully</li> <li>– 0x01 - Information write failed</li> <li>– 0xFF - Not Available.</li> </ul> </li> </ul>

**Note**

Parameters which are mentioned as NULL will be ignored.

**8.565.2 Field Documentation**

8.565.2.1 **BYTE\*** voiceSetConfigResp::pAirTimerStatus

8.565.2.2 **BYTE\*** voiceSetConfigResp::pAutoAnsStatus

8.565.2.3 **BYTE\*** voiceSetConfigResp::pPrefVoiceSOStatus

8.565.2.4 **BYTE\*** voiceSetConfigResp::pRoamTimerStatus

8.565.2.5 **BYTE\*** voiceSetConfigResp::pTTYConfigStatus

8.565.2.6 **BYTE\*** voiceSetConfigResp::pVoiceDomainPrefStatus

**8.566 voiceSetPrefPrivacy Struct Reference****Data Fields**

- [BYTE privacyPref](#)

**8.566.1 Detailed Description**

This structure contains the preferred voice privacy values.

## Parameters

<i>privacyPref</i>	<ul style="list-style-type: none"><li>• Voice Privacy Preference<ul style="list-style-type: none"><li>– 0x00 - VOICE_PRIVACY_STANDARD - Standard privacy</li><li>– 0x01 - VOICE_PRIVACY_ENHANCED - Enhanced privacy</li></ul></li></ul>
--------------------	---

## 8.566.2 Field Documentation

## 8.566.2.1 BYTE voiceSetPrefPrivacy::privacyPref

## 8.567 voiceSetSUPSServiceReq Struct Reference

## Data Fields

- [BYTE voiceSvc](#)
- [BYTE reason](#)
- [BYTE \\* pServiceClass](#)
- [BYTE \\* pCallBarringPasswd](#)
- [BYTE \\* pCallForwardingNumber](#)
- [BYTE \\* pTimerVal](#)
- [callFwdTypeAndPlan \\* pCallFwdTypeAndPlan](#)

## 8.567.1 Detailed Description

This structure contains Supplementary Service request parameters related to different features and their activation, deactivation, registration and erasure (applicable only for 3GPP)

## Parameters

<i>voiceSvc</i>	<ul style="list-style-type: none"><li>• Manages all call-independent supplementary services, such as activation, deactivation, registration, and erasure (mandatory)<ul style="list-style-type: none"><li>– 0x01 - VOICE_SERVICE_ACTIVATE</li><li>– 0x02 - VOICE_SERVICE_DEACTIVATE</li><li>– 0x03 - VOICE_SERVICE_REGISTER</li><li>– 0x04 - VOICE_SERVICE_ERASE</li></ul></li></ul>
-----------------	--

<i>reason</i>	<ul style="list-style-type: none"> <li>• supplementary service reason values (mandatory) <ul style="list-style-type: none"> <li>– 0x01 - QMI_VOICE_REASON_FWD_UNCONDITIONAL Unconditional call forwarding</li> <li>– 0x02 - QMI_VOICE_REASON_FWD_MOBILEBUSY Forward when the mobile is busy</li> <li>– 0x03 - QMI_VOICE_REASON_FWD_NOREPLY Forward when there is no reply</li> <li>– 0x04 - QMI_VOICE_REASON_FWD_UNREACHABLE Forward when the call is unreachable</li> <li>– 0x05 - QMI_VOICE_REASON_FWD_ALLFORWARDING All forwarding</li> <li>– 0x06 - QMI_VOICE_REASON_FWD_ALLCONDITIONAL All conditional forwarding</li> <li>– 0x07 - QMI_VOICE_REASON_BARR_ALLOUTGOING All outgoing calls are barred</li> <li>– 0x08 - QMI_VOICE_REASON_BARR_OUTGOINGINT Outgoing internal calls are barred</li> <li>– 0x09 - QMI_VOICE_REASON_BARR_OUTGOINGINTEXTOHOME Outgoing calls external to home are barred</li> <li>– 0x0A - QMI_VOICE_REASON_BARR_ALLINCOMING All incoming calls are barred</li> <li>– 0x0B - QMI_VOICE_REASON_BARR_INCOMINGROAMING Roaming incoming calls are barred</li> <li>– 0x0C - QMI_VOICE_REASON_BARR_ALLBARRING All calls are barred</li> <li>– 0x0D - QMI_VOICE_REASON_BARR_ALLOUTGOINGBARRING All outgoing calls are barred</li> <li>– 0x0E - QMI_VOICE_REASON_BARR_ALLINCOMINGBARRING All incoming calls are barred</li> <li>– 0x0F - QMI_VOICE_REASON_CALLWAITING Call waiting</li> </ul> </li> </ul>
---------------	---

<i>pServiceClass</i>	<ul style="list-style-type: none"> <li>Service class is a combination (sum) of information class constants (optional) <ul style="list-style-type: none"> <li>See <a href="#">serviceClassInformation</a> for more information</li> </ul> </li> </ul>
<i>pCallBarring-Passwd</i>	<ul style="list-style-type: none"> <li>Password is required if call barring is provisioned using a password. Password consists of 4 ASCII digits. Range: 0000 to 9999 (optional)</li> </ul>
<i>pCallForwarding-Number</i>	<ul style="list-style-type: none"> <li>Call forwarding number to be registered with the network. This has to be included in the request only when the service is set to VOICE_SERVICE_REGISTER. NULL terminated ASCII string. (optional)</li> </ul>
<i>pTimerVal</i>	<ul style="list-style-type: none"> <li>Call forwarding no reply timer value in seconds. This has to be included in the request only when the service is set to VOICE_SERVICE_REGISTER and the reason is QMI_VOICE_REASON_FWD_NOREPLY. (optional) <ul style="list-style-type: none"> <li>Range: 5 to 30 in steps of 5</li> </ul> </li> </ul>
<i>pCallFwdType-AndPlan</i>	<ul style="list-style-type: none"> <li>Information about call forwarding type and plan. This parameter is ignored when the Call Forwarding Number is not included (optional) <ul style="list-style-type: none"> <li>See <a href="#">callFwdTypeAndPlan</a> for more information</li> </ul> </li> </ul>

## 8.567.2 Field Documentation

8.567.2.1 **BYTE\*** voiceSetSUPSServiceReq::pCallBarringPasswd

8.567.2.2 **BYTE\*** voiceSetSUPSServiceReq::pCallForwardingNumber

8.567.2.3 **callFwdTypeAndPlan\*** voiceSetSUPSServiceReq::pCallFwdTypeAndPlan

8.567.2.4 **BYTE\*** voiceSetSUPSServiceReq::pServiceClass

8.567.2.5 **BYTE\*** voiceSetSUPSServiceReq::pTimerVal

8.567.2.6 **BYTE** voiceSetSUPSServiceReq::reason

8.567.2.7 **BYTE** voiceSetSUPSServiceReq::voiceSvc

## 8.568 voiceSetSUPSServiceResp Struct Reference

### Data Fields

- WORD \*** pFailCause
- alphaIDInfo \*** pAlphaIDInfo
- BYTE \*** pCCResultType
- BYTE \*** pCallID

- [ccSUPSType](#) \* [pCCSUPSType](#)

### 8.568.1 Detailed Description

This structure contains Supplementary Service response parameters related to different features and their activation, deactivation, registration and erasure (applicable only for 3GPP)

#### Parameters

<i>pFailCause</i>	<ul style="list-style-type: none"> <li>• Supplementary service failure causes (optional, supply NULL if not required). <ul style="list-style-type: none"> <li>– 0xFFFF is the value when the information is not received from device</li> </ul> </li> </ul>
<i>pAlphaIDInfo</i>	<ul style="list-style-type: none"> <li>• Pointer to structure of <a href="#">alphaIDInfo</a>. The parameter used to pass the alpha (if any) given by the SIM/R-UIM after call control (optional, supply NULL if not required) <ul style="list-style-type: none"> <li>– See <a href="#">alphaIDInfo</a> for more information</li> </ul> </li> </ul>
<i>pCCResultType</i>	<ul style="list-style-type: none"> <li>• Call control result types (optional, supply NULL if not required) <ul style="list-style-type: none"> <li>– 0x00 - CC_RESULT_TYPE_VOICE - Voice</li> <li>– 0x01 - CC_RESULT_TYPE_SUPS - Supplementary service</li> <li>– 0x02 - CC_RESULT_TYPE_USSD - Unstructured supplementary service</li> <li>– 0xFF - if the device does not provide this information</li> </ul> </li> </ul>
<i>pCallID</i>	<ul style="list-style-type: none"> <li>• Unique call identifier for the dialed call (optional, supply NULL if not required) <ul style="list-style-type: none"> <li>– 0x00 - if the device does not provide this information</li> </ul> </li> </ul>
<i>pCCSUPSType</i>	<ul style="list-style-type: none"> <li>• Data is present when pCCResultType is present and is other than Voice. (optional, supply NULL if not required) <ul style="list-style-type: none"> <li>– See <a href="#">ccSUPSType</a> for more information</li> </ul> </li> </ul>

### 8.568.2 Field Documentation

8.568.2.1 [alphaIDInfo](#)\* [voiceSetSUPSServiceResp::pAlphaIDInfo](#)

8.568.2.2 [BYTE](#)\* [voiceSetSUPSServiceResp::pCallID](#)

8.568.2.3 [BYTE](#)\* [voiceSetSUPSServiceResp::pCCResultType](#)

8.568.2.4 [ccSUPSType](#)\* [voiceSetSUPSServiceResp::pCCSUPSType](#)

8.568.2.5 [WORD](#)\* [voiceSetSUPSServiceResp::pFailCause](#)



## 8.569 voiceStopContDTMFInfo Struct Reference

### Data Fields

- [BYTE callID](#)

### 8.569.1 Detailed Description

This structure contains parameters of stop continuous DTMF

#### Parameters

<i>pCallID</i> [IN/OUT]	<ul style="list-style-type: none"> <li>• Call ID associated with call on which the DTMF information has to be sent. Stop continuous DTMF request is sent to the current active/alerting call when pCallId is set to 0xFF.</li> <li>• This is IN/OUT parameter, value passed by user will be packed in request and value received from the device would be returned to the user.</li> <li>• If the call ID value received is 0, no value has been returned by the device</li> </ul>
-------------------------	--

### 8.569.2 Field Documentation

#### 8.569.2.1 BYTE voiceStopContDTMFInfo::callID

## 8.570 voiceSUPSInfo Struct Reference

### Data Fields

- [SUPSInfo SUPSInformation](#)
- [BYTE \\* pSvcClass](#)
- [BYTE \\* pReason](#)
- [BYTE \\* pCallFWNum](#)
- [BYTE \\* pCallFWTimerVal](#)
- [struct USSInfo \\* pUSSInfo](#)
- [BYTE \\* pCallID](#)
- [alphaIDInfo \\* pAlphaIDInfo](#)
- [BYTE \\* pCallBarPasswd](#)
- [newPwdData \\* pNewPwdData](#)
- [BYTE \\* pDataSrc](#)
- [WORD \\* pFailCause](#)
- [getCallFWInfo \\* pCallFwdInfo](#)
- [CLIRResp \\* pCLIRstatus](#)
- [CLIPResp \\* pCLIPstatus](#)
- [COLPResp \\* pCOLPstatus](#)
- [COLRResp \\* pCOLRstatus](#)
- [CNAPResp \\* pCNAPstatus](#)

### 8.570.1 Detailed Description

This structure contains the parameters passed for SLQSVoiceSetSUPSCallBack by the device.

## Parameters

<i>SUPS-Information(mandatory)</i>	<ul style="list-style-type: none"> <li>See <a href="#">SUPSInfo</a> for more information.</li> </ul>
<i>pSvc-Class(optional)</i>	<ul style="list-style-type: none"> <li>Service class is a combination (sum) of information class constants (optional)</li> <li>See <a href="#">qaGobiApiTableSupServiceInfoClasses.h</a> for service classes.</li> </ul>
<i>p-Reason(optional)</i>	<ul style="list-style-type: none"> <li>See <a href="#">qaGobiApiTableCallControlReturnReasons.h</a> for return reasons.</li> </ul>
<i>pCallFW-Num(optional)</i>	<ul style="list-style-type: none"> <li>Call forwarding number to be registered with the network.</li> <li>ASCII String, NULL terminated.</li> </ul>
<i>pCallFWTimer-Val(optional)</i>	<ul style="list-style-type: none"> <li>Call Forwarding No Reply Timer. <ul style="list-style-type: none"> <li>Range: 5 to 30 in steps of 5.</li> </ul> </li> </ul>
<i>pUSS-Info(optional)</i>	<ul style="list-style-type: none"> <li>See <a href="#">USSInfo</a> for more information.</li> </ul>
<i>pCallID(optional)</i>	<ul style="list-style-type: none"> <li>Call identifier of the voice call that has been modified to a supplementary service as a result of call control.</li> </ul>
<i>pAlphaID-Info(optional)</i>	<ul style="list-style-type: none"> <li>See <a href="#">alphaIDInfo</a> for more information.</li> </ul>
<i>pCallBar-Passwd(optional)</i>	<ul style="list-style-type: none"> <li>Password is required if call barring is provisioned using a password. <ul style="list-style-type: none"> <li>Password consists of 4 ASCII digits.</li> <li>Range: 0000 to 9999.</li> </ul> </li> <li>This also serves as the old password in the register password scenario.</li> </ul>
<i>pNewPwd-Data(optional)</i>	<ul style="list-style-type: none"> <li>See <a href="#">newPwdData</a> for more information.</li> </ul>
<i>pData-Src(optional)</i>	<ul style="list-style-type: none"> <li>Sups Data Source.</li> <li>Used to distinguish between the supplementary service data sent to the network and the response received from the network.</li> <li>If absent, the supplementary service data in this indication can be assumed as a request sent to the network.</li> </ul>

<i>pFail-Cause(optional)</i>	<ul style="list-style-type: none"> <li>• Supplementary services failure cause.</li> <li>• See <a href="#">qaGobiApiTableVoiceCallEndReasons.h</a> for more information.</li> </ul>
<i>pCallFwd-Info(optional)</i>	<ul style="list-style-type: none"> <li>• See <a href="#">getCallFWInfo</a> for more information.</li> </ul>
<i>pCLI-Rstatus(optional)</i>	<ul style="list-style-type: none"> <li>• See <a href="#">CLIRResp</a> for more information.</li> </ul>
<i>pCLI-Pstatus(optional)</i>	<ul style="list-style-type: none"> <li>• See <a href="#">CLIPResp</a> for more information.</li> </ul>
<i>pCOL-Pstatus(optional)</i>	<ul style="list-style-type: none"> <li>• See <a href="#">COLPResp</a> for more information.</li> </ul>
<i>pCOL-Rstatus(optional)</i>	<ul style="list-style-type: none"> <li>• See <a href="#">COLRResp</a> for more information.</li> </ul>
<i>pCNA-Pstatus(optional)</i>	<ul style="list-style-type: none"> <li>• See <a href="#">CNAPResp</a> for more information.</li> </ul>

## Note

None

## 8.570.2 Field Documentation

8.570.2.1 **alphaIDInfo\*** voiceSUPSInfo::pAlphaIDInfo8.570.2.2 **BYTE\*** voiceSUPSInfo::pCallBarPasswd8.570.2.3 **getCallFWInfo\*** voiceSUPSInfo::pCallFwdInfo8.570.2.4 **BYTE\*** voiceSUPSInfo::pCallFNum8.570.2.5 **BYTE\*** voiceSUPSInfo::pCallFWTimerVal8.570.2.6 **BYTE\*** voiceSUPSInfo::pCallID8.570.2.7 **CLIPResp\*** voiceSUPSInfo::pCLIPstatus8.570.2.8 **CLIRResp\*** voiceSUPSInfo::pCLIRstatus8.570.2.9 **CNAPResp\*** voiceSUPSInfo::pCNAPstatus8.570.2.10 **COLPResp\*** voiceSUPSInfo::pCOLPstatus8.570.2.11 **COLRResp\*** voiceSUPSInfo::pCOLRstatus8.570.2.12 **BYTE\*** voiceSUPSInfo::pDataSrc

8.570.2.13 **WORD\*** voiceSUPSInfo::pFailCause

8.570.2.14 **newPwdData\*** voiceSUPSInfo::pNewPwdData

8.570.2.15 **BYTE\*** voiceSUPSInfo::pReason

8.570.2.16 **BYTE\*** voiceSUPSInfo::pSvcClass

8.570.2.17 **struct USSInfo\*** voiceSUPSInfo::pUSSInfo

8.570.2.18 **SUPSInfo** voiceSUPSInfo::SUPSInformation

## 8.571 voiceSUPSNotification Struct Reference

### Data Fields

- [BYTE](#) callID
- [BYTE](#) notifType
- [WORD](#) \* pCUGIndex
- [ECTNum](#) \* pECTNum

### 8.571.1 Detailed Description

Contains the parameters passed for SLQSVoiceSetSUPSNotificationCallback by the device.

## Parameters

<i>callID</i>	<ul style="list-style-type: none"> <li>Unique identifier of the call for which the notification is applicable. (mandatory)</li> </ul>
<i>notifType</i>	<ul style="list-style-type: none"> <li>Notification type parameter (mandatory) <ul style="list-style-type: none"> <li>0x01 - NOTIFICATION_TYPE_OUTGOING_CALL_IS_FORWARDED Originated MO call is being forwarded to another user</li> <li>0x02 - NOTIFICATION_TYPE_OUTGOING_CALL_IS_WAITING Originated MO call is waiting at the called user</li> <li>0x03 - NOTIFICATION_TYPE_OUTGOING_CUG_CALL Outgoing call is a CUG call</li> <li>0x04 - NOTIFICATION_TYPE_OUTGOING_CALLS_BARRED Outgoing calls are barred</li> <li>0x05 - NOTIFICATION_TYPE_OUTGOING_CALL_IS_DEFLECTED Outgoing call is deflected</li> <li>0x06 - NOTIFICATION_TYPE_INCOMING_CUG_CALL Incoming call is a CUG call</li> <li>0x07 - NOTIFICATION_TYPE_INCOMING_CALLS_BARRED Incoming calls are barred</li> <li>0x08 - NOTIFICATION_TYPE_INCOMING_FORWARDED_CALL Incoming call received is a forwarded call</li> <li>0x09 - NOTIFICATION_TYPE_INCOMING_DEFLECTED_CALL Incoming call is a deflected call</li> <li>0x0A - NOTIFICATION_TYPE_INCOMING_CALL_IS_FORWARDED Incoming call is forwarded to another user</li> <li>0x0B - NOTIFICATION_TYPE_UNCOND_CALL_FORWARD_ACTIVE Unconditional call forwarding is active</li> <li>0x0C - NOTIFICATION_TYPE_COND_CALL_FORWARD_ACTIVE Conditional call forwarding is active</li> <li>0x0D - NOTIFICATION_TYPE_CLIR_SUPPRESSION_REJECTED CLIR suppression is rejected</li> <li>0x0E - NOTIFICATION_TYPE_CALL_IS_ON_HOLD Call is put on hold at the remote party</li> <li>0x0F - NOTIFICATION_TYPE_CALL_IS_RETRIEVED Call is retrieved at the remote party from the hold state</li> <li>0x10 - NOTIFICATION_TYPE_CALL_IS_IN_MPTY Call is in a conference</li> <li>0x11 - NOTIFICATION_TYPE_INCOMING_CALL_IS_ECT Incoming call is an explicit call transfer</li> </ul> </li> </ul>
<i>pCUGIndex</i>	<ul style="list-style-type: none"> <li>The CUG Index used to indicate that the incoming/outgoing call is a CUG call. (optional, NULL when not present) Range: 0x00 to 0x7FFF.</li> </ul>

<i>pECTNum</i>	<ul style="list-style-type: none"> <li>The ECT Number is used to indicate that the incoming call is an explicitly transferred call. (optional, NULL when not present)</li> </ul> Refer <a href="#">ECTNum</a> for details.
----------------	--

## Note

None

## 8.571.2 Field Documentation

8.571.2.1 BYTE voiceSUPSNotification::callID

8.571.2.2 BYTE voiceSUPSNotification::notifType

8.571.2.3 WORD\* voiceSUPSNotification::pCUGIndex

8.571.2.4 ECTNum\* voiceSUPSNotification::pECTNum

## 8.572 wcdmaCellInfo Struct Reference

## Data Fields

- [WORD](#) psc
- [SHORT](#) cpich\_rscp
- [SHORT](#) cpich\_ecno
- [SHORT](#) srxlev

## 8.572.1 Detailed Description

This structure contains information about the WCDMA Cell.

## Parameters

<i>psc</i>	<ul style="list-style-type: none"> <li>Primary scrambling code.</li> <li>Range: 0 to 511.</li> </ul>
<i>cpich_rscp</i>	<ul style="list-style-type: none"> <li>Absolute power level (in 1/10 dBm) of the common pilot channel as received by the UE.</li> <li>Range: -120.0 dBm to -25.0 dBm</li> </ul>
<i>cpich_ecno</i>	<ul style="list-style-type: none"> <li>CPICH Ec/No; ratio (in 1/10 dB) of the received energy per PN chip for the CPICH to the total received power spectral density at the UE antenna connector.</li> <li>Range: -50.0 dB to 0.</li> </ul>

<i>srxlev</i>	<ul style="list-style-type: none"> <li>• Cell selection Rx level (Srxlev) value.</li> <li>• Range: -128 to 128.</li> <li>• This field is only valid when ue_in_idle is TRUE.</li> </ul>
---------------	---

## 8.572.2 Field Documentation

8.572.2.1 **SHORT** wcdmaCellInfo::cpich\_ecno

8.572.2.2 **SHORT** wcdmaCellInfo::cpich\_rscp

8.572.2.3 **WORD** wcdmaCellInfo::psc

8.572.2.4 **SHORT** wcdmaCellInfo::srxlev

## 8.573 WCDMAECIOThresh Struct Reference

### Data Fields

- [BYTE](#) WCDMAECIOThreshListLen
- [WORD](#) \* pWCDMAECIOThreshList

### 8.573.1 Detailed Description

This structure contains WCDMA ECIO threshold related parameters.

#### Parameters

<i>WCDMAECIO- ThreshListLen</i>	<ul style="list-style-type: none"> <li>• Length of the WCDMA ECIO threshold list parameter to follow</li> </ul>
<i>pWCDMAECIO- ThreshList</i>	<ul style="list-style-type: none"> <li>• Array of ECIO thresholds (in units of 0.1 dB)</li> <li>• Maximum of 32 values</li> <li>• Range for ECIO values: -31.5 to 0 (in dB)</li> </ul>

## 8.573.2 Field Documentation

8.573.2.1 **WORD**\* WCDMAECIOThresh::pWCDMAECIOThreshList

8.573.2.2 **BYTE** WCDMAECIOThresh::WCDMAECIOThreshListLen

## 8.574 WCDMAInfoLTENeighborCell Struct Reference

### Data Fields

- [ULONG](#) wcdmaRRCTest
- [BYTE](#) umtsLTENbrCellLen

- [umtsLTENbrCell](#) [UMTSLTENbrCell](#) [255]

### 8.574.1 Detailed Description

This structure contains information about the WCDMA - LTE Neighboring Cell Info Set.

#### Parameters

<i>wcdmaRRC-State</i>	<ul style="list-style-type: none"> <li>• WCDMA RRC states.</li> <li>• Defined in 3GPP TS 25.331</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0x00 - NAS_WCDMA_RRC_STATE_DISCONNECTED <ul style="list-style-type: none"> <li>* WCDMA RRC State is IDLE</li> </ul> </li> <li>– 0x01 - NAS_WCDMA_RRC_STATE_CELL_PCH <ul style="list-style-type: none"> <li>* WCDMA RRC state is CELL_PCH</li> </ul> </li> <li>– 0x02 - NAS_WCDMA_RRC_STATE_URA_PCH <ul style="list-style-type: none"> <li>* WCDMA RRC state is URA_PCH</li> </ul> </li> <li>– 0x03 - NAS_WCDMA_RRC_STATE_CELL_FACH <ul style="list-style-type: none"> <li>* WCDMA RRC state is CELL_FACH</li> </ul> </li> <li>– 0x04 - NAS_WCDMA_RRC_STATE_CELL_DCH <ul style="list-style-type: none"> <li>* WCDMA RRC state is CELL_DCH</li> </ul> </li> </ul> </li> </ul>
<i>umtsLTENbr-CellLen</i>	<ul style="list-style-type: none"> <li>• Number of sets of UMTS LTE Neighbors.</li> </ul>
<i>UMTSLTENbr-Cell</i>	<ul style="list-style-type: none"> <li>• See <a href="#">umtsLTENbrCell</a> for more information.</li> </ul>

### 8.574.2 Field Documentation

8.574.2.1 [umtsLTENbrCell](#) WCDMAInfoLTENeighborCell::UMTSLTENbrCell[255]

8.574.2.2 **BYTE** WCDMAInfoLTENeighborCell::umtsLTENbrCellLen

8.574.2.3 **ULONG** WCDMAInfoLTENeighborCell::wcdmaRRCState

## 8.575 wcdmaLongMsgDecodingParams Struct Reference

### Data Fields

- **BYTE** \* pMessage
- **BYTE** \* pSenderAddrLength
- **CHAR** \* pSenderAddr
- **BYTE** \* pTextMsgLength
- **CHAR** \* pTextMsg
- **BYTE** \* pScAddrLength
- **CHAR** \* pScAddr



- [BYTE Time](#) [0x09]
- [BYTE Date](#) [0x09]
- [BYTE \\* pReferenceNum](#)
- [BYTE \\* pTotalNum](#)
- [BYTE \\* pPartNum](#)
- [BOOL \\* plsUDHPresent](#)

### 8.575.1 Detailed Description

Structure contains parameters which need to be decoded from message

#### Parameters

<i>pMessage[IN]</i>	<ul style="list-style-type: none"> <li>• Message read off the device via SLQSGetSMS</li> </ul>
<i>pSenderAddrLength[IN/OUT]</i>	<ul style="list-style-type: none"> <li>• Upon input, indicates the maximum number of ASCII characters (including NULL termination) that the pSenderAddr buffer can accommodate. A length with 24 will be much safe since this address field can be up to 12 octets (24 bytes) Upon successful output, returns the length of destination address string.</li> </ul>
<i>pSenderAddr[OUT]</i>	<ul style="list-style-type: none"> <li>• Note that a length with 24 bytes will be much safe. Returns NULL-terminated ASCII String containing destination address</li> </ul>
<i>pTextMsgLength[IN/OUT]</i>	<ul style="list-style-type: none"> <li>• Upon input, specifies the number of characters the given text message buffer can accommodate. Upon successful output, returns the number of characters returns in the given text message buffer.</li> </ul>
<i>pTextMsg[OUT]</i>	<ul style="list-style-type: none"> <li>• Encoded PDU message</li> </ul>
<i>pScAddrLength[IN/OUT]</i>	<ul style="list-style-type: none"> <li>• A length with 24 will be much safe since this address field can be up to 12 octets (24 bytes) Returns NULL-terminated ASCII String containing destination address</li> </ul>
<i>pScAddr[OUT]</i>	<ul style="list-style-type: none"> <li>• Note that a length with 24 bytes will be much safe. Returns NULL-terminated ASCII String containing service center address. This SMSC field contains the Type of Address. To get the exact SMSC address, skip the first two bytes. e.g, 9085290100334, 90 is the Type of Address, indicates international format of phone number</li> </ul>
<i>pTime[OUT]</i>	<ul style="list-style-type: none"> <li>• Time fetched from message</li> </ul>
<i>pReferenceNum[OUT]</i>	<ul style="list-style-type: none"> <li>• Reference number of the sms</li> </ul>
<i>pTotalNum[OUT]</i>	<ul style="list-style-type: none"> <li>• Total number of the concatenated message</li> </ul>

<i>pPartNum[OUT]</i>	<ul style="list-style-type: none"> <li>Sequence number of the current message</li> </ul>
<i>plsUDHPresent</i>	<ul style="list-style-type: none"> <li>Is User Data Header Present in the PDU? If yes, it means it is a concatenated SMS.</li> </ul>

## 8.575.2 Field Documentation

- 8.575.2.1 **BYTE** wcdmaLongMsgDecodingParams::Date[0x09]
- 8.575.2.2 **BOOL\*** wcdmaLongMsgDecodingParams::plsUDHPresent
- 8.575.2.3 **BYTE\*** wcdmaLongMsgDecodingParams::pMessage
- 8.575.2.4 **BYTE\*** wcdmaLongMsgDecodingParams::pPartNum
- 8.575.2.5 **BYTE\*** wcdmaLongMsgDecodingParams::pReferenceNum
- 8.575.2.6 **CHAR\*** wcdmaLongMsgDecodingParams::pScAddr
- 8.575.2.7 **BYTE\*** wcdmaLongMsgDecodingParams::pScAddrLength
- 8.575.2.8 **CHAR\*** wcdmaLongMsgDecodingParams::pSenderAddr
- 8.575.2.9 **BYTE\*** wcdmaLongMsgDecodingParams::pSenderAddrLength
- 8.575.2.10 **CHAR\*** wcdmaLongMsgDecodingParams::pTextMsg
- 8.575.2.11 **BYTE\*** wcdmaLongMsgDecodingParams::pTextMsgLength
- 8.575.2.12 **BYTE\*** wcdmaLongMsgDecodingParams::pTotalNum
- 8.575.2.13 **BYTE** wcdmaLongMsgDecodingParams::Time[0x09]

## 8.576 wcdmaMsgDecodingParams Struct Reference

### Data Fields

- [BYTE \\*](#) [pMessage](#)
- [BYTE \\*](#) [pSenderAddrLength](#)
- [CHAR \\*](#) [pSenderAddr](#)
- [BYTE \\*](#) [pTextMsgLength](#)
- [CHAR \\*](#) [pTextMsg](#)
- [BYTE \\*](#) [pScAddrLength](#)
- [CHAR \\*](#) [pScAddr](#)
- [BYTE](#) [Time](#) [0x09]
- [BYTE](#) [Date](#) [0x09]

### 8.576.1 Detailed Description

Structure contains parameters which need to be decoded from message

## Parameters

<i>pMessage</i> [IN]	<ul style="list-style-type: none"> <li>• Message read off the device via SLQSGetSMS</li> </ul>
<i>pSenderAddrLength</i> [IN/OUT]	<ul style="list-style-type: none"> <li>• Upon input, indicates the maximum number of ASCII characters (including NULL termination) that the pSenderAddr buffer can accommodate. A length with 24 will be much safe since this address field can be up to 12 octets (24 bytes) Upon successful output, returns the length of destination address string.</li> </ul>
<i>pSenderAddr</i> [OUT]	<ul style="list-style-type: none"> <li>• Note that a length with 24 bytes will be much safe. Returns NULL-terminated ASCII String containing destination address</li> </ul>
<i>pTextMsgLength</i> [IN/OUT]	<ul style="list-style-type: none"> <li>• Upon input, specifies the number of characters the given text message buffer can accommodate. Upon successful output, returns the number of characters returns in the given text message buffer.</li> </ul>
<i>pTextMsg</i> [OUT]	<ul style="list-style-type: none"> <li>• Encoded PDU message</li> </ul>
<i>pScAddrLength</i> [IN/OUT]	<ul style="list-style-type: none"> <li>• A length with 24 will be much safe since this address field can be up to 12 octets (24 bytes) Returns NULL-terminated ASCII String containing destination address</li> </ul>
<i>pScAddr</i> [OUT]	<ul style="list-style-type: none"> <li>• Note that a length with 24 bytes will be much safe. Returns NULL-terminated ASCII String containing service center address. This SMSC field contains the Type of Address. To get the exact SMSC address, skip the first two bytes. e.g, 9085290100334, 90 is the Type of Address, indicates international format of phone number</li> </ul>
<i>pTime</i> [OUT]	<ul style="list-style-type: none"> <li>• Time fetched from message</li> </ul>
<i>pDate</i>	<ul style="list-style-type: none"> <li>• Date fetched from message</li> </ul>

## 8.576.2 Field Documentation

8.576.2.1 BYTE wcdmaMsgDecodingParams::Date[0x09]

8.576.2.2 BYTE\* wcdmaMsgDecodingParams::pMessage

8.576.2.3 CHAR\* wcdmaMsgDecodingParams::pScAddr

8.576.2.4 BYTE\* wcdmaMsgDecodingParams::pScAddrLength

8.576.2.5 CHAR\* wcdmaMsgDecodingParams::pSenderAddr

8.576.2.6 BYTE\* wcdmaMsgDecodingParams::pSenderAddrLength

8.576.2.7 **CHAR\*** wcdmaMsgDecodingParams::pTextMsg

8.576.2.8 **BYTE\*** wcdmaMsgDecodingParams::pTextMsgLength

8.576.2.9 **BYTE** wcdmaMsgDecodingParams::Time[0x09]

## 8.577 wcdmaMsgEncodingParams Struct Reference

### Data Fields

- [ULONG](#) messageSize
- [CHAR \\*](#) pDestAddr
- [CHAR \\*](#) pTextMsg
- [CHAR \\*](#) pPDUMessage
- [BYTE](#) alphabet

### 8.577.1 Detailed Description

Structure contains parameters which need to encoded with message

#### Parameters

<i>messageSize</i>	<ul style="list-style-type: none"> <li>• The length of the message contents in bytes</li> </ul>
<i>pDestAddr[IN]</i>	<ul style="list-style-type: none"> <li>• Gives NULL-terminated ASCII String containing destination address</li> </ul>
<i>pTextMsg[IN]</i>	<ul style="list-style-type: none"> <li>• Text message to be encoded, maximum limit is 160 charaters</li> </ul>
<i>pPDUMessage[-OUT]</i>	<ul style="list-style-type: none"> <li>• Encoded PDU message</li> </ul>
<i>alphabet[IN]</i>	<ul style="list-style-type: none"> <li>• Encoding method to generate the PDU <ul style="list-style-type: none"> <li>– 0 - 7 bit encoding</li> <li>– 4 - 8 bit encoding</li> <li>– 8 - 16 bit UCS2 encoding</li> <li>– others value will be treated as default 7 bit encoding</li> </ul> </li> </ul>

### 8.577.2 Field Documentation

8.577.2.1 **BYTE** wcdmaMsgEncodingParams::alphabet

8.577.2.2 **ULONG** wcdmaMsgEncodingParams::messageSize

8.577.2.3 **CHAR\*** wcdmaMsgEncodingParams::pDestAddr

8.577.2.4 **CHAR\*** wcdmaMsgEncodingParams::pPDUMessage

8.577.2.5 CHAR\* wcdmaMsgEncodingParams::pTextMsg

## 8.578 WCDMARSSIThresh Struct Reference

### Data Fields

- [BYTE WCDMARSSIThreshListLen](#)
- [WORD \\* pWCDMARSSIThreshList](#)

### 8.578.1 Detailed Description

This structure contains WCDMA RSSI threshold related parameters.

#### Parameters

<i>WCDMARSSI-ThreshListLen</i>	<ul style="list-style-type: none"> <li>• Length of the WCDMA RSSI threshold list parameter to follow</li> </ul>
<i>pWCDMARSSI-ThreshList</i>	<ul style="list-style-type: none"> <li>• Array of RSSI thresholds (in units of 0.1 dBm)</li> <li>• Maximum of 32 values.</li> <li>• Range for RSSI values: -121 to 0 (in dBm)</li> </ul>

### 8.578.2 Field Documentation

8.578.2.1 WORD\* WCDMARSSIThresh::pWCDMARSSIThreshList

8.578.2.2 BYTE WCDMARSSIThresh::WCDMARSSIThreshListLen

## 8.579 WCDMASysInfo Struct Reference

### Data Fields

- [sysInfoCommon sysInfoWCDMA](#)
- [BYTE lacValid](#)
- [WORD lac](#)
- [BYTE cellIdValid](#)
- [ULONG cellId](#)
- [BYTE regRejectInfoValid](#)
- [BYTE rejectSrvDomain](#)
- [BYTE rejCause](#)
- [BYTE networkIdValid](#)
- [BYTE MCC \[3\]](#)
- [BYTE MNC \[3\]](#)
- [BYTE hsCallStatusValid](#)
- [BYTE hsCallStatus](#)
- [BYTE hsIndValid](#)
- [BYTE hsInd](#)
- [BYTE pscValid](#)
- [WORD psc](#)

### 8.579.1 Detailed Description

Structure for storing the WCDMA System Information.

## Parameters

<i>sysInfoWCDMA</i>	<ul style="list-style-type: none"> <li>• See <a href="#">sysInfoCommon</a> for more information.</li> </ul>
<i>lacValid</i>	<ul style="list-style-type: none"> <li>• Indicates whether the location area code is valid.. <ul style="list-style-type: none"> <li>– 0x00 - Invalid</li> <li>– 0x01 - Valid</li> <li>– 0xFF - Not Available</li> </ul> </li> </ul>
<i>lac</i>	<ul style="list-style-type: none"> <li>• Location area code.</li> <li>• Only applies to 3GPP. <ul style="list-style-type: none"> <li>– 0xFFFF - Not Available</li> </ul> </li> </ul>
<i>cellIdValid</i>	<ul style="list-style-type: none"> <li>• Indicates whether the cell ID is valid. <ul style="list-style-type: none"> <li>– 0x00 - Invalid</li> <li>– 0x01 - Valid</li> <li>– 0xFF - Not Available</li> </ul> </li> </ul>
<i>cellId</i>	<ul style="list-style-type: none"> <li>• Cell ID. <ul style="list-style-type: none"> <li>– 0xFFFFFFFF - Not Available</li> </ul> </li> </ul>
<i>regRejectInfoValid</i>	<ul style="list-style-type: none"> <li>• Indicates whether the registration reject information is valid. <ul style="list-style-type: none"> <li>– 0x00 - Invalid</li> <li>– 0x01 - Valid</li> <li>– 0xFF - Not Available</li> </ul> </li> </ul>
<i>rejectSrvDomain</i>	<ul style="list-style-type: none"> <li>• Type of service domain in which the registration is rejected. <ul style="list-style-type: none"> <li>– 0x00 - SYS_SRV_DOMAIN_NO_SRV - No service</li> <li>– 0x01 - Circuit-switched only</li> <li>– 0x02 - Packet-switched only</li> <li>– 0x03 - Circuit-switched and packet-switched</li> <li>– 0x04 - Camped</li> <li>– 0xFF - Not Available</li> </ul> </li> </ul>
<i>rejCause</i>	<ul style="list-style-type: none"> <li>• Reject cause values sent are specified in [3GPP TS 24.008, Section 10.5.3.6]. <ul style="list-style-type: none"> <li>– 0xFF - Not Available</li> </ul> </li> </ul>

<i>networkIdValid</i>	<ul style="list-style-type: none"> <li>Indicates whether the network ID is valid. <ul style="list-style-type: none"> <li>0x00 - Invalid</li> <li>0x01 - Valid</li> <li>0xFF - Not Available</li> </ul> </li> </ul>
<i>MCC[PLMN_LENGTH]</i>	<ul style="list-style-type: none"> <li>Mobile Country Code.</li> <li>MCC digits in ASCII characters</li> </ul>
<i>MNC[PLMN_LENGTH]</i>	<ul style="list-style-type: none"> <li>Mobile Network Code.</li> <li>MNC digits in ASCII characters</li> <li>An unused byte is set to 0xFF.</li> </ul>
<i>hsCallStatusValid</i>	<ul style="list-style-type: none"> <li>Indicates whether the high-speed call status is valid. <ul style="list-style-type: none"> <li>0x00 - Invalid</li> <li>0x01 - Valid</li> <li>0xFF - Not Available</li> </ul> </li> </ul>
<i>hsCallStatus</i>	<ul style="list-style-type: none"> <li>Call status on high speed.</li> <li>Only applicable for WCDMA. <ul style="list-style-type: none"> <li>0x00 - HSDPA and HSUPA are unsupported</li> <li>0x01 - HSDPA is supported</li> <li>0x02 - HSUPA is supported</li> <li>0x03 - HSDPA and HSUPA are supported</li> <li>0x04 - HSDPA+ is supported</li> <li>0x05 - HSDPA+ and HSUPA are supported</li> <li>0x06 - Dual-cell HSDPA+ is supported</li> <li>0x07 - Dual-cell HSDPA+ and HSUPA are supported</li> <li>0xFF - Not Available</li> </ul> </li> </ul>



<i>hsIndValid</i>	<ul style="list-style-type: none"> <li>Indicates whether high-speed service indication is valid. <ul style="list-style-type: none"> <li>0x00 - Invalid</li> <li>0x01 - Valid</li> <li>0xFF - Not Available</li> </ul> </li> </ul>
<i>hsInd</i>	<ul style="list-style-type: none"> <li>High-speed service indication</li> <li>Only applicable for WCDMA. <ul style="list-style-type: none"> <li>0x00 - HSDPA and HSUPA are unsupported</li> <li>0x01 - HSDPA is supported</li> <li>0x02 - HSUPA is supported</li> <li>0x03 - HSDPA and HSUPA are supported</li> <li>0x04 - HSDPA+ is supported</li> <li>0x05 - HSDPA+ and HSUPA are supported</li> <li>0x06 - Dual-cell HSDPA+ is supported</li> <li>0x07 - Dual-cell HSDPA+ and HSUPA are supported</li> <li>0xFF - Not Available</li> </ul> </li> </ul>
<i>pscValid</i>	<ul style="list-style-type: none"> <li>Indicates whether primary scrambling code is valid. <ul style="list-style-type: none"> <li>0x00 - Invalid</li> <li>0x01 - Valid</li> <li>0xFF - Not Available</li> </ul> </li> </ul>
<i>psc</i>	<ul style="list-style-type: none"> <li>Primary scrambling code. <ul style="list-style-type: none"> <li>0xFFFF - Not Available</li> </ul> </li> </ul>

## 8.579.2 Field Documentation

8.579.2.1 **ULONG** WCDMASysInfo::cellId

8.579.2.2 **BYTE** WCDMASysInfo::cellIdValid

8.579.2.3 **BYTE** WCDMASysInfo::hsCallStatus

8.579.2.4 **BYTE** WCDMASysInfo::hsCallStatusValid

8.579.2.5 **BYTE** WCDMASysInfo::hsInd

8.579.2.6 **BYTE** WCDMASysInfo::hsIndValid

- 8.579.2.7 WORD WCDMASysInfo::lac
- 8.579.2.8 BYTE WCDMASysInfo::lacValid
- 8.579.2.9 BYTE WCDMASysInfo::MCC[3]
- 8.579.2.10 BYTE WCDMASysInfo::MNC[3]
- 8.579.2.11 BYTE WCDMASysInfo::networkIdValid
- 8.579.2.12 WORD WCDMASysInfo::psc
- 8.579.2.13 BYTE WCDMASysInfo::pscValid
- 8.579.2.14 BYTE WCDMASysInfo::regRejectInfoValid
- 8.579.2.15 BYTE WCDMASysInfo::rejCause
- 8.579.2.16 BYTE WCDMASysInfo::rejectSrvDomain
- 8.579.2.17 sysInfoCommon WCDMASysInfo::sysInfoWCDMA

## 8.580 WdsByteTotals Struct Reference

### Data Fields

- [ULONG](#) \* pV4sessionId
- [ULONG](#) \* pV6sessionId
- struct [WdsByteTotalsElmnts](#) ByteTotalsElmntsV4
- struct [WdsByteTotalsElmnts](#) ByteTotalsElmntsV6

### 8.580.1 Detailed Description

WDS ByteTotals request data structure

#### Parameters

<i>pV4sessionId</i>	<ul style="list-style-type: none"> <li>• The v4 session ID for which the byte totals are to be retrieved</li> <li>• provide a NULL pointer if not applicable</li> </ul>
<i>pV6sessionId</i>	<ul style="list-style-type: none"> <li>• The v6 session ID for which the byte totals are to be retrieved</li> <li>• provide a NULL pointer if not applicable</li> </ul>
<i>ByteTotals-ElmntsV4</i>	<ul style="list-style-type: none"> <li>• data structure to be populated with the byte totals for V4 session</li> </ul>
<i>ByteTotals-ElmntsV6</i>	<ul style="list-style-type: none"> <li>• data structure to be populated with the byte totals for V6 session</li> </ul>

## Note

At least one of pV4sessionId and pV6sessionId must point to a valid session ID.

## 8.580.2 Field Documentation

8.580.2.1 struct WdsByteTotalsElmnts WdsByteTotals::ByteTotalsElmntsV4

8.580.2.2 struct WdsByteTotalsElmnts WdsByteTotals::ByteTotalsElmntsV6

8.580.2.3 ULONG\* WdsByteTotals::pV4sessionId

8.580.2.4 ULONG\* WdsByteTotals::pV6sessionId

## 8.581 WdsByteTotalsElmnts Struct Reference

## Data Fields

- [ULONGLONG](#) \* [pTXTotalBytes](#)
- [ULONGLONG](#) \* [pRXTotalBytes](#)

## 8.581.1 Detailed Description

WDS Bytes Totals request data structure for individual session

## Parameters

<i>pTXTotalBytes</i>	<ul style="list-style-type: none"> <li>• No of transmitted bytes without error.</li> </ul>
<i>pRXTotalBytes</i>	<ul style="list-style-type: none"> <li>• No of received bytes without error.</li> </ul>

## 8.581.2 Field Documentation

8.581.2.1 ULONGLONG\* WdsByteTotalsElmnts::pRXTotalBytes

8.581.2.2 ULONGLONG\* WdsByteTotalsElmnts::pTXTotalBytes

## 8.582 WdsConnectionRate Struct Reference

## Data Fields

- [ULONG](#) \* [pV4sessionId](#)
- [ULONG](#) \* [pV6sessionId](#)
- struct [WdsConnectionRateElmnts ConnRateElmntsV4](#)
- struct [WdsConnectionRateElmnts ConnRateElmntsV6](#)

## 8.582.1 Detailed Description

WDS ConnectionRate request data structure

## Parameters

<i>pV4sessionId</i>	<ul style="list-style-type: none"> <li>• The v4 session ID for which the connection rate are to be retrieved</li> <li>• provide a NULL pointer if not applicable</li> </ul>
<i>pV6sessionId</i>	<ul style="list-style-type: none"> <li>• The v6 session ID for which the connection rate are to be retrieved</li> <li>• provide a NULL pointer if not applicable</li> </ul>
<i>ConnRate-ElmntsV4</i>	<ul style="list-style-type: none"> <li>• data structure to be populated with the connection rate for V4 session</li> </ul>
<i>ConnRate-ElmntsV6</i>	<ul style="list-style-type: none"> <li>• data structure to be populated with the connection rate for V6 session</li> </ul>

## Note

At least one of pV4sessionId and pV6sessionId must point to a valid session ID.

## 8.582.2 Field Documentation

8.582.2.1 struct WdsConnectionRateElmnts WdsConnectionRate::ConnRateElmntsV4

8.582.2.2 struct WdsConnectionRateElmnts WdsConnectionRate::ConnRateElmntsV6

8.582.2.3 ULONG\* WdsConnectionRate::pV4sessionId

8.582.2.4 ULONG\* WdsConnectionRate::pV6sessionId

## 8.583 WdsConnectionRateElmnts Struct Reference

## Data Fields

- [ULONG \\* pCurrentChannelTXRate](#)
- [ULONG \\* pCurrentChannelRXRate](#)
- [ULONG \\* pMaxChannelTXRate](#)
- [ULONG \\* pMaxChannelRXRate](#)

## 8.583.1 Detailed Description

WDS Connection rates request data structure for individual session

## Parameters

<i>pCurrent-ChannelTXRate</i>	<ul style="list-style-type: none"> <li>• Instantaneous channel Tx rate in bits per second.</li> </ul>
-------------------------------	---

<i>pCurrent-ChannelRXRate</i>	<ul style="list-style-type: none"> <li>Instantaneous channel Rx rate in bits per second.</li> </ul>
<i>pMaxChannelTXRate</i>	<ul style="list-style-type: none"> <li>Maximum Tx rate that can be assigned to the device by the</li> <li>serving system in bits per second</li> </ul>
<i>pMaxChannelRXRate</i>	<ul style="list-style-type: none"> <li>Maximum Rx rate that can be assigned to the device by the</li> <li>serving system in bits per second</li> </ul>

## 8.583.2 Field Documentation

8.583.2.1 **ULONG\*** WdsConnectionRateElmnts::pCurrentChannelRXRate

8.583.2.2 **ULONG\*** WdsConnectionRateElmnts::pCurrentChannelTXRate

8.583.2.3 **ULONG\*** WdsConnectionRateElmnts::pMaxChannelRXRate

8.583.2.4 **ULONG\*** WdsConnectionRateElmnts::pMaxChannelTXRate

## 8.584 WDSGetLoopbackData Struct Reference

### Data Fields

- [BYTE ByteLoopbackMode](#)
- [BYTE ByteLoopbackMultiplier](#)

### 8.584.1 Detailed Description

This API to Queries Enable/disable Data Loopback Mode and set the value of loopback multiplier.

#### Parameters

<i>pReq</i>	[IN]
	<ul style="list-style-type: none"> <li>See <a href="#">WDSSetLoopbackData</a> for more information</li> </ul>

#### Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

#### See Also

see [qmerrno.h](#) for eQCWWAN\_xxx error values

Timeout: 2 seconds\n

WDS SWI Get Loopback Structure of Packet Data Connection Information.

## Parameters

<i>ByteLoopback-Mode</i>	<ul style="list-style-type: none"> <li>• Loopback Mode. <ul style="list-style-type: none"> <li>– 0 - Disable</li> <li>– 1 - Enable</li> </ul> </li> </ul>
<i>ByteLoopback-Multiplier</i>	<ul style="list-style-type: none"> <li>• Loopback multiplier. Number of downlink bytes to send for each uplink byte.</li> </ul>

## 8.584.2 Field Documentation

8.584.2.1 BYTE WDSGetLoopbackData::ByteLoopbackMode

8.584.2.2 BYTE WDSGetLoopbackData::ByteLoopbackMultiplier

## 8.585 WdsIpAddressInfoReq Struct Reference

## Data Fields

- [ULONG \\* pv4sessionId](#)
- [ULONG \\* pv6sessionId](#)
- [QmiWdsIpAddressInfo ip](#)

## 8.585.1 Field Documentation

8.585.1.1 QmiWdsIpAddressInfo WdsIpAddressInfoReq::ip

8.585.1.2 ULONG\* WdsIpAddressInfoReq::pv4sessionId

8.585.1.3 ULONG\* WdsIpAddressInfoReq::pv6sessionId

## 8.586 WdsPktStatisticsElmnts Struct Reference

## Data Fields

- [ULONG \\* pTXPacketSuccesses](#)
- [ULONG \\* pRXPacketSuccesses](#)
- [ULONG \\* pTXPacketErrors](#)
- [ULONG \\* pRXPacketErrors](#)
- [ULONG \\* pTXPacketOverflows](#)
- [ULONG \\* pRXPacketOverflows](#)
- [ULONGLONG \\* pTXOkBytesCount](#)
- [ULONGLONG \\* pRXOkBytesCount](#)
- [ULONGLONG \\* pTXOKBytesLastCall](#)
- [ULONGLONG \\* pRXOKBytesLastCall](#)
- [ULONG \\* pTXDroppedCount](#)
- [ULONG \\* pRXDroppedCount](#)

### 8.586.1 Detailed Description

WDS Pkt Statistics request data structure for individual session

## Parameters

<i>pTXPacket-Successes</i>	<ul style="list-style-type: none"> <li>No of transmitted Packets without error.</li> </ul>
<i>pRXPacket-Successes</i>	<ul style="list-style-type: none"> <li>No of received Packets without error.</li> </ul>
<i>pTXPacketErrors</i>	<ul style="list-style-type: none"> <li>Number of outgoing packets with framing errors.</li> </ul>
<i>pRXPacket-Errors</i>	<ul style="list-style-type: none"> <li>Number of incoming packets with framing errors.</li> </ul>
<i>pTXPacket-Overflows</i>	<ul style="list-style-type: none"> <li>Number of packets dropped because Tx buffer overflowed (out of memory).</li> </ul>
<i>pRXPacket-Overflows</i>	<ul style="list-style-type: none"> <li>Number of packets dropped because Rx buffer overflowed (out of memory).</li> </ul>
<i>pTXOkBytes-Count</i>	<ul style="list-style-type: none"> <li>No of bytes transmitted without error.</li> </ul>
<i>pRXOkBytes-Count</i>	<ul style="list-style-type: none"> <li>No of bytes received without error.</li> </ul>
<i>pTXOKBytes-LastCall</i>	<ul style="list-style-type: none"> <li>No of bytes transmitted without error during the last data call (0 if no call was made earlier). Returned only if not in a call, and when the previous call was made using RmNet (for any devices that support</li> </ul>
<i>pRXOKBytes-LastCall</i>	<ul style="list-style-type: none"> <li>Number of bytes received without error during the last data call (0 if no call was made earlier). Returned only if not in a call, and when the previous call was made using RmNet (for any devices that support</li> </ul>
<i>pTXDropped-Count</i>	<ul style="list-style-type: none"> <li>Number of outgoing packets dropped.</li> </ul>
<i>pRXDropped-Count</i>	<ul style="list-style-type: none"> <li>Number of incoming packets dropped.</li> </ul>

## 8.586.2 Field Documentation

8.586.2.1 **ULONG\*** WdsPktStatisticsElmnts::pRXDroppedCount8.586.2.2 **ULONGLONG\*** WdsPktStatisticsElmnts::pRXOkBytesCount8.586.2.3 **ULONGLONG\*** WdsPktStatisticsElmnts::pRXOKBytesLastCall8.586.2.4 **ULONG\*** WdsPktStatisticsElmnts::pRXPacketErrors



- 8.586.2.5 `ULONG*` WdsPktStatisticsElmnts::pRXPacketOverflows
- 8.586.2.6 `ULONG*` WdsPktStatisticsElmnts::pRXPacketSuccesses
- 8.586.2.7 `ULONG*` WdsPktStatisticsElmnts::pTXDroppedCount
- 8.586.2.8 `ULONGLONG*` WdsPktStatisticsElmnts::pTXOkBytesCount
- 8.586.2.9 `ULONGLONG*` WdsPktStatisticsElmnts::pTXOkBytesLastCall
- 8.586.2.10 `ULONG*` WdsPktStatisticsElmnts::pTXPacketErrors
- 8.586.2.11 `ULONG*` WdsPktStatisticsElmnts::pTXPacketOverflows
- 8.586.2.12 `ULONG*` WdsPktStatisticsElmnts::pTXPacketSuccesses

## 8.587 WdsPktStatisticsReq Struct Reference

### Data Fields

- [ULONG](#) \* [pStatMask](#)

### 8.587.1 Detailed Description

WDS PktStatistics request data structure

#### Parameters

<i>pStatMask</i>	<ul style="list-style-type: none"><li>• Packet Statistics Mask 0x00000001 - Tx packets OK 0x00000002 - Rx packets OK 0x00000004 - Tx packet errors 0x00000008 - Rx packet errors 0x00000010 - Tx overflows 0x00000020 - Rx overflows 0x00000040 - Tx bytes OK 0x00000080 - Rx bytes OK</li></ul>
------------------	--

### 8.587.2 Field Documentation

- 8.587.2.1 `ULONG*` WdsPktStatisticsReq::pStatMask

## 8.588 WdsPktStatisticsResp Struct Reference

### Data Fields

- [ULONG](#) \* [pV4sessionId](#)
- [ULONG](#) \* [pV6sessionId](#)
- struct [WdsPktStatisticsElmnts PktStatElmntsV4](#)
- struct [WdsPktStatisticsElmnts PktStatElmntsV6](#)

### 8.588.1 Detailed Description

WDS PktStatistics response data structure

## Parameters

<i>pV4sessionId</i>	<ul style="list-style-type: none"> <li>• The v4 session ID for which the byte totals are to be retrieved</li> <li>• provide a NULL pointer if not applicable</li> </ul>
<i>pV6sessionId</i>	<ul style="list-style-type: none"> <li>• The v6 session ID for which the byte totals are to be retrieved</li> <li>• provide a NULL pointer if not applicable</li> </ul>
<i>PktStatElmntsV4</i>	<ul style="list-style-type: none"> <li>• data structure to be populated with the Pkt Statistics for V4 session</li> </ul>
<i>PktStatElmntsV6</i>	<ul style="list-style-type: none"> <li>• data structure to be populated with the Pkt Statistics for V6 session</li> </ul>

## Note

At least one of pV4sessionId and pV6sessionId must point to a valid session ID.

## 8.588.2 Field Documentation

8.588.2.1 struct WdsPktStatisticsElmnts WdsPktStatisticsResp::PktStatElmntsV4

8.588.2.2 struct WdsPktStatisticsElmnts WdsPktStatisticsResp::PktStatElmntsV6

8.588.2.3 ULONG\* WdsPktStatisticsResp::pV4sessionId

8.588.2.4 ULONG\* WdsPktStatisticsResp::pV6sessionId

## 8.589 WdsProfileParam Union Reference

## Data Fields

- struct [Profile3GPP](#) SlqsProfile3GPP
- struct [Profile3GPP2](#) SlqsProfile3GPP2

## 8.589.1 Detailed Description

This union [WdsProfileParam](#) consist of [Profile3GPP](#) and [Profile3GPP2](#) out of which one will be used to create profile.

## 8.589.2 Field Documentation

8.589.2.1 struct Profile3GPP WdsProfileParam::SlqsProfile3GPP

8.589.2.2 struct Profile3GPP2 WdsProfileParam::SlqsProfile3GPP2

## 8.590 WdsRunTimeSettings Struct Reference

## Data Fields

- [ULONG](#) \* [v4sessionId](#)
- [ULONG](#) \* [v6sessionId](#)
- struct [qmiWdsRunTimeSettings](#) [rts](#)

## 8.590.1 Detailed Description

WDS runtime settings request data structure

## Parameters

<a href="#">v4sessionId</a>	<ul style="list-style-type: none"> <li>• The v4 session ID for which the runtime settings are to be retrieved</li> <li>• provide a NULL pointer if not applicable</li> </ul>
<a href="#">v6sessionId</a>	<ul style="list-style-type: none"> <li>• The v6 session ID for which the runtime settings are to be retrieved</li> <li>• provide a NULL pointer if not applicable</li> </ul>
<a href="#">qmiWdsRunTimeSettings</a>	<ul style="list-style-type: none"> <li>• data structure to be populated with the runtime settings</li> </ul>

## Note

At least one of [v4sessionId](#) and [v6sessionId](#) must point to a valid session ID.

## 8.590.2 Field Documentation

8.590.2.1 struct [qmiWdsRunTimeSettings](#) [WdsRunTimeSettings::rts](#)

8.590.2.2 [ULONG](#)\* [WdsRunTimeSettings::v4sessionId](#)

8.590.2.3 [ULONG](#)\* [WdsRunTimeSettings::v6sessionId](#)

## 8.591 wdsSetEventReportReq Struct Reference

## Data Fields

- [BYTE](#) \* [pCurrChannelRateInd](#)
- [TrStatInd](#) \* [pTransferStatInd](#)
- [BYTE](#) \* [pDataBearerTechInd](#)
- [BYTE](#) \* [pDormancyStatusInd](#)
- [BYTE](#) \* [pMIPStatusInd](#)
- [BYTE](#) \* [pCurrDataBearerTechInd](#)
- [BYTE](#) \* [pDataCallStatusChangeInd](#)
- [BYTE](#) \* [pCurrPrefDataSysInd](#)
- [BYTE](#) \* [pEVDOPageMonPerChangeInd](#)
- [BYTE](#) \* [pDataSystemStatusChangeInd](#)

### 8.591.1 Detailed Description

This structure contains the information about the Set Event Report Request parameters.

## Parameters

<i>pCurrChannel-RateInd</i>	(optional) <ul style="list-style-type: none"> <li>Current Channel Rate Indicator. <ul style="list-style-type: none"> <li>0 - Do not report</li> <li>1 - Report channel rate when it changes</li> </ul> </li> </ul>
<i>pTransferStatInd</i>	(optional) <ul style="list-style-type: none"> <li>See <a href="#">TrStatInd</a> for more information.</li> </ul>
<i>pDataBearer-TechInd</i>	(optional) <ul style="list-style-type: none"> <li>Data Bearer Technology Indicator. <ul style="list-style-type: none"> <li>0 - Do not report</li> <li>1 - Report radio interface used for data transfer when it changes</li> </ul> </li> </ul>
<i>pDormancy-StatusInd</i>	(optional) <ul style="list-style-type: none"> <li>Dormancy Status indicator. <ul style="list-style-type: none"> <li>0 - Do not report</li> <li>1 - Report traffic channel state of interface used for data connection</li> </ul> </li> </ul>
<i>pMIPStatusInd</i>	(optional) <ul style="list-style-type: none"> <li>MIP Status Indicator. <ul style="list-style-type: none"> <li>0 - Do not report</li> <li>1 - Report MIP status</li> </ul> </li> </ul>
<i>pCurrData-BearerTechInd</i>	(optional) <ul style="list-style-type: none"> <li>Current Data Bearer Technology Indicator. <ul style="list-style-type: none"> <li>0 - Do not report</li> <li>1 - Report current data bearer technology when it changes</li> </ul> </li> </ul>
<i>pDataCallStatus-ChangeInd</i>	(optional) <ul style="list-style-type: none"> <li>Data Call Status Change Indicator. <ul style="list-style-type: none"> <li>0 - Do not report</li> <li>1 - Report data call status change when it changes</li> </ul> </li> </ul>

<i>pCurrPrefData-SysInd</i>	(optional) <ul style="list-style-type: none"> <li>Current Preferred Data System Indicator. <ul style="list-style-type: none"> <li>0 - Do not report</li> <li>1 - Report preferred data system when it changes</li> </ul> </li> </ul>
<i>pEVDOPage-MonPerChange-Ind</i>	(optional) <ul style="list-style-type: none"> <li>EV-DO Page Monitor Period Change Indicator. <ul style="list-style-type: none"> <li>0 - Do not report</li> <li>1 - Report EV-DO page monitor period change event</li> </ul> </li> </ul>
<i>pDataSystem-StatusChange-Ind</i>	(optional) <ul style="list-style-type: none"> <li>Data System Status Change Indicator. <ul style="list-style-type: none"> <li>0 - Do not report</li> <li>1 - Report data system status change event</li> </ul> </li> </ul>

**Note**

At least one parameter should be present.

**8.591.2 Field Documentation**

8.591.2.1 **BYTE\*** wdsSetEventReportReq::pCurrChannelRateInd

8.591.2.2 **BYTE\*** wdsSetEventReportReq::pCurrDataBearerTechInd

8.591.2.3 **BYTE\*** wdsSetEventReportReq::pCurrPrefDataSysInd

8.591.2.4 **BYTE\*** wdsSetEventReportReq::pDataBearerTechInd

8.591.2.5 **BYTE\*** wdsSetEventReportReq::pDataCallStatusChangeInd

8.591.2.6 **BYTE\*** wdsSetEventReportReq::pDataSystemStatusChangeInd

8.591.2.7 **BYTE\*** wdsSetEventReportReq::pDormancyStatusInd

8.591.2.8 **BYTE\*** wdsSetEventReportReq::pEVDOPageMonPerChangeInd

8.591.2.9 **BYTE\*** wdsSetEventReportReq::pMIPStatusInd

8.591.2.10 **TrStatInd\*** wdsSetEventReportReq::pTransferStatInd

**8.592 WDSSetLoopbackData Struct Reference****Data Fields**

- BYTE \*** pLoopbackMode
- BYTE \*** pLoopbackMultiplier

### 8.592.1 Detailed Description

WDS SWI Set Loopback Structure of Set Loopback Information.

## Parameters

<i>pLoopbackMode</i>	<ul style="list-style-type: none"> <li>• Loopback Mode. <ul style="list-style-type: none"> <li>– 0 - Disable</li> <li>– 1 - Enable</li> </ul> </li> </ul>
<i>pLoopback-Multiplier</i>	<ul style="list-style-type: none"> <li>• Loopback multiplier. Number of downlink bytes to send for each uplink byte.</li> </ul>

## 8.592.2 Field Documentation

8.592.2.1 BYTE\* WDSSetLoopbackData::pLoopbackMode

8.592.2.2 BYTE\* WDSSetLoopbackData::pLoopbackMultiplier

## 8.593 WDSSWICurrentChannelRates Struct Reference

## Data Fields

- unsigned long [current\\_channel\\_tx\\_rate](#)
- unsigned long [current\\_channel\\_rx\\_rate](#)
- unsigned long [max\\_channel\\_tx\\_rate](#)
- unsigned long [max\\_channel\\_rx\\_rate](#)

## 8.593.1 Detailed Description

WDS SWI Current Channel Rates Structure of Packet Data Connection Information.

## Parameters

<i>current_channel-_tx_rate</i>	<ul style="list-style-type: none"> <li>• Current Channel Tx Rate.</li> </ul>
<i>current_channel-_rx_rate</i>	<ul style="list-style-type: none"> <li>• Current Channel Rx Rate.</li> </ul>
<i>max_channel_-tx_rate</i>	<ul style="list-style-type: none"> <li>• Max Channel Tx Rate.</li> </ul>
<i>max_channel_-rx_rate</i>	<ul style="list-style-type: none"> <li>• Max Channel Rx Rate.</li> </ul>

## 8.593.2 Field Documentation

8.593.2.1 unsigned long WDSSWICurrentChannelRates::current\_channel\_rx\_rate

8.593.2.2 unsigned long WDSSWICurrentChannelRates::current\_channel\_tx\_rate



8.593.2.3 unsigned long WDSSWICurrentChannelRates::max\_channel\_rx\_rate

8.593.2.4 unsigned long WDSSWICurrentChannelRates::max\_channel\_tx\_rate



## Chapter 9

# File Documentation

### 9.1 apdoxypages.c File Reference

Contains the module declaration for the Doxygen output. Also contains the content of the main page and related pages.

#### Namespaces

- [Tables](#)

#### 9.1.1 Detailed Description

Contains the module declaration for the Doxygen output. Also contains the content of the main page and related pages.

### 9.2 qaCbkCatEventReportInd.h File Reference

#### Data Structures

- struct [CatEventIDDataTlv](#)
- struct [CatAlPhalIdentifierTlv](#)
- struct [CatEventListTlv](#)
- struct [CatRefreshTlv](#)
- struct [CatEndProactiveSessionTlv](#)
- union [currentCatEvent](#)
- struct [CatCommonEventTlv](#)
- struct [QmiCbkCatEventStatusReportInd](#)

#### Macros

- [#define QMI\\_MAX\\_CAT\\_EVENT\\_DATA\\_LENGTH 255](#)
- [#define QMI\\_CAN\\_COMMON\\_EVENT\\_TLV\\_NUMBER 11](#)

## Enumerations

- enum [eQMI\\_CAT\\_EVENT\\_REPORT\\_IND\\_TLV](#) {  
[eTLV\\_CBK\\_DISPLAY\\_TEXT](#) = 0x10,  
[eTLV\\_CBK\\_GET\\_IN\\_KEY](#) = 0x11,  
[eTLV\\_CBK\\_GET\\_INPUT](#) = 0x12,  
[eTLV\\_CBK\\_SETUP\\_MENU](#) = 0x13,  
[eTLV\\_CBK\\_SELECT\\_ITEM](#) = 0x14,  
[eTLV\\_CBK\\_ALPHA\\_IDENTIFIER](#) = 0x15,  
[eTLV\\_CBK\\_SETUP\\_EVENT\\_LIST](#) = 0x16,  
[eTLV\\_CBK\\_SETUP\\_IDLE\\_MODE\\_TEXT](#) = 0x17,  
[eTLV\\_CBK\\_LANGUAGE\\_NOTIFICATION](#) = 0x18,  
[eTLV\\_CBK\\_REFRESH](#) = 0x19,  
[eTLV\\_CBK\\_END\\_PROACTIVE\\_SESSION](#) = 0x1A }
- enum [eQMI\\_CAT\\_EVENT\\_REPORT\\_IND\\_TLV\\_LENGTH](#) {  
[eTLV\\_SETUP\\_EVENT\\_LIST\\_LENGTH](#) = 0x04,  
[eTLV\\_REFRESH\\_LENGTH](#) = 0x03,  
[eTLV\\_END\\_PROACTIVE\\_SESSION\\_LENGTH](#) = 0x01 }

## Functions

- enum [eQCWWANError UpkQmiCbkCatEventReportInd](#) ([BYTE](#) \*pMdmResp, struct [QmiCbkCatEventStatus-ReportInd](#) \*pAipResp)

### 9.2.1 Macro Definition Documentation

9.2.1.1 `#define QMI_CAN_COMMON_EVENT_TLV_NUMBER 11`

9.2.1.2 `#define QMI_MAX_CAT_EVENT_DATA_LENGTH 255`

### 9.2.2 Enumeration Type Documentation

9.2.2.1 enum [eQMI\\_CAT\\_EVENT\\_REPORT\\_IND\\_TLV](#)

Enumerator

**[eTLV\\_CBK\\_DISPLAY\\_TEXT](#)**  
**[eTLV\\_CBK\\_GET\\_IN\\_KEY](#)**  
**[eTLV\\_CBK\\_GET\\_INPUT](#)**  
**[eTLV\\_CBK\\_SETUP\\_MENU](#)**  
**[eTLV\\_CBK\\_SELECT\\_ITEM](#)**  
**[eTLV\\_CBK\\_ALPHA\\_IDENTIFIER](#)**  
**[eTLV\\_CBK\\_SETUP\\_EVENT\\_LIST](#)**  
**[eTLV\\_CBK\\_SETUP\\_IDLE\\_MODE\\_TEXT](#)**  
**[eTLV\\_CBK\\_LANGUAGE\\_NOTIFICATION](#)**  
**[eTLV\\_CBK\\_REFRESH](#)**  
**[eTLV\\_CBK\\_END\\_PROACTIVE\\_SESSION](#)**

9.2.2.2 enum [eQMI\\_CAT\\_EVENT\\_REPORT\\_IND\\_TLV\\_LENGTH](#)

Enumerator

**[eTLV\\_SETUP\\_EVENT\\_LIST\\_LENGTH](#)**  
**[eTLV\\_REFRESH\\_LENGTH](#)**  
**[eTLV\\_END\\_PROACTIVE\\_SESSION\\_LENGTH](#)**

### 9.2.3 Function Documentation

- 9.2.3.1 enum eQCWWANError UpkQmiCbkCatEventReportInd ( BYTE \* pMdmResp, struct QmiCbkCatEventStatusReportInd \* pAipResp )

## 9.3 qaCbkSwiOmaDmEventReportInd.h File Reference

### Data Structures

- struct [sessionInfoTlv](#)
- struct [sessionInfoTlvExt](#)
- struct [QmiCbkSwiOmaDmEventStatusReportInd](#)
- struct [QmiCbkSwiOmaDmEventStatusReportIndExt](#)

### Macros

- #define [QMI\\_SWIOMA\\_DM\\_FOTA](#) 0x00
- #define [QMI\\_SWIOMA\\_DM\\_CONFIG](#) 0x01
- #define [QMI\\_SWIOMA\\_DM\\_NOT](#) 0x02

### Enumerations

- enum [eQMI\\_SWIOMA\\_DM\\_EVENT\\_REPORT\\_IND](#) {  
[eTLV\\_IND\\_OMA\\_DM\\_FOTA](#) = 0x10,  
[eTLV\\_IND\\_OMA\\_DM\\_CONFIG](#) = 0x11,  
[eTLV\\_IND\\_OMA\\_DM\\_NOT](#) = 0x12 }

### Functions

- enum [eQCWWANError](#) UpkQmiCbkSwiOmaDmEventReportInd (BYTE \*pMdmResp, struct [QmiCbkSwiOmaDmEventStatusReportInd](#) \*pApiResp)
- package enum [eQCWWANError](#) UpkQmiCbkSwiOmaDmEventReportIndExt (BYTE \*pMdmResp, struct [QmiCbkSwiOmaDmEventStatusReportInd](#) \*pApiResp)

### 9.3.1 Macro Definition Documentation

9.3.1.1 #define [QMI\\_SWIOMA\\_DM\\_CONFIG](#) 0x01

9.3.1.2 #define [QMI\\_SWIOMA\\_DM\\_FOTA](#) 0x00

9.3.1.3 #define [QMI\\_SWIOMA\\_DM\\_NOT](#) 0x02

### 9.3.2 Enumeration Type Documentation

9.3.2.1 enum [eQMI\\_SWIOMA\\_DM\\_EVENT\\_REPORT\\_IND](#)

#### Enumerator

***eTLV\_IND\_OMA\_DM\_FOTA***  
***eTLV\_IND\_OMA\_DM\_CONFIG***  
***eTLV\_IND\_OMA\_DM\_NOT***

### 9.3.3 Function Documentation

- 9.3.3.1 `enum eQCWWANError UpkQmiCbkSwiOmaDmEventReportInd ( BYTE * pMdmResp, struct QmiCbkSwiOmaDmEventStatusReportInd * pApiResp )`
- 9.3.3.2 `package enum eQCWWANError UpkQmiCbkSwiOmaDmEventReportIndExt ( BYTE * pMdmResp, struct QmiCbkSwiOmaDmEventStatusReportInd * pApiResp )`

## 9.4 qaGobiApiAudio.h File Reference

Audio Service API function prototypes.

### Data Structures

- struct [GetAudioProfileReq](#)
- struct [GetAudioProfileResp](#)
- struct [SetAudioProfileReq](#)
- struct [GetAudioPathConfigReq](#)
- struct [TXPCMIIRFtr](#)
- struct [RXPCMIIRFtr](#)
- struct [RXAGCList](#)
- struct [RXAVCList](#)
- struct [TXAGCList](#)
- struct [GetAudioPathConfigResp](#)
- struct [SetAudioPathConfigReq](#)
- struct [GetAudioVolTLBConfigReq](#)
- struct [GetAudioVolTLBConfigResp](#)
- struct [SetAudioVolTLBConfigReq](#)
- struct [SetAudioVolTLBConfigResp](#)

### Functions

- `ULONG SLQSGetAudioProfile (GetAudioProfileReq *pGetAudioProfileReq, GetAudioProfileResp *pGetAudioProfileResp)`
- `ULONG SLQSSetAudioProfile (SetAudioProfileReq *pSetAudioProfileReq)`
- `ULONG SLQSGetAudioPathConfig (GetAudioPathConfigReq *pGetAudioPathConfigReq, GetAudioPathConfigResp *pGetAudioPathConfigResp)`
- `ULONG SLQSSetAudioPathConfig (SetAudioPathConfigReq *pSetAudioPathConfigReq)`
- `ULONG SLQSGetAudioVolTLBConfig (GetAudioVolTLBConfigReq *pGetAudioVolTLBConfigReq, GetAudioVolTLBConfigResp *pGetAudioVolTLBConfigResp)`
- `ULONG SLQSSetAudioVolTLBConfig (SetAudioVolTLBConfigReq *pSetAudioVolTLBConfigReq, SetAudioVolTLBConfigResp *pSetAudioVolTLBConfigResp)`

#### 9.4.1 Detailed Description

Audio Service API function prototypes.

#### 9.4.2 Function Documentation

- 9.4.2.1 `ULONG SLQSGetAudioPathConfig ( GetAudioPathConfigReq * pGetAudioPathConfigReq, GetAudioPathConfigResp * pGetAudioPathConfigResp )`

This API gets the audio path configuration parameters.

## Parameters

<i>pGetAudioPath-ConfigReq</i> [IN]	<ul style="list-style-type: none"> <li>See <a href="#">GetAudioPathConfigReq</a> for more information</li> </ul>
<i>pGetAudioPath-ConfigResp</i> [OUT]	<ul style="list-style-type: none"> <li>See <a href="#">GetAudioPathConfigResp</a> for more information</li> </ul>

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Device Supported: SL9090

Timeout: 5 seconds

#### 9.4.2.2 ULONG SLQSGetAudioProfile ( GetAudioProfileReq \* *pGetAudioProfileReq*, GetAudioProfileResp \* *pGetAudioProfileResp* )

This API get the profile content of the requested audio generator.

## Parameters

<i>pGetAudio-ProfileReq</i> [IN]	<ul style="list-style-type: none"> <li>See <a href="#">GetAudioProfileReq</a> for more information</li> </ul>
<i>pGetAudio-ProfileResp</i> [OUT]	<ul style="list-style-type: none"> <li>See <a href="#">GetAudioProfileResp</a> for more information</li> </ul>

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Device Supported: SL9090

Timeout: 5 seconds

#### 9.4.2.3 ULONG SLQSGetAudioVoITLBConfig ( GetAudioVoITLBConfigReq \* *pGetAudioVoITLBConfigReq*, GetAudioVoITLBConfigResp \* *pGetAudioVoITLBConfigResp* )

This API gets the audio path configuration parameters.

## Parameters

<i>pGetAudioVolTL-BCfgReq[IN]</i>	<ul style="list-style-type: none"> <li>• See <a href="#">GetAudioVolTLBCfgReq</a> for more information</li> </ul>
<i>pGetAudioVolTL-BCfgResp[OUT]</i>	<ul style="list-style-type: none"> <li>• See <a href="#">GetAudioVolTLBCfgResp</a> for more information</li> </ul>

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Device Supported: SL9090

Timeout: 5 seconds

#### 9.4.2.4 ULONG SLQSSetAudioPathConfig ( SetAudioPathConfigReq \* pSetAudioPathConfigReq )

This API sets the audio path configuration parameters.

## Parameters

<i>pSetAudioPath-ConfigReq[IN]</i>	<ul style="list-style-type: none"> <li>• See <a href="#">SetAudioPathConfigReq</a> for more information</li> </ul>
------------------------------------	--

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Device Supported: SL9090

Timeout: 5 seconds

#### 9.4.2.5 ULONG SLQSSetAudioProfile ( SetAudioProfileReq \* pSetAudioProfileReq )

This API sets an audio profile.

## Parameters



<i>pSetAudioProfileReq[IN]</i>	<ul style="list-style-type: none"> <li>See <a href="#">SetAudioProfileReq</a> for more information</li> </ul>
--------------------------------	---

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Device Supported: SL9090  
Timeout: 5 seconds

#### 9.4.2.6 ULONG SLQSSetAudioVoTLBConfig ( SetAudioVoTLBConfigReq \* pSetAudioVoTLBCfgReq, SetAudioVoTLBConfigResp \* pSetAudioVoTLBCfgResp )

This API sets the audio path configuration parameters.

**Parameters**

<i>pSetAudioVoTLBCfgReq[IN]</i>	<ul style="list-style-type: none"> <li>See <a href="#">SetAudioVoTLBConfigReq</a> for more information</li> </ul>
<i>pSetAudioVoTLBCfgResp[OUT]</i>	<ul style="list-style-type: none"> <li>See <a href="#">SetAudioVoTLBConfigResp</a> for more information</li> </ul>

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Device Supported: SL9090  
Timeout: 5 seconds

## 9.5 qaGobiApiCat.h File Reference

Card Application Toolkit API function headers.

**Functions**

- [ULONG CATSendEnvelopeCommand](#) (ULONG cmdID, ULONG dataLen, BYTE \*pData)
- [ULONG CATSendTerminalResponse](#) (ULONG refID, ULONG dataLen, BYTE \*pData)

### 9.5.1 Detailed Description

Card Application Toolkit API function headers.

### 9.5.2 Function Documentation

#### 9.5.2.1 `ULONG CATSendEnvelopeCommand ( ULONG cmdID, ULONG dataLen, BYTE * pData )`

Sends the envelope command to the device.

##### Parameters

<i>cmdID</i>	<ul style="list-style-type: none"> <li>Envelope command type <ul style="list-style-type: none"> <li>0x01 - Menu Selection</li> <li>0x02 - Event DL User activity</li> <li>0x03 - Event DL Idle Screen Available</li> <li>0x04 - Event DL Language Selection</li> </ul> </li> </ul>
<i>dataLen</i>	<ul style="list-style-type: none"> <li>Length of <i>pData</i> in bytes</li> </ul>
<i>pData</i> [IN]	<ul style="list-style-type: none"> <li>Encoded envelope data as defined in ETSI TS 102 223, section 7 [Smart Cards: Card Application Toolkit (CAT) – Release 4]</li> </ul>

##### Returns

`eQCWWAN_ERR_NONE` on success, `eQCWWAN_xxx` error value otherwise

##### See Also

See [qmerrno.h](#) for `eQCWWAN_xxx` error values

##### Note

Technology Supported: UMTS  
Timeout: 2 seconds

#### 9.5.2.2 `ULONG CATSendTerminalResponse ( ULONG refID, ULONG dataLen, BYTE * pData )`

Sends the terminal response to the device.

##### Parameters

<i>refID</i>	<ul style="list-style-type: none"> <li>Proactive command reference ID. The value should be the same as indicated in the CAT event callback data for the relevant proactive command.</li> </ul>
--------------	--

<i>dataLen</i>	<ul style="list-style-type: none"> <li>Terminal response data length</li> </ul>
<i>pData[!N]</i>	<ul style="list-style-type: none"> <li>Terminal response for the relevant proactive command encoded as per ETSI TS 102 223, section 6.8 [Smart Cards: Card Application Toolkit (CAT) – Release 4]</li> </ul>

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Technology Supported: UMTS  
Timeout: 2 seconds

## 9.6 qaGobiApiCbk.h File Reference

Callback Service API function prototypes.

**Data Structures**

- struct [qaQmiInterfaceInfo](#)
- struct [slqsSessionStateInfo](#)
- struct [slqsWdsEventInfo](#)
- struct [TransferStatsDataType](#)
- struct [SignalStrengthDataType](#)
- struct [SMSMTMessage](#)
- struct [SMSTransferRouteMTMessage](#)
- struct [SMSMessageMode](#)
- struct [SMSEtwsMessage](#)
- struct [SMSEtwsPlmn](#)
- struct [SMSCAddress](#)
- struct [SMSONIMS](#)
- struct [SMSEventInfo\\_s](#)
- struct [CATEventDataType](#)
- struct [ServingSystemInfo](#)
- struct [RoamingInfo](#)
- struct [SLQSSignalStrengthsInformation](#)
- struct [SLQSSignalStrengthsIndReq](#)
- struct [ECTNum](#)
- struct [voiceSUPSNotification](#)
- struct [voiceSetAllCallStatusCbInfo](#)
- struct [\\_transLayerInfoNotification](#)
- struct [\\_transNWRegInfoNotification](#)
- struct [UIMStatusChangeInfo](#)

- struct [voicePrivacyInfo](#)
- struct [voiceDTMFEventInfo](#)
- struct [voiceSUPSInfo](#)
- struct [nasSysInfo](#)
- struct [UniversalTime](#)
- struct [nasNetworkTime](#)
- struct [omaDmFotaTlv](#)
- struct [omaDmFotaTlvExt](#)
- struct [omaDmConfigTlvExt](#)
- struct [omaDmConfigTlv](#)
- struct [omaDmNotificationsTlv](#)
- union [sessionInfo](#)
- union [sessionInfoExt](#)
- struct [SMSMemoryInfo](#)
- struct [voiceOTASPStatusInfo](#)
- struct [signalInfo](#)
- struct [callerIDInfo](#)
- struct [callingPartyInfo](#)
- struct [calledPartyInfo](#)
- struct [redirNumInfo](#)
- struct [NSSAudioCtrl](#)
- struct [lineCtrlInfo](#)
- struct [extDispRecInfo](#)
- struct [voiceInfoRec](#)
- struct [msgWaitingInfo](#)
- struct [QosFlowInfoState](#)
- struct [QosFlowInfo](#)
- struct [QosEventInfo](#)
- struct [nasSigInfo](#)
- struct [\\_modemTempNotification](#)
- struct [\\_packetSrvStatus](#)
- struct [HDRPersonalityInd](#)
- struct [imsSIPConfigInfo](#)
- struct [imsRegMgrConfigInfo](#)
- struct [imsSMSConfigInfo](#)
- struct [imsUserConfigInfo](#)
- struct [imsVoIPConfigInfo](#)
- struct [USSDNoWaitIndicationInfo](#)
- struct [channelRate](#)
- struct [DUNCallInfoInd](#)
- struct [SMSAsyncRawSend\\_s](#)
- struct [LteNasReleaseInfo\\_s](#)
- struct [SwiOTAMsg\\_s](#)
- struct [QmiCbkLocCradleMountInd](#)
- struct [QmiCbkLocEventTimeSyncInd](#)
- struct [QmiCbkLocInjectTimeInd](#)
- struct [accelAcceptReady\\_s](#)
- struct [gyroAcceptReady\\_s](#)
- struct [accelTempAcceptReady\\_s](#)
- struct [gyroTempAcceptReady\\_s](#)
- struct [QmiCbkLocSensorStreamingInd](#)
- struct [QmiCbkLocInjectSensorDataInd](#)
- struct [precisionDilution\\_s](#)
- struct [gpsTime\\_s](#)
- struct [sensorDataUsage\\_s](#)

- struct [svUsedforFix\\_s](#)
- struct [QmiCbKLocPositionReportInd](#)
- struct [imsaRegStatusInfo](#)
- struct [imsaSvcStatusInfo](#)
- struct [imsaRatStatusInfo](#)
- struct [imsaPdpStatusInfo](#)
- struct [satelliteInfo](#)
- struct [gnssSvInfoNotification](#)
- struct [delAssistDataStatus](#)
- struct [QmiCbKNasLTECphyCalInfo](#)

## Macros

- #define [SIGSTRENGTH\\_THRESHOLD\\_ARR\\_SZ](#) 5
- #define [QMI\\_WMS\\_MAX\\_PAYLOAD\\_LENGTH](#) 256
- #define [QMI\\_ETWS\\_MAX\\_PAYLOAD\\_LENGTH](#) 1254 /\* Qualcomm defined max \*/
- #define [QMI\\_MAX\\_VOICE\\_NUMBER\\_LENGTH](#) 81
- #define [MAX\\_NO\\_OF\\_UUSINFO](#) 20
- #define [MAXUSSDLENGTH](#) 182
- #define [MAX\\_NO\\_OF\\_CALLS](#) 20
- #define [CBK\\_ENABLE\\_EVENT](#) 0x01
- #define [CBK\\_DISABLE\\_EVENT](#) 0x00
- #define [CBK\\_NOCHANGE](#) 0xFF
- #define [MAX\\_NO\\_OF\\_APPLICATIONS](#) 10
- #define [MAX\\_NO\\_OF\\_SLOTS](#) 5
- #define [MAX\\_NO\\_OF\\_FILES](#) 255
- #define [MAX\\_PATH\\_LENGTH](#) 255
- #define [EVENT\\_MASK\\_CARD](#) 0x00000001
- #define [EVENT\\_MASK\\_DEREGISTER\\_ALL](#) 0x00000000
- #define [REGISTER\\_EVENT](#) 0x01
- #define [DEREGISTER\\_EVENT](#) 0x00
- #define [FIRST\\_INSTANCE](#) 0x00
- #define [SECOND\\_INSTANCE](#) 0x01
- #define [THIRD\\_INSTANCE](#) 0x02
- #define [INVALID\\_INSTACNE](#) 0x08
- #define [REGISTER\\_SRV](#) 0x01
- #define [DEREGISTER\\_SRV](#) 0x00
- #define [WDS\\_SRV](#) 0x01
- #define [NAS\\_SRV](#) 0x02
- #define [PDS\\_SRV](#) 0x04
- #define [VOICE\\_SRV](#) 0x08
- #define [NUM\\_OF\\_SET](#) 0xFF
- #define [IPV4](#) 4
- #define [IPV6](#) 6
- #define [IPV4V6](#) 7
- #define [LOC\\_EVENT\\_MASK\\_ENG\\_STATE](#) 0x00000080
- #define [LOC\\_EVENT\\_MASK\\_TIME\\_SYNC](#) 0x00000800
- #define [LOC\\_EVENT\\_MASK\\_INJECT\\_TIME](#) 0x00000010
- #define [LOC\\_EVENT\\_MASK\\_SENSOR\\_STREAM](#) 0x00000400
- #define [LOC\\_EVENT\\_POSITION\\_REPORT](#) 0x00000001
- #define [LOC\\_EVENT\\_MASK\\_GNSS\\_SV\\_INFO](#) 0x00000002
- #define [MAX\\_RADIO\\_INTERFACE\\_LIST](#) 255
- #define [USSD\\_DCS\\_ASCII](#) 0x01 /\* ASCII coding scheme \*/
- #define [USSD\\_DCS\\_8BIT](#) 0x02 /\* 8-bit coding scheme \*/
- #define [USSD\\_DCS\\_UCS2](#) 0x03 /\* UCS2 coding scheme \*/

## Typedefs

- typedef void(\* tFNSLQSSessionState )(slqsSessionStateInfo \*pSessionStateInfo)
- typedef void(\* tFNSLQSWDSEvent )(slqsWdsEventInfo \*pWdsEventInfo)
- typedef void(\* tFNPower )(ULONG operatingMode)
- typedef void(\* tFNActivationStatus )(ULONG activationStatus)
- typedef void(\* tFNMobileIPStatus )(ULONG mipStatus)
- typedef void(\* tFNRoamingIndicator )(ULONG roaming)
- typedef void(\* tFNDataCapabilities )(BYTE dataCapsSize, BYTE \*pDataCaps)
- typedef void(\* tFNSignalStrength )(INT8 signalStrength, ULONG radiolInterface)
- typedef void(\* tFNRInfo )(ULONG radiolInterface, ULONG activeBandClass, ULONG activeChannel)
- typedef void(\* tFNUReject )(ULONG serviceDomain, ULONG rejectCause)
- typedef void(\* tFNNewSMS )(ULONG storageType, ULONG messageIndex)
- typedef enum SMSEventType eSMSEventType
- typedef struct SMSMTMessage SMSMTMessageInfo
- typedef struct  
    SMSTransferRouteMTMessage SMSTransferRouteMTMessageInfo
- typedef struct SMSMessageMode SMSMessageModeInfo
- typedef struct SMSEtwsMessage SMSEtwsMessageInfo
- typedef struct SMSEtwsPlmn SMSEtwsPlmnInfo
- typedef struct SMSCAddress SMSCAddressInfo
- typedef struct SMSOnIMS SMSOnIMSInfo
- typedef struct SMSEventInfo\_s SMSEventInfo
- typedef void(\* tFNSMSEvents )(SMSEventInfo \*pSMSEventInfo)
- typedef void(\* tFNNewNMEA )(LPCSTR pNMEA)
- typedef void(\* tFNPDSSState )(ULONG enabledStatus, ULONG trackingStatus)
- typedef void(\* tFNCATEvent )(ULONG eventId, ULONG eventLen, BYTE \*pEventData)
- typedef enum device\_state\_enum eDevState
- typedef void(\* tFNDeviceStateChange )(eDevState device\_state)
- typedef void(\* tFNNet )(ULONG q\_depth, BYTE isThrottle, BYTE instanceId)
- typedef void(\* tFNFWdldCompletion )(ULONG fwdld\_completion\_status)
- typedef void(\* tFNSLQSOMADMAAlert )(ULONG eventType, BYTE \*pEventFields)
- typedef void(\* tFNOMADMState )(ULONG sessionState, ULONG failureReason)
- typedef void(\* tFNServingSystem )(struct ServingSystemInfo \*pServingSystem, struct RoamingInfo \*pRoamingInfo)
- typedef void(\* tFNBandPreference )(ULONGLONG band\_pref)
- typedef void(\* tFNUSSDRelease )(void)
- typedef void(\* tFNUSSDNotification )(ULONG type, BYTE \*pNetworkInfo)
- typedef void(\* tFNSLQSSignalStrengths )(struct SLQSSignalStrengthsInformation sSLQSSignalStrengthsInfo)
- typedef void(\* tFNSUPSNotification )(voiceSUPSNotification \*pVoiceSUPSNotification)
- typedef void(\* tFNSDKTerminated )(BYTE \*psReason)
- typedef void(\* tFNAllCallStatus )(voiceSetAllCallStatusCbInfo \*pVoiceSetAllCallStatusCbInfo)
- typedef struct  
    \_transLayerInfoNotification transLayerNotification
- typedef void(\* tFNtransLayerInfo )(transLayerNotification \*pTransLayerNotification)
- typedef struct  
    \_transNWRegInfoNotification transNWRegInfoNotification
- typedef void(\* tFNtransNWRegInfo )(transNWRegInfoNotification \*pTransNWRegInfoNotification)
- typedef void(\* tFNSysSelectionPref )(sysSelectPrefInfo \*pSysSelectPrefInfo)
- typedef void(\* tFNUIMRefresh )(UIMRefreshEvent \*pUIMRefreshEvent)
- typedef void(\* tFNUIMStatusChangeInfo )(UIMStatusChangeInfo \*pUIMStatusChangeInfo)
- typedef void(\* tFNPrivacyChange )(voicePrivacyInfo \*pVoicePrivacyInfo)
- typedef void(\* tFNDTMFEvent )(voiceDTMFEventInfo \*pVoiceDTMFEventInfo)
- typedef void(\* tFNSUPSInfo )(voiceSUPSInfo \*pVoiceSUPSInfo)

- typedef void(\* tFNSysInfo )(nasSysInfo \*pNasSysInfo)
- typedef void(\* tFNNetworkTime )(nasNetworkTime \*pNasNetworkTime)
- typedef union sessionInfo sessionInformation
- typedef union sessionInfoExt sessionInformationExt
- typedef void(\* tFNMemoryFull )(SMSMemoryInfo \*pSMSMemoryFullInfo)
- typedef void(\* tFNOTASPStatus )(voiceOTASPStatusInfo \*pVoiceOTASPStatusInfo)
- typedef void(\* tFNInfoRec )(voiceInfoRec \*pVoiceInfoRec)
- typedef void(\* tFNMessageWaiting )(msgWaitingInfo \*pSMSMessageWaitingInfo)
- typedef void(\* tFNSLQSQOSEvent )(BYTE instance, QosFlowInfo \*pFlowInfo)
- typedef void(\* tFNQosStatus )(BYTE instance, ULONG id, BYTE status, BYTE event, BYTE reason)
- typedef void(\* tFNQosNWStatus )(BYTE status)
- typedef void(\* tFNQosPriEvent )(WORD event)
- typedef void(\* tFNSigInfo )(nasSigInfo \*pNasSigInfo)
- typedef struct
  - \_modemTempNotification modemTempNotification
- typedef void(\* tFNModemTempInfo )(modemTempNotification \*pModemTempNotification)
- typedef struct \_packetSrvStatus packetSrvStatus
- typedef void(\* tFNPacketSrvState )(packetSrvStatus \*pPacketSrvStatus)
- typedef void(\* tFNHDRPersonality )(HDRPersonalityInd \*pHDRPers)
- typedef void(\* tFNImSIPConfig )(imsSIPConfigInfo \*pImSIPConfigInfo)
- typedef void(\* tFNImRegMgrConfig )(imsRegMgrConfigInfo \*pImRegMgrConfigInfo)
- typedef void(\* tFNImSMSConfig )(imsSMSConfigInfo \*pImSMSConfigInfo)
- typedef void(\* tFNImUserConfig )(imsUserConfigInfo \*pImUserConfigInfo)
- typedef void(\* tFNImVoIPConfig )(imsVoIPConfigInfo \*pImVoIPConfigInfo)
- typedef void(\* tFNUSSDNoWaitIndication )(USSDNoWaitIndicationInfo \*pNetworkInfo)
- typedef void(\* tFNDUNCallInfo )(DUNCallInfoInd \*pDUNCallInfo)
- typedef void(\* tFNDataSysStatus )(CurrDataSysStat \*pCurrDataSysStat)
- typedef struct SMSAsyncRawSend\_s SMSAsyncRawSend
- typedef void(\* tFNAsyncRawSend )(SMSAsyncRawSend \*pSMSAsyncRawSend)
- typedef struct LteNasReleaseInfo\_s LteNasReleaseInfo
- typedef struct SwiOTAMsg\_s SwiOTAMsg
- typedef void(\* tFNASwiOTAMsg )(SwiOTAMsg \*pSwiOTAMsg)
- typedef void(\* tFNNewGPS )(double dLongitude, double dLatitude)
- typedef void(\* tFNNewRMTransferStatistics )(QmiCbkWdsStatisticsIndState \*pMsg)
- typedef void(\* tFNSetCradleMount )(QmiCbkLocCradleMountInd \*pSetLocCradleMount)
- typedef void(\* tFNSetEventTimeSync )(QmiCbkLocEventTimeSyncInd \*pSetLocEventTimeSync)
- typedef void(\* tFNInjectTimeStatus )(QmiCbkLocInjectTimeInd \*pLocInjectTime)
- typedef struct accelAcceptReady\_s accelAcceptReady
- typedef struct gyroAcceptReady\_s gyroAcceptReady
- typedef struct
  - accelTempAcceptReady\_s accelTempAcceptReady
- typedef struct
  - gyroTempAcceptReady\_s gyroTempAcceptReady
- typedef void(\* tFNSensorStreaming )(QmiCbkLocSensorStreamingInd \*pLocSensorStream)
- typedef void(\* tFNInjectSensorData )(QmiCbkLocInjectSensorDataInd \*pLocInjectSensorData)
- typedef struct precisionDilution\_s precisionDilution
- typedef struct gpsTime\_s gpsTime
- typedef struct sensorDataUsage\_s sensorDataUsage
- typedef struct svUsedforFix\_s svUsedforFix
- typedef void(\* tFNEventPosition )(QmiCbkLocPositionReportInd \*pLocPositionReport)
- typedef void(\* tFNOpMode )(ULONG mode)
- typedef void(\* tFNImsaRegStatus )(imsaRegStatusInfo \*pImsaRegStatusInfo)
- typedef void(\* tFNImsaSvcStatus )(imsaSvcStatusInfo \*pImsaSvcStatusInfo)
- typedef void(\* tFNImsaRatStatus )(imsaRatStatusInfo \*pImsaRatStatusInfo)
- typedef void(\* tFNImsaPdpStatus )(imsaPdpStatusInfo \*pImsaPdpStatusInfo)
- typedef void(\* tFNGnssSvInfo )(gnssSvInfoNotification \*pGnssSvInfoNotification)
- typedef void(\* tFNDelAssistData )(delAssistDataStatus \*pAssistDataNotification)
- typedef void(\* tFNASwiLTECphyCallInfo )(QmiCbkNasLTECphyCaInfo \*pQmiCbkNasLTECphyCaInfo)

## Enumerations

- enum `eQaQMIService` {  
`eQA_QMI_SVC_WDS` = 0x01,  
`eQA_QMI_SVC_NAS` = 0x03,  
`eQA_QMI_SVC_NA` = 0xFF }
- enum `SMSEventType` {  
`SMS_EVENT_MT_MESSAGE` = 0x01,  
`SMS_EVENT_TRANSFER_ROUTE_MT_MESSAGE` = 0x02,  
`SMS_EVENT_MESSAGE_MODE` = 0x04,  
`SMS_EVENT_ETWS` = 0x08,  
`SMS_EVENT_ETWS_PLMN` = 0x10,  
`SMS_EVENT_SMSC_ADDRESS` = 0x20,  
`SMS_EVENT_SMS_ON_IMS` = 0x40 }
- enum `device_state_enum` {  
`DEVICE_STATE_DISCONNECTED`,  
`DEVICE_STATE_READY`,  
`DEVICE_STATE_BOOT` }

## Functions

- `ULONG SLQSSetSessionStateCallback` (`tFNSLQSSessionState` pCallback)
- `ULONG SLQSSetWdsEventCallback` (`tFNSLQSWDSEvent` pCallback, `BYTE` interval, `BYTE` instanceid, `BYTE` ipfamily)
- `ULONG SLQSSetWdsTransferStatisticCallback` (`tFNSLQSWDSEvent` pXferStatsCb, `BYTE` interval, `BYTE` instanceid, `BYTE` ipfamily)
- `ULONG iSLQSSetWdsFirstInstEventCallback` (`tFNSLQSWDSEvent` pCallback)
- `ULONG iSLQSSetWdsSecondInstEventCallback` (`tFNSLQSWDSEvent` pCallback)
- `ULONG iSLQSSetWdsThirdInstEventCallback` (`tFNSLQSWDSEvent` pCallback)
- `ULONG iSLQSSetWdsXferStatsFirstInstCallback` (`tFNSLQSWDSEvent` pCallback)
- `ULONG iSLQSSetWdsXferStatsSecondInstCallback` (`tFNSLQSWDSEvent` pCallback)
- `ULONG SetPowerCallback` (`tFNPower` pCallback)
- `ULONG SetActivationStatusCallback` (`tFNActivationStatus` pCallback)
- `ULONG SetMobileIPStatusCallback` (`tFNMobileIPStatus` pCallback)
- `ULONG SetRoamingIndicatorCallback` (`tFNRoamingIndicator` pCallback)
- `ULONG SetDataCapabilitiesCallback` (`tFNDataCapabilities` pCallback)
- `ULONG SetSignalStrengthCallback` (`tFNSignalStrength` pCallback, `BYTE` thresholdsSize, `INT8` \*p-Thresholds)
- `ULONG iSetSignalStrengthCallback` (`tFNSignalStrength` pCallback)
- `ULONG SetRFInfoCallback` (`tFNRInfo` pCallback)
- `ULONG SetLURRejectCallback` (`tFNLURReject` pCallback)
- `ULONG SetNewSMSCallback` (`tFNNewSMS` pCallback)
- `ULONG SLQSSetSMSEventCallback` (`tFNSMSEvents` pCallback)
- `ULONG SetNMEACallback` (`tFNNewNMEA` pCallback)
- `ULONG SetPDSSStateCallback` (`tFNPDSState` pCallback)
- `ULONG SetCATEventCallback` (`tFNCATEvent` pCallback, `ULONG` eventMask, `ULONG` \*pErrorMask)
- `ULONG iSetCATEventCallback` (`tFNCATEvent` pCallback)
- `ULONG SetDeviceStateChangeCb` (`tFNDeviceStateChange` pCallback)
- `ULONG SetNetChangeCb` (`BYTE` instance, `tFNNet` pCallback, `ULONG` loMark, `ULONG` hiMark, `ULONG` period)
- `ULONG SetFwDldCompletionCb` (`tFNFwDldCompletion` pCallback)
- `ULONG SetSLQSOMADMAAlertCallback` (`tFNSLQSOMADMAAlert` pCallback)
- `ULONG SetSLQSOMADMAAlertCallbackExt` (`tFNSLQSOMADMAAlert` pCallback)
- `ULONG SetOMADMStateCallback` (`tFNOMADMState` pCallback)
- `ULONG SLQSSetServingSystemCallback` (`tFNServingSystem` pCallback)



- [ULONG SLQSSetBandPreferenceCbK](#) (tFNBandPreference pCallback)
- [ULONG SetUSSDReleaseCallback](#) (tFNUSSDRelease pCallback)
- [ULONG SetUSSDNotificationCallback](#) (tFNUSSDNotification pCallback)
- [ULONG SLQSSetSignalStrengthsCallback](#) (tFNSLQSSignalStrengths pCallback, struct [SLQSSignalStrengthsIndReq](#) \*pSLQSSignalStrengthsIndReq)
- [ULONG iSLQSSetSignalStrengthsCallback](#) (tFNSLQSSignalStrengths pCallback)
- [ULONG SLQSVoiceSetSUPSNotificationCallback](#) (tFNSUPSNotification pCallback)
- [ULONG SLQSSetSDKTerminatedCallback](#) (tFNSDKTerminated pCallback)
- [ULONG SLQSVoiceSetAllCallStatusCallBack](#) (tFNAIAllCallStatus pCallback)
- [ULONG SLQSSetTransLayerInfoCallback](#) (tFNtransLayerInfo pCallback)
- [ULONG SLQSSetTransNWRegInfoCallback](#) (tFNtransNWRegInfo pCallback)
- [ULONG SLQSSetSysSelectionPrefCallBack](#) (tFNSysSelectionPref pCallback)
- [ULONG SLQSUIMSetRefreshCallBack](#) (tFNUIMRefresh pCallback)
- [ULONG SLQSUIMSetStatusChangeCallBack](#) (tFNUIMStatusChangeInfo pCallback)
- [ULONG SLQSVoiceSetPrivacyChangeCallBack](#) (tFNPrivacyChange pCallback)
- [ULONG SLQSVoiceSetDTMFEventCallBack](#) (tFNDTMFEvent pCallback)
- [ULONG SLQSVoiceSetSUPSCallBack](#) (tFNSUPSInfo pCallback)
- [ULONG SLQSNasSysInfoCallBack](#) (tFNSysInfo pCallback)
- [ULONG SLQSNasNetworkTimeCallBack](#) (tFNNetworkTime pCallback)
- [ULONG SLQSWmsMemoryFullCallBack](#) (tFNMemoryFull pCallback)
- [ULONG SLQSVoiceSetOTASPStatusCallBack](#) (tFNOTASPStatus pCallback)
- [ULONG SLQSVoiceInfoRecCallback](#) (tFNInfoRec pCallback)
- [ULONG SLQSWmsMessageWaitingCallBack](#) (tFNMessageWaiting pCallback)
- [ULONG SLQSSetQosEventCallback](#) (BYTE instance, tFNSLQSQOSEvent pCallback)
- [ULONG SLQSSetQosStatusCallback](#) (BYTE instance, tFNQosStatus pCallback)
- [ULONG SLQSSetQosNWStatusCallback](#) (tFNQosNWStatus pCallback)
- [ULONG SLQSSetQosPriEventCallback](#) (tFNQosPriEvent pCallback)
- [ULONG SLQSNasSigInfoCallBack](#) (tFNSigInfo pCallback, sigInfo \*pSigInfo)
- [ULONG SLQSSetModemTempCallback](#) (tFNModemTempInfo pCallback)
- [ULONG SLQSSetPacketSrvStatusCallback](#) (tFNPacketSrvState pCallback)
- [ULONG SLQSSetSwtHDRPersCallback](#) (tFNHDRPersonality pCallback)
- [ULONG SLQSSetSIPConfigCallback](#) (tFNImSIPConfig pCallback)
- [ULONG SLQSSetRegMgrConfigCallback](#) (tFNImRegMgrConfig pCallback)
- [ULONG SLQSSetIMSSMSConfigCallback](#) (tFNImSMSConfig pCallback)
- [ULONG SLQSSetIMSUserConfigCallback](#) (tFNImUserConfig pCallback)
- [ULONG SLQSSetIMSVoIPConfigCallback](#) (tFNImVoIPConfig pCallback)
- [ULONG SetUSSDNoWaitIndicationCallback](#) (tFNUSSDNoWaitIndication pCallback)
- [ULONG SLQSSetDUNCallInfoCallback](#) (BYTE StatsPeriod, tFNDUNCallInfo pCallback)
- [ULONG iSLQSSetDUNCallInfoCallback](#) (tFNDUNCallInfo pCallback)
- [ULONG SLQSSetDataSystemStatusCallback](#) (tFNDataSysStatus pCallback)
- [ULONG SLQSWmsAsyncRawSendCallBack](#) (tFNAsyncRawSend pCallback)
- [ULONG SLQSNasSwtOTAMessageCallback](#) (NasSwtIndReg \*req, tFNASwtOTAMsg pCallback)
- [ULONG SetGPSCallback](#) (tFNNewGPS pCallback)
- [ULONG SetRMTransferStatisticsCallback](#) (tFNNewRMTransferStatistics pCallback)
- [ULONG SetLocCradleMountCallback](#) (tFNSetCradleMount pCallback)
- [ULONG SetLocEventTimeSyncCallback](#) (tFNSetEventTimeSync pCallback)
- [ULONG SetLocInjectTimeCallback](#) (tFNInjectTimeStatus pCallback)
- [ULONG SetLocSensorStreamingCallback](#) (tFNSensorStreaming pCallback)
- [ULONG SetLocInjectSensorDataCallback](#) (tFNInjectSensorData pCallback)
- [ULONG SetLocEventPositionCallback](#) (tFNEventPosition pCallback)
- [ULONG SetLocOpModeCallback](#) (tFNOpMode pCallback)
- [ULONG SLQSSetIMSARegStatusCallback](#) (tFNImsaRegStatus pCallback)
- [ULONG SLQSSetIMSASvcStatusCallback](#) (tFNImsaSvcStatus pCallback)
- [ULONG SLQSSetIMSARatStatusCallback](#) (tFNImsaRatStatus pCallback)
- [ULONG SLQSSetIMSAPdpStatusCallback](#) (tFNImsaPdpStatus pCallback)

- [ULONG SLQSNasSigInfo2Callback](#) (tFNSigInfo pCallback, [setSignalStrengthInfo](#) \*pSigInfo2)
- [ULONG SetLocGnssSvInfoCallback](#) (tFNGnssSvInfo pCallback)
- [ULONG SetLocDeleteAssistDataCallback](#) (tFNDeIAssistData pCallback)
- [ULONG SetNasLTECphyCaIndCallback](#) (tFNASwiLTECphyCalInfo pCallback)

### 9.6.1 Detailed Description

Callback Service API function prototypes.

### 9.6.2 Macro Definition Documentation

9.6.2.1 `#define CBK_DISABLE_EVENT 0x00`

9.6.2.2 `#define CBK_ENABLE_EVENT 0x01`

9.6.2.3 `#define CBK_NOCHANGE 0xFF`

9.6.2.4 `#define DEREGISTER_EVENT 0x00`

9.6.2.5 `#define DEREGISTER_SRV 0x00`

9.6.2.6 `#define EVENT_MASK_CARD 0x00000001`

9.6.2.7 `#define EVENT_MASK_DEREGISTER_ALL 0x00000000`

9.6.2.8 `#define FIRST_INSTANCE 0x00`

9.6.2.9 `#define INVALID_INSTACNE 0x08`

9.6.2.10 `#define IPV4 4`

9.6.2.11 `#define IPV4V6 7`

9.6.2.12 `#define IPV6 6`

9.6.2.13 `#define LOC_EVENT_MASK_ENG_STATE 0x00000080`

9.6.2.14 `#define LOC_EVENT_MASK_GNSS_SV_INFO 0x00000002`

9.6.2.15 `#define LOC_EVENT_MASK_INJECT_TIME 0x00000010`

9.6.2.16 `#define LOC_EVENT_MASK_SENSOR_STREAM 0x00000400`

9.6.2.17 `#define LOC_EVENT_MASK_TIME_SYNC 0x00000800`

9.6.2.18 `#define LOC_EVENT_POSITION_REPORT 0x00000001`

9.6.2.19 `#define MAX_NO_OF_APPLICATIONS 10`

9.6.2.20 `#define MAX_NO_OF_CALLS 20`

9.6.2.21 `#define MAX_NO_OF_FILES 255`

9.6.2.22 `#define MAX_NO_OF_SLOTS 5`

- 9.6.2.23 `#define MAX_NO_OF_UUSINFO 20`
- 9.6.2.24 `#define MAX_PATH_LENGTH 255`
- 9.6.2.25 `#define MAX_RADIO_INTERFACE_LIST 255`
- 9.6.2.26 `#define MAXUSSDLENGTH 182`
- 9.6.2.27 `#define NAS_SRV 0x02`
- 9.6.2.28 `#define NUM_OF_SET 0xFF`
- 9.6.2.29 `#define PDS_SRV 0x04`
- 9.6.2.30 `#define QMI_ETWS_MAX_PAYLOAD_LENGTH 1254 /* Qualcomm defined max */`
- 9.6.2.31 `#define QMI_MAX_VOICE_NUMBER_LENGTH 81`
- 9.6.2.32 `#define QMI_WMS_MAX_PAYLOAD_LENGTH 256`
- 9.6.2.33 `#define REGISTER_EVENT 0x01`
- 9.6.2.34 `#define REGISTER_SRV 0x01`
- 9.6.2.35 `#define SECOND_INSTANCE 0x01`
- 9.6.2.36 `#define SIGSTRENGTH_THRESHOLD_ARR_SZ 5`
- 9.6.2.37 `#define THIRD_INSTANCE 0x02`
- 9.6.2.38 `#define USSD_DCS_8BIT 0x02 /* 8-bit coding scheme */`
- 9.6.2.39 `#define USSD_DCS_ASCII 0x01 /* ASCII coding scheme */`
- 9.6.2.40 `#define USSD_DCS_UCS2 0x03 /* UCS2 coding scheme */`
- 9.6.2.41 `#define VOICE_SRV 0x08`
- 9.6.2.42 `#define WDS_SRV 0x01`

### 9.6.3 Typedef Documentation

#### 9.6.3.1 `typedef struct accelAcceptReady_s accelAcceptReady`

This structure contains Accelerometer Accept Ready Info

## Parameters

<i>injectEnable</i>	<ul style="list-style-type: none"> <li>GNSS location engine is ready to accept data from sensor.</li> <li>Values</li> <li>0x01 - Ready to accept sensor data</li> <li>0x00 - Not ready to accept sensor data</li> </ul>
<i>samplesPerBatch</i>	<ul style="list-style-type: none"> <li>number of samples per batch the GNSS location engine is to receive.</li> <li><math>\text{samplingFrequency} = \text{samplesPerBatch} * \text{batchesPerSecond}</math></li> <li><i>samplesPerBatch</i> must be a nonzero positive value.</li> </ul>
<i>batchPerSec</i>	<ul style="list-style-type: none"> <li>LTE NAS version minor</li> <li>Number of sensor-data batches the GNSS location engine is to receive per second.</li> <li><i>BatchesPerSecond</i> must be a nonzero positive value.</li> </ul>

9.6.3.2 `typedef struct accelTempAcceptReady_s accelTempAcceptReady`

This structure contains Accelerometer Temperature Accept Ready Info

## Parameters

<i>injectEnable</i>	<ul style="list-style-type: none"> <li>GNSS location engine is ready to accept data from sensor.</li> <li>Values</li> <li>0x01 - Ready to accept sensor data</li> <li>0x00 - Not ready to accept sensor data</li> </ul>
<i>samplesPerBatch</i>	<ul style="list-style-type: none"> <li>number of samples per batch the GNSS location engine is to receive.</li> <li><math>\text{samplingFrequency} = \text{samplesPerBatch} * \text{batchesPerSecond}</math></li> <li><i>samplesPerBatch</i> must be a nonzero positive value.</li> </ul>
<i>batchPerSec</i>	<ul style="list-style-type: none"> <li>LTE NAS version minor</li> <li>Number of sensor-data batches the GNSS location engine is to receive per second.</li> <li><i>BatchesPerSecond</i> must be a nonzero positive value.</li> </ul>

9.6.3.3 `typedef enum device_state_enum eDevState`

Device State enumeration

#### 9.6.3.4 typedef enum SMSEventType eSMSEventType

This enumeration defines the different type of SMS events that are received

#### 9.6.3.5 typedef struct gpsTime\_s gpsTime

This structure contains GPS Time info.

##### Parameters

<i>gpsWeek</i>	<ul style="list-style-type: none"> <li>• Current GPS week as calculated from midnight, Jan. 6, 1980.</li> <li>• Units - Weeks</li> </ul>
<i>gpsTimeOf-WeekMs</i>	<ul style="list-style-type: none"> <li>• Amount of time into the current GPS week.</li> <li>• Units - Milliseconds</li> </ul>

#### 9.6.3.6 typedef struct gyroAcceptReady\_s gyroAcceptReady

This structure contains Gyroscope Accept Ready Info

##### Parameters

<i>injectEnable</i>	<ul style="list-style-type: none"> <li>• GNSS location engine is ready to accept data from sensor.</li> <li>• Values</li> <li>• 0x01 - Ready to accept sensor data</li> <li>• 0x00 - Not ready to accept sensor data</li> </ul>
<i>samplesPer-Batch</i>	<ul style="list-style-type: none"> <li>• number of samples per batch the GNSS location engine is to receive.</li> <li>• <math>\text{samplingFrequency} = \text{samplesPerBatch} * \text{batchesPerSecond}</math></li> <li>• samplesPerBatch must be a nonzero positive value.</li> </ul>
<i>batchPerSec</i>	<ul style="list-style-type: none"> <li>• LTE NAS version minor</li> <li>• Number of sensor-data batches the GNSS location engine is to receive per second.</li> <li>• BatchesPerSecond must be a nonzero positive value.</li> </ul>

#### 9.6.3.7 typedef struct gyroTempAcceptReady\_s gyroTempAcceptReady

This structure contains Gyroscope Temperature Accept Ready Info

## Parameters

<i>injectEnable</i>	<ul style="list-style-type: none"> <li>GNSS location engine is ready to accept data from sensor.</li> <li>Values</li> <li>0x01 - Ready to accept sensor data</li> <li>0x00 - Not ready to accept sensor data</li> </ul>
<i>samplesPerBatch</i>	<ul style="list-style-type: none"> <li>number of samples per batch the GNSS location engine is to receive.</li> <li><math>\text{samplingFrequency} = \text{samplesPerBatch} * \text{batchesPerSecond}</math></li> <li><i>samplesPerBatch</i> must be a nonzero positive value.</li> </ul>
<i>batchPerSec</i>	<ul style="list-style-type: none"> <li>LTE NAS version minor</li> <li>Number of sensor-data batches the GNSS location engine is to receive per second.</li> <li><i>BatchesPerSecond</i> must be a nonzero positive value.</li> </ul>

9.6.3.8 typedef struct **LteNasReleaseInfo\_s** **LteNasReleaseInfo**

This structure contains LTE Nas Release Information

## Parameters

<i>nas_release</i>	<ul style="list-style-type: none"> <li>LTE NAS release</li> </ul>
<i>nas_major</i>	<ul style="list-style-type: none"> <li>LTE NAS version major</li> </ul>
<i>nas_minor</i>	<ul style="list-style-type: none"> <li>LTE NAS version minor</li> </ul>

9.6.3.9 typedef struct **\_modemTempNotification** **modemTempNotification**

Contains the parameters passed for `SLQSSetModemTempCallback` by the device.

## Parameters

<i>ModemTemp-State</i>	<ul style="list-style-type: none"> <li>• provides the temperature state of the modem</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0 - unknown</li> <li>– 1 - normal</li> <li>– 2 - high(warning)</li> <li>– 3 - high(critical)</li> <li>– 4 - low(critical)</li> </ul> </li> </ul>
<i>Modem-Temperature</i>	<ul style="list-style-type: none"> <li>• provides the temperature of the modem</li> </ul>

## Note

None

## 9.6.3.10 typedef struct \_packetSrvStatus packetSrvStatus

Contains the parameters passed for SLQSSetPacketSrvStatusCallback by the device.

## Parameters

<i>pQmiInterface-Info</i>	<ul style="list-style-type: none"> <li>• See <a href="#">qaQmiInterfaceInfo</a> for more information</li> </ul>
<i>connStatus</i>	<ul style="list-style-type: none"> <li>• Current Link Status <ul style="list-style-type: none"> <li>– 1 - Disconnected</li> <li>– 2 - Connected</li> <li>– 3 - Suspended</li> <li>– 4 - Authenticating</li> </ul> </li> </ul>
<i>reconfigReqd</i>	<ul style="list-style-type: none"> <li>• Indicates if the network interface on the host needs to be reconfigured <ul style="list-style-type: none"> <li>– 0 - No need to reconfigure</li> <li>– 1 - Reconfiguration required</li> </ul> </li> </ul>

<i>sessionEnd-Reason</i>	<ul style="list-style-type: none"> <li>• See <a href="#">qaGobiApiTableCallEndReasons.h</a> for Call End Reason, 0xFFFF means invalid value</li> </ul>
<i>verboseSessn-EndReasonType</i>	<ul style="list-style-type: none"> <li>• Call End Reason Type <ul style="list-style-type: none"> <li>– 0 - Unspecified</li> <li>– 1 - Mobile IP</li> <li>– 2 - Internal</li> <li>– 3 - Call Manager defined</li> <li>– 6 - 3GPP Specification defined</li> <li>– 7 - PPP</li> <li>– 8 - EHRPD</li> <li>– 9 - IPv6</li> <li>– 0xFFFF - invalid value</li> </ul> </li> </ul>
<i>verboseSessn-EndReason</i>	<ul style="list-style-type: none"> <li>• See <a href="#">qaGobiApiTableCallEndReasons.h</a> for verbose Call End Reason. The values depend on verboseSessnEndReasonType parameter 0xFFFF means invalid value</li> </ul>
<i>ipFamily</i>	<ul style="list-style-type: none"> <li>• IP Family of the packet data connection <ul style="list-style-type: none"> <li>– 4 - IPv4</li> <li>– 6 - IPv6</li> <li>– 0xFF - invalid value</li> </ul> </li> </ul>
<i>techName</i>	<ul style="list-style-type: none"> <li>• Technology name of the packet data connection. <ul style="list-style-type: none"> <li>– 32767 - CDMA</li> <li>– 32764 - UMTS</li> <li>– 30592 - EPC</li> <li>– 30590 - EMBMS</li> <li>– 30584 - Modem Link Local</li> <li>– 0xFFFF - invalid value EPC is a logical interface to support LTE/eHRPD handoff. Modem Link is an interface for transferring data between entities on the AP and modem.</li> </ul> </li> </ul>



<i>bearerID</i>	<ul style="list-style-type: none"> <li>• Bearer ID (3GPP) or RLP ID (3GPP2) of the packet data connection 0xFF means invalid value</li> </ul>
-----------------	---

**Note**

Any parameter not returned by the device is returned as its maximum unsigned value by the callback.

**9.6.3.11 typedef struct precisionDilution\_s precisionDilution**

This structure contains Dilution of precision associated with this position.

**Parameters**

<i>PDOP</i>	<ul style="list-style-type: none"> <li>• Position dilution of precision.</li> <li>• Range - 1 (highest accuracy) to 50 (lowest accuracy)</li> <li>• PDOP = square root of (Square of HDOP + Square of VDOP2 )</li> </ul>
<i>HDOP</i>	<ul style="list-style-type: none"> <li>• Horizontal dilution of precision.</li> <li>• Range - 1 (highest accuracy) to 50 (lowest accuracy)</li> </ul>
<i>VDOP</i>	<ul style="list-style-type: none"> <li>• Vertical dilution of precision.</li> <li>• Range- 1 (highest accuracy) to 50 (lowest accuracy)</li> </ul>

**9.6.3.12 typedef struct sensorDataUsage\_s sensorDataUsage**

This structure contains Sensor Data Usage info.

**Parameters**

<i>usageMask</i>	<ul style="list-style-type: none"> <li>• Specifies which sensors were used in calculating the position in the position report.</li> </ul>
------------------	---

- Value

- 0x00000001 - Accelerometer used
- 0x00000002 - Gyroscope used

**Parameters**

<i>aidingIndicator-Mask</i>	
-----------------------------	--

- Specifies which results were aided by sensors.

- Value

- 0x00000001 - AIDED\_HEADING

- 0x00000002 - AIDED\_SPEED
- 0x00000004 - AIDED\_POSITION
- 0x00000008 - AIDED\_VELOCITY

#### 9.6.3.13 typedef union sessionInfo sessionInformation

This union [sessionInfo](#) consist of [omaDmFotaTlv](#), [omaDmConfigTlv](#) and [omaDmNotificationsTlv](#), out of which one will be unpacked against pEventFields.

#### 9.6.3.14 typedef union sessionInfoExt sessionInformationExt

This union [sessionInfo](#) consist of [omaDmFotaTlv](#) and [omaDmConfigTlv](#), out of which one will be unpacked against pEventFields.

#### 9.6.3.15 typedef struct SMSAsyncRawSend\_s SMSAsyncRawSend

This structure contains SMS parameters

##### Parameters

<i>sendStatus</i>	<ul style="list-style-type: none"> <li>• Send Status</li> <li>• Values: <ul style="list-style-type: none"> <li>– QMI_ERR_NONE – No error in the request</li> <li>– QMI_ERR_CAUSE_CODE - SMS cause code</li> <li>– QMI_ERR_MESSAGE_DELIVERY_FAILURE - Message could not be delivered</li> <li>– QMI_ERR_NO_MEMORY - Device could not allocate memory to formulate a response</li> </ul> </li> </ul>
<i>messageID</i>	<ul style="list-style-type: none"> <li>• Unique ID assigned by WMS for non-retry messages.</li> </ul>
<i>causeCode</i>	<ul style="list-style-type: none"> <li>• WMS cause code</li> </ul>
<i>errorClass</i>	<ul style="list-style-type: none"> <li>• Error Class</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0x00 - ERROR_CLASS_TEMPORARY</li> <li>– 0x01 - ERROR_CLASS_PERMANENT</li> </ul> </li> </ul>

<i>RPCause</i>	<ul style="list-style-type: none"> <li>• GW RP cause</li> </ul>
<i>TPCause</i>	<ul style="list-style-type: none"> <li>• GW TP Cause</li> </ul>
<i>msgDelFailure-Type</i>	<ul style="list-style-type: none"> <li>• Message delivery failure type</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0x00 - WMS_MESSAGE_DELIVERY_FAILURE_TEMPORARY</li> <li>– 0x01 - WMS_MESSAGE_DELIVERY_FAILURE_PERMANENT</li> </ul> </li> </ul>
<i>msgDelFailure-Cause</i>	<ul style="list-style-type: none"> <li>• Message delivery failure cause</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0x00 - WMS_MESSAGE_BLOCKED_DUE_TO_CALL_CONTROL</li> </ul> </li> </ul>
<i>alphaIDLen</i>	<ul style="list-style-type: none"> <li>• Number of sets of the pAlphaID</li> </ul>
<i>pAlphaID</i>	<ul style="list-style-type: none"> <li>• Alpha ID</li> </ul>
<i>userData</i>	<ul style="list-style-type: none"> <li>• Identifies the request associated with this indication.</li> </ul>

#### 9.6.3.16 typedef struct SMSCAddress SMSCAddressInfo

This structure holds SMSC information

##### Parameters

<i>length</i>	<ul style="list-style-type: none"> <li>• Number of sets of following element</li> </ul>
<i>data</i>	<ul style="list-style-type: none"> <li>• SMSC address</li> </ul>

#### 9.6.3.17 typedef struct SMSEtwsMessage SMSEtwsMessageInfo

This structure holds information related earthquake and Tsunami warning system

**Parameters**

<i>notificationType</i>	<ul style="list-style-type: none"> <li>• Message mode 0x00 - Primary 0x01 - Secondary GSM 0x02 - Secondary UMTS</li> </ul>
<i>length</i>	<ul style="list-style-type: none"> <li>• Number of sets of following elements</li> </ul>
<i>data</i>	<ul style="list-style-type: none"> <li>• Raw message data</li> </ul>

**9.6.3.18 typedef struct SMSEtwsPlmn SMSEtwsPlmnInfo**

This structure holds information related ETWS PLMN

**Parameters**

<i>mobileCountry-Code</i>	<ul style="list-style-type: none"> <li>• 16 bit representation of MCC value range : 0 -999</li> </ul>
<i>mobileNetwork-Code</i>	<ul style="list-style-type: none"> <li>• 16 bit representation of MNC value range : 0 -999</li> </ul>

**9.6.3.19 typedef struct SMSEventInfo\_s SMSEventInfo**

This structure will hold the information related to received SMS events

**Parameters**

<i>smsEventType</i>	<ul style="list-style-type: none"> <li>• Type of the SMS events that are received. This is a bit map of <a href="#">SMSEventType</a>. Only the parameters (which follows) related to the events received would be filled, and the rest of the parameters would be NULL</li> </ul>
<i>pMTMessage-Info</i>	<ul style="list-style-type: none"> <li>• pointer to the <a href="#">SMSMTMessageInfo</a> structure NULL, if this event is not present in the smsEventType parameter</li> </ul>
<i>pTransferRoute-MTMessageInfo</i>	<ul style="list-style-type: none"> <li>• pointer to the <a href="#">SMSTransferRouteMTMessageInfo</a> structure . NULL, if this event is not present in the smsEventType parameter</li> </ul>
<i>pMessageMode-Info</i>	<ul style="list-style-type: none"> <li>• pointer to the <a href="#">SMSMessageModeInfo</a> structure NULL, if this event is not present in the smsEventType parameter</li> </ul>

<i>pEtwsMessage-Info</i>	<ul style="list-style-type: none"> <li>pointer to the <a href="#">SMSEtwsMessageInfo</a> structure</li> <li>NULL, if this event is not present in the smsEventType parameter</li> </ul>
<i>pEtwsPlmnInfo</i>	<ul style="list-style-type: none"> <li>pointer to the <a href="#">SMSEtwsPlmnInfo</a> structure</li> <li>NULL, if this event is not present in the smsEventType parameter</li> </ul>
<i>pSMSCAddress-Info</i>	<ul style="list-style-type: none"> <li>pointer to the <a href="#">SMSCAddressInfo</a> structure</li> <li>NULL, if this event is not present in the smsEventType parameter</li> </ul>
<i>pSMSOnIMSInfo</i>	<ul style="list-style-type: none"> <li>pointer to the <a href="#">SMSOnIMSInfo</a> structure</li> <li>NULL, if this event is not present in the smsEventType parameter Note: None</li> </ul>

#### 9.6.3.20 typedef struct SMSMessageMode SMSMessageModelInfo

This structure holds information related to message mode

##### Parameters

<i>messageMode</i>	<ul style="list-style-type: none"> <li>Message mode 0x00 - CDMA 0x01 - GW</li> </ul>
--------------------	--

#### 9.6.3.21 typedef struct SMSMTMessage SMSMTMessageInfo

This structure holds information related to MT SMS

##### Parameters

<i>storageType</i>	<ul style="list-style-type: none"> <li>SMS message storage type for the new message</li> <li>0 - UIM 1 - NV</li> </ul>
<i>messageIndex</i>	<ul style="list-style-type: none"> <li>Index of the new message</li> </ul>

#### 9.6.3.22 typedef struct SMSOnIMS SMSOnIMSInfo

This structure holds information related to message mode

## Parameters

<i>smsOnIMS</i>	<ul style="list-style-type: none"> <li>Indicates whether the message is received from IMS 0x00 - Message is not received from IMS 0x01 - Message is received from IMS 0x02-0xFF - Reserved Note: In multiple modem solutions, this TLV may be used to help the client determine with which modem to communicate. This TLV may not be supported on all implementations.</li> </ul>
-----------------	---

## 9.6.3.23 typedef struct SMSTransferRouteMTMessage SMSTransferRouteMTMessageInfo

This structure holds information related to transfer route MT SMS

## Parameters

<i>ackIndicator</i>	<ul style="list-style-type: none"> <li>Parameter to indicate if ACK must be sent by the control point 0x00 - Send ACK 0x01 - Do not send ACK</li> </ul>
<i>transactionID</i>	<ul style="list-style-type: none"> <li>Transaction ID of the message</li> </ul>
<i>format</i>	<ul style="list-style-type: none"> <li>Message format 0x00 - CDMA 0x02 - 0x05 - Reserved 0x06 - GW_PP 0x07 - GW_BC</li> </ul>
<i>length</i>	<ul style="list-style-type: none"> <li>Length of the raw message. This length should not exceed the maximum WMS payload length of 256 bytes</li> </ul>
<i>data</i>	<ul style="list-style-type: none"> <li>Raw message data</li> </ul>

## 9.6.3.24 typedef struct svUsedforFix\_s svUsedforFix

This structure contains SVs Used to Calculate the Fix.

## Parameters

<i>gnssSvUsedList- _len</i>	<ul style="list-style-type: none"> <li>Number of sets of gnssSvUsedList</li> </ul>
<i>pGnssSvUsed- List</i>	<ul style="list-style-type: none"> <li>Entry in the list contains the <a href="#">SV</a> ID of a satellite used for calculating this position report.</li> <li>Following information is associated with each <a href="#">SV</a> ID: <ul style="list-style-type: none"> <li>GPS - 1 to 32</li> <li>SBAS - 33 to 64</li> <li>GLONASS - 65 to 96</li> <li>QZSS - 193 to 197</li> <li>BDS - 201 to 237</li> </ul> </li> </ul>

## 9.6.3.25 typedef struct SwiOTAMsg\_s SwiOTAMsg

This structure contains OTA message

## Parameters

<i>type</i>	<ul style="list-style-type: none"><li>• message type<ul style="list-style-type: none"><li>– 0 - LTE ESM uplink</li><li>– 1 - LTE ESM downlink</li><li>– 2 - LTE EMM uplink</li><li>– 3 - LTE EMM downlink</li><li>– 4 - GSM/UMTS uplink</li><li>– 5 - GSM/UMTS downlink</li></ul></li></ul>
<i>data_len</i>	<ul style="list-style-type: none"><li>• OTA Message Content Length</li></ul>
<i>data</i>	<ul style="list-style-type: none"><li>• OTA Message Content</li></ul>
<i>pLteNasRelInfo</i>	<ul style="list-style-type: none"><li>• LTE NAS Release Info</li><li>• see <a href="#">LteNasReleaseInfo</a> for details</li></ul>
<i>pTime</i>	<ul style="list-style-type: none"><li>• Seconds in local time since Jan. 6th 1980 00:00:00 UTC</li></ul>

## 9.6.3.26 typedef void(\* tFNActivationStatus)(ULONG activationStatus)

Activation status callback function.

## Parameters

<i>activationStatus</i>	<ul style="list-style-type: none"> <li>• Service Activation Code <ul style="list-style-type: none"> <li>– 0 - Service not activated</li> <li>– 1 - Service activated</li> <li>– 2 - Activation connecting</li> <li>– 3 - Activation connected</li> <li>– 4 - OTASP security authenticated</li> <li>– 5 - OTASP NAM downloaded</li> <li>– 6 - OTASP MDN downloaded</li> <li>– 7 - OTASP IMSI downloaded</li> <li>– 8 - OTASP PRL downloaded</li> <li>– 9 - OTASP SPC downloaded</li> <li>– 10 - OTASP settings committed</li> </ul> </li> </ul>
-------------------------	--

9.6.3.27 `typedef void(* tFNAllCallStatus)(voiceSetAllCallStatusCbInfo *pVoiceSetAllCallStatusCbInfo)`

Voice Call Status Callback function. This function pointer will be executed to process received Indication.

## Parameters

<i>pVoiceSetAll- CallStatusCb- Info</i>	<ul style="list-style-type: none"> <li>• Call back will populated memory pointed by this parameter when a call is originated, connected, or ended. See <a href="#">voiceSetAllCallStatusCbInfo</a> for more information.</li> </ul>
---	---

9.6.3.28 `typedef void(* tFNASwiLTECphyCalInfo)(QmiCbK NasLTECphyCalInfo *pQmiCbK NasLTECphyCalInfo)`

LTE CPHY CA message callback function.

## Parameters

<i>pQmiCbK NasLT- ECphyCalInfo[O- UT]</i>	<ul style="list-style-type: none"> <li>• Events related to NAS, see <a href="#">QmiCbK NasLTECphyCalInfo</a> for details.</li> </ul>
---	--

9.6.3.29 `typedef void(* tFNASwiOTAMsg)(SwiOTAMsg *pSwiOTAMsg)`

OTA message callback function.

## Parameters

<i>pSwiOTAMsg[O- UT]</i>	<ul style="list-style-type: none"> <li>• Events related to NAS, see <a href="#">SwiOTAMsg</a> for details</li> </ul>
------------------------------	--



9.6.3.30 `typedef void(* tFNAsyncRawSend)(SMSAsyncRawSend *pSMSAsyncRawSend)`

SMS event related callback function.

## Parameters

<i>pSMSEventInfo[OUT]</i>	<ul style="list-style-type: none"><li>• Events related to SMS, see <a href="#">SMSEventInfo</a> for details</li></ul>
---------------------------	---

9.6.3.31 `typedef void(* tFNBandPreference)(ULONGLONG band_pref)`

Band Preference Callback function

## Parameters

<i>pBandPref</i>	<p>- Bit mask representing the current band preference Bit position meanings:</p> <ul style="list-style-type: none"> <li>• 0 - BC0_A - Band Class 0, A-System</li> <li>• 1 - BC0_B - Band Class 0, B-System, Band Class 0 AB , GSM 850 Band</li> <li>• 2 - BC1 - Band Class 1, all blocks</li> <li>• 3 - BC2 - Band Class 2 place holder</li> <li>• 4 - BC3 - Band Class 3, A-System</li> <li>• 5 - BC4 - Band Class 4, all blocks</li> <li>• 6 - BC5 - Band Class 5, all blocks</li> <li>• 7 - GSM_DCS_1800 - GSM DCS band</li> <li>• 8 - GSM_EGSM_900 - GSM Extended GSM (E-GSM) band</li> <li>• 9 - GSM_PGSM_900 - GSM Primary GSM (P-GSM) band</li> <li>• 10 - BC6 - Band Class 6</li> <li>• 11 - BC7 - Band Class 7</li> <li>• 12 - BC8 - Band Class 8</li> <li>• 13 - BC9 - Band Class 9</li> <li>• 14 - BC10 - Band Class 10</li> <li>• 15 - BC11 - Band Class 11</li> <li>• 16 - GSM_450 - GSM 450 band</li> <li>• 17 - GSM_480 - GSM 480 band</li> <li>• 18 - GSM_750 - GSM 750 band</li> <li>• 19 - GSM_850 - GSM 850 band</li> <li>• 20 - GSM_RGSM_900 - GSM Railways GSM Band</li> <li>• 21 - GSM_PCS_1900 - GSM PCS band</li> <li>• 22 - WCDMA_I_IMT_2000 - WCDMA EUROPE JAPAN and CHINA IMT 2100 band</li> <li>• 23 - WCDMA_II_PCS_1900 - WCDMA US PCS 1900 band</li> <li>• 24 - WCDMA_III_1700 - WCDMA EUROPE and CHINA DCS 1800 band</li> <li>• 25 - WCDMA_IV_1700 - WCDMA US 1700 band</li> <li>• 26 - WCDMA_V_850 - WCDMA US 850 band</li> <li>• 27 - WCDMA_VI_800 - WCDMA JAPAN 800 band</li> <li>• 28 - BC12 - Band Class 12</li> <li>• 29 - BC14 - Band Class 14</li> <li>• 30 - RESERVED_2 - Reserved 2</li> <li>• 31 - BC15 - Band Class 15</li> <li>• 32 - 47 - Reserved</li> <li>• 48 - WCDMA_VII_2600 - WCDMA EUROPE 2600 band</li> <li>• 49 - WCDMA_VIII_900 - WCDMA EUROPE and JAPAN 900 band</li> <li>• 50 - WCDMA_IX_1700 - WCDMA JAPAN 1700 band</li> <li>• 51 to 55 - Reserved</li> </ul>
------------------	---

**Note**

Timeout: NA To set the band preference the API [SLQSSetBandPreference\(\)](#) should be used

**9.6.3.32** typedef void(\* tFNCATEvent)(ULONG eventID, ULONG eventLen, BYTE \*pEventData)

CAT event callback function.

**Parameters**

<i>eventID</i>	<ul style="list-style-type: none"> <li>• Event ID <ul style="list-style-type: none"> <li>– 16 - Display Text</li> <li>– 17 - Get In-Key</li> <li>– 18 - Get Input</li> <li>– 19 - Setup Menu</li> <li>– 20 - Select Item</li> <li>– 21 - Send SMS - Alpha Identifier</li> <li>– 22 - Setup Event List</li> <li>– 23 - Setup Idle Mode Text</li> <li>– 24 - Language Notification</li> <li>– 25 - Refresh</li> <li>– 26 - End Proactive Session</li> </ul> </li> </ul>
<i>eventLen</i>	<ul style="list-style-type: none"> <li>• Length of pData (in bytes)</li> </ul>
<i>pEventData</i>	<ul style="list-style-type: none"> <li>• Data specific to the CAT event ID See <a href="#">currentCatEvent</a> for details</li> </ul>

**Note**

Technology Supported: UMTS

**9.6.3.33** typedef void(\* tFNDataCapabilities)(BYTE dataCapsSize, BYTE \*pDataCaps)

Serving system data capabilities callback function.

## Parameters

<i>dataCapsSize</i>	<ul style="list-style-type: none"> <li>• Number of elements the data capability array contains</li> </ul>
<i>pDataCaps</i>	<ul style="list-style-type: none"> <li>• Data Capabilities Array. <ul style="list-style-type: none"> <li>– 1 - GPRS</li> <li>– 2 - EDGE</li> <li>– 3 - HSDPA</li> <li>– 4 - HSUPA</li> <li>– 5 - WCDMA</li> <li>– 6 - CDMA 1xRTT</li> <li>– 7 - CDMA 1xEV-DO Rev 0</li> <li>– 8 - CDMA 1xEV-DO Rev. A</li> <li>– 9 - GSM</li> <li>– 10 - EVDO Rev. B</li> <li>– 11 - LTE</li> <li>– 12 - HSDPA Plus</li> <li>– 13 - Dual Carrier HSDPA Plus</li> </ul> </li> </ul>

## 9.6.3.34 typedef void(\* tFNDataSysStatus)(CurrDataSysStat \*pCurrDataSysStat)

Data System Status callback.

## Parameters

<i>pCurrDataSys-Stat</i>	<ul style="list-style-type: none"> <li>• See <a href="#">CurrDataSysStat</a> for more information.</li> </ul>
--------------------------	---

## 9.6.3.35 typedef void(\* tFNDeAssistData)(delAssistDataStatus \*pAssistDataNotification)

Delete Assist Data Notification callback.

## Parameters

<i>pAssistData-Notification</i>	<ul style="list-style-type: none"> <li>• See <a href="#">delAssistDataStatus</a> for more information.</li> </ul>
---------------------------------	---

## 9.6.3.36 typedef void(\* tFNDeviceStateChange)(eDevState device\_state)

Device State Change callback function prototype

## Parameters

<i>device_state</i>	<ul style="list-style-type: none"> <li>the current state of the device</li> </ul>
---------------------	---

## Note

Does not require communication with the device

### 9.6.3.37 typedef void( \* tFNDTMFEvent)(voiceDTMFEventInfo \*pVoiceDTMFEventInfo)

Preferred DTMF event indication callback.

## Parameters

<i>pVoiceDTMF-EventInfo</i>	<ul style="list-style-type: none"> <li>See <a href="#">voiceDTMFEventInfo</a> for more information.</li> </ul>
-----------------------------	--

### 9.6.3.38 typedef void( \* tFNDUNCallInfo)(DUNCallInfoInd \*pDUNCallInfo)

DUN Call Info indication callback.

## Parameters

<i>pDUNCallInfo</i>	<ul style="list-style-type: none"> <li>See <a href="#">DUNCallInfoInd</a> for more information.</li> </ul>
---------------------	--

### 9.6.3.39 typedef void(\* tFNEventPosition)(QmiCbkLocPositionReportInd \*pLocPositionReport)

### 9.6.3.40 typedef void(\* tFNFwDldCompletion)(ULONG fwdld\_completion\_status)

Firmware Download Completion callback function prototype

## Parameters

<i>error_code</i>	<ul style="list-style-type: none"> <li>error code returned from firmware download operation</li> </ul>
-------------------	--

## Note

Does not require communication with the device

### 9.6.3.41 typedef void(\* tFNGnssSvInfo)(gnssSvInfoNotification \*pGnssSvInfoNotification)

GNSS SVN Information Notification callback.

## Parameters

<i>pGnssSvInfo-Notification</i>	<ul style="list-style-type: none"> <li>See <a href="#">gnssSvInfoNotification</a> for more information.</li> </ul>
---------------------------------	--

**9.6.3.42** `typedef void( * tFNHDRPersonaity)(HDRPersonalityInd *pHDRPers)`

HDR Personality indication callback.

## Parameters

<i>pHDRPers</i>	<ul style="list-style-type: none"> <li>See <a href="#">HDRPersonalityInd</a> for more information.</li> </ul>
-----------------	---

## Note

Technology Supported: CDMA

**9.6.3.43** `typedef void( * tFNImsaPdpStatus)(imsaPdpStatusInfo *plmsaPdpStatusInfo)`

IMSA PDP status indication callback.

## Parameters

<i>plmsaPdp-StatusInfo</i>	<ul style="list-style-type: none"> <li>See <a href="#">imsaPdpStatusInfo</a> for more information.</li> </ul>
----------------------------	---

**9.6.3.44** `typedef void( * tFNImsaRatStatus)(imsaRatStatusInfo *plmsaRatStatusInfo)`

IMSA RAT handover status indication callback.

## Parameters

<i>plmsaRatStatus-Info</i>	<ul style="list-style-type: none"> <li>See <a href="#">imsaRatStatusInfo</a> for more information.</li> </ul>
----------------------------	---

**9.6.3.45** `typedef void( * tFNImsaRegStatus)(imsaRegStatusInfo *plmsaRegStatusInfo)`

IMSA Registration Status indication callback.

## Parameters

<i>plmsaReg-StatusInfo</i>	<ul style="list-style-type: none"> <li>See <a href="#">imsaRegStatusInfo</a> for more information.</li> </ul>
----------------------------	---

**9.6.3.46** `typedef void( * tFNImsaSvcStatus)(imsaSvcStatusInfo *plmsaSvcStatusInfo)`

IMSA Service Status indication callback.

## Parameters

<i>plmsaSvcStatus-Info</i>	<ul style="list-style-type: none"> <li>See <a href="#">imsaSvcStatusInfo</a> for more information.</li> </ul>
----------------------------	---

9.6.3.47 `typedef void( * tFNImRegMgrConfig)(imsRegMgrConfigInfo *plmsRegMgrConfigInfo)`

IMS Reg Mgr Config indication callback.

## Parameters

<i>plmsRegMgr-ConfigInfo</i>	<ul style="list-style-type: none"> <li>See <a href="#">imsRegMgrConfigInfo</a> for more information.</li> </ul>
------------------------------	---

9.6.3.48 `typedef void( * tFNImSIPConfig)(imsSIPConfigInfo *plmsSIPConfigInfo)`

IMS SIP Config indication callback.

## Parameters

<i>plmsSIPConfig-Info</i>	<ul style="list-style-type: none"> <li>See <a href="#">imsSIPConfigInfo</a> for more information.</li> </ul>
---------------------------	--

9.6.3.49 `typedef void( * tFNImSMSConfig)(imsSMSConfigInfo *plmsSMSConfigInfo)`

IMS SMS Config indication callback.

## Parameters

<i>plmsSMSConfig-Info</i>	<ul style="list-style-type: none"> <li>See <a href="#">imsSMSConfigInfo</a> for more information.</li> </ul>
---------------------------	--

9.6.3.50 `typedef void( * tFNImUserConfig)(imsUserConfigInfo *plmsUserConfigInfo)`

IMS User Config indication callback.

## Parameters

<i>plmsUserConfig-Info</i>	<ul style="list-style-type: none"> <li>See <a href="#">imsUserConfigInfo</a> for more information.</li> </ul>
----------------------------	---

9.6.3.51 `typedef void( * tFNImVoIPConfig)(imsVoIPConfigInfo *plmsVoIPConfigInfo)`

IMS VoIP Config indication callback.



## Parameters

<i>plmsVoIPConfig-Info</i>	<ul style="list-style-type: none"> <li>See <a href="#">imsVoIPConfigInfo</a> for more information.</li> </ul>
----------------------------	---

9.6.3.52 `typedef void(* tFNInfoRec)(voiceInfoRec *pVoiceInfoRec)`

Voice Information Record callback.

## Parameters

<i>pVoiceInfoRec</i>	<ul style="list-style-type: none"> <li>See <a href="#">voiceInfoRec</a> for more information.</li> </ul>
----------------------	--

## Note

Technology Supported: CDMA  
Device Supported: MC7750

9.6.3.53 `typedef void(* tFNInjectSensorData)(QmiCbkLocInjectSensorDataInd *pLocInjectSensorData)`

9.6.3.54 `typedef void(* tFNInjectTimeStatus)(QmiCbkLocInjectTimeInd *pLocInjectTime)`

9.6.3.55 `typedef void(* tFNLUReject)(ULONG serviceDomain, ULONG rejectCause)`

LU reject callback function.

## Parameters

<i>serviceDomain</i>	<ul style="list-style-type: none"> <li>Service domain 1 - Circuit Switched</li> </ul>
<i>rejectCause</i>	<ul style="list-style-type: none"> <li>Reject cause See 3GPP TS 24.008, Section 4.4.4.7 See <a href="#">qaGobiApiTableCallEndReasons.h</a> for Call End reasons</li> </ul>

## Note

Technology Supported: UMTS

9.6.3.56 `typedef void(* tFNMemoryFull)(SMSMemoryInfo *pSMSMemoryFullInfo)`

SMS Memory related callback function.

## Parameters

<i>pSMSMemory-FullInfo[OUT]</i>	<ul style="list-style-type: none"> <li>• pointer to <a href="#">SMSMemoryInfo</a>.</li> <li>• see <a href="#">SMSMemoryInfo</a> for details.</li> </ul>
---------------------------------	---

#### 9.6.3.57 typedef void(\* tFNMessageWaiting)(msgWaitingInfo \*pSMSMessageWaitingInfo)

SMS Memory related callback function.

##### Parameters

<i>pSMSMessage-WaitingInfo[OUT]</i>	<ul style="list-style-type: none"> <li>• pointer to <a href="#">msgWaitingInfo</a>.</li> <li>• see <a href="#">msgWaitingInfo</a> for details.</li> </ul>
-------------------------------------	---

#### 9.6.3.58 typedef void(\* tFNMobileIPStatus)(ULONG mipStatus)

Mobile IP status callback function.

##### Parameters

<i>mipStatus</i>	<ul style="list-style-type: none"> <li>• Mobile IP Status <ul style="list-style-type: none"> <li>– 0 - success</li> <li>– All others error codes as defined in RFC 2002 See <a href="#">qaGobiApiTableCallEndReasons.h</a> for mobile IP error codes</li> </ul> </li> </ul>
------------------	---

#### 9.6.3.59 typedef void(\* tFNModemTempInfo)(modemTempNotification \*pModemTempNotification)

Modem Temperature Information callback.

##### Parameters

<i>pModemTemp-Notification</i>	<ul style="list-style-type: none"> <li>• See <a href="#">modemTempNotification</a> for more information.</li> </ul>
--------------------------------	---

#### 9.6.3.60 typedef void(\* tFNNet)(ULONG q\_depth, BYTE isThrottle, BYTE instanceId)

Transmit Queue Length Change callback function prototype

##### Parameters

<i>q_depth</i>	<ul style="list-style-type: none"> <li>• transmit queue length</li> </ul>
----------------	---

<i>isThrottle</i>	<ul style="list-style-type: none"> <li>• 0: unthrottle</li> <li>• 1: throttle</li> </ul>
<i>instanceId</i>	<ul style="list-style-type: none"> <li>• qmi instance id</li> </ul>

**Note**

Does not require communication with the device

**9.6.3.61 typedef void( \* tFNNetworkTime)(nasNetworkTime \*pNasNetworkTime)**

Network Time indication callback.

**Parameters**

<i>pNasNetworkTime</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nasNetworkTime</a> for more information.</li> </ul>
------------------------	--

**9.6.3.62 typedef void(\* tFNNewGPS)(double dLongitude, double dLatitude)**

Set Current Location Data.

**Parameters**

<i>dLongitude[IN]</i>	<ul style="list-style-type: none"> <li>• Current Longitude Value</li> </ul>
<i>dLatitude[IN]</i>	<ul style="list-style-type: none"> <li>• Current Latitude Value</li> </ul>

**9.6.3.63 typedef void(\* tFNNewNMEA)(LPCSTR pNMEA)**

New NMEA sentence callback function.

**Parameters**

<i>pNMEA</i>	<ul style="list-style-type: none"> <li>• NULL-terminated string containing the position data in NMEA sentence format</li> </ul>
--------------	---

**9.6.3.64 typedef void(\* tFNNewRMTransferStatistics)(QmiCbKWdsStatisticsIndState \*pMsg)**

PDS session state callback function.

## Parameters

<i>enabledStatus</i>	<ul style="list-style-type: none"> <li>• GPS enabled status <ul style="list-style-type: none"> <li>– 0 - Disable</li> <li>– 1 - Enable</li> </ul> </li> </ul>
<i>trackingStatus</i>	<ul style="list-style-type: none"> <li>• GPS tracking status <ul style="list-style-type: none"> <li>– 0 - Unknown</li> <li>– 1 - Inactive</li> <li>– 2 - Active RM Transfer Statistics message callback function.</li> </ul> </li> </ul>
<i>pMsg[OUT]</i>	<ul style="list-style-type: none"> <li>• Events related to NAS, see <a href="#">QmiCbkWdsStatisticsIndState</a> for details</li> </ul>

9.6.3.65 `typedef void(* tFNNewSMS)(ULONG storageType, ULONG messageIndex)`

New SMS message callback function.

## Parameters

<i>storageType</i>	<ul style="list-style-type: none"> <li>• SMS message storage type for the new message <ul style="list-style-type: none"> <li>0 - UIM 1 - NV</li> </ul> </li> </ul>
<i>messageIndex</i>	<ul style="list-style-type: none"> <li>• Index of the new message</li> </ul>

9.6.3.66 `typedef void(* tFNOMADMState)(ULONG sessionState, ULONG failureReason)`

OMA-DM state callback function

## Parameters

<i>sessionState</i>	<ul style="list-style-type: none"> <li>• Session state <ul style="list-style-type: none"> <li>– 0x00 - Complete, information was updated</li> <li>– 0x01 - Complete, update information is unavailable</li> <li>– 0x02 - Failed</li> <li>– 0x03 - Retrying</li> <li>– 0x04 - Connecting</li> <li>– 0x05 - Connected</li> <li>– 0x06 - Authenticated</li> <li>– 0x07 - Mobile Directory Number (MDN) downloaded</li> <li>– 0x08 - Mobile Station Identifier (MSID) downloaded</li> <li>– 0x09 - PRL downloaded</li> <li>– 0x0A - Mobile IP (MIP) profile downloaded</li> </ul> </li> </ul>
<i>failureReason</i>	<ul style="list-style-type: none"> <li>• Session failure reason (when state indicates failure) <ul style="list-style-type: none"> <li>– 0x00 - Unknown</li> <li>– 0x01 - Network is unavailable</li> <li>– 0x02 - Server is unavailable</li> <li>– 0x03 - Authentication failed</li> <li>– 0x04 - Maximum retry exceeded</li> <li>– 0x05 - Session is cancelled</li> </ul> </li> </ul>

## Note

Technology Supported: CDMA

9.6.3.67 `typedef void(* tFNOpMode)(ULONG mode)`

9.6.3.68 `typedef void( * tFNOTASPStatus)(voiceOTASPStatusInfo *pVoiceOTASPStatusInfo)`

OTASP or OTAPA event Indication Callback function

## Parameters

<i>pVoiceOTASP-StatusInfo</i>	<ul style="list-style-type: none"> <li>• OTASP Status Information.</li> <li>• See <a href="#">voiceOTASPStatusInfo</a> for more information</li> </ul>
-------------------------------	--

## Note

Technology Supported: CDMA

9.6.3.69 `typedef void(* tFNPacketSrvState)(packetSrvStatus *pPacketSrvStatus)`

Packet Service state callback function.

## Parameters

<i>pPacketSrv-Status</i>	<ul style="list-style-type: none"> <li>• See <a href="#">packetSrvStatus</a> for more details</li> </ul>
--------------------------	--

## 9.6.3.70 typedef void(\* tFNPDSState)(ULONG enabledStatus, ULONG trackingStatus)

PDS session state callback function.

## Parameters

<i>enabledStatus</i>	<ul style="list-style-type: none"> <li>• GPS enabled status <ul style="list-style-type: none"> <li>– 0 - Disable</li> <li>– 1 - Enable</li> </ul> </li> </ul>
<i>trackingStatus</i>	<ul style="list-style-type: none"> <li>• GPS tracking status <ul style="list-style-type: none"> <li>– 0 - Unknown</li> <li>– 1 - Inactive</li> <li>– 2 - Active</li> </ul> </li> </ul>

## 9.6.3.71 typedef void(\* tFNPower)(ULONG operatingMode)

Power operating mode callback function.

## Parameters

<i>operatingMode</i>	<ul style="list-style-type: none"> <li>• Service Operating mode See <a href="#">Tables</a> for Operating Modes</li> </ul>
----------------------	---

## Note

Technology Supported: UMTS/CDMA  
Device Supported: MC83x5, MC7700/50

## 9.6.3.72 typedef void( \* tFNPrivacyChange)(voicePrivacyInfo \*pVoicePrivacyInfo)

Preferred voice privacy indication callback.

## Parameters

<i>pVoicePrivacy-Info</i>	<ul style="list-style-type: none"> <li>• See <a href="#">voicePrivacyInfo</a> for more information.</li> </ul>
---------------------------	--

## Note

Technology Supported: CDMA

### 9.6.3.73 typedef void(\* tFNQosNWStatus)(BYTE status)

QOS Network status callback function.

#### Parameters

<i>status</i>	Network QoS support status <ul style="list-style-type: none"> <li>• 0x00 – Current network does not support QoS</li> <li>• 0x01 – Current network supports QoS</li> </ul>
---------------	---

### 9.6.3.74 typedef void(\* tFNQosPriEvent)(WORD event)

QOS primary flow callback function.

#### Parameters

<i>event</i>	Event which causes this indication: <ul style="list-style-type: none"> <li>• 0x0001 – Primary flow QoS modify operation success</li> <li>• 0x0002 – Primary flow QoS modify operation failure</li> </ul>
--------------	--

### 9.6.3.75 typedef void(\* tFNQosStatus)(BYTE instance, ULONG id, BYTE status, BYTE event, BYTE reason)

QOS Status callback function.

#### Parameters

<i>instance</i>	<ul style="list-style-type: none"> <li>• QMI instance</li> </ul>
<i>id</i>	<ul style="list-style-type: none"> <li>• Index identifying the QoS flow whose status is being reported</li> </ul>
<i>status</i>	Current QoS flow status: <ul style="list-style-type: none"> <li>• 0x01 – QMI_QOS_STATUS_ACTIVATED</li> <li>• 0x02 – QMI_QOS_STATUS_SUSPENDED</li> <li>• 0x03 – QMI_QOS_STATUS_GONE</li> </ul>



<i>event</i>	<ul style="list-style-type: none"> <li>• 0x01 – QMI_QOS_ACTIVATED_EV</li> <li>• 0x02 – QMI_QOS_SUSPENDED_EV</li> <li>• 0x03 – QMI_QOS_GONE_EV</li> <li>• 0x04 – QMI_QOS_MODIFY_ACCEPTED_EV</li> <li>• 0x05 – QMI_QOS_MODIFY_REJECTED_EV</li> <li>• 0x06 – QMI_QOS_INFO_CODE_UPDATED_EV</li> </ul>
<i>reason</i>	<ul style="list-style-type: none"> <li>• 0x01 - QMI_QOS_INVALID_PARAMS</li> <li>• 0x02 - QMI_QOS_INTERNAL_CALL_ENDED</li> <li>• 0x03 - QMI_QOS_INTERNAL_ERROR</li> <li>• 0x04 - QMI_QOS_INSUFFICIENT_LOCAL_Resources</li> <li>• 0x05 - QMI_QOS_TIMED_OUT_OPERATION</li> <li>• 0x06 - QMI_QOS_INTERNAL_UNKNOWN_CAUSE_CODE</li> <li>• 0x07 - QMI_QOS_INTERNAL_MODIFY_IN_PROGRESS</li> <li>• 0x08 - QMI_QOS_NOT_SUPPORTED</li> <li>• 0x09 - QMI_QOS_NOT_AVAILABLE</li> <li>• 0x0A - QMI_QOS_NOT_GUARANTEED</li> <li>• 0x0B - QMI_QOS_INSUFFICIENT_NETWORK_RESOURCES</li> <li>• 0x0C - QMI_QOS_AWARE_SYSTEM</li> <li>• 0x0D - QMI_QOS_UNAWARE_SYSTEM</li> <li>• 0x0E - QOS_REJECTED_OPERATION</li> <li>• 0x0F - QMI_QOS_WILL_GRANT_WHEN_QOS_RESUMED</li> <li>• 0x10 - QMI_QOS_NETWORK_CALL_ENDED</li> <li>• 0x11 - QMI_QOS_NETWORK_SERVICE_NOT_AVAILABLE</li> <li>• 0x12 - QMI_QOS_NETWORK_L2_LINK_RELEASED</li> <li>• 0x13 - QMI_QOS_NETWORK_L2_LINK_REESTAB_REJ</li> <li>• 0x14 - QMI_QOS_NETWORK_L2_LINK_REESTAB_IND</li> <li>• 0x15 - QMI_QOS_NETWORK_UNKNOWN_CAUSE_CODE</li> <li>• 0x16 - QMI_NETWORK_BUSY</li> </ul>

**9.6.3.76** `typedef void(* tFNRFInfo)(ULONG radiolInterface, ULONG activeBandClass, ULONG activeChannel)`

RF information callback function.

## Parameters

<i>radioInterface</i>	<ul style="list-style-type: none"> <li>Radio interface technology of the signal being measured See <a href="#">Tables</a> for Radio Interface</li> </ul>
<i>activeBandClass</i>	<ul style="list-style-type: none"> <li>Active band class See <a href="#">Tables</a> for Active Band Class</li> </ul>
<i>activeChannel</i>	<ul style="list-style-type: none"> <li>Active channel <ul style="list-style-type: none"> <li>0 - Channel is not relevant to the reported technology</li> </ul> </li> </ul>

9.6.3.77 `typedef void(* tFNRoamingIndicator)(ULONG roaming)`

Roaming indicator callback function.

## Parameters

<i>roaming</i>	<ul style="list-style-type: none"> <li>Roaming Indication <ul style="list-style-type: none"> <li>0 - Roaming</li> <li>1 - Home</li> <li>2 - Roaming partner</li> <li>&gt;2 - Operator defined values</li> </ul> </li> </ul>
----------------	---

9.6.3.78 `typedef void(* tFNSDKTerminated)(BYTE *psReason)`

SDK terminated callback function prototype

## Parameters

<i>psReason</i>	<ul style="list-style-type: none"> <li>sdk termination reason string</li> </ul>
-----------------	---

## Note

Timeout: None

Does not require communication with the device

9.6.3.79 `typedef void(* tFNSensorStreaming)(QmiCbkLocSensorStreamingInd *pLocSensorStream)`9.6.3.80 `typedef void(* tFNServingSystem)(struct ServingSystemInfo *pServingSystem, struct RoamingInfo *pRoamingInfo)`

Serving System callback function

## Parameters

<i>pServingSystem</i>	<ul style="list-style-type: none"><li>• <a href="#">ServingSystemInfo</a> structure</li></ul>
-----------------------	---

9.6.3.81 `typedef void(* tFNSetCradleMount)(QmiCbkLocCradleMountInd *pSetLocCradleMount)`

9.6.3.82 `typedef void(* tFNSetEventTimeSync)(QmiCbkLocEventTimeSyncInd *pSetLocEventTimeSync)`

9.6.3.83 `typedef void(* tFNSigInfo)(nasSigInfo *pNasSigInfo)`

Signal Strength Information indication callback.

## Parameters

<i>pNasSigInfo</i>	<ul style="list-style-type: none"><li>• See <a href="#">nasSigInfo</a> for more information.</li></ul>
--------------------	--

9.6.3.84 `typedef void(* tFNSignalStrength)(INT8 signalStrength, ULONG radioInterface)`

Signal strength callback function.

## Parameters

<i>signalStrength</i>	<ul style="list-style-type: none"><li>• Received signal strength (in dBm)</li></ul>
<i>radioInterface</i>	<ul style="list-style-type: none"><li>• Radio interface technology of the signal being measured See <a href="#">Tables</a> for Radio Interface</li></ul>

9.6.3.85 `typedef void(* tFNSLQSOMADMAAlert)(ULONG eventType, BYTE *pEventFields)`

SWIOMA-DM network-initiated alert callback function

## Parameters

<i>eventType</i>	<ul style="list-style-type: none"><li>• 0x00 - SWIOMA-DM FOTA</li><li>• 0x01 - SWIOMA-DM Config</li><li>• 0x02 - SWIOMA-DM Notification</li></ul>
------------------	---

<i>pEventFields</i>	<ul style="list-style-type: none"> <li>• Pointer to structure containing info for that session type</li> <li>• See <a href="#">sessionInfo</a> for more details</li> </ul>
---------------------	--

**9.6.3.86** `typedef void(* tFNSLQSQOSEvent)(BYTE instance, QosFlowInfo *pFlowInfo)`

QOS Event callback function.

Parameters

<i>instance</i>	<ul style="list-style-type: none"> <li>• QMI instance</li> </ul>
<i>pFlowInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">QosFlowInfo</a> for more information</li> </ul>

**9.6.3.87** `typedef void(* tFNSLQSSessionState)(slqsSessionStateInfo *pSessionStateInfo)`

Session state callback function.

Parameters

<i>pSessionState-Info</i>	<ul style="list-style-type: none"> <li>• See <a href="#">slqsSessionStateInfo</a> for more details</li> </ul>
---------------------------	---

**9.6.3.88** `typedef void(* tFNSLQSSignalStrengths)(struct SLQSSignalStrengthsInformation sSLQSSignalStrengthsInfo)`

Received Signal Strength Information callback function.

Parameters

<i>sSLQSSignal-StrengthsInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">SLQSSignalStrengthsInformation</a> for more information.</li> </ul>
----------------------------------	--

**9.6.3.89** `typedef void(* tFNSLQSWDSEvent)(slqsWdsEventInfo *pWdsEventInfo)`

WDS Event callback function.

Parameters

<i>pWdsEventInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">slqsWdsEventInfo</a> for more details</li> </ul>
----------------------	---

**9.6.3.90** `typedef void(* tFNSMSEvents)(SMSEventInfo *pSMSEventInfo)`

SMS event related callback function.

## Parameters

<i>pSMSEventInfo</i> [OUT]	<ul style="list-style-type: none"> <li>Events related to SMS, see <a href="#">SMSEventInfo</a> for details</li> </ul>
-------------------------------	---

## 9.6.3.91 typedef void( \* tFNSUPSInfo)(voiceSUPSInfo \*pVoiceSUPSInfo)

Preferred SUPS indication callback.

## Parameters

<i>pVoiceSUPSInfo</i>	<ul style="list-style-type: none"> <li>See <a href="#">voiceSUPSInfo</a> for more information.</li> </ul>
-----------------------	---

## Note

Technology Supported: GSM

## 9.6.3.92 typedef void(\* tFNSUPSNotification)(voiceSUPSNotification \*pVoiceSUPSNotification)

Supplementary service notification callback.

## Parameters

<i>pVoiceSUPS-Notification</i>	<ul style="list-style-type: none"> <li>See <a href="#">voiceSUPSNotification</a> for more information.</li> </ul>
--------------------------------	---

## 9.6.3.93 typedef void( \* tFNSysInfo)(nasSysInfo \*pNasSysInfo)

System Information indication callback.

## Parameters

<i>pNasSysInfo</i>	<ul style="list-style-type: none"> <li>See <a href="#">nasSysInfo</a> for more information.</li> </ul>
--------------------	--

## 9.6.3.94 typedef void(\* tFNSysSelectionPref)(sysSelectPrefInfo \*pSysSelectPrefInfo)

System Selection Preference Callback function

## Parameters

<i>pSysSelectPref-Info</i>	<ul style="list-style-type: none"> <li>Current System Selection preferences for the device.</li> <li>See <a href="#">sysSelectPrefInfo</a> for more information</li> </ul>
----------------------------	--

9.6.3.95 `typedef void(* tFNtransLayerInfo)(transLayerNotification *pTransLayerNotification)`

Transport Layer Information callback.

## Parameters

<i>transLayer-Notification</i>	<ul style="list-style-type: none"> <li>See <a href="#">transLayerNotification</a> for more information.</li> </ul>
--------------------------------	--

**9.6.3.96** `typedef void(* tFNtransNWRegInfo)(transNWRegInfoNotification *pTransNWRegInfoNotification)`

Transport Network Registration Information callback.

## Parameters

<i>pTransNWReg-InfoNotification</i>	<ul style="list-style-type: none"> <li>See <a href="#">transNWRegInfoNotification</a> for more information.</li> </ul>
-------------------------------------	--

**9.6.3.97** `typedef void(* tFNUIMRefresh)(UIMRefreshEvent *pUIMRefreshEvent)`

UIM Refresh Callback function

## Parameters

<i>pUIMRefresh-Event</i>	<ul style="list-style-type: none"> <li>Pointer to Refresh Event structure.</li> <li>See <a href="#">UIMRefreshEvent</a> for more information</li> </ul>
--------------------------	---

**9.6.3.98** `typedef void( * tFNUIMStatusChangeInfo)(UIMStatusChangeInfo *pUIMStatusChangeInfo)`

UIM Status Change Callback function

## Parameters

<i>pUIMStatus-ChangeInfo</i>	<ul style="list-style-type: none"> <li>Pointer to UIM status change structure.</li> <li>See <a href="#">UIMStatusChangeInfo</a> for more information</li> </ul>
------------------------------	---

**9.6.3.99** `typedef void(* tFNUSSDNotification)(ULONG type, BYTE *pNetworkInfo)`

SetUSSDNotificationCallback function prototype

## Parameters

<i>type</i>	<ul style="list-style-type: none"> <li>- Notification type <ul style="list-style-type: none"> <li>• 0x01 - No action required</li> <li>• 0x02 - Action required</li> </ul> </li> </ul>
<i>pNetworkInfo</i>	<ul style="list-style-type: none"> <li>• USS information from the network (0 indicates that no info was received) <ul style="list-style-type: none"> <li>– See <a href="#">USSInfo</a> for more details</li> </ul> </li> </ul>

## Note

Technology Supported: UMTS

**9.6.3.100** `typedef void(* tFNUSSDNoWaitIndication)(USSDNoWaitIndicationInfo *pNetworkInfo)`

**9.6.3.101** `typedef void(* tFNUSSDRelease)(void)`

USSD releaserecallback function prototype

## Note

Technology Supported: UMTS

**9.6.3.102** `typedef struct _transLayerInfoNotification transLayerNotification`

Contains the parameters passed for SLQSSetTransLayerInfoCallback by the device.

## Parameters

<i>regInd</i>	<ul style="list-style-type: none"> <li>• Indicates whether the transport layer is registered or not</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0x00 - Transport layer is not registered</li> <li>– 0x01 - Transport layer is registered</li> </ul> </li> </ul>
<i>pTransLayerInfo</i>	<ul style="list-style-type: none"> <li>• Optional parameter</li> <li>• See <a href="#">transLayerInfo</a> for more information</li> </ul>

## Note

None

**9.6.3.103** `typedef struct _transNWRegInfoNotification transNWRegInfoNotification`

Contains the parameters passed for SLQSSetTransNWRegInfoCallback by the device.



## Parameters

<i>NWRegStat</i>	<ul style="list-style-type: none"> <li>• provides the transport network registration information</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0x00 - No Service</li> <li>– 0x01 - In Progress</li> <li>– 0x02 - Failed</li> <li>– 0x03 - Limited Service</li> <li>– 0x04 - Full Service</li> </ul> </li> </ul>
------------------	---

## Note

None

## 9.6.4 Enumeration Type Documentation

## 9.6.4.1 enum device\_state\_enum

Device State enumeration

## Enumerator

***DEVICE\_STATE\_DISCONNECTED***  
***DEVICE\_STATE\_READY***  
***DEVICE\_STATE\_BOOT***

## 9.6.4.2 enum eQaQMIService

The QMI service information which is exposed to the application, only the services which are relevant to multiple PDP are listed in this enumeration as these are the only required services to be exposed.

## Enumerator

***eQA\_QMI\_SVC\_WDS***  
***eQA\_QMI\_SVC\_NAS***  
***eQA\_QMI\_SVC\_NA***

## 9.6.4.3 enum SMSEventType

This enumeration defines the different type of SMS events that are received

## Enumerator

***SMS\_EVENT\_MT\_MESSAGE***  
***SMS\_EVENT\_TRANSFER\_ROUTE\_MT\_MESSAGE***  
***SMS\_EVENT\_MESSAGE\_MODE***  
***SMS\_EVENT\_ETWS***  
***SMS\_EVENT\_ETWS\_PLMN***  
***SMS\_EVENT\_SMSC\_ADDRESS***  
***SMS\_EVENT\_SMS\_ON\_IMS***

### 9.6.5 Function Documentation

9.6.5.1 **ULONG** iSetCATEventCallback ( **tFNCATEvent** *pCallback* )

9.6.5.2 **ULONG** iSetSignalStrengthCallback ( **tFNSignalStrength** *pCallback* )

9.6.5.3 **ULONG** iLQSSetDUNCallInfoCallback ( **tFNDUNCallInfo** *pCallback* )

9.6.5.4 **ULONG** iLQSSetSignalStrengthsCallback ( **tFNSLQSSignalStrengths** *pCallback* )

9.6.5.5 **ULONG** iLQSSetWdsFirstInstEventCallback ( **tFNSLQSWDSEvent** *pCallback* )

9.6.5.6 **ULONG** iLQSSetWdsSecondInstEventCallback ( **tFNSLQSWDSEvent** *pCallback* )

9.6.5.7 **ULONG** iLQSSetWdsThirdInstEventCallback ( **tFNSLQSWDSEvent** *pCallback* )

9.6.5.8 **ULONG** iLQSSetWdsXferStatsFirstInstCallback ( **tFNSLQSWDSEvent** *pCallback* )

9.6.5.9 **ULONG** iLQSSetWdsXferStatsSecondInstCallback ( **tFNSLQSWDSEvent** *pCallback* )

9.6.5.10 **ULONG** SetActivationStatusCallback ( **tFNActivationStatus** *pCallback* )

Enables/disables the Activation Status callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

#### Parameters

<i>pCallback</i> [IN]	<ul style="list-style-type: none"> <li>• Callback function pointer (0 - disable)</li> </ul>
-----------------------	---

#### Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

#### See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

9.6.5.11 **ULONG** SetCATEventCallback ( **tFNCATEvent** *pCallback*, **ULONG** *eventMask*, **ULONG** \* *pErrorMask* )

Enables/disables the CAT event callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

## Parameters

<i>pCallback</i> [IN]	<ul style="list-style-type: none"> <li>• Callback function pointer (0 - Disable)</li> </ul>
<i>eventMask</i>	<ul style="list-style-type: none"> <li>• bitmask of CAT events to register for             <ul style="list-style-type: none"> <li>– 0x00000001 - Display Text</li> <li>– 0x00000002 - Get In-Key</li> <li>– 0x00000004 - Get Input</li> <li>– 0x00000008 - Setup Menu</li> <li>– 0x00000010 - Select Item</li> <li>– 0x00000020 - Send SMS - Alpha Identifier</li> <li>– 0x00000040 - Setup Event: User Activity</li> <li>– 0x00000080 - Setup Event: Idle Screen Notify</li> <li>– 0x00000100 - Setup Event: Language Sel Notify</li> <li>– 0x00000200 - Setup Idle Mode Text</li> <li>– 0x00000400 - Language Notification</li> <li>– 0x00000800 - Refresh</li> <li>– 0x00001000 - End Proactive Session</li> </ul> </li> </ul>
<i>pErrorMask</i> [OUT]	<ul style="list-style-type: none"> <li>• error bitmask. Each bit set indicates the proactive command that caused the error             <ul style="list-style-type: none"> <li>– 0x00000001 - Display Text</li> <li>– 0x00000002 - Get In-Key</li> <li>– 0x00000004 - Get Input</li> <li>– 0x00000008 - Setup Menu</li> <li>– 0x00000010 - Select Item</li> <li>– 0x00000020 - Send SMS - Alpha Identifier</li> <li>– 0x00000040 - Setup Event: User Activity</li> <li>– 0x00000080 - Setup Event: Idle Screen Notify</li> <li>– 0x00000100 - Setup Event: Language Sel Notify</li> <li>– 0x00000200 - Setup Idle Mode Text</li> <li>– 0x00000400 - Language Notification</li> <li>– 0x00000800 - Refresh</li> <li>– 0x00001000 - End Proactive Session</li> </ul> </li> </ul>

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**Note**

Technology Supported: UMTS  
Timeout: 2 seconds

**9.6.5.12 ULONG SetDataCapabilitiesCallback ( tFNDataCapabilities pCallback )**

Enables/disables the data capabilities callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

**Parameters**

<i>pCallback[IN]</i>	<ul style="list-style-type: none"><li>• Callback function pointer (0 - disable)</li></ul>
----------------------	---

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Does not require communication with the device

**9.6.5.13 ULONG SetDeviceStateChangeCbk ( tFNDeviceStateChange pCallback )**

Used by the client application to register a Callback function for Device State Change (DSC) event notifications. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

**Parameters**

<i>pCallback[IN]</i>	<ul style="list-style-type: none"><li>• a valid function pointer to be notified of DSC events</li><li>• NULL to disable DSC event notification</li></ul>
----------------------	--

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**9.6.5.14 ULONG SetFwDidCompletionCbk ( tFNFwDidCompletion pCallback )**

Used by the client application to register a Callback function for a Firmware Download Completion (FDC) event notification. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

**Parameters**

<i>pCallback[IN]</i>	<ul style="list-style-type: none"><li>• a valid function pointer to enable FDC event notification</li><li>• NULL to disable FDC event notification</li></ul>
----------------------	--

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Timeout: N/A

**9.6.5.15 ULONG SetGPSCallback ( tFNNewGPS pCallback )**

Enables/disables the NMEA sentence callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

**Parameters**

<i>pCallback[IN]</i>	<ul style="list-style-type: none"><li>• Callback function pointer (0 - Disable)</li></ul>
----------------------	---

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**9.6.5.16 ULONG SetLocCradleMountCallback ( tFNSetCradleMount pCallback )**

Enables/disables the Cradle Mount callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

**Parameters**

<i>pCallback[IN]</i>	<ul style="list-style-type: none"><li>• Callback function pointer (0 - Disable)</li></ul>
----------------------	---

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**9.6.5.17    ULONG SetLocDeleteAssistDataCallback ( tFNDelAssistData *pCallback* )**

Enables/disables Delete Assist Data callback function. This API is used to receive the SUCCESS/FAILURE status of API [SLQSLCDeIAssData\(\)](#).

## Parameters

<i>pCallback</i> [IN]	<ul style="list-style-type: none"> <li>• Callback function pointer (0-Disable)</li> </ul>
-----------------------	---

**9.6.5.18    ULONG SetLocEventPositionCallback ( tFNEventPosition *pCallback* )**

Enables/disables the Event Position Report callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

## Parameters

<i>pCallback</i> [IN]	<ul style="list-style-type: none"> <li>• Callback function pointer (0 - Disable)</li> </ul>
-----------------------	---

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**9.6.5.19    ULONG SetLocEventTimeSyncCallback ( tFNSetEventTimeSync *pCallback* )**

Enables/disables the Event Time Sync callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

## Parameters

<i>pCallback</i> [IN]	<ul style="list-style-type: none"> <li>• Callback function pointer (0 - Disable)</li> </ul>
-----------------------	---

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**9.6.5.20    ULONG SetLocGnssSvInfoCallback ( tFNGnssSvInfo *pCallback* )**

Enables/disables the GNSS [SV](#) Info callback function. This API is used to send the satellite report to the application. The satellite reports are sent only to the application that invoked API [SLQSLOCStart\(\)](#) that generated the satellite report.

## Parameters

<i>pCallback</i> [IN]	<ul style="list-style-type: none"> <li>• Callback function pointer (0-Disable)</li> </ul>
-----------------------	---

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Timeout: 2 seconds

**9.6.5.21 ULONG SetLocInjectSensorDataCallback ( tFNInjectSensorData pCallback )**

Enables/disables the Inject Sensor Data callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

**Parameters**

<i>pCallback</i> [IN]	<ul style="list-style-type: none"> <li>• Callback function pointer (0 - Disable)</li> </ul>
-----------------------	---

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**9.6.5.22 ULONG SetLocInjectTimeCallback ( tFNInjectTimeStatus pCallback )**

Enables/disables the Inject Time Sync Data callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

**Parameters**

<i>pCallback</i> [IN]	<ul style="list-style-type: none"> <li>• Callback function pointer (0 - Disable)</li> </ul>
-----------------------	---

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**9.6.5.23 ULONG SetLocOpModeCallback ( tFNOpMode pCallback )**

Enables/disables Set Operating Mode callback function. This API is used to receive the SUCCESS/FAILURE status of API [SLQSLOCSetOpMode\(\)](#).

**Parameters**

<i>pCallback</i> [IN]	<ul style="list-style-type: none"> <li>• Callback function pointer (0-Disable)</li> </ul>
-----------------------	---

**9.6.5.24 ULONG SetLocSensorStreamingCallback ( tFNSensorStreaming pCallback )**

Enables/disables the Event Sensor Streaming Ready Status callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.



## Parameters

<i>pCallback</i> [IN]	<ul style="list-style-type: none"><li>• Callback function pointer (0 - Disable)</li></ul>
-----------------------	---

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**9.6.5.25   ULONG SetLURejectCallback ( tFNLUReject *pCallback* )**

Enables/disables the LU reject callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

## Parameters

<i>pCallback</i> [IN]	<ul style="list-style-type: none"><li>• Callback function pointer (0 - disable)</li></ul>
-----------------------	---

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 2 seconds

**9.6.5.26   ULONG SetMobileIPStatusCallback ( tFNMobileIPStatus *pCallback* )**

Enables/disables the Mobile IP Status callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

## Parameters

<i>pCallback</i> [IN]	<ul style="list-style-type: none"><li>• Callback function pointer (0 - disable)</li></ul>
-----------------------	---

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Technology Supported: CDMA  
Timeout: 2 seconds

9.6.5.27 **ULONG** SetNasLTECphyCalndCallback ( **tFNASwiLTECphyCalInfo** *pCallback* )

Enables/disables the LTE NAS CA Info callback function.

## Parameters

<i>pCallback</i> [IN]	<ul style="list-style-type: none"> <li>• Callback function pointer (0-Disable)</li> </ul>
-----------------------	---

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 2 seconds

#### 9.6.5.28 ULONG SetNetChangeCbK ( BYTE *instance*, tFNNet *pCallback*, ULONG *loMark*, ULONG *hiMark*, ULONG *period* )

Used by the client application to register a Callback function for USB Transmit Queue Length Change event notifications. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

## Parameters

<i>instance</i> [IN]	<ul style="list-style-type: none"> <li>• PDP instance</li> </ul>
<i>pCallback</i> [IN]	<ul style="list-style-type: none"> <li>• a valid function pointer to be notified of the event</li> <li>• NULL to disable the event notification</li> </ul>
<i>loMark</i> [IN]	<ul style="list-style-type: none"> <li>• Transmit queue length smaller will trigger unthrottle event notification</li> </ul>
<i>hiMark</i> [IN]	<ul style="list-style-type: none"> <li>• Transmit queue length larger will trigger throttle event notification</li> </ul>
<i>period</i> [IN]	<ul style="list-style-type: none"> <li>• monitoring period in seconds, minimum 1 second</li> </ul>

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

#### 9.6.5.29 ULONG SetNewSMSCallback ( tFNNewSMS *pCallback* )

Enables/disables the new SMS callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

**Parameters**

<i>pCallback</i> [IN]	<ul style="list-style-type: none"> <li>• Callback function pointer (0 - Disable)</li> </ul>
-----------------------	---

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**Note**

Timeout: 2 seconds

**9.6.5.30 ULONG SetNMEACallback ( tFNNewNMEA pCallback )**

Enables/disables the NMEA sentence callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

This API is deprecated on all MC/EM74xx firmware versions. Please use [SetLocEventPositionCallback](#)

**Parameters**

<i>pCallback</i> [IN]	<ul style="list-style-type: none"> <li>• Callback function pointer (0 - Disable)</li> </ul>
-----------------------	---

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**9.6.5.31 ULONG SetOMADMStateCallback ( tFNOMADMState pCallback )**

Enables/disables the OMADM state callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

This API is deprecated on all MC/EM74xx firmware versions. Please use [SetSLQSOMADMAAlertCallback](#)

**Parameters**

<i>pCallback</i> [IN]	<ul style="list-style-type: none"> <li>• a valid function pointer to enable OMADMState notification</li> <li>• NULL to disable OMADMState notification</li> </ul>
-----------------------	---

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**Note**

Technology Supported: CDMA

Timeout: 2 seconds

**9.6.5.32 ULONG SetPDSStateCallback ( tFNPDSState pCallback )**

Enables/disables the PDS service state callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

## Parameters

<i>pCallback</i> [IN]	<ul style="list-style-type: none"> <li>• Callback function pointer (0 - Disable)</li> </ul>
-----------------------	---

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

9.6.5.33 ULONG SetPowerCallback ( tFNPower *pCallback* )

Enables/disables the Operating Mode callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

## Parameters

<i>pCallback</i> [IN]	<ul style="list-style-type: none"> <li>• Callback function pointer (0 - disable)</li> </ul>
-----------------------	---

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 2 seconds

9.6.5.34 ULONG SetRFInfoCallback ( tFNRInfo *pCallback* )

Enables/disables the radio frequency information callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

## Parameters

<i>pCallback</i> [IN]	<ul style="list-style-type: none"> <li>• Callback function pointer (0 - Disable)</li> </ul>
-----------------------	---

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 2 seconds

**9.6.5.35    ULONG SetRMTransferStatisticsCallback ( tFNNewRMTransferStatistics *pCallback* )**

Enables/disables the RM Transfer Statistics callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

## Parameters

<i>pCallback</i> [IN]	<ul style="list-style-type: none"> <li>• Callback function pointer (0 - Disable)</li> </ul>
-----------------------	---

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

9.6.5.36 ULONG SetRoamingIndicatorCallback ( tFNRoamingIndicator *pCallback* )

Enables/disables the Roaming Indicator callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

This API is deprecated on all MC/EM74xx firmware versions. Please use [SLQSNasSysInfoCallBack](#)

## Parameters

<i>pCallback</i> [IN]	<ul style="list-style-type: none"> <li>• Callback function pointer (0 - disable)</li> </ul>
-----------------------	---

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 2 seconds

9.6.5.37 ULONG SetSignalStrengthCallback ( tFNSignalStrength *pCallback*, BYTE *thresholdsSize*, INT8 \* *pThresholds* )

Enables/disables the Signal Strength callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs. This API is deprecated on MC73xx/-EM73xx modules since firmware version SWI9X15C\_05\_xx\_xx\_xx and all EM74xx firmware versions. Please use API [SLQSNasIndicationRegisterExt\(\)](#) for new firmware versions and new modules

## Parameters

<i>pCallback</i> [IN]	<ul style="list-style-type: none"> <li>• Callback function pointer (0-Disable)</li> </ul>
<i>thresholdsSize</i>	<ul style="list-style-type: none"> <li>• Number of elements threshold array contains; a maximum of five thresholds is supported;</li> <li>• This parameter is not used when disabling the callback.</li> </ul>

<i>pThresholds[IN]</i>	<ul style="list-style-type: none"> <li>• Signal threshold array for each entry (in dBm).</li> <li>• This parameter is not used when disabling the callback.</li> </ul>
------------------------	--

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Timeout: 2 seconds

The signal strength callback function is called when a threshold in the threshold array is crossed.

**9.6.5.38 ULONG SetSLQSOMADMAAlertCallback ( tFNSLQSOMADMAAlert pCallback )**

Enables/disables the SWIOMADM network-initiated alert callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

**Parameters**

<i>pCallback[IN]</i>	<ul style="list-style-type: none"> <li>• a valid function pointer to enable SLQSOMADMAAlert notification</li> <li>• NULL to disable SLQSOMADMAAlert notification</li> </ul>
----------------------	---

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**9.6.5.39 ULONG SetSLQSOMADMAAlertCallbackExt ( tFNSLQSOMADMAAlert pCallback )**

Enables/disables the SWIOMADM network-initiated alert callback function for SL9090 module. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

**Parameters**

<i>pCallback[IN]</i>	<ul style="list-style-type: none"> <li>• a valid function pointer to enable SLQSOMADMAAlert notification</li> <li>• NULL to disable SLQSOMADMAAlert notification</li> </ul>
----------------------	---

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**Note**

Technology Supported: UMTS/CDMA

Device Supported: SL9090

Timeout: 2 seconds



**9.6.5.40 ULONG SetUSSDNotificationCallback ( tFNUSSDNotification pCallback )**

Enables/disables the USSDNotification callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

**Parameters**

<i>pCallback[IN]</i>	<ul style="list-style-type: none"><li>• a valid function pointer to enable ServingSystem notification</li><li>• NULL to disable ServingSystem notification</li></ul>
----------------------	--

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**Note**

Technology Supported: UMTS

Timeout: Does not require communication with device

**9.6.5.41 ULONG SetUSSDNoWaitIndicationCallback ( tFNUSSDNoWaitIndication pCallback )**

SetUSSDNoWaitIndicationCallback

**Parameters**

<i>pNetworkInfo</i>	<ul style="list-style-type: none"><li>• Data from the network.</li><li>• See <a href="#">USSDNoWaitIndicationInfo</a> for more details.</li></ul>
---------------------	---

**Note**

Technology Supported: UMTS

Device Supported: MC83x5

**9.6.5.42 ULONG SetUSSDReleaseCallback ( tFNUSSDRelease pCallback )**

Enables/disables the USSD release callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

**Parameters**

<i>pCallback[IN]</i>	<ul style="list-style-type: none"><li>• a valid function pointer to enable ServingSystem notification</li><li>• NULL to disable ServingSystem notification</li></ul>
----------------------	--

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**Note**

Technology Supported: UMTS

Timeout: Does not require communication with the device

#### 9.6.5.43 ULONG SLQSNasNetworkTimeCallBack ( tFNNetworkTime *pCallback* )

Enables/disables the Network Time callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

##### Parameters

<i>pCallback</i> [IN]	<ul style="list-style-type: none"> <li>• Callback function pointer (0-Disable)</li> </ul>
-----------------------	---

##### Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

##### See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

##### Note

Timeout: 2 seconds

This callback is sent when the 3GPP or 3GPP2 network sends time information to the User Equipment.

#### 9.6.5.44 ULONG SLQSNasSigInfo2CallBack ( tFNSigInfo *pCallback*, setSignalStrengthInfo \* *pSigInfo2* )

Enables/disables the Signal Info callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

##### Parameters

<i>pCallback</i> [IN]	<ul style="list-style-type: none"> <li>• Callback function pointer (0-Disable)</li> </ul>
<i>pSigInfo2</i> [IN]	<ul style="list-style-type: none"> <li>• Structure containing the threshold values beyond which signal information is to be reported</li> <li>• See <a href="#">setSignalStrengthInfo</a> for more details</li> </ul>

##### Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

##### See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

##### Note

Timeout: 2 seconds

This callback is sent when the signal strength change occurs

**9.6.5.45    ULONG SLQSNasSigInfoCallBack ( tFNSigInfo *pCallback*, sigInfo \* *pSigInfo* )**

Enables/disables the Signal Info callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs. This callback is deprecated on MC73xx/-EM73xx modules since firmware version SWI9X15C\_05\_xx\_xx\_xx and all EM74xx firmware versions. Please use callback [SLQSNasSigInfo2CallBack\(\)](#) for new firmware versions and new modules

## Parameters

<i>pCallback</i> [IN]	<ul style="list-style-type: none"> <li>• Callback function pointer (0-Disable)</li> </ul>
<i>pSigInfo</i> [IN]	<ul style="list-style-type: none"> <li>• Structure containing the threshold values beyond which signal information is to be reported</li> <li>• See <a href="#">sigInfo</a> for more details</li> </ul>

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 2 seconds

This callback is sent when the signal strength change occurs

#### 9.6.5.46 ULONG SLQSNasSwiOTAMessageCallback ( NasSwiIndReg \* req, tFNASwiOTAMsg pCallback )

Enables/disables the SLQSNasSwiOTAMessageCallback callback function. To disable the callback, provide both req and pCallback as NULL pointer to the API

## Parameters

<i>req</i> [IN]	<ul style="list-style-type: none"> <li>• the request to which kind of message type should be enabled, see <a href="#">NasSwiIndReg</a> for details</li> </ul>
<i>pCallback</i> [IN]	<ul style="list-style-type: none"> <li>• Callback function pointer (0-Disable)</li> </ul>

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 2 seconds

#### 9.6.5.47 ULONG SLQSNasSysInfoCallBack ( tFNSysInfo pCallback )

Enables/disables the Sys Info callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

## Parameters

<i>pCallback</i> [IN]	<ul style="list-style-type: none"> <li>• Callback function pointer (0-Disable)</li> </ul>
-----------------------	---

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 2 seconds

This callback provides current serving system information, including registration information and system property. The serving system information of the radio interfaces specified in mode\_pref are included in the response message. When any value in the sys\_info message changes, an indication message is sent. Indications contain all the values for all active RATs.

#### 9.6.5.48 ULONG SLQSSetBandPreferenceCbk ( tFNBandPreference *pCallback* )

Enables/disables the Band Preference callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

## Parameters

<i>pCallback</i> [IN]	<ul style="list-style-type: none"> <li>• a valid function pointer to enable Band Preference Indication notification</li> <li>• NULL to disable Band Preference notification</li> </ul>
-----------------------	--

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## Note

Timeout: NA To set the band preference the API [SLQSSetBandPreference\(\)](#) should be used

#### 9.6.5.49 ULONG SLQSSetDataSystemStatusCallback ( tFNDataSysStatus *pCallback* )

Enables/disables the Data System Status callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

## Parameters

<i>pCallback</i> [IN]	<ul style="list-style-type: none"> <li>• Callback function pointer (0-Disable)</li> </ul>
-----------------------	---

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Timeout: 5 seconds

#### 9.6.5.50 **ULONG** SLQSSetDUNCallInfoCallback ( **BYTE** *StatsPeriod*, **tFNDUNCallInfo** *pCallback* )

Enables/disables the DUN Call Info callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

**Parameters**

<i>StatsPeriod</i> [IN]	<ul style="list-style-type: none"> <li>• Period between reports(seconds)</li> <li>• 0 - Do not report</li> <li>• Only applicable to pTXOKBytesCount and pRXOKBytesCount parameters</li> </ul>
<i>pCallback</i> [IN]	<ul style="list-style-type: none"> <li>• Callback function pointer (0-Disable)</li> </ul>

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Timeout: 5 seconds

#### 9.6.5.51 **ULONG** SLQSSetIMSAPdpStatusCallback ( **tFNImsaPdpStatus** *pCallback* )

SLQSSetIMSAPdpStatusCallback

**Parameters**

<i>pCallback</i> [IN]	<ul style="list-style-type: none"> <li>• Callback function pointer (0-Disable)</li> </ul>
-----------------------	---

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 5 seconds

**9.6.5.52   ULONG SLQSSetIMSAStatusCallback ( tFNImsaStatus pCallback )**

SLQSSetIMSAStatusCallback

## Parameters

<i>pCallback</i> [IN]	<ul style="list-style-type: none"><li>• Callback function pointer (0-Disable)</li></ul>
-----------------------	---

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 5 seconds

**9.6.5.53   ULONG SLQSSetIMSARegStatusCallback ( tFNImsaRegStatus pCallback )**

SLQSSetIMSARegStatusCallback

## Parameters

<i>pCallback</i> [IN]	<ul style="list-style-type: none"><li>• Callback function pointer (0-Disable)</li></ul>
-----------------------	---

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 5 seconds

**9.6.5.54   ULONG SLQSSetIMSASvcStatusCallback ( tFNImsaSvcStatus pCallback )**

SLQSSetIMSASvcStatusCallback

## Parameters

<i>pCallback</i> [IN]	<ul style="list-style-type: none"><li>• Callback function pointer (0-Disable)</li></ul>
-----------------------	---

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 5 seconds

**9.6.5.55 ULONG SLQSSetIMSSMSConfigCallback ( tFNImSMSConfig pCallback )**

Enables/disables the SMS Config callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

## Parameters

<i>pCallback</i> [IN]	<ul style="list-style-type: none"><li>• Callback function pointer (0-Disable)</li></ul>
-----------------------	---

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 2 seconds

**9.6.5.56 ULONG SLQSSetIMSUserConfigCallback ( tFNImUserConfig pCallback )**

Enables/disables the User Config callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

## Parameters

<i>pCallback</i> [IN]	<ul style="list-style-type: none"><li>• Callback function pointer (0-Disable)</li></ul>
-----------------------	---

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise



## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 2 seconds

**9.6.5.57 ULONG SLQSSetIMSVoIPConfigCallback ( tFNImSVoIPConfig pCallback )**

Enables/disables the VoIP Config callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

## Parameters

<i>pCallback</i> [IN]	<ul style="list-style-type: none"><li>• Callback function pointer (0-Disable)</li></ul>
-----------------------	---

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 2 seconds

**9.6.5.58 ULONG SLQSSetModemTempCallback ( tFNModemTempInfo pCallback )**

Enables/disables the Modem Temperature information callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

## Parameters

<i>pCallback</i> [IN]	<ul style="list-style-type: none"><li>• Callback function pointer (0-Disable)</li></ul>
-----------------------	---

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 5 seconds

**9.6.5.59 ULONG SLQSSetPacketSrvStatusCallback ( tFNPacketSrvState pCallback )**

Enables/disables the session state callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

## Parameters

<i>pCallback</i> [IN]	<ul style="list-style-type: none"> <li>• Callback function pointer (0 - disable)</li> </ul>
-----------------------	---

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: none; does not require communication with the device

#### 9.6.5.60 ULONG SLQSSetQosEventCallback ( BYTE *instance*, tFNSLQSQOSEvent *pCallback* )

Enables/disables the QoS event callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs. This callback provide extra information regarding the QoS flow state

## Parameters

in	<i>instance</i>	<ul style="list-style-type: none"> <li>• QMI instance</li> </ul>
in	<i>pCallback</i>	<ul style="list-style-type: none"> <li>• Callback function pointer (0 - disable)</li> </ul>

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

maximum number of tx/rx filters supported is 25 (pTxQFilter/pRxQFilter)

#### 9.6.5.61 ULONG SLQSSetQosNWStatusCallback ( tFNQosNWStatus *pCallback* )

Enables/disables the QoS event callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs. This callback provide extra information regarding the QoS Network supports status

## Parameters

in	<i>pCallback</i> [IN]	<ul style="list-style-type: none"><li>• Callback function pointer (0 - disable)</li></ul>
----	-----------------------	---

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**9.6.5.62 ULONG SLQSSetQosPriEventCallback ( tFNQosPriEvent *pCallback* )**

Enables/disables the QoS event callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs. This callback provide extra information regarding the QoS Primary flow event

## Parameters

in	<i>pCallback</i>	<ul style="list-style-type: none"><li>• Callback function pointer (0 - disable)</li></ul>
----	------------------	---

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Technology Supported: UMTS

**9.6.5.63 ULONG SLQSSetQosStatusCallback ( BYTE *instance*, tFNQosStatus *pCallback* )**

Enables/disables the QoS event callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs. This callback provide extra information regarding the QoS status

## Parameters

in	<i>instance</i>	<ul style="list-style-type: none"><li>• QMI instance</li></ul>
----	-----------------	--

<i>in</i>	<i>pCallback[IN]</i>	<ul style="list-style-type: none"> <li>• Callback function pointer (0 - disable)</li> </ul>
-----------	----------------------	---

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**9.6.5.64 ULONG SLQSSetRegMgrConfigCallback ( tFNlmsRegMgrConfig pCallback )**

Enables/disables the Reg Mgr Config callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

**Parameters**

<i>pCallback[IN]</i>	<ul style="list-style-type: none"> <li>• Callback function pointer (0-Disable)</li> </ul>
----------------------	---

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Timeout: 2 seconds

**9.6.5.65 ULONG SLQSSetSDKTerminatedCallback ( tFNSDKTerminated pCallback )**

Used by the client application to register a Callback function for SDK terminated event notifications. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

**Parameters**

<i>pCallback[IN]</i>	<ul style="list-style-type: none"> <li>• a valid function pointer to be notified of SWI events</li> <li>• NULL to disable SWI event notification</li> </ul>
----------------------	---

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

#### See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

#### Note

Technology Supported: N/A

Device Supported: N/A

Timeout: N/A

The following signals will trigger this callback:

2 INT	4 ILL	5 TRAP	6 ABRT	7 BUS
8 FPE	11 SEGV	13 PIPE	15 TERM	31 SYS

#### 9.6.5.66 ULONG SLQSSetServingSystemCallback ( tFNServingSystem pCallback )

Enables/disables the Serving System callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

This API is deprecated on all MC/EM74xx firmware versions. Please use [SLQSNasSysInfoCallBack](#)

##### Parameters

<i>pCallback[IN]</i>	<ul style="list-style-type: none"><li>• a valid function pointer to enable ServingSystem notification</li><li>• NULL to disable ServingSystem notification</li></ul>
----------------------	--

##### Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

#### 9.6.5.67 ULONG SLQSSetSessionStateCallback ( tFNSLQSSessionState pCallback )

Enables/disables the session state callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs. This callback provide extra information regarding the multiple PDP interface

##### Parameters

<i>pCallback[IN]</i>	<ul style="list-style-type: none"><li>• Callback function pointer (0 - disable)</li></ul>
----------------------	---

##### Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

#### See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

#### Note

Timeout: none; does not require communication with the device

#### 9.6.5.68 **ULONG** SLQSSetSignalStrengthsCallback ( **tFNSLQSSignalStrengths** *pCallback*, **struct** **SLQSSignalStrengthsIndReq** \* *pSLQSSignalStrengthsIndReq* )

Enables/disables the Received Signal Strength Information callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs. This API is same as API SetSignalStrengthsCallback() except providing more information of signal such as ECIO, SNR etc. This API is deprecated on MC73xx/EM73xx modules since firmware version SWI9X15C\_05\_xx\_xx\_xx and all E-M74xx firmware versions. Please use API [SLQSNasIndicationRegisterExt\(\)](#) for new firmware versions and new modules

##### Parameters

<i>pCallback</i> [IN]	<ul style="list-style-type: none"> <li>• Callback function pointer (0-Disable)</li> </ul>
<i>pSLQSSignalStrengthsIndReq</i>	<ul style="list-style-type: none"> <li>• See <a href="#">SLQSSignalStrengthsIndReq</a> for more information</li> <li>• This parameter is not used when disabling the callback.</li> </ul>

##### Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

##### See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

##### Note

Timeout: 2 seconds

The signal strength callback function is called when a threshold in the threshold array is crossed.

#### 9.6.5.69 **ULONG** SLQSSetSIPConfigCallback ( **tFNImSIPConfig** *pCallback* )

Enables/disables the SIP Config callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

##### Parameters

<i>pCallback</i> [IN]	<ul style="list-style-type: none"> <li>• Callback function pointer (0-Disable)</li> </ul>
-----------------------	---

##### Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

##### See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

##### Note

Timeout: 2 seconds

#### 9.6.5.70 ULONG SLQSSetSMSEventCallback ( tFNSMSEvents *pCallback* )

Enables/disables the events related to SMS callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

## Parameters

<i>pCallback</i> [IN]	<ul style="list-style-type: none"> <li>• Callback function pointer (0 - Disable)</li> </ul>
-----------------------	---

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

#### 9.6.5.71 ULONG SLQSSetSwtHdRPersCallback ( tFNHdRPersnaity *pCallback* )

Enables/disables the HDR Personality callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

## Parameters

<i>pCallback</i> [IN]	<ul style="list-style-type: none"> <li>• Callback function pointer (0-Disable)</li> </ul>
-----------------------	---

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Technology Supported: CDMA  
Timeout: 5 seconds

#### 9.6.5.72 ULONG SLQSSetSysSelectionPrefCallBack ( tFNSysSelectionPref *pCallback* )

Enables/disables the System Selection Preference callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

## Parameters

<i>pCallback</i> [IN]	<ul style="list-style-type: none"> <li>• a valid function pointer to enable System Selection Preference Indication notification</li> <li>• NULL to disable Band Preference notification</li> </ul>
-----------------------	--

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## Note

Timeout: 2 seconds  
To set the system selection preferences the API [SLQSSetSysSelectionPref\(\)](#) should be used



**9.6.5.73    ULONG SLQSSetTransLayerInfoCallback ( tFNtransLayerInfo *pCallback* )**

Enables/disables the Transport Layer information callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

## Parameters

<i>pCallback</i> [IN]	<ul style="list-style-type: none"> <li>• Callback function pointer (0-Disable)</li> </ul>
-----------------------	---

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 2 seconds

#### 9.6.5.74 ULONG SLQSSetTransNWRegInfoCallback ( tFNtransNWRegInfo *pCallback* )

Enables/disables the Transport Network Registration information callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

## Parameters

<i>pCallback</i> [IN]	<ul style="list-style-type: none"> <li>• Callback function pointer (0-Disable)</li> </ul>
-----------------------	---

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 2 seconds

#### 9.6.5.75 ULONG SLQSSetWdsEventCallback ( tFNSLQSWDSEvent *pCallback*, BYTE *interval*, BYTE *instanceid*, BYTE *ipfamily* )

Enables/disables the WDS event callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs. This callback provide extra information regarding the multiple PDP interface. Transfer statistic are reported only when changed.

## Parameters

<i>pCallback</i> [!N]	<ul style="list-style-type: none"> <li>• Callback function pointer (0 - disable)</li> </ul>
<i>interval</i>	<ul style="list-style-type: none"> <li>• Interval in seconds.</li> <li>• ignored when disabling, should be non-zero when enabling</li> <li>• period only affect transfer statistic attributes</li> </ul>
<i>instanceid</i>	<ul style="list-style-type: none"> <li>• PDP instance id 0 - First PDP instance 1 - Second PDP instance 2 - Third PDP instance</li> </ul>
<i>ipfamily</i>	<ul style="list-style-type: none"> <li>• 4 for an IPv4 data session</li> <li>• 6 for an IPv6 data session</li> <li>• 7 for an IPv4v6 data session</li> </ul>

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 2 seconds Currently 3 PDP instances are supported in device. user of this callback can subscribe by passing instanceid of particular instance. All PDP instance can be subscribed by passing instanceid sequentially.

#### 9.6.5.76 ULONG SLQSSetWdsTransferStatisticCallback ( tFNSLQSWDSEvent *pXferStatsCb*, BYTE *interval*, BYTE *instanceid*, BYTE *ipfamily* )

Enables/disables the WDS transfer statistic callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs. This callback provide extra information regarding the multiple PDP interface. Transfer statistic are reported only when changed.

## Parameters

<i>pCallback</i> [IN]	<ul style="list-style-type: none"> <li>• Callback function pointer (0 - disable)</li> </ul>
<i>interval</i>	<ul style="list-style-type: none"> <li>• Interval in seconds.</li> <li>• ignored when disabling, should be non-zero when enabling</li> <li>• period only affect transfer statistic attributes</li> </ul>
<i>instanceid</i>	<ul style="list-style-type: none"> <li>• PDP instance id 0 - First PDP instance 1 - Second PDP instance 2 - Third PDP instance</li> </ul>
<i>ipfamily</i>	<ul style="list-style-type: none"> <li>• 4 for an IPv4 data session</li> <li>• 6 for an IPv6 data session</li> <li>• 7 for an IPv4v6 data session</li> </ul>

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 2 seconds Currently 3 PDP instances are supported in device. User of this callback can subscribe by passing instance id of particular instance. All PDP instance can be subscribed by passing instance id sequentially.

#### 9.6.5.77 ULONG SLQSUIMSetRefreshCallBack ( tFNUIMRefresh pCallback )

Enables/disables the UIM refresh callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

## Parameters

<i>pCallback</i> [IN]	<ul style="list-style-type: none"> <li>• a valid function pointer to enable UIM Refresh Indication notification</li> <li>• NULL to disable Band Preference notification</li> </ul>
-----------------------	--

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## Note

Timeout: 2 seconds  
[SLQSUIMRefreshRegister\(\)](#) API should be invoked prior to the invocation of the callback for the events to be registered.

**9.6.5.78 ULONG SLQSUIMSetStatusChangeCallBack ( tFNUIMStatusChangeInfo *pCallback* )**

Enables/disables the UIM Status Change Callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

**Parameters**

<i>pCallback</i> [IN]	<ul style="list-style-type: none"> <li>• a valid function pointer to enable UIM Status Change Indication notification</li> <li>• NULL to disable Band Preference notification</li> </ul>
-----------------------	--

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**Note**

Timeout: 2 seconds

**9.6.5.79 ULONG SLQSVoiceInfoRecCallback ( tFNInfoRec *pCallback* )**

Enables/disables the Voice information Record callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs. (Applicable only for 3GPP2)

**Parameters**

<i>pCallback</i> [IN]	<ul style="list-style-type: none"> <li>• Callback function pointer (0-Disable)</li> </ul>
-----------------------	---

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Technology Supported: CDMA  
Timeout: 2 seconds

**9.6.5.80 ULONG SLQSVoiceSetAllCallStatusCallBack ( tFNAllCallStatus *pCallback* )**

Enables/disables Voice Call Status Callback function. User can subscribe this callback get the call state change notifications. eg:- Call originated, connected, or ended. Whenever there is a change in the call information, there will be a indication with the information.

**Parameters**

<i>pCallback</i> [IN]	<ul style="list-style-type: none"> <li>• Callback function pointer (0 - Disable)</li> </ul>
-----------------------	---

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**9.6.5.81 ULONG SLQSVoiceSetDTMFEventCallBack ( tFNDTMFEvent pCallback )**

Enables/disables the DTMF Event callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

**Parameters**

<i>pCallback</i> [IN]	<ul style="list-style-type: none"> <li>• Callback function pointer (0-Disable)</li> </ul>
-----------------------	---

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Timeout: 2 seconds

This callback communicates that a DTMF event has been received.

**9.6.5.82 ULONG SLQSVoiceSetOTASPStatusCallBack ( tFNOTASPStatus pCallback )**

Enables/disables OTASP(Over-The-Air Service Provisioning) or OTAPA(Over-The-Air Parameter Administration) event CallBack Function (applicable only for 3GPP2). The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

**Parameters**

<i>pCallback</i> [IN]	<ul style="list-style-type: none"> <li>• a valid function pointer to enable OTASP or OTAPA event Indication notification</li> <li>• NULL to disable OTASP or OTAPA event, Indication notification</li> </ul>
-----------------------	--

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**Note**

Technology Supported: CDMA

Timeout: 10 seconds

This indication communicates the occurrence of an OTASP or OTAPA event. This indication is only applicable for 3GPP2 devices.

**9.6.5.83 ULONG SLQSVoiceSetPrivacyChangeCallBack ( tFNPrivacyChange pCallback )**

Enables/disables the voice privacy change callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

**Parameters**

<i>pCallback</i> [IN]	<ul style="list-style-type: none"><li>• Callback function pointer (0-Disable)</li></ul>
-----------------------	---

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Technology Supported: CDMA

Timeout: 2 seconds

This callback communicates a change in the voice privacy of a call. This is applicable only in 3GPP2 devices.

**9.6.5.84 ULONG SLQSVoiceSetSUPSCallBack ( tFNSUPSInfo pCallback )**

Enables/disables the SUPS callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

**Parameters**

<i>pCallback</i> [IN]	<ul style="list-style-type: none"><li>• Callback function pointer (0-Disable)</li></ul>
-----------------------	---

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Technology Supported: GSM

Timeout: 2 seconds

This callback notifies clients about the modem-originated supplementary service requests and the responses received from the network.

**9.6.5.85    ULONG SLQSVoiceSetSUPSNotificationCallback ( tFNSUPSNotification *pCallback* )**

Enables/disables the supplementary service notification callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.



## Parameters

<i>pCallback</i> [IN]	<ul style="list-style-type: none"> <li>• Callback function pointer (0-Disable)</li> </ul>
-----------------------	---

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 2 seconds

#### 9.6.5.86 ULONG SLQSWmsAsyncRawSendCallBack ( tFNAsyncRawSend *pCallback* )

Enables/disables the SLQSWmsAsyncRawSendCallBack callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

## Parameters

<i>pCallback</i> [IN]	<ul style="list-style-type: none"> <li>• Callback function pointer (0-Disable)</li> </ul>
-----------------------	---

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 2 seconds

#### 9.6.5.87 ULONG SLQSWmsMemoryFullCallBack ( tFNMemoryFull *pCallback* )

Enables/disables the event related to memory full status callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

## Parameters

<i>pCallback</i> [IN]	<ul style="list-style-type: none"> <li>• Callback function pointer (0-Disable)</li> </ul>
-----------------------	---

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**9.6.5.88   ULONG SLQSWmsMessageWaitingCallBack ( tFNMessageWaiting *pCallback* )**

Enables/disables the event related to message waiting information callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

## Parameters

<i>pCallback</i> [IN]	<ul style="list-style-type: none"> <li>• Callback function pointer (0-Disable)</li> </ul>
-----------------------	---

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## 9.7 qaGobiApiDcs.h File Reference

Device Connectivity Service API function prototypes.

### Data Structures

- struct [DcsUsbPortNames](#)
- struct [QosMap](#)
- struct [NetStats](#)

### Macros

- #define [LEN](#) 10
- #define [PORTNAM\\_LEN](#) 32

### Functions

- [ULONG QCWWAN2kEnumerateDevices](#) ([BYTE](#) \*pDevicesSize, [BYTE](#) \*pDevices)
- [ULONG QCWWAN2kConnect](#) ([CHAR](#) \*pDeviceID, [CHAR](#) \*pDeviceKey)
- [ULONG QCWWANDisconnect](#) ()
- [ULONG QCWWAN2kGetConnectedDeviceID](#) ([ULONG](#) deviceIDSize, [CHAR](#) \*pDeviceID, [ULONG](#) device-KeySize, [CHAR](#) \*pDeviceKey)
- [ULONG QCWWANEnumerateDevices](#) ([BYTE](#) \*pDevicesSize, [BYTE](#) \*pDevices)
- [ULONG QCWWANConnect](#) ([CHAR](#) \*pDeviceID, [CHAR](#) \*pDeviceKey)
- [ULONG SetSDKImagePath](#) ([LPCSTR](#) pPath)
- [ULONG SLQSGetUsbPortNames](#) (struct [DcsUsbPortNames](#) \*pUsbPortNames)
- [ULONG SLQSStart\\_AVAgent](#) ([BYTE](#) modem\_index)
- [ULONG SLQSStart](#) ([BYTE](#) modem\_index, [CHAR](#) \*usb\_path)
- [ULONG SLQSKillSDKProcess](#) ()
- [ULONG SLQSGetDeviceMode](#) ([BYTE](#) \*pDeviceMode)
- [ULONG SLQSStartSrv](#) ([BYTE](#) action, [BYTE](#) mask)
- [ULONG SLQSSetLoggingMask](#) ([BYTE](#) mask)
- [ULONG SLQSQosMap](#) ([BYTE](#) instance, [BYTE](#) dscp, [ULONG](#) qos\_id)
- [ULONG SLQSQosEditMap](#) ([BYTE](#) instance, [BYTE](#) dscp, [ULONG](#) qos\_id)
- [ULONG SLQSQosReadMap](#) ([BYTE](#) instance, [BYTE](#) dscp, [ULONG](#) \*qos\_id)
- [ULONG SLQSQosDumpMap](#) ([BYTE](#) instance, struct [QosMap](#) \*pmap, [BYTE](#) \*plen)
- [ULONG SLQSQosUnmap](#) ([BYTE](#) instance, [BYTE](#) dscp)
- [ULONG SLQSQosClearMap](#) ([BYTE](#) instance)
- [ULONG SLQSGetNetStatistic](#) (struct [NetStats](#) \*pNetStatistic, [BYTE](#) instance)

### 9.7.1 Detailed Description

Device Connectivity Service API function prototypes.

### 9.7.2 Macro Definition Documentation

#### 9.7.2.1 #define LEN 10

#### 9.7.2.2 #define PORTNAM\_LEN 32

This structure contains the SLQSGetUsbPortNames Information

#### Parameters

<i>AtCmdPort</i>	[OUT] <ul style="list-style-type: none"><li>Name of AT command port</li></ul>
<i>NmeaPort</i>	[OUT] <ul style="list-style-type: none"><li>Name of NMEA port</li></ul>
<i>DmPort</i>	[OUT] <ul style="list-style-type: none"><li>Name of DM port</li></ul>

#### Note

Technology Supported: UMTS/CDMA  
Device Supported: MC83x5, MC7700/10/50  
Timeout: 2 seconds  
[Port](#) names are limited to 32 characters.

### 9.7.3 Function Documentation

#### 9.7.3.1 ULONG QCWWAN2kConnect ( CHAR \* *pDeviceID*, CHAR \* *pDeviceKey* )

Connects the Connection Manager API to the first detected QC WWAN device. This function MUST be called after QCWWAN2kEnumerateDevices has been called.

#### Parameters

<i>pDeviceID</i> [IN]	<ul style="list-style-type: none"><li>Device path pertaining to the device for which the API is being invoked e.g. /dev/qcqm0.</li></ul>
<i>pDeviceKey</i> [IN]	<ul style="list-style-type: none"><li>Device key pertaining to the device for which the API is being invoked</li></ul>

#### Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_ERR\_NO\_DEVICE otherwise

#### Note

Timeout: 2 seconds

### 9.7.3.2 ULONG QCWWAN2kEnumerateDevices ( BYTE \* *pDevicesSize*, BYTE \* *pDevices* )

Enumerates the QC WWAN devices currently attached to the host. This API MUST be called before any other API.

## Parameters

<i>pDeviceSize</i> [IN/OUT]	<ul style="list-style-type: none"> <li>• Upon input, maximum number of elements that the device array can contain.</li> <li>• Upon successful output, actual number of elements in the device array.</li> </ul>
<i>pDevices</i> [IN/OUT]	<ul style="list-style-type: none"> <li>• Device array; array elements are structures with the following elements:  CHAR deviceId[256] - Device path (e.g. /dev/qcqmio)  CHAR deviceKey[16] - Device key stored in the device (e.g. A1000004B01051)</li> </ul>

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_ERR\_NO\_DEVICE on otherwise

## Note

Timeout: 2 seconds

### 9.7.3.3 ULONG QCWWAN2kGetConnectedDeviceID ( ULONG *deviceIdSize*, CHAR \* *pDeviceID*, ULONG *deviceKeySize*, CHAR \* *pDeviceKey* )

Returns the device ID and device key of the currently connected QC WWAN device.

## Parameters

<i>deviceIdSize</i>	<ul style="list-style-type: none"> <li>• Maximum number of characters (including NULL terminator) that the device ID array can contain.</li> </ul>
<i>pDeviceID</i> [OUT]	<ul style="list-style-type: none"> <li>• Device path string</li> </ul>
<i>deviceKeySize</i>	<ul style="list-style-type: none"> <li>• Maximum number of characters (including NULL terminator) that the device key array can contain.</li> </ul>
<i>pDeviceKey</i> [OUT]	<ul style="list-style-type: none"> <li>• Device key string</li> </ul>

## Returns

eQCWWAN\_ERR\_NONE if device found, eQCWWAN\_ERR\_NO\_DEVICE otherwise

## Note

Timeout: 2 seconds

### 9.7.3.4 ULONG QCWWANConnect ( CHAR \* *pDeviceID*, CHAR \* *pDeviceKey* )

Enumerates the QC WWAN devices currently attached to the host. This API MUST be called before any other API.

## Parameters

<i>pDeviceID</i> [IN]	<ul style="list-style-type: none"> <li>Device path pertaining to the device for which the API is being invoked e.g. /dev/qcqmio.</li> </ul>
<i>pDeviceKey</i> [IN]	<ul style="list-style-type: none"> <li>Device key pertaining to the device for which the API is being invoked</li> </ul>

## Returns

eQCWWAN\_ERR\_NONE if device found, eQCWWAN\_ERR\_NO\_DEVICE otherwise

## Note

Timeout: 2 seconds

This API is deprecated; use QCWWAN2kConnect instead

## 9.7.3.5 ULONG QCWWANDisconnect ( )

Disconnects the Connection Manager API from a previously connected QC device. This function de-registers all the callback functions that have been registered.

## Parameters

<i>none</i>	
-------------	--

## Returns

eQCWWAN\_ERR\_NONE

## Note

Timeout: 2 seconds

9.7.3.6 ULONG QCWWANEnumerateDevices ( BYTE \* *pDevicesSize*, BYTE \* *pDevices* )

Enumerates the QC WWAN devices currently attached to the host. This API is deprecated; use QCWWAN2kEnumerateDevices instead.

## Parameters

<i>pDeviceSize</i> [IN/-OUT]	<ul style="list-style-type: none"> <li>Upon input, maximum number of elements that the device array can contain.</li> <li>Upon successful output, actual number of elements in the device array.</li> </ul>
------------------------------	---

<i>pDevices</i> [IN/OUT]	<ul style="list-style-type: none"> <li>Device array; array elements are structures with the following elements:  CHAR deviceId[256] - Device path (e.g. /dev/qcqmio)  CHAR deviceKey[16] - Device key stored in the device</li> </ul>
--------------------------	---

**Returns**

eQCWWAN\_ERR\_NONE

**Note**

Timeout: 2 seconds

This API must be called prior to any other APIs.

**9.7.3.7 ULONG SetSDKImagePath ( LPCSTR pPath )**

Set the location of the SLQS executable

**Parameters**

<i>pPath</i> [IN]	- Pointer to fully qualified path of SLQS executable (includes the executable file's name)
-------------------	--

**Returns**

eQCWWAN\_ERR\_NONE

**Note**

Timeout: None

**9.7.3.8 ULONG SLQSGetDeviceMode ( BYTE \* pDeviceMode )**

Returns the Device Mode

**Parameters**

<i>pDeviceMode</i> [OUT]	<ul style="list-style-type: none"> <li>Pointer to SLQS Device Mode of type <a href="#">eDevState</a></li> </ul>
--------------------------	---

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**see [qmerrno.h](#) for eQCWWAN\_xxx error values**Note**

Timeout: 2 seconds

**9.7.3.9 ULONG SLQSGetNetStatistic ( struct NetStats \* pNetStatistic, BYTE instance )**

Returns the usbnet statistics for a particular PDN.



## Parameters

	<i>pNetStatistic[OUT]</i>	<ul style="list-style-type: none"> <li>• Pointer to the structure <a href="#">NetStats</a> which the value of every member is to be retrieved</li> </ul>
in	<i>instance</i>	<ul style="list-style-type: none"> <li>• PDP Instance id</li> </ul>

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

see [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 5 seconds

#### 9.7.3.10 ULONG SLQSGetUsbPortNames ( struct DcsUsbPortNames \* pUsbPortNames )

Returns the Usb [Port](#) Names currently in use.

## Parameters

<i>pUsbPortNames[OUT]</i>	<ul style="list-style-type: none"> <li>• Pointer to SLQS USB <a href="#">Port</a> Names Information</li> </ul>
---------------------------	--

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

see [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 2 seconds

#### 9.7.3.11 ULONG SLQSKillSDKProcess ( )

Kill the SDK process

## Parameters

<i>none</i>	
-------------	--

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

see [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Timeout: N/A

This API useful if the application was started with non-root privileges as subsequent attempt to start any application will fail because the SDK requires root permission to access /dev/qcqm device special files.

**9.7.3.12 ULONG SLQSQosClearMap ( BYTE *instance* )**

Clear Differential Service Code Point(DSCP) to QoS identifier association

**Parameters**

<i>in</i>	<i>instance</i>	<ul style="list-style-type: none"> <li>PDP Instance id</li> </ul>
-----------	-----------------	---

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**9.7.3.13 ULONG SLQSQosDumpMap ( BYTE *instance*, struct QosMap \* *pmap*, BYTE \* *plen* )**

Dump all association of Differential Service Code Point(DSCP) with QoS identifier

**Parameters**

<i>in</i>	<i>instance</i>	<ul style="list-style-type: none"> <li>PDP Instance id</li> </ul>
<i>out</i>	<i>pmap</i>	<ul style="list-style-type: none"> <li>Pointer to <a href="#">QosMap</a> struct</li> </ul>
<i>out</i>	<i>plen</i>	<ul style="list-style-type: none"> <li>number of QoSMap element</li> </ul>

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

#### 9.7.3.14 ULONG SLQSQosEditMap ( BYTE *instance*, BYTE *dscp*, ULONG *qos\_id* )

Edit association of Differential Service Code Point(DSCP) with QoS identifier

**Parameters**

in	<i>instance</i>	• PDP Instance id
in	<i>dscp</i>	
in	<i>qos_id</i>	

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**9.7.3.15 ULONG SLQSQosMap ( BYTE *instance*, BYTE *dscp*, ULONG *qos\_id* )**

Associate Differential Service Code Point(DSCP) with QoS identifier

**Parameters**

in	<i>instance</i>	• PDP Instance id
in	<i>dscp</i>	
in	<i>qos_id</i>	

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**9.7.3.16 ULONG SLQSQosReadMap ( BYTE *instance*, BYTE *dscp*, ULONG \* *qos\_id* )**

Read association of Differential Service Code Point(DSCP) with QoS identifier

**Parameters**

in	<i>instance</i>	• PDP Instance id
in	<i>dscp</i>	
out	<i>qos_id</i>	

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

#### 9.7.3.17 ULONG SLQSQosUnmap ( BYTE *instance*, BYTE *dscp* )

Remove Differential Service Code Point(DSCP) to QoS identifier association

## Parameters

in	<i>instance</i>	<ul style="list-style-type: none"> <li>• PDP Instance id</li> </ul>
in	<i>dscp</i>	

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

### 9.7.3.18 ULONG SLQSSetLoggingMask ( BYTE mask )

Limit Syslog messages according to the Mask provided by user

## Parameters

<i>mask</i>	<ul style="list-style-type: none"> <li>• Mask 0x01: disable all log</li> <li>• Mask 0xFF: enable all log</li> </ul>
-------------	---

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

see [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 5 seconds

### 9.7.3.19 ULONG SLQSStart ( BYTE modem\_index, CHAR \* usb\_path )

Create the SDK process and IPC sockets for the Application and SDK processes to communicate over.

## Parameters

in	<i>modem_index</i>	<ul style="list-style-type: none"> <li>• 0: first modem detected</li> <li>• 1: second modem detected</li> <li>• 2: third modem detected</li> <li>• ...</li> <li>• 7: seventh modem detected</li> </ul>
in	<i>usb_path</i>	optional usb path for multi modem scenario when specified, the modem_index will mapping to usb_path e.g.: set usb_path to '2-2' to match modem at /sys/bus/usb/devices/2-2/ Please set to NULL when not used

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

see [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Timeout: N/A

**9.7.3.20 ULONG SLQSStart\_AVAgent ( BYTE modem\_index )**

Create IPC sockets for AirVantage Agent and SDK processes to communicate over

**Parameters**

<i>none</i>	
-------------	--

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

see [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Timeout: N/A

**9.7.3.21 ULONG SLQSStartSrv ( BYTE action, BYTE mask )**

Registers/deregisters for service with unsolicited notifications

**Parameters**

<i>action, 1</i> <i>mask</i>	for register, 0 for deregister  <ul style="list-style-type: none"><li>• Bit mask for unsolicited notifications<ul style="list-style-type: none"><li>– Bit0 - WDS</li><li>– Bit1 - NAS</li><li>– Bit2 - PDS</li><li>– Bit3 - VOICE</li></ul></li></ul>
---------------------------------	---

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

see [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Timeout: 2 seconds API is useful to register for the services which supports unsolicited notifications. Registration/deregistration can be done by using parameter action if action is set then the mask (set bits) will be used for registering service and if action is "0" mask(set bits) will be used to deregister services. For example : bit mask 0x03 - Registers for services WDS and NAS if action is "1" and deregisters WDS and NAS if action is "0".

## 9.8 qaGobiApiDms.h File Reference

Device Management Service API function prototypes.

### Data Structures

- struct [serialNumbersInfo](#)
- struct [ERIFileparams](#)
- struct [custFeaturesInfo](#)
- struct [custFeaturesSetting](#)
- struct [dmsCurrentPRLInfo](#)
- struct [FactorySequenceNumber](#)
- struct [CurrImageInfo](#)
- struct [CurrentImgList](#)
- struct [FirmwareUpdatStat](#)
- struct [USBCompParams](#)
- struct [USBCompConfig](#)
- struct [CrashInfo](#)
- struct [CrashInfoParams](#)
- struct [\\_SLQSSwiGetHostDevInfoParams](#)
- struct [\\_SLQSSwiSetHostDevInfoParams](#)
- struct [\\_SLQSSwiGetOSInfoParams](#)
- struct [\\_SLQSSwiSetOSInfoParams](#)
- struct [\\_SLQSSwiGetSerialNoExtParams](#)
- struct [setCustomSettingV2](#)
- struct [getCustomInput](#)
- struct [custSettingInfo](#)
- struct [custSettingList](#)
- struct [getCustomFeatureV2](#)

### Macros

- #define [MAX\\_FSN\\_LENGTH](#) 255
- #define [MAX\\_BUILD\\_ID\\_LEN](#) 255
- #define [UNIQUE\\_ID\\_LEN](#) 16
- #define [IMGDETAILS\\_LEN](#) 16
- #define [MAX\\_CUST\\_ID\\_LEN](#) 64
- #define [MAX\\_CUST\\_VALUE\\_LEN](#) 8



## Typedefs

- typedef struct [serialNumbersInfo](#) serialNumbersInfo
- typedef struct [ERIFileparams](#) ERIFileparams
- typedef struct [custFeaturesInfo](#) custFeaturesInfo
- typedef struct [custFeaturesSetting](#) custFeaturesSetting
- typedef struct [dmsCurrentPRLInfo](#) dmsCurrentPRLInfo
- typedef struct  
    [\\_SLQSSwiGetHostDevInfoParams](#) SLQSSwiGetHostDevInfoParams
- typedef struct  
    [\\_SLQSSwiSetHostDevInfoParams](#) SLQSSwiSetHostDevInfoParams
- typedef struct  
    [\\_SLQSSwiGetOSInfoParams](#) SLQSSwiGetOSInfoParams
- typedef struct  
    [\\_SLQSSwiSetOSInfoParams](#) SLQSSwiSetOSInfoParams
- typedef struct  
    [\\_SLQSSwiGetSerialNoExtParams](#) SLQSSwiGetSerialNoExtParams

## Functions

- [ULONG GetManufacturer](#) (BYTE stringSize, CHAR \*pString)
- [ULONG GetModelID](#) (BYTE stringSize, CHAR \*pString)
- [ULONG GetFirmwareRevision](#) (BYTE stringSize, CHAR \*pString)
- [ULONG GetFirmwareRevisions](#) (BYTE amssSize, CHAR \*pAMSSString, BYTE bootSize, CHAR \*pBootString, BYTE priSize, CHAR \*pPRIString)
- [ULONG GetPRLVersion](#) (WORD \*pPRLVersion)
- [ULONG GetIMSI](#) (BYTE stringSize, CHAR \*pString)
- [ULONG GetSerialNumbers](#) (BYTE esnSize, CHAR \*pESNString, BYTE imeiSize, CHAR \*pIMEIString, BYTE meidSize, CHAR \*pMEIDString)
- [ULONG SLQSGetSerialNumbers](#) ([serialNumbersInfo](#) \*pSerialNumbersInfo)
- [ULONG GetHardwareRevision](#) (BYTE stringSize, CHAR \*pString)
- [ULONG GetNetworkTime](#) (ULONGLONG \*pTimeStamp, ULONG \*pTimeSource)
- [ULONG UIMSetPINProtection](#) (ULONG id, ULONG bEnable, CHAR \*pValue, ULONG \*pVerifyRetriesLeft, ULONG \*pUnblockRetriesLeft)
- [ULONG UIMUnblockPIN](#) (ULONG id, CHAR \*pPUKValue, CHAR \*pNewValue, ULONG \*pVerifyRetriesLeft, ULONG \*pUnblockRetriesLeft)
- [ULONG UIMVerifyPIN](#) (ULONG id, CHAR \*pValue, ULONG \*pVerifyRetriesLeft, ULONG \*pUnblockRetriesLeft)
- [ULONG UIMChangePIN](#) (ULONG id, CHAR \*pOldValue, CHAR \*pNewValue, ULONG \*pVerifyRetriesLeft, ULONG \*pUnblockRetriesLeft)
- [ULONG GetVoiceNumber](#) (BYTE voiceNumberSize, CHAR \*pVoiceNumber, BYTE minSize, CHAR \*pMIN)
- [ULONG SetPower](#) (ULONG powerMode)
- [ULONG GetPower](#) (ULONG \*pPowerMode)
- [ULONG UIMGetControlKeyStatus](#) (ULONG id, ULONG \*pStatus, ULONG \*pVerifyRetriesLeft, ULONG \*pUnblockRetriesLeft)
- [ULONG UIMGetICCID](#) (BYTE stringSize, CHAR \*pString)
- [ULONG UIMGetPINStatus](#) (ULONG id, ULONG \*pStatus, ULONG \*pVerifyRetriesLeft, ULONG \*pUnblockRetriesLeft)
- [ULONG GetOfflineReason](#) (ULONG \*pReasonMask, ULONG \*pbPlatform)
- [ULONG UIMSetControlKeyProtection](#) (ULONG id, ULONG status, CHAR \*pValue, ULONG \*pVerifyRetriesLeft)
- [ULONG UIMUnblockControlKey](#) (ULONG id, CHAR \*pValue, ULONG \*pUnblockRetriesLeft)
- [ULONG GetDeviceCapabilities](#) (ULONG \*pMaxTXChannelRate, ULONG \*pMaxRXChannelRate, ULONG \*pDataServiceCapability, ULONG \*pSimCapability, ULONG \*pRadioIfacesSize, BYTE \*pRadioIfaces)
- [ULONG ResetToFactoryDefaults](#) (CHAR \*pSPC)

- [ULONG ValidateSPC](#) ([CHAR](#) \*pSPC)
- [ULONG ActivateAutomatic](#) ([CHAR](#) \*pActivationCode)
- [ULONG SLQSGetERIFile](#) ([ERIFileparams](#) \*pERIFileparams)
- [ULONG GetActivationState](#) ([ULONG](#) \*pActivationState)
- [ULONG SLQSUIMGetState](#) ([ULONG](#) \*pUIMState)
- [ULONG SLQSGetBandCapability](#) ([ULONGLONG](#) \*pBandCapability)
- [ULONG SLQSGetCustFeatures](#) ([custFeaturesInfo](#) \*pCustFeaturesInfo)
- [ULONG SLQSSetCustFeatures](#) ([custFeaturesSetting](#) \*pCustFeaturesSetting)
- [ULONG SLQSGetCurrentPRLInfo](#) ([dmsCurrentPRLInfo](#) \*pCurrentPRLInfo)
- [ULONG SLQSSwiGetFSN](#) ([FactorySequenceNumber](#) \*pFSNumber)
- [ULONG SLQSSwiGetFirmwareCurr](#) ([CurrentImgList](#) \*pCurrentImgList)
- [ULONG SLQSSwiGetFwUpdateStatus](#) ([FirmwareUpdatStat](#) \*pFirmwareUpdatStat)
- [ULONG SLQSSwiGetUSBComp](#) ([USBCompParams](#) \*pUSBCompParams)
- [ULONG SLQSSwiSetUSBComp](#) ([USBCompConfig](#) \*pUSBCompConfig)
- [ULONG SLQSSwiGetCrashInfo](#) ([BYTE](#) \*pClear, [CrashInfoParams](#) \*pCrashInfoParams)
- [ULONG SLQSSwiGetCrashAction](#) ([BYTE](#) \*pDevCrashState)
- [ULONG SLQSSwiSetCrashAction](#) ([BYTE](#) crashActionParams)
- [ULONG SLQSSwiGetHostDevInfo](#) ([SLQSSwiGetHostDevInfoParams](#) \*pGetHostDevInfoParams)
- [ULONG SLQSSwiSetHostDevInfo](#) ([SLQSSwiSetHostDevInfoParams](#) \*pSetHostDevInfoParams)
- [ULONG SLQSSwiGetOSInfo](#) ([SLQSSwiGetOSInfoParams](#) \*pParams)
- [ULONG SLQSSwiSetOSInfo](#) ([SLQSSwiSetOSInfoParams](#) \*pParams)
- [ULONG SLQSSwiGetSerialNoExt](#) ([SLQSSwiGetSerialNoExtParams](#) \*pParams)
- [ULONG SLQSSetCustFeaturesV2](#) ([setCustomSettingV2](#) \*pSetCustSetting)
- [ULONG SLQSGetCustFeaturesV2](#) ([getCustomFeatureV2](#) \*pGetCustomFeatureV2)

### 9.8.1 Detailed Description

Device Management Service API function prototypes.

### 9.8.2 Macro Definition Documentation

9.8.2.1 `#define IMGDETAILS_LEN 16`

9.8.2.2 `#define MAX_BUILD_ID_LEN 255`

9.8.2.3 `#define MAX_CUST_ID_LEN 64`

9.8.2.4 `#define MAX_CUST_VALUE_LEN 8`

9.8.2.5 `#define MAX_FSN_LENGTH 255`

9.8.2.6 `#define UNIQUE_ID_LEN 16`

### 9.8.3 Typedef Documentation

9.8.3.1 `typedef struct custFeaturesInfo custFeaturesInfo`

This structure contains current settings of custom features

Parameters

---

<i>GpsEnable</i>	<ul style="list-style-type: none"> <li>describes if GPS is enabled or disabled</li> <li>values: <ul style="list-style-type: none"> <li>0x00 - GPS is disabled</li> <li>0x01 - GPS is enabled</li> </ul> </li> <li>function <a href="#">SLQSGetCustFeatures()</a> returns a default value FFFFFFFF if no value is returned by the modem</li> </ul>
<i>pDisableIMSI</i>	<ul style="list-style-type: none"> <li>optional 1 byte parameter</li> <li>describes if IMSI display is enabled or disabled</li> <li>values: <ul style="list-style-type: none"> <li>0x00 - Allow display of IMSI</li> <li>0x01 - Do not display IMSI</li> </ul> </li> <li>function <a href="#">SLQSGetCustFeatures()</a> returns a default value FF if no value is returned by the modem</li> </ul>
<i>pIPFamSupport</i>	<ul style="list-style-type: none"> <li>optional 2 byte BitMask</li> <li>bitmask representing the IP families supported</li> <li>values: <ul style="list-style-type: none"> <li>0x01 - IPv4</li> <li>0x02 - IPv6</li> <li>0x04 - IPv4v6</li> </ul> </li> <li>function <a href="#">SLQSGetCustFeatures()</a> returns a default value FFFF if no value is returned by the modem</li> </ul>
<i>pRMAuto-Connect</i>	<ul style="list-style-type: none"> <li>optional 1 byte parameter</li> <li>QMI Mode RM Net Auto Connect Support</li> <li>values: <ul style="list-style-type: none"> <li>0x00 - Not Supported</li> <li>0x01 - Supported</li> </ul> </li> <li>function <a href="#">SLQSGetCustFeatures()</a> returns a default value FF if no value is returned by the modem</li> </ul>

<i>pGPSSel</i>	<ul style="list-style-type: none"> <li>• optional 1 byte parameter</li> <li>• GPS Antenna Select</li> <li>• values: <ul style="list-style-type: none"> <li>– 0x00 - Dedicated GPS <a href="#">Port</a></li> <li>– 0x01 - GPS Rx over AUX <a href="#">Port</a></li> <li>– 0x02 - GPS Rx over dedicated GPS port with no bias voltage applied</li> </ul> </li> <li>• function <a href="#">SLQSGetCustFeatures()</a> returns a default value FF if no value is returned by the modem</li> </ul>
<i>pSMSSupport</i>	<ul style="list-style-type: none"> <li>• optional 1 byte parameter</li> <li>• SMS support</li> <li>• values: <ul style="list-style-type: none"> <li>– 0x00 - Not supported</li> <li>– 0x01 - supported</li> </ul> </li> <li>• Used to determine whether or not to hide SMS from user</li> <li>• function <a href="#">SLQSGetCustFeatures()</a> returns a default value FF if no value is returned by the modem. In this case assume, SMS is supported.</li> </ul>
<i>plsVoiceEnabled</i>	<ul style="list-style-type: none"> <li>• optional 1 byte parameter</li> <li>• Voice support</li> <li>• values: <ul style="list-style-type: none"> <li>– 0x00 - Enable voice on both AT and QMI interface (default)</li> <li>– 0x01 - Reserved</li> <li>– 0x02 - Disable voice on both AT and QMI interface</li> </ul> </li> </ul>
<i>pDHCPRelay-Enabled</i>	<ul style="list-style-type: none"> <li>• optional 1 byte parameter</li> <li>• DHCP Relay support</li> <li>• values: <ul style="list-style-type: none"> <li>– 0x00 - Disable DHCP relay</li> <li>– 0x01 - Enable DHCP relay</li> </ul> </li> </ul>

<i>pGPSLPM</i>	<ul style="list-style-type: none"> <li>• optional 1 byte parameter</li> <li>• GPSLPM support</li> <li>• values: <ul style="list-style-type: none"> <li>– 0x00 - Enable GPS in Low Power Mode</li> <li>– 0x01 - Disable GPS in Low Power Mode</li> </ul> </li> </ul>
----------------	---

### 9.8.3.2 typedef struct custFeaturesSetting custFeaturesSetting

This structure contains settings to be used for custom features

#### Parameters

<i>pGPSSel</i>	<ul style="list-style-type: none"> <li>• optional 1 byte parameter</li> <li>• GPS Antenna Select</li> <li>• values: <ul style="list-style-type: none"> <li>– 0x00 - Dedicated GPS <a href="#">Port</a></li> <li>– 0x01 - GPS Rx over AUX <a href="#">Port</a></li> <li>– 0x02 - GPS Rx over dedicated GPS port with no bias voltage applied</li> </ul> </li> </ul>
<i>pGPSEnable</i>	<ul style="list-style-type: none"> <li>• optional 4 byte parameter</li> <li>• GPS Enable/Disable</li> <li>• values: The value of 7 least significant bits: <ul style="list-style-type: none"> <li>– 0 - Disabled</li> <li>– 1 - MT &amp; MO enabled</li> <li>– 2 - MO enabled</li> <li>– 3 - MT enabled</li> <li>– 4 - MT &amp; MO enabled if GPS_DISABLE pin is not asserted</li> <li>– 5 - MO GPS enabled if GPS_DISABLE pin is not asserted</li> <li>– 6 - MT GPS enabled if GPS_DISABLE pin is not asserted</li> </ul> </li> </ul>

**Note**

Only MC7750 3.5.x firmware supports above 0x04, 0x05 and 0x06 settings. To disable GLONASS, set the most significant bit - 0x80. This setting is only applicable if GPS is not Disabled.

**Parameters**

<i>pIsVoiceEnabled</i>	<ul style="list-style-type: none"> <li>• optional 1 byte parameter</li> <li>• voice enabled/disabled</li> <li>• values: <ul style="list-style-type: none"> <li>– 0 - Enable voice on both AT and QMI interface (default)</li> <li>– 1 - Reserved</li> <li>– 2 - Disable voice on both AT and QMI interface</li> </ul> </li> </ul>
<i>pDHCPRelay-Enabled</i>	<ul style="list-style-type: none"> <li>• optional 1 byte parameter</li> <li>• DHCPRELAYENABLE support</li> <li>• values: <ul style="list-style-type: none"> <li>– 0 - Disable DHCP relay</li> <li>– 1 - Enable DHCP relay</li> </ul> </li> </ul>
<i>pGPSLPM</i>	<ul style="list-style-type: none"> <li>• optional 1 byte parameter</li> <li>• GPSLPM support</li> <li>• values: <ul style="list-style-type: none"> <li>– 0 - Enable GPS in Low Power Mode</li> <li>– 1 - Disable GPS in Low Power Mode</li> </ul> </li> </ul>

**9.8.3.3 typedef struct dmsCurrentPRLInfo dmsCurrentPRLInfo**

This structure contains GetCurrentPRLInfo response parameter

**Parameters**

<i>pPRLVersion[OUT]</i>	- Optional <ul style="list-style-type: none"> <li>• PRL version of device.</li> </ul>
<i>pPRLPreference</i>	[OUT]- Optional <ul style="list-style-type: none"> <li>• PRL Preference <ul style="list-style-type: none"> <li>– 0 - Unset</li> <li>– 1 - Set</li> </ul> </li> </ul>

#### 9.8.3.4 typedef struct ERIFileparams ERIFileparams

This structure contains Extended Roaming Indicator(ERI) file parameters

## Parameters

<i>pFileSize[IN/OUT]</i>	<ul style="list-style-type: none"> <li>• Upon input, the maximum number of bytes that file contents array can contain.</li> <li>• Upon successful output, actual number of bytes written to file contents array</li> </ul>
<i>pFile[OUT]</i>	<ul style="list-style-type: none"> <li>• ERI data read from persistent storage( Max size is 1024 )</li> </ul>

## 9.8.3.5 typedef struct serialNumbersInfo serialNumbersInfo

Returns all the serial numbers assigned to the device. These serial numbers include the ESN (Electronic serial number of the device), the IMEI (International Mobile Equipment Identity) and MEID (Mobile Equipment Identifier).

## Parameters

<i>esnSize</i>	<ul style="list-style-type: none"> <li>• The maximum number of characters (including NULL terminator) that the ESN string array can contain</li> </ul>
<i>pESNString[OUT]</i>	<ul style="list-style-type: none"> <li>• NULL-terminated ESN string. Empty string is returned when ESN is not supported/programmed</li> </ul>
<i>imeiSize</i>	<ul style="list-style-type: none"> <li>• The maximum number of characters (including NULL terminator) that the IMEI string array can contain</li> </ul>
<i>pIMEIString[OUT]</i>	<ul style="list-style-type: none"> <li>• NULL terminated IMEI string. Empty string is returned when IMEI is not supported/programmed</li> </ul>
<i>meidSize</i>	<ul style="list-style-type: none"> <li>• The maximum number of characters (including NULL terminator) that the MEID string array can contain</li> </ul>
<i>pMEIDString[OUT]</i>	<ul style="list-style-type: none"> <li>• NULL-terminated MEID string. Empty string is returned when MEID is not supported/programmed</li> </ul>
<i>imeiSvnSize</i>	<ul style="list-style-type: none"> <li>• The maximum number of characters (including NULL terminator) that the IMEI SVN string array can contain.</li> </ul>
<i>pImeiSvnString[OUT]</i>	<ul style="list-style-type: none"> <li>• NULL-terminated IMEI SVN string. Empty string is returned when IMEI SVN is not supported/programmed.</li> </ul>



9.8.3.6 `typedef struct _SLQSSwiGetHostDevInfoParams SLQSSwiGetHostDevInfoParams`

9.8.3.7 `typedef struct _SLQSSwiGetOSInfoParams SLQSSwiGetOSInfoParams`

9.8.3.8 `typedef struct _SLQSSwiGetSerialNoExtParams SLQSSwiGetSerialNoExtParams`

This structure is used to store MEID Information

#### Parameters

<i>meidLength</i> [OUT]	<ul style="list-style-type: none"> <li>String length of the of MEID received</li> </ul>
<i>pMeidString</i> [OUT]	<ul style="list-style-type: none"> <li>Pointer to receive String containing the Mobile Equipment Identifier(MEID) of the device.</li> </ul>

9.8.3.9 `typedef struct _SLQSSwiSetHostDevInfoParams SLQSSwiSetHostDevInfoParams`

9.8.3.10 `typedef struct _SLQSSwiSetOSInfoParams SLQSSwiSetOSInfoParams`

## 9.8.4 Function Documentation

9.8.4.1 **ULONG** ActivateAutomatic ( **CHAR** \* *pActivationCode* )

Requests the device to perform automatic service activation

#### Parameters

<i>pActivationCode</i> [IN]	<ul style="list-style-type: none"> <li>NULL-terminated string representing activation code (maximum string length of 12); specific carrier requirements may dictate actual activation code that is applicable, e.g., "*22899"</li> </ul>
-----------------------------	--

#### Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

#### See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

#### Note

Timeout: 5 Minutes

9.8.4.2 **ULONG** GetActivationState ( **ULONG** \* *pActivationState* )

Returns the device activation state.

## Parameters

<i>pActivation-State[OUT]</i>	<ul style="list-style-type: none"> <li>• Service Activation Code</li> <li>0 - Service not activated</li> <li>1 - Service activated</li> <li>2 - Activation connecting</li> <li>3 - Activation connected</li> <li>4 - OTASP security authenticated</li> <li>5 - OTASP NAM downloaded</li> <li>6 - OTASP MDN downloaded</li> <li>7 - OTASP IMSI downloaded</li> <li>8 - OTASP PRL downloaded</li> <li>9 - OTASP SPC downloaded</li> <li>10 - OTASP settings committed</li> </ul>
-------------------------------	--

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Technology Supported: CDMA  
Timeout: 2 Seconds

#### 9.8.4.3 ULONG GetDeviceCapabilities ( ULONG \* pMaxTXChannelRate, ULONG \* pMaxRXChannelRate, ULONG \* pDataServiceCapability, ULONG \* pSimCapability, ULONG \* pRadiofacesSize, BYTE \* pRadiofaces )

Gets the device capabilities

## Parameters

<i>pMaxTX-ChannelRate[OUT]</i>	<ul style="list-style-type: none"> <li>• Maximum transmission rate (in bps) supported by the device</li> <li>• In multi-technology devices, this value will be the greatest rate among all supported technologies</li> </ul>
<i>pMaxRX-ChannelRate[OUT]</i>	<ul style="list-style-type: none"> <li>• Maximum reception rate (in bps) supported by the device</li> <li>• In multi-technology devices, this value will be the greatest rate among all supported technologies</li> </ul>

<i>pDataService-Capability[OUT]</i>	<ul style="list-style-type: none"> <li>CS/PS data service capability <ul style="list-style-type: none"> <li>0 - No data services supported</li> <li>1 - Only Circuit Switched (CS) services supported</li> <li>2 - Only Packet Switched (PS) services supported</li> <li>3 - Simultaneous CS and PS</li> <li>4 - Non-simultaneous CS and PS</li> </ul> </li> </ul>
<i>pSimCapability[-OUT]</i>	<ul style="list-style-type: none"> <li>Device SIM capability <ul style="list-style-type: none"> <li>0 - SIM not supported</li> <li>1 - SIM supported</li> </ul> </li> </ul>
<i>pRadioIfaces-Size[IN/OUT]</i>	<ul style="list-style-type: none"> <li>Upon input, the maximum number of elements that the radio interface array can contain</li> <li>Upon successful output, actual number of elements in the radio interface array</li> </ul>
<i>pRadioIfaces[OUT]</i>	<ul style="list-style-type: none"> <li>Radio interface array. This is a structure of array containing the elements below. ULONG radioInterface <ul style="list-style-type: none"> <li>See <a href="#">qaGobiApiTableRadioInterfaces.h</a> for Radio Interfaces</li> </ul> </li> </ul>

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Timeout: 2 seconds

**9.8.4.4 ULONG GetFirmwareRevision ( BYTE stringSize, CHAR \* pString )**

Returns the device firmware revision

**Parameters**

<i>stringSize</i>	<ul style="list-style-type: none"> <li>The maximum number of characters (including NULL terminator) that the string array can contain</li> </ul>
<i>pString[OUT]</i>	<ul style="list-style-type: none"> <li>NULL terminated string</li> </ul>

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Timeout: 2 seconds

#### 9.8.4.5 **ULONG** GetFirmwareRevisions ( **BYTE** *amssSize*, **CHAR** \* *pAMSSString*, **BYTE** *bootSize*, **CHAR** \* *pBootString*, **BYTE** *priSize*, **CHAR** \* *pPRIString* )

Returns the device firmware revisions (AMSS, boot, and PRI)

**Parameters**

<i>amssSize</i>	<ul style="list-style-type: none"> <li>Maximum number of characters (including NULL terminator) that the AMSS string array can contain</li> </ul>
<i>pAMSSString</i> [O-UT]	<ul style="list-style-type: none"> <li>NULL-terminated AMSS revision string</li> </ul>
<i>bootSize</i>	<ul style="list-style-type: none"> <li>Maximum number of characters (including NULL terminator) that the boot string array can contain</li> </ul>
<i>pBootString</i> [O-UT]	<ul style="list-style-type: none"> <li>NULL-terminated boot code revision string</li> </ul>
<i>priSize</i>	<ul style="list-style-type: none"> <li>Maximum number of characters (including NULL terminator) that the PRI string array can contain</li> </ul>
<i>pPRIString</i> [OUT]	<ul style="list-style-type: none"> <li>NULL-terminated PRI revision string</li> </ul>

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Timeout: 2 seconds

#### 9.8.4.6 **ULONG** GetHardwareRevision ( **BYTE** *stringSize*, **CHAR** \* *pString* )

Returns the hardware revision of the device

## Parameters

<i>stringSize</i>	<ul style="list-style-type: none"><li>• The maximum number of characters (including NULL terminator) that the string array can contain</li></ul>
<i>pString[OUT]</i>	<ul style="list-style-type: none"><li>• NULL terminated string</li></ul>

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 2 seconds

**9.8.4.7 ULONG GetIMSI ( BYTE *stringSize*, CHAR \* *pString* )**

Returns the device IMSI. This API is deprecated on MC73xx/EM73xx modules since firmware version SWI9X15C-05\_xx\_xx\_xx and all EM74xx firmware versions.

## Parameters

<i>stringSize</i>	<ul style="list-style-type: none"><li>• The maximum number of characters (including NULL terminator) that the string array can contain</li></ul>
<i>pString[OUT]</i>	<ul style="list-style-type: none"><li>• NULL terminated string</li></ul>

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 2 seconds

**9.8.4.8 ULONG GetManufacturer ( BYTE *stringSize*, CHAR \* *pString* )**

Returns the device manufacturer name

## Parameters

<i>stringSize</i>	<ul style="list-style-type: none"> <li>The maximum number of characters (including NULL terminator) that the string array can contain.</li> </ul>
<i>pString[OUT]</i>	<ul style="list-style-type: none"> <li>NULL terminated string</li> </ul>

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 2 seconds

#### 9.8.4.9 ULONG GetModelID ( BYTE *stringSize*, CHAR \* *pString* )

Returns the device model ID

## Parameters

<i>stringSize</i>	<ul style="list-style-type: none"> <li>The maximum number of characters (including NULL terminator) that the string array can contain</li> </ul>
<i>pString[OUT]</i>	<ul style="list-style-type: none"> <li>NULL terminated string</li> </ul>

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 2 seconds

#### 9.8.4.10 ULONG GetNetworkTime ( ULONGLONG \* *pTimeStamp*, ULONG \* *pTimeSource* )

Returns the current time of the device based on the value supported by the network.

## Parameters

<i>pTimeStamp</i> [O-UT]	<ul style="list-style-type: none"> <li>Count of 1.25 ms that have elapsed from the start of GPS time (Jan 6, 1980)</li> </ul>
<i>pTimeSource</i> [O-UT]	<ul style="list-style-type: none"> <li>Source of timestamp               <ul style="list-style-type: none"> <li>0 - 32 kHz device clock</li> <li>1 - CDMA network</li> <li>2 - cdma2000 1xEV-DO network</li> </ul> </li> </ul>

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 2 seconds

The source of the timestamp provided specifies how the timestamp was determined. The first network time that is available will be returned. If no network time is available, the timestamp is taken from the 32 kHz slow-clock of the device.

#### 9.8.4.11 ULONG GetOfflineReason ( ULONG \* *pReasonMask*, ULONG \* *pbPlatform* )

Returns reason why the operating mode of the device is currently offline.

## Parameters

<i>pReasonMask</i> [OUT]	<ul style="list-style-type: none"> <li>Bitmask of offline reasons               <ul style="list-style-type: none"> <li>0x00000001 - Host image configuration issue</li> <li>0x00000002 - PRI image configuration issue</li> <li>0x00000004 - PRI version incompatible</li> <li>0x00000008 - PRI copy issue</li> <li>All others - Reserved</li> </ul> </li> </ul>
<i>pbPlatform</i> [OUT]	<ul style="list-style-type: none"> <li>Is the device offline due to a platform restriction?               <ul style="list-style-type: none"> <li>0 - No</li> <li>1 - Yes</li> </ul> </li> </ul>

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Timeout: 2 Seconds

**9.8.4.12 ULONG GetPower ( ULONG \* pPowerMode )**

Returns the operating mode of the device.

**Parameters**

<i>pPowerMode</i> [O-UT]	<ul style="list-style-type: none"> <li>Selected operating mode               <ul style="list-style-type: none"> <li>See <a href="#">qaGobiApiTablePowerModes.h</a> for power modes</li> </ul> </li> </ul>
--------------------------	---

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Timeout: 2 Seconds

**9.8.4.13 ULONG GetPRLVersion ( WORD \* pPRLVersion )**

Returns the version of the active Preferred Roaming List (PRL) in use by the device.

**Parameters**

<i>pPRLVersion</i> [O-UT]	<ul style="list-style-type: none"> <li>PRL version number</li> </ul>
---------------------------	--

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Timeout: 2 seconds

**9.8.4.14 ULONG GetSerialNumbers ( BYTE esnSize, CHAR \* pESNString, BYTE imeiSize, CHAR \* pIMEIString, BYTE meidSize, CHAR \* pMEIDString )**

Returns all the serial numbers assigned to the device. These serial numbers include the ESN (Electronic serial number of the device), the IMEI (International Mobile Equipment Identity) and MEID (Mobile Equipment Identifier).



## Parameters

<i>esnSize</i>	<ul style="list-style-type: none"> <li>The maximum number of characters (including NULL terminator) that the ESN string array can contain</li> </ul>
<i>pESNString[OUT]</i>	<ul style="list-style-type: none"> <li>NULL-terminated ESN string. Empty string is returned when ESN is not supported/programmed</li> </ul>
<i>imeiSize</i>	<ul style="list-style-type: none"> <li>The maximum number of characters (including NULL terminator) that the IMEI string array can contain</li> </ul>
<i>pIMEIString[OUT]</i>	<ul style="list-style-type: none"> <li>NULL terminated IMEI string. Empty string is returned when IMEI is not supported/programmed</li> </ul>
<i>meidSize</i>	<ul style="list-style-type: none"> <li>The maximum number of characters (including NULL terminator) that the MEID string array can contain</li> </ul>
<i>pMEIDString[OUT]</i>	<ul style="list-style-type: none"> <li>NULL-terminated MEID string. Empty string is returned when MEID is not supported/programmed</li> </ul>

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

For CDMA devices that use a RUIM, the MEID of the RUIM (if present) will be returned. Use [SLQSSwiGetSerialNoExt\(\)](#) to get MEID of CDMA modems. Timeout: 2 seconds

#### 9.8.4.15 ULONG GetVoiceNumber ( BYTE voiceNumberSize, CHAR \* pVoiceNumber, BYTE minSize, CHAR \* pMIN )

Returns the voice number in use by the device

## Parameters

<i>voiceNumber-Size</i>	<ul style="list-style-type: none"> <li>Maximum number of characters (including NULL terminator) that the voice number array can contain.</li> </ul>
-------------------------	---

<i>pVoiceNumber[-OUT]</i>	<ul style="list-style-type: none"> <li>Voice number string: MDN or MS ISDN</li> </ul>
<i>minSize</i>	<ul style="list-style-type: none"> <li>Maximum number of characters (including NULL terminator) that the MIN array can contain.</li> </ul>
<i>pMIN</i>	<ul style="list-style-type: none"> <li>MIN string: Empty string returned when MIN is not supported/ programmed.</li> </ul>

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Timeout: 2 seconds

**9.8.4.16 ULONG ResetToFactoryDefaults ( CHAR \* pSPC )**

Resets to default factory settings of the device

**Parameters**

<i>pSPC[IN]</i>	<ul style="list-style-type: none"> <li>NULL-terminated string representing a six-digit service programming code</li> </ul>
-----------------	--

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Timeout: 5 minutes

**9.8.4.17 ULONG SetPower ( ULONG powerMode )**

Sets the operating mode of the device.

## Parameters

<i>powerMode</i>	<ul style="list-style-type: none"><li>• Selected operating mode<ul style="list-style-type: none"><li>– See <a href="#">qaGobiApiTablePowerModes.h</a> for power modes</li></ul></li></ul>
------------------	---

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Technology Supported: UMTS/CDMA  
Device Supported: MC83x5, MC7700/50  
Timeout: 2 seconds

**9.8.4.18 ULONG SLQSGetBandCapability ( ULONGLONG \* pBandCapability )**

Returns the band capability of the device.

## Parameters

<p><i>pBand-Capability</i>[OUT]</p>	<p>Bitmask of bands supported by the device</p> <ul style="list-style-type: none"> <li>• Bit 0 - Band class 0, A-system</li> <li>• Bit 1 - Band class 0, B-system</li> <li>• Bit 2 - Band class 1, all blocks</li> <li>• Bit 3 - Band class 2</li> <li>• Bit 4 - Band class 3, A-system</li> <li>• Bit 5 - Band class 4, all blocks</li> <li>• Bit 6 - Band class 5, all blocks</li> <li>• Bit 7 - GSM DCS band (1800)</li> <li>• Bit 8 - GSM Extended GSM (E-GSM) band (900)</li> <li>• Bit 9 - GSM Primary GSM (P-GSM) band (900)</li> <li>• Bit 10 - Band class 6</li> <li>• Bit 11 - Band class 7</li> <li>• Bit 12 - Band class 8</li> <li>• Bit 13 - Band class 9</li> <li>• Bit 14 - Band class 10</li> <li>• Bit 15 - Band class 11</li> <li>• Bit 16 - GSM 450 band</li> <li>• Bit 17 - GSM 480 band</li> <li>• Bit 18 - GSM 750 band</li> <li>• Bit 19 - GSM 850 band</li> <li>• Bit 20 - GSM railways GSM band (900)</li> <li>• Bit 21 - GSM PCS band (1900)</li> <li>• Bit 22 - WCDMA (Europe, Japan, and China) 2100 band</li> <li>• Bit 23 - WCDMA US PCS 1900 band</li> <li>• Bit 24 - WCDMA (Europe and China) DCS 1800 band</li> <li>• Bit 25 - WCDMA US 1700 band</li> <li>• Bit 26 - WCDMA US 850 band</li> <li>• Bit 27 - WCDMA Japan 800 band</li> <li>• Bit 28 - Band class 12</li> <li>• Bit 29 - Band class 14</li> <li>• Bit 30 - Reserved</li> <li>• Bit 31 - Band class 15</li> <li>• Bits 32 through 47 - Reserved</li> <li>• Bit 48 - WCDMA Europe 2600 band</li> </ul>
	<ul style="list-style-type: none"> <li>• Bit 49 - WCDMA Europe and Japan 900 band</li> <li>• Bit 50 - WCDMA Japan 1700 band</li> <li>• Bits 51 through 55 - Reserved</li> </ul>

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Timeout: 2 Seconds

**9.8.4.19 ULONG SLQSGGetCurrentPRLInfo ( dmsCurrentPRLInfo \* pCurrentPRLInfo )**

This API get the currently active PRL information of the device.

**Parameters**

<i>pCurrentPRLInfo</i>	<ul style="list-style-type: none"><li>• Pointer to structure <a href="#">dmsCurrentPRLInfo</a></li><li>• See <a href="#">dmsCurrentPRLInfo</a> for more information</li></ul>
------------------------	---

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Technology Supported: CDMA  
Timeout: 2 Secs

**9.8.4.20 ULONG SLQSGgetCustFeatures ( custFeaturesInfo \* pCustFeaturesInfo )**

This API fetches the current settings of custom features

**Parameters**

<i>pCustFeatures-Info</i>	<ul style="list-style-type: none"><li>• Structure containing settings of custom features.</li><li>• See <a href="#">custFeaturesInfo</a> for more information</li></ul>
---------------------------	---

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Timeout: 2 Secs

#### 9.8.4.21 **ULONG SLQSGetCustFeaturesV2** ( **getCustomFeatureV2** \* *pGetCustomFeatureV2* )

This function queries the modem for a list of supported features. This function is for firmware version 2.0 and newer. Currently supported Customization features:

- GPIOARENABLE
- GPSSEL
- IMSWITCHHIDE
- IPV6ENABLE
- WAKEHOSTEN

##### Parameters

<i>pGetCustomFeatureV2</i>	<ul style="list-style-type: none"> <li>• See <a href="#">getCustomFeatureV2</a> for more information of the input structure</li> </ul>
----------------------------	--

#### 9.8.4.22 **ULONG SLQSGetERIFile** ( **ERIFileparams** \* *pERIFileparams* )

Returns the Extended Roaming Indicator (ERI) file that is stored in EFS on the device at a predetermined location. See the carrier requirements for specific details.

##### Parameters

<i>pERIFileparams</i>	<ul style="list-style-type: none"> <li>• Pointer to structure <a href="#">ERIFileparams</a></li> <li>• See <a href="#">ERIFileparams</a> for more information</li> </ul>
-----------------------	--

##### Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

##### See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

##### Note

Technology Supported: CDMA  
Timeout: 5 Seconds

#### 9.8.4.23 **ULONG SLQSGetSerialNumbers** ( **serialNumbersInfo** \* *pSerialNumbersInfo* )

Returns all the serial numbers assigned to the device. These serial numbers include the ESN (Electronic serial number of the device), the IMEI (International Mobile Equipment Identity), MEID (Mobile Equipment Identifier) and IMEI SVN (IMEI software version number).

##### Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

For CDMA devices that use a RUIM, the MEID of the RUIM (if present) will be returned. Use [SLQSSwiGet-SerialNoExt\(\)](#) to get MEID of CDMA modems. Timeout: 2 seconds

#### 9.8.4.24 ULONG SLQSSetCustFeatures ( custFeaturesSetting \* pCustFeaturesSetting )

This API changes the settings of custom features, a reset is required for any settings that are changed to take effect.

## Parameters

<i>pCustFeaturesSetting</i>	<ul style="list-style-type: none"> <li>• Structure containing settings of custom features.</li> <li>• See <a href="#">custFeaturesSetting</a> for more information</li> </ul>
-----------------------------	---

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 2 Secs

#### 9.8.4.25 ULONG SLQSSetCustFeaturesV2 ( setCustomSettingV2 \* pSetCustSetting )

This function sets the modem for a list of supported features. This function is for firmware version 2.0 and newer. Currently supported customization features:

- GPIOSARENABLE
- GPSSEL
- IMSWITCHHIDE
- IPV6ENABLE
- WAKEHOSTEN

## Parameters

<i>pSetCustSetting</i>	<ul style="list-style-type: none"> <li>• See <a href="#">setCustomSettingV2</a> for more information</li> </ul>
------------------------	---

#### 9.8.4.26 ULONG SLQSSwiGetCrashAction ( BYTE \* pDevCrashState )

This API queries the Crash State from the device.

## Parameters

<i>pDevCrashState</i>	<ul style="list-style-type: none"> <li>• Device Crash State</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0 - USB Memory Download Modem will reset after a crash and will stay in USB download mode with only ttyUSB0 enumerated. ramdump tool is to be used to recover the crash dump. Modem needs to be reset again to come back in ONLINE mode.</li> <li>– 1 - Reset Modem will reset and come back in ONLINE mode. Minimal crash data will be available and can be extracted with at!gcdump? AT command or <a href="#">SLQSSwiGetCrashInfo()</a> SDK API</li> <li>– 2 - No action</li> </ul> </li> </ul>
-----------------------	--

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Technology Supported: NA

Please free two buffers after get crash report successfully

1. pCrashInfoParams->pCrashInfo->pCrashString
2. pCrashInfoParams->pCrashInfo->pGCDumpString Timeout: 5 Secs

#### 9.8.4.27 ULONG SLQSSwiGetCrashInfo ( BYTE \* pClear, CrashInfoParams \* pCrashInfoParams )

This API queries the Crash Information from the device.

## Parameters

<i>pClear[IN]</i>	<ul style="list-style-type: none"> <li>• request parameter Clear(Optional parameter)</li> <li>• Values: 0 - Do not clear crash data after response 1 - Clear crash data after response</li> </ul>
<i>pCrashInfo-Params[Out]</i>	<ul style="list-style-type: none"> <li>• Pointer to structure <a href="#">CrashInfoParams</a></li> <li>• See <a href="#">CrashInfoParams</a> for more information</li> </ul>

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values



**Note**

Technology Supported: NA  
 Timeout: 5 Secs

**9.8.4.28 ULONG SLQSSwiGetFirmwareCurr ( CurrentImgList \* pCurrentImgList )**

This API gets the currently active images on the device.

**Parameters**

<i>pCurrentImgList</i>	<ul style="list-style-type: none"> <li>• Pointer to structure <a href="#">CurrentImgList</a></li> <li>• See <a href="#">CurrentImgList</a> for more information</li> </ul>
------------------------	--

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Technology Supported: NA  
 Device Supported: MC73xx  
 Timeout: 5 Secs

**9.8.4.29 ULONG SLQSSwiGetFSN ( FactorySequenceNumber \* pFSNumber )**

This API get the Factory Sequence Number of the device.

**Parameters**

<i>pFSNumber</i>	<ul style="list-style-type: none"> <li>• Pointer to structure <a href="#">FactorySequenceNumber</a></li> <li>• See <a href="#">FactorySequenceNumber</a> for more information</li> </ul>
------------------	--

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Technology Supported: CDMA  
 Timeout: 5 Secs

**9.8.4.30 ULONG SLQSSwiGetFwUpdateStatus ( FirmwareUpdatStat \* pFirmwareUpdatStat )**

This API will be used to query last firmware update status. The firmware status is stored in RAM and can be retained over warm resets but not power off resets.

## Parameters

<i>pFirmware- UpdatStat</i>	<ul style="list-style-type: none"> <li>• Pointer to structure <a href="#">FirmwareUpdatStat</a></li> <li>• See <a href="#">FirmwareUpdatStat</a> for more information</li> </ul>
---------------------------------	--

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Technology Supported: NA  
Device Supported: MC73xx  
Timeout: 5 Secs

#### 9.8.4.31 ULONG SLQSSwiGetHostDevInfo ( SLQSSwiGetHostDevInfoParams \* pGetHostDevInfoParams )

This API Get Host Information from the device.

## Parameters

<i>SLQSSwiGet- HostDevInfo- Params</i>	<ul style="list-style-type: none"> <li>• Crash Action</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0 - USB Memory Download</li> <li>– 1 - Reset</li> <li>– 2 - No action</li> </ul> </li> </ul>
--	--

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Technology Supported: NA  
Timeout: 5 Secs

#### 9.8.4.32 ULONG SLQSSwiGetOSInfo ( SLQSSwiGetOSInfoParams \* pParams )

This API Get OS Information to the device.

## Parameters

<i>SLQSSwiSet-HostDevInfo-Params</i>	<ul style="list-style-type: none"><li>• Crash Action</li><li>• Values:<ul style="list-style-type: none"><li>– 0 - USB Memory Download</li><li>– 1 - Reset</li><li>– 2 - No action</li></ul></li></ul>
--------------------------------------	---

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Technology Supported: NA  
Timeout: 5 Secs

**9.8.4.33   ULONG SLQSSwiGetSerialNoExt ( SLQSSwiGetSerialNoExtParams \* pParams )**

This API is used to get the MEID of the modem. For CDMA devices that use a RUIM, the MEID of the modem will always be returned.

## Parameters

<i>SLQSSwiGet-SerialNoExt-Params</i>	
--------------------------------------	--

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Technology Supported: NA  
Timeout: 5 Secs

**9.8.4.34   ULONG SLQSSwiGetUSBComp ( USBCompParams \* pUSBCompParams )**

This API queries the modem's USB interface configuration and supported configuration parameters.

## Parameters

<i>pUSBCompParams</i>	<ul style="list-style-type: none"> <li>• Pointer to structure <a href="#">USBCompParams</a></li> <li>• See <a href="#">USBCompParams</a> for more information</li> </ul>
-----------------------	--

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Technology Supported: NA  
Timeout: 5 Secs

#### 9.8.4.35 ULONG SLQSSwiSetCrashAction ( BYTE *crashActionParams* )

This API set the Crash Action to the device.

## Parameters

<i>crashActionParams</i>	<ul style="list-style-type: none"> <li>• Crash Action</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0 - USB Memory Download Modem will reset after a crash and will stay in USB download mode with only ttyUSB0 enumerated. ramdump tool is to be used to recover the crash dump. Modem needs to be reset again to come back in ONLINE mode.</li> <li>– 1 - Reset Modem will reset and come back in ONLINE mode. Minimal crash data will be available and can be extracted with at!gcdump? AT command or <a href="#">SLQSSwiGetCrashInfo()</a> SDK API</li> <li>– 2 - No action</li> </ul> </li> </ul>
--------------------------	--

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Technology Supported: NA  
Timeout: 5 Secs

#### 9.8.4.36 ULONG SLQSSwiSetHostDevInfo ( SLQSSwiSetHostDevInfoParams \* *pSetHostDevInfoParams* )

This API Set Host Information to the device.

## Parameters

<i>SLQSSwiSet-HostDevInfo-Params</i>	<ul style="list-style-type: none"> <li>• Crash Action</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0 - USB Memory Download</li> <li>– 1 - Reset</li> <li>– 2 - No action</li> </ul> </li> </ul>
--------------------------------------	--

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Technology Supported: NA  
Timeout: 5 Secs

#### 9.8.4.37 ULONG SLQSSwiSetOSInfo ( SLQSSwiSetOSInfoParams \* pParams )

This API Set OS Information to the device.

## Parameters

<i>SLQSSwiSet-HostDevInfo-Params</i>	<ul style="list-style-type: none"> <li>• Crash Action</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0 - USB Memory Download</li> <li>– 1 - Reset</li> <li>– 2 - No action</li> </ul> </li> </ul>
--------------------------------------	--

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Technology Supported: NA  
Timeout: 5 Secs

#### 9.8.4.38 ULONG SLQSSwiSetUSBComp ( USBCompConfig \* pUSBCompConfig )

This API is used to change the modem's USB interface configuration thus allowing a device to have multiple USB compositions. Devices will, by default, be configured to support a minimal set of interfaces to reduce end user modem installation time. Developers and some customers, however, require access to a custom set of interfaces. A reset is required for any change in the USB composition to take effect.

##### Parameters

<i>pUSBComp-Config</i>	<ul style="list-style-type: none"> <li>• Pointer to structure <a href="#">USBCompConfig</a></li> <li>• See <a href="#">USBCompConfig</a> for more information</li> </ul>
------------------------	--

##### Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

##### See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

##### Note

Technology Supported: NA  
Timeout: 5 Secs

#### 9.8.4.39 ULONG SLQSUIMGetState ( ULONG \* pUIMState )

Returns the UIM state. This API is deprecated on MC73xx/EM73xx modules since firmware version SWI9X15C\_05-\_xx\_xx\_xx and all EM74xx firmware versions. Please use API [SLQSUIMGetCardStatus\(\)](#) for new firmware versions and new modules

##### Parameters

<i>pUIMState[OUT]</i>	<ul style="list-style-type: none"> <li>• UIM state: <ul style="list-style-type: none"> <li>– 0x00 - UIM initialization completed</li> <li>– 0x01 - UIM locked or failed</li> <li>– 0x02 - UIM not present</li> <li>– 0x03 - 0xFE - Reserved</li> <li>– 0xFF - UIM state currently unavailable</li> </ul> </li> </ul>
-----------------------	--

##### Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

##### See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

##### Note

Timeout: 2 Seconds

#### 9.8.4.40 **ULONG** UIMChangePIN ( **ULONG** *id*, **CHAR** \* *pOldValue*, **CHAR** \* *pNewValue*, **ULONG** \* *pVerifyRetriesLeft*, **ULONG** \* *pUnblockRetriesLeft* )

Changes the PIN value for a given PIN. This API is deprecated on MC73xx/EM73xx modules since firmware version SWI9X15C\_05\_xx\_xx\_xx and all EM74xx firmware versions. Please use API [SLQSUIMChangePin\(\)](#) for new firmware versions and new modules

##### Parameters

<i>id</i>	<ul style="list-style-type: none"> <li>PIN ID <ul style="list-style-type: none"> <li>1 ( PIN1 / CHV1 )</li> <li>2 ( PIN2 / CHV2 )</li> </ul> </li> </ul>
<i>pOldValue[IN]</i>	<ul style="list-style-type: none"> <li>Old PIN value of PIN to change</li> </ul>
<i>pNewValue[IN]</i>	<ul style="list-style-type: none"> <li>New PIN value of PIN to change</li> </ul>
<i>pVerifyRetriesLeft[OUT]</i>	<ul style="list-style-type: none"> <li>Upon operational failure, this will indicate number of retries left, after which PIN will be blocked. <ul style="list-style-type: none"> <li>0xFFFFFFFF - Unknown</li> </ul> </li> </ul>
<i>pUnblockRetriesLeft[OUT]</i>	<ul style="list-style-type: none"> <li>Upon operational failure, this will indicate number of unblock retries left, after which the PIN will be permanently blocked; i.e. UIM is unusable. <ul style="list-style-type: none"> <li>0xFFFFFFFF - Unknown</li> </ul> </li> </ul>

##### Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

##### See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

##### Note

Timeout: 5 seconds

#### 9.8.4.41 **ULONG** UIMGetControlKeyStatus ( **ULONG** *id*, **ULONG** \* *pStatus*, **ULONG** \* *pVerifyRetriesLeft*, **ULONG** \* *pUnblockRetriesLeft* )

Returns the status of the specified UIM facility control key. This API is deprecated on MC73xx/EM73xx modules since firmware version SWI9X15C\_05\_xx\_xx\_xx and all EM74xx firmware versions.

## Parameters

<i>id</i>	<ul style="list-style-type: none"> <li>• Facility ID <ul style="list-style-type: none"> <li>– 0 - Network Personalization (PN)</li> <li>– 1 - Network Subset Personalization (PU)</li> <li>– 2 - Service Provider Personalization (PP)</li> <li>– 3 - Corporate Personalization (PC)</li> <li>– 4 - UIM Personalization (PF)</li> </ul> </li> </ul>
<i>pStatus[OUT]</i>	<ul style="list-style-type: none"> <li>• Control key status <ul style="list-style-type: none"> <li>– 0 - Deactivated</li> <li>– 1 - Activated</li> <li>– 2 - Blocked</li> </ul> </li> </ul>
<i>pVerifyRetriesLeft[OUT]</i>	<ul style="list-style-type: none"> <li>• The number of retries left, after which the control key will be blocked <ul style="list-style-type: none"> <li>– 0xFFFFFFFF - Unknown</li> </ul> </li> </ul>
<i>pUnblockRetriesLeft[OUT]</i>	<ul style="list-style-type: none"> <li>• The number of unblock retries left, after which the control key will be permanently blocked <ul style="list-style-type: none"> <li>– 0xFFFFFFFF - Unknown</li> </ul> </li> </ul>

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 5 seconds

#### 9.8.4.42 ULONG UIMGetICCID ( BYTE stringSize, CHAR \* pString )

Returns the UIM ICCID. This API is deprecated on MC73xx/EM73xx modules since firmware version SWI9X15C\_05\_xx\_xx\_xx and all EM74xx firmware versions.

## Parameters



<i>stringSize</i>	<ul style="list-style-type: none"> <li>The maximum number of characters (including NULL terminator) that the string array can contain.</li> </ul>
<i>pString[OUT]</i>	<ul style="list-style-type: none"> <li>NULL terminated string</li> </ul>

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Timeout: 2 seconds

#### 9.8.4.43 ULONG UIMGetPINStatus ( ULONG id, ULONG \* pStatus, ULONG \* pVerifyRetriesLeft, ULONG \* pUnblockRetriesLeft )

Gets the status of the SIM PINs. This API is deprecated on MC73xx/EM73xx modules since firmware version SWI9X15C\_05\_xx\_xx\_xx and all EM74xx firmware versions. Please use API [SLQSUIGetCardStatus\(\)](#) for new firmware versions and new modules

**Parameters**

<i>id</i>	<ul style="list-style-type: none"> <li>PIN ID <ul style="list-style-type: none"> <li>1 ( PIN1 / CHV1 )</li> <li>2 ( PIN2 / CHV2 )</li> </ul> </li> </ul>
<i>pStatus[OUT]</i>	<ul style="list-style-type: none"> <li>PIN status(0xFFFFFFFF - Unknown) <ul style="list-style-type: none"> <li>0 - PIN not initialized</li> <li>1 - PIN enabled, not verified</li> <li>2 - PIN enabled, verified</li> <li>3 - PIN disabled</li> <li>4 - PIN blocked</li> <li>5 - PIN permanently blocked</li> </ul> </li> </ul>

<i>pVerifyRetriesLeft</i> [OUT]	<ul style="list-style-type: none"> <li>Upon operational failure, this will indicate number of retries left, after which PIN will be blocked. <ul style="list-style-type: none"> <li>– 0xFFFFFFFF - Unknown</li> </ul> </li> </ul>
<i>pUnblockRetriesLeft</i> [OUT]	<ul style="list-style-type: none"> <li>Upon operational failure, this will indicate number of unblock retries left, after which the PIN will be permanently blocked; i.e., UIM is unusable. <ul style="list-style-type: none"> <li>– 0xFFFFFFFF - Unknown</li> </ul> </li> </ul>

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Timeout: 5 seconds

#### 9.8.4.44 **ULONG** UIMSetControlKeyProtection ( **ULONG** *id*, **ULONG** *status*, **CHAR** \* *pValue*, **ULONG** \* *pVerifyRetriesLeft* )

Changes the specified UIM facility control key. This API is deprecated on MC73xx/EM73xx modules since firmware version SWI9X15C\_05\_xx\_xx\_xx and all EM74xx firmware versions. Please use API [SLQSUIMDepersonalization\(\)](#) for new firmware versions and new modules

**Parameters**

<i>id</i>	<ul style="list-style-type: none"> <li>Facility ID <ul style="list-style-type: none"> <li>– 0 - Network Personalization (PN)</li> <li>– 1 - Network Subset Personalization (PU)</li> <li>– 2 - Service Provider Personalization (PP)</li> <li>– 3 - Corporate Personalization (PC)</li> <li>– 4 - UIM Personalization (PF)</li> </ul> </li> </ul>
-----------	---

<i>status</i>	<ul style="list-style-type: none"> <li>Control key status <ul style="list-style-type: none"> <li>0 - Deactivated</li> </ul> </li> </ul>
<i>pValue[IN]</i>	<ul style="list-style-type: none"> <li>Control key de-personalization string (maximum length of 8 characters)</li> </ul>
<i>pVerifyRetries-Left[OUT]</i>	<ul style="list-style-type: none"> <li>Upon operational failure, this will indicate number of retries left, after which the control key will be blocked <ul style="list-style-type: none"> <li>0xFFFFFFFF - Unknown</li> </ul> </li> </ul>

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Timeout: 5 seconds

#### 9.8.4.45 **ULONG UIMSetPINProtection ( ULONG id, ULONG bEnable, CHAR \* pValue, ULONG \* pVerifyRetriesLeft, ULONG \* pUnblockRetriesLeft )**

Enables or disables protection of SIM contents for a given PIN, This API is deprecated on MC73xx/EM73xx modules since firmware version SWI9X15C\_05\_xx\_xx\_xx and all EM74xx firmware versions. Please use API [SLQSUIMSet-PinProtection\(\)](#) for new firmware versions and new modules

**Parameters**

<i>id</i>	<ul style="list-style-type: none"> <li>PIN ID <ul style="list-style-type: none"> <li>1 ( PIN1 / CHV1 )</li> <li>2 ( PIN2 / CHV2 )</li> </ul> </li> </ul>
<i>bEnable</i>	<ul style="list-style-type: none"> <li>Enable/disable PIN protection, 0 = Disable</li> </ul>
<i>pValue[IN]</i>	<ul style="list-style-type: none"> <li>PIN value of the PIN to be enabled/disabled</li> </ul>

<i>pVerifyRetriesLeft[OUT]</i>	<ul style="list-style-type: none"> <li>Upon operational failure, this will indicate number of retries left, after which PIN will be blocked. <ul style="list-style-type: none"> <li>– 0xFFFFFFFF - Unknown</li> </ul> </li> </ul>
<i>pUnblockRetriesLeft[OUT]</i>	<ul style="list-style-type: none"> <li>Upon operational failure, this will indicate number of unblock retries left, after which the PIN will be permanently blocked i.e. UIM is unusable. <ul style="list-style-type: none"> <li>– 0xFFFFFFFF - Unknown</li> </ul> </li> </ul>

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Timeout: 5 seconds

#### 9.8.4.46 ULONG UIMUnblockControlKey ( ULONG *id*, CHAR \* *pValue*, ULONG \* *pUnblockRetriesLeft* )

Unblocks the specified UIM facility control key. This API is deprecated on MC73xx/EM73xx modules since firmware version SWI9X15C\_05\_xx\_xx\_xx and all EM74xx firmware versions. Please use API [SLQSUIMDepersonalization\(\)](#) for new firmware versions and new modules

**Parameters**

<i>id</i>	<ul style="list-style-type: none"> <li>Facility ID <ul style="list-style-type: none"> <li>– 0 - Network Personalization (PN)</li> <li>– 1 - Network Subset Personalization (PU)</li> <li>– 2 - Service Provider Personalization (PP)</li> <li>– 3 - Corporate Personalization (PC)</li> <li>– 4 - UIM Personalization (PF)</li> </ul> </li> </ul>
-----------	---

<i>pValue[IN]</i>	<ul style="list-style-type: none"> <li>Control key de-personalization string (maximum length of 8 characters)</li> </ul>
<i>pUnblockRetriesLeft[OUT]</i>	<ul style="list-style-type: none"> <li>Upon operational failure, this will indicate number of unblock retries left, after which the control key will be blocked <ul style="list-style-type: none"> <li>– 0xFFFFFFFF - Unknown</li> </ul> </li> </ul>

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Timeout: 5 seconds

#### 9.8.4.47 **ULONG** UIMUnblockPIN ( **ULONG** *id*, **CHAR** \* *pPUKValue*, **CHAR** \* *pNewValue*, **ULONG** \* *pVerifyRetriesLeft*, **ULONG** \* *pUnblockRetriesLeft* )

Unlocks a blocked SIM. This API is deprecated on MC73xx/EM73xx modules since firmware version SWI9X15C\_05\_xx\_xx\_xx and all EM74xx firmware versions. Please use API [SLQSUIMUnblockPin\(\)](#) for new firmware versions and new modules

**Parameters**

<i>id</i>	<ul style="list-style-type: none"> <li>PIN ID <ul style="list-style-type: none"> <li>– 1 ( PIN1 / CHV1 )</li> <li>– 2 ( PIN2 / CHV2 )</li> </ul> </li> </ul>
<i>pPUKValue[IN]</i>	<ul style="list-style-type: none"> <li>PUK value of PIN to unblock</li> </ul>
<i>pNewValue[IN]</i>	<ul style="list-style-type: none"> <li>New PIN value of PIN to unblock</li> </ul>
<i>pVerifyRetriesLeft[OUT]</i>	<ul style="list-style-type: none"> <li>Upon operational failure, this will indicate number of retries left, after which the PIN will be blocked. <ul style="list-style-type: none"> <li>– 0xFFFFFFFF - Unknown</li> </ul> </li> </ul>

<i>pUnblockRetriesLeft</i> [OUT]	<ul style="list-style-type: none"> <li>Upon operational failure, this will indicate number of unblock retries left, after which the PIN will be permanently blocked; i.e. UIM is unusable <ul style="list-style-type: none"> <li>0xFFFFFFFF - Unknown</li> </ul> </li> </ul>
----------------------------------	--

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Timeout: 5 seconds

#### 9.8.4.48 **ULONG** UIMVerifyPIN ( **ULONG** *id*, **CHAR** \* *pValue*, **ULONG** \* *pVerifyRetriesLeft*, **ULONG** \* *pUnblockRetriesLeft* )

Verifies a SIM PIN. This API is deprecated on MC73xx/EM73xx modules since firmware version SWI9X15C\_05\_XX\_XX\_XX and all EM74xx firmware versions. Please use API [SLQSUIMVerifyPin\(\)](#) for new firmware versions and new modules

**Parameters**

<i>id</i>	<ul style="list-style-type: none"> <li>PIN ID <ul style="list-style-type: none"> <li>1 ( PIN1 / CHV1 )</li> <li>2 ( PIN2 / CHV2 )</li> </ul> </li> </ul>
<i>pValue</i> [IN]	<ul style="list-style-type: none"> <li>Value of PIN to verify</li> </ul>
<i>pVerifyRetriesLeft</i> [OUT]	<ul style="list-style-type: none"> <li>Upon operational failure, this will indicate number of retries left, after which the PIN will be blocked. <ul style="list-style-type: none"> <li>0xFFFFFFFF - Unknown</li> </ul> </li> </ul>
<i>pUnblockRetriesLeft</i> [OUT]	<ul style="list-style-type: none"> <li>Upon operational failure, this will indicate number of unblock retries left, after which the PIN will be permanently blocked; i.e. UIM is unusable <ul style="list-style-type: none"> <li>0xFFFFFFFF - Unknown</li> </ul> </li> </ul>

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 5 seconds

## 9.8.4.49 ULONG ValidateSPC ( CHAR \* pSPC )

This function Validates Service Programming code of the device

## Parameters

<i>pSPC</i> [IN]	<ul style="list-style-type: none"> <li>• NULL-terminated string representing a six-digit service programming code</li> </ul>
------------------	--

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Technology Supported: CDMA  
Device Supported: MC83x5, MC7750  
Timeout: 2 seconds

## 9.9 qaGobiApiFms.h File Reference

Firmware Management Service API function prototypes.

### Data Structures

- struct [fwinfo\\_s](#)
- struct [slqsfwinfo\\_s](#)
- struct [qmifwinfo\\_s](#)
- struct [ImageElement](#)
- struct [PrefImageList](#)
- struct [ImageIdElement](#)
- struct [ImageIDEntries](#)
- struct [ImageList](#)
- struct [sGetDeviceSeriesResult](#)
- struct [SWI\\_STRUCT\\_CarrierImage](#)

## Macros

- `#define SLQSFWINFO_MODELID_SZ 20`
- `#define SLQSFWINFO_BOOTVERSION_SZ 85`
- `#define SLQSFWINFO_APPVERSION_SZ 85`
- `#define SLQSFWINFO_SKU_SZ 15`
- `#define SLQSFWINFO_PACKAGEID_SZ 85`
- `#define SLQSFWINFO_CARRIER_SZ 20`
- `#define SLQSFWINFO_PRIVERSION_SZ 16`
- `#define SLQSFWINFO_CUR_CARR_NAME 17`
- `#define SLQSFWINFO_CUR_CARR_REV 13`
- `#define GOBI_MBN_IMG_ID_STR_LEN 16`
- `#define GOBI_MBN_BUILD_ID_STR_LEN 100`
- `#define GOBI_LISTENTRIES_MAX 2`
- `#define GOBI_SET_IMG_PREF_RSPLEN 40`
- `#define DEVICE_SHUTDOWN 5`
- `#define DEVICE_RESET 4`
- `#define FIRMWARE_UPDATE_SUCCESS 0x01`
- `#define FIRMWARE_UPDATE_FAIL 0x01`
- `#define PRI_UPDATE_FAIL 0x02`
- `#define FIRMWARE_UPGRADE_SUCCESS 0x00`

## Enumerations

- `enum eGobiImageTech {  
    eGOBI_IMG_TECH_CDMA = 0,  
    eGOBI_IMG_TECH_UMTS }`
- `enum eGobiImageCarrier {`



```
eGOBI_IMG_CAR_GENERIC = 1,  
eGOBI_IMG_CAR_FACTORY,  
eGOBI_IMG_CAR_NORF,  
eGOBI_IMG_CAR_VERIZON = 101,  
eGOBI_IMG_CAR_SPRINT,  
eGOBI_IMG_CAR_ALLTEL,  
eGOBI_IMG_CAR_BELL,  
eGOBI_IMG_CAR_TELUS,  
eGOBI_IMG_CAR_US,  
eGOBI_IMG_CAR_TELSTRA1,  
eGOBI_IMG_CAR_CHINA_UNICOM,  
eGOBI_IMG_CAR_TELCOM_NZ,  
eGOBI_IMG_CAR_SK_TELCOM1,  
eGOBI_IMG_CAR_RELIANCE1,  
eGOBI_IMG_CAR_TATA,  
eGOBI_IMG_CAR_METROPCS,  
eGOBI_IMG_CAR_LEAP,  
eGOBI_IMG_CAR_KDDI,  
eGOBI_IMG_CAR_IUSACELL,  
eGOBI_IMG_CAR_CHINA_TELECOM,  
eGOBI_IMG_CAR_OMH,  
eGOBI_IMG_CAR_GENERIC_CDMA,  
eGOBI_IMG_CAR_ATT = 201,  
eGOBI_IMG_CAR_VODAFONE,  
eGOBI_IMG_CAR_TMOBILE,  
eGOBI_IMG_CAR_ORANGE,  
eGOBI_IMG_CAR_TELEFONICA,  
eGOBI_IMG_CAR_TELCOM_ITALIA,  
eGOBI_IMG_CAR_3,  
eGOBI_IMG_CAR_O2,  
eGOBI_IMG_CAR_SFR,  
eGOBI_IMG_CAR_SWISSCOM,  
eGOBI_IMG_CAR_CHINA_MOBILE,  
eGOBI_IMG_CAR_TELSTRA2,  
eGOBI_IMG_CAR_SINGTEL_OPTUS,  
eGOBI_IMG_CAR_RELIANCE2,  
eGOBI_IMG_CAR_BHARTI,  
eGOBI_IMG_CAR_NTT_DOCOMO,  
eGOBI_IMG_CAR_EMOBILE,  
eGOBI_IMG_CAR_SOFTBANK,  
eGOBI_IMG_CAR_KT_FREETEL,  
eGOBI_IMG_CAR_SK_TELCOM2,  
eGOBI_IMG_CAR_TELENOR,  
eGOBI_IMG_CAR_NETCOM,  
eGOBI_IMG_CAR_TELIASONERA,  
eGOBI_IMG_CAR_AMX_TELCEL,  
eGOBI_IMG_CAR_BRASIL_VIVO,  
eGOBI_IMG_CAR_AERIS,  
eGOBI_IMG_CAR_ROGERS }  
  
• enum eGobiImageRegion {  
    eGOBI_IMG_REG_NA = 0,  
    eGOBI_IMG_REG_LA,  
    eGOBI_IMG_REG_EU,  
    eGOBI_IMG_REG_ASIA,  
    eGOBI_IMG_REG_AUS,  
    eGOBI_IMG_REG_GLOBAL }  
  
• enum eGobiImageGPS {
```

```

eGOBI_IMG_GPS_NONE = 0,
eGOBI_IMG_GPS_STAND_ALONE,
eGOBI_IMG_GPS_ASSISTED,
eGOBI_IMG_GPS_NO_XTRA }
• enum eGobiDeviceSeries {
eGOBI_DEV_SERIES_UNKNOWN = -1,
eGOBI_DEV_SERIES_NON_GOBI = 0,
eGOBI_DEV_SERIES_G3K,
eGOBI_DEV_SERIES_SIERRA_GOBI,
eGOBI_DEV_SERIES_9X15,
eGOBI_DEV_SERIES_9X30,
eGobi_DEV_SERIES_MC83 }

```

## Functions

- [ULONG GetImageStore](#) ([WORD](#) imageStorePathSize, [CHAR](#) \*pImageStorePath)
- [ULONG SLQSGetFirmwareInfo](#) (struct [qmifwinfo\\_s](#) \*pinfo)
- [ULONG SLQSGetImageInfoMC77xx](#) ([LPCSTR](#) path, struct [qmifwinfo\\_s](#) \*pinfo)
- [ULONG SLQSGetImageInfoMC83xx](#) ([LPCSTR](#) path, struct [qmifwinfo\\_s](#) \*pinfo)
- [ULONG SLQSGetImageInfo](#) ([LPCSTR](#) path, struct [qmifwinfo\\_s](#) \*pinfo)
- [ULONG UpgradeFirmware2k](#) ([CHAR](#) \*pDestinationPath)
- [ULONG GetImagesPreference](#) ([ULONG](#) \*pImageListSize, struct [PrefImageList](#) \*pImageList)
- [ULONG SetImagesPreference](#) ([ULONG](#) imageListSize, [BYTE](#) \*pImageList, [ULONG](#) bForceDownload, [BYTE](#) modemIndex, [ULONG](#) \*pImageTypesSize, [BYTE](#) \*pImageTypes)
- [ULONG GetStoredImages](#) ([ULONG](#) \*pImageListSize, [BYTE](#) \*pImageList)
- [ULONG DeleteStoredImage](#) ([ULONG](#) imageInfoSize, [BYTE](#) \*pImageInfo)
- [ULONG SLQSGetImageInfo\\_9x15](#) ([LPCSTR](#) path, [BYTE](#) imgType, struct [slqsfwinfo\\_s](#) \*pinfo)
- [ULONG SLQSUpgradeFirmware9x15](#) ([CHAR](#) \*pDestinationPath)
- [ULONG SLQSGetBootVersionNumber](#) ([ULONG](#) \*bootversion)
- [BOOL SLQSIspkgFormatRequired](#) (void)
- [ULONG upgrade\\_mc77xx\\_fw](#) ([LPCSTR](#) path)
- [void eGetDeviceSeries](#) (struct [sGetDeviceSeriesResult](#) \*)
- [ULONG SLQSSwiGetAllCarrierImages](#) ([ULONG](#) \*pNumOfItems, struct [SWI\\_STRUCT\\_CarrierImage](#) \*pCarrierImages, [wchar\\_t](#) \*pFolderPath)
- [ULONG SLQSDownloadFirmwareToSlot](#) ([CHAR](#) \*pPath, [BYTE](#) slot\_index, [BYTE](#) force\_download)

### 9.9.1 Detailed Description

Firmware Management Service API function prototypes.

### 9.9.2 Macro Definition Documentation

9.9.2.1 `#define DEVICE_RESET 4`

9.9.2.2 `#define DEVICE_SHUTDOWN 5`

9.9.2.3 `#define FIRMWARE_UPDATE_FAIL 0x01`

9.9.2.4 `#define FIRMWARE_UPDATE_SUCCESS 0x01`

9.9.2.5 `#define FIRMWARE_UPGRADE_SUCCESS 0x00`

9.9.2.6 `#define GOBI_LISTENTRIES_MAX 2`

9.9.2.7 `#define GOBI_MBN_BUILD_ID_STR_LEN 100`

9.9.2.8 `#define GOBI_MBN_IMG_ID_STR_LEN 16`

9.9.2.9 `#define GOBI_SET_IMG_PREF_RSPLN 40`

9.9.2.10 `#define PRI_UPDATE_FAIL 0x02`

9.9.2.11 `#define SLQSFWINFO_APPVERSION_SZ 85`

9.9.2.12 `#define SLQSFWINFO_BOOTVERSION_SZ 85`

9.9.2.13 `#define SLQSFWINFO_CARRIER_SZ 20`

9.9.2.14 `#define SLQSFWINFO_CUR_CARR_NAME 17`

9.9.2.15 `#define SLQSFWINFO_CUR_CARR_REV 13`

9.9.2.16 `#define SLQSFWINFO_MODELID_SZ 20`

9.9.2.17 `#define SLQSFWINFO_PACKAGEID_SZ 85`

9.9.2.18 `#define SLQSFWINFO_PRIVERSION_SZ 16`

9.9.2.19 `#define SLQSFWINFO_SKU_SZ 15`

### 9.9.3 Enumeration Type Documentation

#### 9.9.3.1 `enum eGobiDeviceSeries`

enumeration which lists the Device Series

Enumerator

***eGOBI\_DEV\_SERIES\_UNKNOWN***  
***eGOBI\_DEV\_SERIES\_NON\_GOBI***  
***eGOBI\_DEV\_SERIES\_G3K***  
***eGOBI\_DEV\_SERIES\_SIERRA\_GOBI***  
***eGOBI\_DEV\_SERIES\_9X15***  
***eGOBI\_DEV\_SERIES\_9X30***  
***eGobi\_DEV\_SERIES\_MC83***

#### 9.9.3.2 `enum eGobiImageCarrier`

enumeration which lists the carrier supported by the image

Enumerator

***eGOBI\_IMG\_CAR\_GENERIC***  
***eGOBI\_IMG\_CAR\_FACTORY***  
***eGOBI\_IMG\_CAR\_NORF***  
***eGOBI\_IMG\_CAR\_VERIZON***  
***eGOBI\_IMG\_CAR\_SPRINT***  
***eGOBI\_IMG\_CAR\_ALLTEL***

*eGOBI\_IMG\_CAR\_BELL*  
*eGOBI\_IMG\_CAR\_TELUS*  
*eGOBI\_IMG\_CAR\_US*  
*eGOBI\_IMG\_CAR\_TELSTRA1*  
*eGOBI\_IMG\_CAR\_CHINA\_UNICOM*  
*eGOBI\_IMG\_CAR\_TELCOM\_NZ*  
*eGOBI\_IMG\_CAR\_SK\_TELCOM1*  
*eGOBI\_IMG\_CAR\_RELIANCE1*  
*eGOBI\_IMG\_CAR\_TATA*  
*eGOBI\_IMG\_CAR\_METROPCS*  
*eGOBI\_IMG\_CAR\_LEAP*  
*eGOBI\_IMG\_CAR\_KDDI*  
*eGOBI\_IMG\_CAR\_IUSACELL*  
*eGOBI\_IMG\_CAR\_CHINA\_TELECOM*  
*eGOBI\_IMG\_CAR\_OMH*  
*eGOBI\_IMG\_CAR\_GENERIC\_CDMA*  
*eGOBI\_IMG\_CAR\_ATT*  
*eGOBI\_IMG\_CAR\_VODAFONE*  
*eGOBI\_IMG\_CAR\_TMOBILE*  
*eGOBI\_IMG\_CAR\_ORANGE*  
*eGOBI\_IMG\_CAR\_TELEFONICA*  
*eGOBI\_IMG\_CAR\_TELCOM\_ITALIA*  
*eGOBI\_IMG\_CAR\_3*  
*eGOBI\_IMG\_CAR\_O2*  
*eGOBI\_IMG\_CAR\_SFR*  
*eGOBI\_IMG\_CAR\_SWISSCOM*  
*eGOBI\_IMG\_CAR\_CHINA\_MOBILE*  
*eGOBI\_IMG\_CAR\_TELSTRA2*  
*eGOBI\_IMG\_CAR\_SINGTEL\_OPTUS*  
*eGOBI\_IMG\_CAR\_RELIANCE2*  
*eGOBI\_IMG\_CAR\_BHARTI*  
*eGOBI\_IMG\_CAR\_NTT\_DOCOMO*  
*eGOBI\_IMG\_CAR\_EMOBILE*  
*eGOBI\_IMG\_CAR\_SOFTBANK*  
*eGOBI\_IMG\_CAR\_KT\_FREETEL*  
*eGOBI\_IMG\_CAR\_SK\_TELCOM2*  
*eGOBI\_IMG\_CAR\_TELNOR*  
*eGOBI\_IMG\_CAR\_NETCOM*  
*eGOBI\_IMG\_CAR\_TELIASONERA*  
*eGOBI\_IMG\_CAR\_AMX\_TELCEL*  
*eGOBI\_IMG\_CAR\_BRASIL\_VIVO*  
*eGOBI\_IMG\_CAR\_AERIS*  
*eGOBI\_IMG\_CAR\_ROGERS*

## 9.9.3.3 enum eGobiImageGPS

enumeration which lists the GPS type supported by the image

Enumerator

***eGOBI\_IMG\_GPS\_NONE***  
***eGOBI\_IMG\_GPS\_STAND\_ALONE***  
***eGOBI\_IMG\_GPS\_ASSISTED***  
***eGOBI\_IMG\_GPS\_NO\_XTRA***

## 9.9.3.4 enum eGobiImageRegion

enumeration which lists the region supported by the image

Enumerator

***eGOBI\_IMG\_REG\_NA***  
***eGOBI\_IMG\_REG\_LA***  
***eGOBI\_IMG\_REG\_EU***  
***eGOBI\_IMG\_REG\_ASIA***  
***eGOBI\_IMG\_REG\_AUS***  
***eGOBI\_IMG\_REG\_GLOBAL***

## 9.9.3.5 enum eGobiImageTech

enumeration which lists the technology supported by the image

Enumerator

***eGOBI\_IMG\_TECH\_CDMA***  
***eGOBI\_IMG\_TECH\_UMTS***

## 9.9.4 Function Documentation

9.9.4.1 ULONG DeleteStoredImage ( ULONG *imageInfoSize*, BYTE \* *plmageInfo* )

Used to delete the specified image from the device. This API function is only relevant to devices with the ability to store multiple firmware images(see Device Supported section below).

Parameters

<i>imageInfoSize</i> [I-N]	<ul style="list-style-type: none"> <li>The size in BYTES of the image info array</li> </ul>
<i>plmageInfo</i> [IN]	<ul style="list-style-type: none"> <li>The image info list array containing information about the image to be deleted.</li> <li>See <a href="#">ImageElement</a></li> </ul>

Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Device Supported: MC83x5/SL9090  
Timeout: 2 Secs

**9.9.4.2 void eGetDeviceSeries ( struct sGetDeviceSeriesResult \* )**

Name : eGetDeviceSeries

**Parameters**

<i>none</i>	
-------------	--

**Returns**

[sGetDeviceSeriesResult](#)

**Note**

Get Devie Series

**9.9.4.3 ULONG GetImagesPreference ( ULONG \* *plmImageListSize*, struct PrefImageList \* *plmImageList* )**

restore original alignment from stack Gets the current images preference from the device.

**Parameters**

<i>plmImageListSize</i> [- IN/OUT]	<ul style="list-style-type: none"> <li>• Upon input, the size of structure <a href="#">ImageList</a> <a href="#">ImageList</a></li> <li>• Upon successful output, the number of BYTEs copied to the image list array</li> </ul>
<i>plmImageList</i> [OUT]	<ul style="list-style-type: none"> <li>• The caller must supply a pointer to a <a href="#">ImageList</a> structure typecast as a BYTE pointer</li> </ul>

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Timeout: 2 seconds

**9.9.4.4 ULONG GetImageStore ( WORD *imageStorePathSize*, CHAR \* *plmImageStorePath* )**

Returns the image store folder, i.e., the folder containing one or more carrier-specific image subfolders compatible with the currently connected QC WWAN device.

## Parameters

<i>imageStorePath-Size</i>	<ul style="list-style-type: none"> <li>Maximum number of characters (including NULL terminator) that can be copied to the image store path array.</li> </ul>
<i>plmageStore-Path[OUT]</i>	<ul style="list-style-type: none"> <li>The path to the image store</li> </ul>

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

#### 9.9.4.5 ULONG GetStoredImages ( ULONG \* plmageListSize, BYTE \* plmageList )

restore original alignment from stack Gets the list of images stored on the device.

## Parameters

<i>plmageListSize[IN/OUT]</i>	<ul style="list-style-type: none"> <li>Upon input, the size of structure <a href="#">ImageList ImageList</a></li> <li>Upon successful output, the number of BYTES copied to the image list array</li> </ul>
<i>plmageList[OUT]</i>	<ul style="list-style-type: none"> <li>The caller must supply a pointer to a <a href="#">ImageList</a> structure typecast as a BYTE pointer</li> </ul>

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Device Supported: MC83x5/SL9090  
Timeout: 2 seconds

#### 9.9.4.6 ULONG SetImagesPreference ( ULONG imageListSize, BYTE \* plmageList, ULONG bForceDownload, BYTE modemIndex, ULONG \* plmageTypesSize, BYTE \* plmageTypes )

Sets the current images preference on the device. After this function successfully completes, the device must be reset for the selected image preference to be realized. Additionally, when the returned list of image types that require downloading is not empty, the device opens in QDL mode after the reset. At that point, the QDL portion of this API must be used to download the selected image preference to the device.

## Parameters

<i>imageListSize</i>	<ul style="list-style-type: none"> <li>The size in BYTES of the image list array</li> </ul>
<i>pImageList[IN]</i>	<ul style="list-style-type: none"> <li>The image info list array containing Image Elements <ul style="list-style-type: none"> <li>– See <a href="#">PrefImageList</a></li> </ul> </li> </ul>
<i>bForce-Download[IN]</i>	<ul style="list-style-type: none"> <li>Force device to download images from host? 0 - No Nonzero - Yes</li> </ul>
<i>modemIndex</i>	<ul style="list-style-type: none"> <li>Desired storage index for downloaded modem image (optional, a value of 0xFF indicates unspecified)</li> </ul>
<i>pImageTypes-Size[IN/OUT]</i>	<ul style="list-style-type: none"> <li>Upon input, maximum number of elements that download image types array can contain</li> <li>Upon successful output, number of elements in download image types array</li> </ul>
<i>pImageTypes[OUT]</i>	-The download image types array.

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 2 seconds

#### 9.9.4.7 ULONG SLQSDownloadFirmwareToSlot ( CHAR \* *pPath*, BYTE *slot\_index*, BYTE *force\_download* )

This API is used to download firmware to a specific slot id of the modem. It is only applicable for EM74xx variant. This API encapsulates all steps involved in the firmware download process. Hence it is a blocking API call. This API will not return until the entire process has been completed. This API will takes significant amount of time (in order of minutes, normally should be less than 10 minutes).

## Parameters

<i>pPath[IN]</i>	<ul style="list-style-type: none"> <li>fully qualified path to firmware image to download.</li> </ul>
------------------	---



<i>slot_index</i> [IN]	<ul style="list-style-type: none"> <li>slot id in the modem to store the firmware</li> </ul>
<i>force_download</i> [IN]	<ul style="list-style-type: none"> <li>a flag to force download take place. this feature is not supported</li> <li>currently. so just pass the argument as 0 when invoke this API.</li> </ul>

**Returns**

- eQCWWAN\_ERR\_INVALID\_ARG - The path input does not contain any image
- eQCWWAN\_ERR\_SWIIM\_FW\_UPDATE\_FAIL - Upgrade( i.e. download to device ) failed
- eQCWWAN\_ERR\_SWIIM\_FW\_UPDATE\_SUCCESS - Upgrade succeeded and device online.
- eQCWWAN\_ERR\_SWIIM\_FW\_PREFERENCE\_MISMATCH - Upgrade succeeded but device offline due to image preference mismatch ( ref. syslogs for cause )

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Timeout: NA

**9.9.4.8 ULONG SLQSGetBootVersionNumber ( ULONG \* bootversion )**

Gets the boot loader version number

**Parameters**

<i>bootversion</i> [OUT]	<ul style="list-style-type: none"> <li>boot loader version presented by a 4 byte integer</li> </ul>
--------------------------	---

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Device Supported: MC9090/SL9090  
Timeout: 2 seconds

**9.9.4.9 ULONG SLQSGetFirmwareInfo ( struct qmifwinfo\_s \* pinfo )**

Returns firmware image information from the connected device

## Parameters

<i>pinfo</i> [OUT]	<ul style="list-style-type: none"> <li>firmware image information record</li> </ul>
--------------------	---

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values  
 struct [qmifwinfo\\_s](#)

## Note

Timeout: 2 Seconds.

#### 9.9.4.10 ULONG SLQSGetImageInfo ( LPCSTR *path*, struct *qmifwinfo\_s* \* *pinfo* )

Returns firmware image information from a CWE file or mbn files stored on the host. For CWE, information is returned for the first CWE image found at the specified path. For MBN, the provided path must be located under the image store for the currently connected QC WWAN device. Note that as this API supports multiple firmware image types, it relies on the presence of a supported device. Otherwise, refer to SLQSGetImageInfoMC83xx and SLQSGetImageInfoMC77xx for APIs which do not rely on the presence of a supported device.

## Parameters

<i>path</i> [IN]	<ul style="list-style-type: none"> <li>fully qualified path to folder containing CWE image or MBN images</li> <li>should use a "/" at the end of the path.</li> </ul>
<i>pinfo</i> [OUT]	<ul style="list-style-type: none"> <li>firmware image information record</li> </ul>

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values  
 struct [qmifwinfo\\_s](#)

## Note

Timeout: N/A

#### 9.9.4.11 ULONG SLQSGetImageInfo\_9x15 ( LPCSTR *path*, BYTE *imgType*, struct *slqsfwinfo\_s* \* *pinfo* )

Returns firmware image information from a CWE file(s) stored on the host. It does not rely on the presence of a supported device.

## Parameters

<i>path</i> [IN]	<ul style="list-style-type: none"> <li>fully qualified path to folder containing the image(s)</li> <li>should use a "/" at the end of the path.</li> </ul>
<i>imgType</i> [IN]	<ul style="list-style-type: none"> <li>2 - Firmware Image( .cwe extension )</li> <li>3 - PRI Image ( .nvu extension )</li> </ul>
<i>pinfo</i> [OUT]	<ul style="list-style-type: none"> <li>firmware image information record</li> </ul>

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values  
 struct [qmifwinfo\\_s](#)

## Note

Device Supported: MC73xx  
 Timeout: N/A

#### 9.9.4.12 ULONG SLQSGetImageInfoMC77xx ( LPCSTR *path*, struct *qmifwinfo\_s* \* *pinfo* )

Returns firmware image information from a SPKGS CWE file stored on the host. The information is returned for the first SPKGS CWE image found at the specified path. This API executes independent of a MC77xx being connected to the target.

## Parameters

<i>path</i> [IN]	<ul style="list-style-type: none"> <li>fully qualified path to folder containing SPKG CWE image</li> <li>should use a "/" at the end of the path.</li> </ul>
<i>pinfo</i> [OUT]	<ul style="list-style-type: none"> <li>firmware image information record</li> </ul>

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values  
 struct [qmifwinfo\\_s](#)

**Note**

Device Supported: MC77xx  
 Timeout: N/A

**9.9.4.13 ULONG SLQSGetImageInfoMC83xx ( LPCSTR path, struct qmifwinfo\_s \* pinfo )**

Returns firmware image information from an MBN file located on the host. This API executes independent of a MC83xx being connected to the target.

**Parameters**

<i>path</i> [IN]	<ul style="list-style-type: none"> <li>fully qualified path to folder containing MBN file</li> <li>should use a "/" at the end of the path.</li> </ul>
<i>pinfo</i> [OUT]	<ul style="list-style-type: none"> <li>firmware image information record</li> </ul>

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values  
 struct [qmifwinfo\\_s](#)

**Note**

Device Supported: MC83xx/SL9090  
 Timeout: N/A

**9.9.4.14 BOOL SLQSIspkgFormatRequired ( void )**

Check if SPKG format download is required for SL9090/MC9090

**Parameters**

<i>none</i>
-------------

**Returns**

return TRUE if required, otherwise, return FALSE

**Note**

Device Supported: MC9090/SL9090  
 Timeout: 2 seconds

**9.9.4.15 ULONG SLQSSwiGetAllCarrierImages ( ULONG \* pNumOfItems, struct SWI\_STRUCT\_CarrierImage \* pCarrierImages, wchar\_t \* pFolderPath )**

This API gets a list of all images stored on both the host and the device

## Parameters

<i>pNumOfItems</i>	<ul style="list-style-type: none"> <li>Number of Images{I/O}</li> </ul>
<i>pCarrierImages[O]</i>	<ul style="list-style-type: none"> <li>See <a href="#">SWI_STRUCT_CarrierImage</a></li> </ul>
<i>pFolderPath</i>	<ul style="list-style-type: none"> <li>Path of Input folder [I]</li> </ul>

## Returns

TRUE/FALSE

## Note

In case pFolderPath is invalid, API does not return invalid path error as SLQSSwiGetAllCarrierImages get carrier images from device also.

## 9.9.4.16 ULONG SLQSSUpgradeFirmware9x15 ( CHAR \* pDestinationPath )

This API is used to upgrade firmware on a MC73xx device. This API encapsulates all steps involved in the firmware download process. It is an alternative to any firmware download application. Hence it is a blocking API call. This API will not return until the entire process has been completed.

This API Performs the following steps:

1. Verifies arguments.
2. Retrieve and store the details of the firmware and the PRI file
3. Enable device state change callback.
4. Enable firmware download callback.
5. Set Image preference on the device and reset the device.
6. Wait for the firmware to download and device to become ready.
7. Check the firmware update status. If fail, return an error.
8. If update status is OK, check if current image preference and preferred image preference( from step 2 ) match
9. If match, firmware download is successful.
10. If do not match, repeat from step 5 once more.
11. Disable callbacks and exit.

The call to this API blocks until step 7 or 11. This could be a significant amount of time ( in order of minutes ). Also note that the device state change callback and firmware download callback are used internally within this API. Hence the user application's instance of these callbacks (if any) are cleared. The user must re-enable these callbacks after a call to this API in order to use them.

## Parameters

<i>pDestination-Path[IN]</i>	<ul style="list-style-type: none"> <li>fully qualified path to firmware image to download. The path must end with a forward slash.</li> </ul>
------------------------------	---

## Returns

- eQCWWAN\_ERR\_INVALID\_ARG - The path input does not contain any image
- eQCWWAN\_ERR\_SWIIM\_FW\_UPDATE\_FAIL - Upgrade( i.e. download to device ) failed
- eQCWWAN\_ERR\_SWIIM\_FW\_UPDATE\_SUCCESS - Upgrade succeeded and device online.
- eQCWWAN\_ERR\_SWIIM\_FW\_PREFERENCE\_MISMATCH - Upgrade succeeded but device offline due to image preference mismatch ( ref. syslogs for cause )

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: NA

## 9.9.4.17 ULONG upgrade\_mc77xx\_fw ( LPCSTR path )

## 9.9.4.18 ULONG UpgradeFirmware2k ( CHAR \* pDestinationPath )

This API is used to download firmware to a MC77xx or Gobi 3000 device.

This API Performs the following steps:

1. Verifies arguments.
2. Informs the SDK of the firmware upgrade path
3. Updates the images preference on the currently connected device.
4. Requests the device reset (device will reset after all open handles are released).

Upon successful completion, the above steps will have been completed, however, the actual upgrade of the firmware will necessarily then follow.

## Parameters

<i>pDestination-Path[IN]</i>	<ul style="list-style-type: none"> <li>fully qualified path to firmware image to download. The path must end with a forward slash. For a Gobi 3000 device the path should specify the carrier image folder index i.e.  "<code>&lt;path&gt;\to\carrier\image&gt;\/&lt;carrier index&gt;\/</code>" where <code>&lt;carrier index&gt;=""</code> is a valid sub-directory entry. For 9x30 devices if pDestinationPath is not valid on host, it will use pseudo path for image switching.</li> </ul>
------------------------------	---

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_XXX error values

**Note**

Timeout: 12 seconds

## 9.10 qaGobiApilms.h File Reference

IMS Service API function prototypes.

**Data Structures**

- struct [SetSIPConfigReq](#)
- struct [SetSIPConfigResp](#)
- struct [SetRegMgrConfigReq](#)
- struct [SetRegMgrConfigResp](#)
- struct [SetIMSSMSConfigReq](#)
- struct [SetIMSSMSConfigResp](#)
- struct [SetIMSUserConfigReq](#)
- struct [SetIMSUserConfigResp](#)
- struct [SetIMSVoIPConfigReq](#)
- struct [SetIMSVoIPConfigResp](#)
- struct [GetSIPConfigResp](#)
- struct [GetRegMgrConfigParams](#)
- struct [GetIMSSMSConfigParams](#)
- struct [GetIMSUserConfigParams](#)
- struct [GetIMSVoIPConfigResp](#)
- struct [imsCfgIndRegisterInfo](#)

**Functions**

- [ULONG SLQSSetSIPConfig](#) ([SetSIPConfigReq](#) \*pSetSIPConfigReq, [SetSIPConfigResp](#) \*pSetSIPConfigResp)
- [ULONG SLQSSetRegMgrConfig](#) ([SetRegMgrConfigReq](#) \*pSetRegMgrConfigReq, [SetRegMgrConfigResp](#) \*pSetRegMgrConfigResp)
- [ULONG SLQSSetIMSSMSConfig](#) ([SetIMSSMSConfigReq](#) \*pSetIMSSMSConfigReq, [SetIMSSMSConfigResp](#) \*pSetIMSSMSConfigResp)
- [ULONG SLQSSetIMSUserConfig](#) ([SetIMSUserConfigReq](#) \*pSetIMSUserConfigReq, [SetIMSUserConfigResp](#) \*pSetIMSUserConfigResp)
- [ULONG SLQSSetIMSVoIPConfig](#) ([SetIMSVoIPConfigReq](#) \*pSetIMSVoIPConfigReq, [SetIMSVoIPConfigResp](#) \*pSetIMSVoIPConfigResp)
- [ULONG SLQSGetSIPConfig](#) ([GetSIPConfigResp](#) \*pGetSIPConfigResp)
- [ULONG SLQSGetRegMgrConfig](#) ([GetRegMgrConfigParams](#) \*pGetRegMgrConfigParams)
- [ULONG SLQSGetIMSSMSConfig](#) ([GetIMSSMSConfigParams](#) \*pGetIMSSMSConfigParams)
- [ULONG SLQSGetIMSUserConfig](#) ([GetIMSUserConfigParams](#) \*pGetIMSUserConfigParams)
- [ULONG SLQSGetIMSVoIPConfig](#) ([GetIMSVoIPConfigResp](#) \*pGetIMSVoIPConfigResp)
- [ULONG SLQSImConfigIndicationRegister](#) ([imsCfgIndRegisterInfo](#) \*pImCfgIndRegisterInfo)

### 9.10.1 Detailed Description

IMS Service API function prototypes.

## 9.10.2 Function Documentation

### 9.10.2.1 ULONG SLQSGetIMSSMSConfig ( GetIMSSMSConfigParams \* pGetIMSSMSConfigParams )

This API retrieves the SMS configuration parameters.

#### Parameters

<i>pGetIMSSMS-ConfigParams</i> [I-/O/OUT]	<ul style="list-style-type: none"> <li>• See <a href="#">GetIMSSMSConfigParams</a> for more information</li> </ul>
---	--

#### Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

#### See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

#### Note

Technology Supported: NA  
Device Supported: MC7800  
Timeout: 5 seconds

### 9.10.2.2 ULONG SLQSGetIMSUserConfig ( GetIMSUserConfigParams \* pGetIMSUserConfigParams )

This API retrieves the IMS User configuration parameters.

#### Parameters

<i>pGetIMSUser-ConfigParams</i> [I-/O/OUT]	<ul style="list-style-type: none"> <li>• See <a href="#">GetIMSUserConfigParams</a> for more information</li> </ul>
--	---

#### Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

#### See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

#### Note

Technology Supported: NA  
Device Supported: MC7800  
Timeout: 5 seconds

### 9.10.2.3 ULONG SLQSGetIMSVoIPConfig ( GetIMSVoIPConfigResp \* pGetIMSVoIPConfigResp )

This API retrieves the IMS VoIP configuration parameters.



## Parameters

<i>GetIMSVolP-ConfigResp[OUT]</i>	<ul style="list-style-type: none"> <li>• See <a href="#">GetIMSVolPConfigResp</a> for more information</li> </ul>
-----------------------------------	---

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Technology Supported: NA  
Device Supported: MC7800  
Timeout: 5 seconds

#### 9.10.2.4 ULONG SLQSGetRegMgrConfig ( GetRegMgrConfigParams \* pGetRegMgrConfigParams )

This API retrieves the registration manager configuration parameters.

## Parameters

<i>pGetRegMgr-ConfigParams[IN/OUT]</i>	<ul style="list-style-type: none"> <li>• See <a href="#">GetRegMgrConfigResp</a> for more information</li> </ul>
--	--

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Technology Supported: NA  
Device Supported: MC7800  
Timeout: 5 seconds

#### 9.10.2.5 ULONG SLQSGetSIPConfig ( GetSIPConfigResp \* pGetSIPConfigResp )

This API retrieves the Session Initiation Protocol(SIP) configuration parameters.

## Parameters

<i>pGetSIPConfig-Resp[OUT]</i>	<ul style="list-style-type: none"> <li>• See <a href="#">GetSIPConfigResp</a> for more information</li> </ul>
--------------------------------	---

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Technology Supported: NA  
 Device Supported: MC7800  
 Timeout: 5 seconds

### 9.10.2.6 ULONG SLQSImsConfigIndicationRegister ( *imsCfgIndRegisterInfo* \* *plmsCfgIndRegisterInfo* )

Sets the registration state for different QMI\_IMS indications for the requesting control point

**Parameters**

<i>plmsCfgIndRegisterInfo</i> [IN]	<ul style="list-style-type: none"> <li>Structure containing Indication Register Information.               <ul style="list-style-type: none"> <li>See <a href="#">imsCfgIndRegisterInfo</a> for more information.</li> </ul> </li> </ul>
------------------------------------	--

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Technology Supported: UMTS  
 Device Supported: MC7800  
 Timeout: 10 Secs

This API is used by a device to register/deregister for different QMI IMS indications. The device's registration state variables that control registration for indications will be modified to reflect the settings indicated in the request message. At least one optional parameter must be present in the request.

### 9.10.2.7 ULONG SLQSSetIMSSMSConfig ( *SetIMSSMSConfigReq* \* *pSetIMSSMSConfigReq*, *SetIMSSMSConfigResp* \* *pSetIMSSMSConfigResp* )

This API sets the IMS SMS configuration parameters for the requesting control point.

**Parameters**

<i>pSetIMSSMSConfigReq</i> [IN]	<ul style="list-style-type: none"> <li>See <a href="#">SetIMSSMSConfigReq</a> for more information</li> </ul>
<i>pSetIMSSMSConfigResp</i> [OUT]	<ul style="list-style-type: none"> <li>See <a href="#">SetIMSSMSConfigResp</a> for more information</li> </ul>

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Technology Supported: NA  
Device Supported: MC7800  
Timeout: 5 seconds

#### 9.10.2.8 **ULONG SLQSSetIMSUserConfig ( SetIMSUserConfigReq \* pSetIMSUserConfigReq, SetIMSUserConfigResp \* pSetIMSUserConfigResp )**

This API sets the IMS user configuration parameters for the requesting control point.

**Parameters**

<i>pSetIMSUserConfigReq[IN]</i>	<ul style="list-style-type: none"> <li>See <a href="#">SetIMSUserConfigReq</a> for more information</li> </ul>
<i>pSetIMSUserConfigResp[OUT]</i>	<ul style="list-style-type: none"> <li>See <a href="#">SetIMSUserConfigResp</a> for more information</li> </ul>

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Technology Supported: NA  
Device Supported: MC7800  
Timeout: 5 seconds

#### 9.10.2.9 **ULONG SLQSSetIMSVoIPConfig ( SetIMSVoIPConfigReq \* pSetIMSVoIPConfigReq, SetIMSVoIPConfigResp \* pSetIMSVoIPConfigResp )**

This API sets the IMS Voice over Internet Protocol (VoIP) configuration parameters for the requesting control point.

**Parameters**

<i>pSetIMSVoIPConfigReq[IN]</i>	<ul style="list-style-type: none"> <li>See <a href="#">SetIMSVoIPConfigReq</a> for more information</li> </ul>
---------------------------------	--

<i>pSetIMSVoIP-ConfigResp[OUT]</i>	<ul style="list-style-type: none"> <li>• See <a href="#">SetIMSVoIPConfigResp</a> for more information</li> </ul>
------------------------------------	---

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Technology Supported: NA  
 Device Supported: MC7800  
 Timeout: 5 seconds

#### 9.10.2.10 **ULONG SLQSSetRegMgrConfig ( SetRegMgrConfigReq \* pSetRegMgrConfigReq, SetRegMgrConfigResp \* pSetRegMgrConfigResp )**

This API sets the IMS registration manager configuration parameters for the requesting control point.

**Parameters**

<i>pSetRegMgr-ConfigReq[IN]</i>	<ul style="list-style-type: none"> <li>• See <a href="#">SetRegMgrConfigReq</a> for more information</li> </ul>
<i>pSetRegMgr-ConfigResp[OUT]</i>	<ul style="list-style-type: none"> <li>• See <a href="#">SetRegMgrConfigResp</a> for more information</li> </ul>

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Technology Supported: NA  
 Device Supported: MC7800  
 Timeout: 5 seconds

#### 9.10.2.11 **ULONG SLQSSetSIPConfig ( SetSIPConfigReq \* pSetSIPConfigReq, SetSIPConfigResp \* pSetSIPConfigResp )**

This API sets the IMS Session Initiation Protocol(SIP) configuration parameters for the requesting control point.

## Parameters

<i>pSetSIPConfigReq</i> [IN]	<ul style="list-style-type: none"> <li>See <a href="#">SetSIPConfigReq</a> for more information</li> </ul>
<i>pSetSIPConfigResp</i> [OUT]	<ul style="list-style-type: none"> <li>See <a href="#">SetSIPConfigResp</a> for more information</li> </ul>

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Technology Supported: NA  
Device Supported: MC7800  
Timeout: 5 seconds

## 9.11 qaGobiApilmsa.h File Reference

IMSA Service API function prototypes.

### Data Structures

- struct [IMSAIndRegisterInfo](#)
- struct [SupportedMsgList](#)
- struct [IMSASupportedMsgInfo](#)
- struct [ReqFieldsList](#)
- struct [RespFieldsList](#)
- struct [IndFieldsList](#)
- struct [IMSASupportedFieldsResp](#)
- struct [IMSARegistrationStatus](#)
- struct [IMSAServiceStatus](#)

### Functions

- [ULONG SLQSRegisterIMSAIndication](#) ([IMSAIndRegisterInfo](#) \*pImsaIndRegisterInfo)
- [ULONG SLQSGetIMSASupportedMsg](#) ([IMSASupportedMsgInfo](#) \*pIMSASupportedMsgInfo)
- [ULONG SLQSGetIMSASupportedFields](#) ([WORD](#) messageID, [IMSASupportedFieldsResp](#) \*pIMSASupportedFieldsResp)
- [ULONG SLQSGetIMSARegStatus](#) ([IMSARegistrationStatus](#) \*pIMSARegistrationStatus)
- [ULONG SLQSGetIMSAServiceStatus](#) ([IMSAServiceStatus](#) \*pIMSAServiceStatus)

#### 9.11.1 Detailed Description

IMSA Service API function prototypes.

## 9.11.2 Function Documentation

### 9.11.2.1 `ULONG SLQSGetIMSARegStatus ( IMSARegistrationStatus * pIMSARegistrationStatus )`

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

## Parameters

<i>pIMSARegistration-Status</i> [OUT]	<ul style="list-style-type: none"> <li>Structure containing response parameters for registration status. <ul style="list-style-type: none"> <li>See <a href="#">IMSARegistrationStatus</a> for more information.</li> </ul> </li> </ul>
---------------------------------------	---

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 5 Secs

This API is used by a device to get the registration status for various IMS services for the requesting control point.

### 9.11.2.2 ULONG SLQSGetIMSAServiceStatus ( IMSAServiceStatus \* *pIMSAServiceStatus* )

Gets the service status for various IMS services for the requesting control point.

## Parameters

<i>pIMSAService-Status</i> [OUT]	<ul style="list-style-type: none"> <li>Structure containing response parameters for service status. <ul style="list-style-type: none"> <li>See <a href="#">IMSAServiceStatus</a> for more information.</li> </ul> </li> </ul>
----------------------------------	---

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 5 Secs

This API is used by a device to Gets the service status for various IMS services for the requesting control point.

### 9.11.2.3 ULONG SLQSGetIMSASupportedFields ( WORD *messageID*, IMSASupportedFieldsResp \* *pIMSASupportedFieldsResp* )

Queries the set of supported fields implemented by the currently running software.

## Parameters

<i>messageID[IN]</i>	<ul style="list-style-type: none"> <li>• Service Message ID.</li> </ul>
<i>pIMSA-Supported-FieldsResp[OUT]</i>	<ul style="list-style-type: none"> <li>• Structure containing Supported Fields Response. <ul style="list-style-type: none"> <li>– See <a href="#">IMSASupportedFieldsResp</a> for more information.</li> </ul> </li> </ul>

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 5 Secs

This API is used by a device to query the fields supported for a single command as implemented by the currently running software.

#### 9.11.2.4 ULONG SLQSGetIMSASupportedMsg ( IMSASupportedMsgInfo \* pIMSASupportedMsgInfo )

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

## Parameters

<i>pIMSA-SupportedMsg-Info[OUT]</i>	<ul style="list-style-type: none"> <li>• Structure containing Supported Messages Information. <ul style="list-style-type: none"> <li>– See <a href="#">IMSASupportedMsgInfo</a> for more information.</li> </ul> </li> </ul>
-------------------------------------	--

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 5 Secs

This API is used by a device to query the set of messages implemented by the currently running software

#### 9.11.2.5 ULONG SLQSRegisterIMSAIndication ( IMSAIndRegisterInfo \* plmsalndRegisterInfo )

Sets the registration state for different QMI\_IMSA indications for the requesting control point



## Parameters

<i>plmsaInd-RegisterInfo</i> [IN]	<ul style="list-style-type: none"> <li>Structure containing Indication Register Information. <ul style="list-style-type: none"> <li>See <a href="#">IMSAIndRegisterInfo</a> for more information.</li> </ul> </li> </ul>
-----------------------------------	--

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 5 Secs

This API is used by a device to register/deregister for different QMI IMSA indications. The device's registration state variables that control registration for indications will be modified to reflect the settings indicated in the request message. At least one optional parameter must be present in the request.

## 9.12 qaGobiApiLoc.h File Reference

Location API function prototypes.

## Data Structures

- struct [LOCEventRegisterReqResp](#)
- struct [LOCExtPowerStateReqResp](#)
- struct [LocApplicationInfo](#)
- struct [LOCStartReq](#)
- struct [LOCStopReq](#)
- struct [SV](#)
- struct [SVInfo](#)
- struct [GnssData](#)
- struct [CellIDb](#)
- struct [CikInfo](#)
- struct [BdsSV](#)
- struct [BdsSVInfo](#)
- struct [LocDelAssDataReq](#)

## Functions

- [ULONG SLQSLOCEventRegister](#) ([LOCEventRegisterReqResp](#) \*pLOCEventRegisterReqResp)
- [ULONG SLQSLOCSetExtPowerState](#) ([LOCExtPowerStateReqResp](#) \*pLOCExtPowerStateReqResp)
- [ULONG SLQSLOCStart](#) ([LOCStartReq](#) \*pLOCStartReq)
- [ULONG SLQSLOCStop](#) ([LOCStopReq](#) \*pLOCStopReq)
- [ULONG SLQSLOCSetOpMode](#) ([ULONG](#) mode)
- [ULONG SLQSLOCDelAssData](#) ([LocDelAssDataReq](#) request)

### 9.12.1 Detailed Description

Location API function prototypes.

### 9.12.2 Function Documentation

#### 9.12.2.1 ULONG SLQSLOCDeAssData ( LocDeAssDataReq *request* )

Used by the control point to delete the location engine assistance data

##### Parameters

<i>request</i> [IN]	<ul style="list-style-type: none"> <li>Input a NULL pointer to delete all assistance data. Otherwise, specify optional fields to be deleted. See <a href="#">LocDeAssDataReq</a> for more information</li> </ul>
---------------------	--

##### Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

##### See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

##### Note

Timeout: 5 seconds

#### 9.12.2.2 ULONG SLQSLOCEventRegister ( LOCEventRegisterReqResp \* *pLOCEventRegisterReqResp* )

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

##### Parameters

<i>pLOCEvent-RegisterReq-Resp</i> [IN]	<ul style="list-style-type: none"> <li>See <a href="#">LOCEventRegisterReqResp</a> for more information</li> </ul>
--	--

##### Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

##### See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

##### Note

Timeout: 5 seconds

#### 9.12.2.3 ULONG SLQSLOCSetExtPowerState ( LOCExtPowerStateReqResp \* *pLOCExtPowerStateReqResp* )

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

## Parameters

<i>pLOCExtPowerStateReqResp</i> [!-N]	<ul style="list-style-type: none"> <li>See <a href="#">LOCExtPowerStateReqResp</a> for more information</li> </ul>
---------------------------------------	--

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 5 seconds

## 9.12.2.4 ULONG SLQSLOCSetOpMode ( ULONG mode )

Used by the control point to tells the engine to use the specified operation mode while making the position fixes

## Parameters

<i>mode</i> [!N]	<ul style="list-style-type: none"> <li>Valid values: <ul style="list-style-type: none"> <li>eQMI_LOC_OPER_MODE_DEFAULT (1) - Use the default engine mode</li> <li>eQMI_LOC_OPER_MODE_MSB (2) - Use the MS-based mode</li> <li>eQMI_LOC_OPER_MODE_MSA (3) - Use the MS-assisted mode</li> <li>eQMI_LOC_OPER_MODE_STANDALONE (4) - Use Standalone mode</li> <li>eQMI_LOC_OPER_MODE_CELL_ID (5) - Use cell ID; this mode is only valid for GSM/UMTS networks</li> <li>eQMI_LOC_OPER_MODE_WWAN (6) - Use WWAN measurements to calculate the position; if this mode is set, AFLT will be used for 1X networks and OTDOA will be used for LTE networks</li> </ul> </li> </ul>
------------------	---

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 5 seconds

## 9.12.2.5 ULONG SLQSLOCStart ( LOCStartReq \* pLOCStartReq )

Used by the control point to initiate a GPS session.

## Parameters

<i>pLOCStartReq-Req[IN]</i>	<ul style="list-style-type: none"> <li>See <a href="#">LOCStartReq</a> for more information</li> </ul>
-----------------------------	--

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 5 seconds

### 9.12.2.6 ULONG SLQSLOCStop ( LOCStopReq \* pLOCStopReq )

Used by the control point to stop a GPS session.

## Parameters

<i>pLOCStopReq-Resp[IN]</i>	<ul style="list-style-type: none"> <li>See <a href="#">LOCStopReq</a> for more information</li> </ul>
-----------------------------	---

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 5 seconds

## 9.13 qaGobiApiNas.h File Reference

Network Access Service API function prototypes.

## Data Structures

- struct [RFBandInfoElements](#)
- struct [servSystem](#)
- struct [dataSrvCapabilities](#)
- struct [currentPLMN](#)
- struct [roamIndList](#)
- struct [qaQmi3Gpp2TimeZone](#)
- struct [detailSvcInfo](#)

- struct [CDMASysInfoExt](#)
- struct [callBarStatus](#)
- struct [qaQmiServingSystemParam](#)
- struct [rxSignalStrengthListElement](#)
- struct [ecioListElement](#)
- struct [errorRateListElement](#)
- struct [rsrqInformation](#)
- struct [lteSnrinformation](#)
- struct [lteRsrpinformation](#)
- struct [slqsSignalStrengthInfo](#)
- struct [SlqsNas3GppNetworkInfo](#)
- struct [SlqsNas3GppNetworkRAT](#)
- struct [SlqsNasPcsDigit](#)
- struct [\\_slqsNetworkScanInfo](#)
- struct [netSelectionPref](#)
- struct [acqOrderPref](#)
- struct [CSGID](#)
- struct [\\_sysSelectPrefParams](#)
- struct [\\_sysSelectPrefInfo](#)
- struct [SrvStatusInfo](#)
- struct [GSMSrvStatusInfo](#)
- struct [sysInfoCommon](#)
- struct [CDMASysInfo](#)
- struct [HDRSysInfo](#)
- struct [GSMSysInfo](#)
- struct [WCDMASysInfo](#)
- struct [LTESysInfo](#)
- struct [AddCDMASysInfo](#)
- struct [AddSysInfo](#)
- struct [CallBarringSysInfo](#)
- struct [nasGetSysInfoResp](#)
- struct [CommInfo](#)
- struct [LTEInfo](#)
- struct [swiModemStatusResp](#)
- struct [nasGetHDRColorCodeResp](#)
- struct [nasGetTxRxInfoReq](#)
- struct [rxInfo](#)
- struct [txInfo](#)
- struct [nasGetTxRxInfoResp](#)
- struct [CDMASSInfo](#)
- struct [HDRSSInfo](#)
- struct [LTSSInfo](#)
- struct [TDSCDMASigInfoExt](#)
- struct [nasGetSigInfoResp](#)
- struct [nasIndicationRegisterReq](#)
- struct [nasPLMNNameReq](#)
- struct [nasPLMNNameResp](#)
- struct [OperatorPLMNData](#)
- struct [operatorPLMNList](#)
- struct [serviceProviderName](#)
- struct [PLMNNetworkNameData](#)
- struct [PLMNNetworkName](#)
- struct [operatorNameString](#)
- struct [nasOperatorNameResp](#)
- struct [nasGet3GPP2SubscriptionInfoReq](#)

- struct [namName](#)
- struct [dirNum](#)
- struct [sidNid](#)
- struct [homeSIDNID](#)
- struct [minBasedIMSI](#)
- struct [trueIMSI](#)
- struct [CDMAChannel](#)
- struct [nasGet3GPP2SubscriptionInfoResp](#)
- struct [nmrCellInfo](#)
- struct [GERANInfo](#)
- struct [geranInstInfo](#)
- struct [UMTSinstInfo](#)
- struct [UMTSInfo](#)
- struct [CDMAInfo](#)
- struct [cellParams](#)
- struct [LTEInfoIntrafreq](#)
- struct [infoInterFreq](#)
- struct [LTEInfoInterfreq](#)
- struct [gsmCellInfo](#)
- struct [lteGsmCellInfo](#)
- struct [LTEInfoNeighboringGSM](#)
- struct [wcdmaCellInfo](#)
- struct [lteWcdmaCellInfo](#)
- struct [LTEInfoNeighboringWCDMA](#)
- struct [umtsLTENbrCell](#)
- struct [WCDMAInfoLTENeighborCell](#)
- struct [nasCellLocationInfoResp](#)
- struct [MNRInfo](#)
- struct [nasInitNetworkReg](#)
- struct [protocolSubtypeElement](#)
- struct [HDRPersonalityResp](#)
- struct [HDRProtSubtypResp](#)
- struct [PSDetachReq](#)
- struct [GetErrRateResp](#)
- struct [DRCParams](#)
- struct [PilotSetParams](#)
- struct [PilotSetData](#)
- struct [GetHRPDStatsResp](#)
- struct [ActPilotPNElement](#)
- struct [NetworkStat1x](#)
- struct [NetworkStatEVDO](#)
- struct [DeviceConfigDetail](#)
- struct [DataStatusDetail](#)
- struct [NetworkDebugResp](#)
- struct [LteCQIParm](#)
- struct [RSSIThresh](#)
- struct [ECIOThresh](#)
- struct [HDRSINRThresh](#)
- struct [LTESNRThresh](#)
- struct [IOTThresh](#)
- struct [RSRQThresh](#)
- struct [RSRPThresh](#)
- struct [LTSigRptCfg](#)
- struct [sigInfo](#)
- struct [NasSwilndReg](#)

- struct [CDMARSSIThresh](#)
- struct [CDMAECIOThresh](#)
- struct [HRRSSIThresh](#)
- struct [HDRECIOThresh](#)
- struct [HRSINRThreshold](#)
- struct [HDRIOThresh](#)
- struct [GSMRSSIThresh](#)
- struct [WCDMARSSIThresh](#)
- struct [WCDMAECIOThresh](#)
- struct [LTERSSIThresh](#)
- struct [LTERSNRThreshold](#)
- struct [LTERSRQThresh](#)
- struct [LTERSRPThresh](#)
- struct [LTERSigRptConfig](#)
- struct [TDSCDMARSCPTthresh](#)
- struct [TDSCDMARSSIThresh](#)
- struct [TDSCDMAECIOThresh](#)
- struct [TDSCDMASINRThresh](#)
- struct [setSignalStrengthInfo](#)
- struct [PhyCaAggScellIndType](#)
- struct [PhyCaAggScellIDBw](#)
- struct [PhyCaAggScellInfo](#)
- struct [PhyCaAggPcellInfo](#)
- struct [PhyCaAggScellIndex](#)
- struct [nasGetLTECphyCaResp](#)
- struct [nasGetLTECphyCa](#)

## Macros

- [#define SLQS\\_SS\\_INFO\\_LIST\\_MAX\\_ELEMENTS](#) 18
- [#define MAX\\_DESCRIPTION\\_LENGTH](#) 255
- [#define SLQS\\_SYSTEM\\_ID\\_SIZE](#) 16
- [#define PLMN\\_LENGTH](#) 3
- [#define MAX\\_SERV\\_SYSTEM\\_RADIO\\_INTERFACES](#) 0x0A
- [#define MAX\\_DATA\\_SRV\\_CAPABILITIES](#) 0x20
- [#define NAM\\_NAME\\_LENGTH](#) 12
- [#define IMSI\\_M\\_S1\\_LENGTH](#) 7
- [#define IMSI\\_M\\_S2\\_LENGTH](#) 3
- [#define MAX\\_PILOT\\_SETS](#) 0xFF
- [#define UATISIZE](#) 16

## Typedefs

- typedef struct [\\_slqsNetworkScanInfo](#) [slqsNetworkScanInfo](#)
- typedef struct [\\_sysSelectPrefParams](#) [sysSelectPrefParams](#)
- typedef struct [\\_sysSelectPrefInfo](#) [sysSelectPrefInfo](#)

## Enumerations

- enum `_NAMS_RADIO_IF_TECHNOLOGY_` {  
`eNAS_RADIO_IF_GSM` = 0x04,  
`eNAS_RADIO_IF_UMTS` = 0x05,  
`eNAS_RADIO_IF_LTE` = 0x08,  
`eNAS_RADIO_IF_TDSCDMA` = 0x09 }
- enum `NAS_LTE_CPHY_SCELL_STATE` {  
`eNAS_LTE_CPHY_SCELL_STATE_DECONFIGURED` = 0x00,  
`eNAS_LTE_CPHY_SCELL_STATE_CONFIGURED_DEACTIVATED` = 0x01,  
`eNAS_LTE_CPHY_SCELL_STATE_CONFIGURED_ACTIVATED` = 0x02 }
- enum `NAS_LTE_CPHY_CA_BW_NRB` {  
`eNAS_LTE_CPHY_CA_BW_NRB_6` = 0x00,  
`eNAS_LTE_CPHY_CA_BW_NRB_15` = 0x01,  
`eNAS_LTE_CPHY_CA_BW_NRB_25` = 0x02,  
`eNAS_LTE_CPHY_CA_BW_NRB_50` = 0x03,  
`eNAS_LTE_CPHY_CA_BW_NRB_75` = 0x04,  
`eNAS_LTE_CPHY_CA_BW_NRB_100` = 0x05 }

## Functions

- `ULONG GetSignalStrengths (ULONG *pArraySizes, INT8 *pSignalStrength, ULONG *pRadioInterface)`
- `ULONG PerformNetworkScan (BYTE *pInstanceSize, BYTE *pInstances)`
- `ULONG InitiateNetworkRegistration (ULONG regType, WORD mcc, WORD mnc, ULONG rat)`
- `ULONG GetServingNetwork (ULONG *pRegistrationState, ULONG *pCSDomain, ULONG *pPSDomain, ULONG *pRAN, BYTE *pRadiofacesSize, BYTE *pRadiofaces, ULONG *pRoaming, WORD *pMCC, WORD *pMNC, BYTE nameSize, CHAR *pName)`
- `ULONG GetHomeNetwork (WORD *pMCC, WORD *pMNC, BYTE nameSize, CHAR *pName, WORD *pSID, WORD *pNID)`
- `ULONG GetServingNetworkCapabilities (BYTE *pDataCapsSize, BYTE *pDataCaps)`
- `ULONG SetNetworkPreference (ULONG technologyPref, ULONG duration)`
- `ULONG GetNetworkPreference (ULONG *pTechnologyPref, ULONG *pDuration, ULONG *pPersistentTechnologyPref)`
- `ULONG GetRFInfo (BYTE *pInstanceSize, struct RFBandInfoElements *pRFBandInfo)`
- `ULONG InitiateDomainAttach (ULONG action)`
- `ULONG GetACCOLC (BYTE *pACCOLC)`
- `ULONG SetACCOLC (CHAR *spc, BYTE accolc)`
- `ULONG SetCDMANetworkParameters (CHAR *pSPC, BYTE *pForceRev0, BYTE *pCustomSCP, ULONG *pProtocol, ULONG *pBroadcast, ULONG *pApplication, ULONG *pRoaming)`
- `ULONG GetCDMANetworkParameters (BYTE *pSCI, BYTE *pSCM, BYTE *pRegHomeSID, BYTE *pRegForeignSID, BYTE *pRegForeignNID, BYTE *pForceRev0, BYTE *pCustomSCP, ULONG *pProtocol, ULONG *pBroadcast, ULONG *pApplication, ULONG *pRoaming)`
- `ULONG GetANAAAAAuthenticationStatus (ULONG *pStatus)`
- `ULONG SLQSGetServingSystem (qmiServingSystemParam *pServingSystem)`
- `ULONG SLQSSetBandPreference (ULONG bandpreference)`
- `ULONG SLQSNasIndicationRegister (BYTE systemSelectionInd, BYTE DDTMInd, BYTE servingSystemInd)`
- `ULONG SLQSGetSignalStrength (struct slqsSignalStrengthInfo *pSignalInfo)`
- `ULONG SLQSPerformNetworkScan (slqsNetworkScanInfo *pNetworkInfo)`
- `ULONG SLQSSetSysSelectionPref (sysSelectPrefParams *pSysSelectPrefParams)`
- `ULONG SLQSGetSysSelectionPref (sysSelectPrefInfo *pSysSelectPrefInfo)`
- `ULONG SLQSNasGetSysInfo (nasGetSysInfoResp *pGetSysInfoResp)`
- `ULONG SLQSNasSwiModemStatus (swiModemStatusResp *pModemStatusResp)`
- `ULONG SLQSNasGetHDRColorCode (nasGetHDRColorCodeResp *pGetHDRColorCodeResp)`
- `ULONG SLQSNasGetTxRxInfo (nasGetTxRxInfoReq *pGetTxRxInfoReq, nasGetTxRxInfoResp *pGetTxRxInfoResp)`
- `ULONG SLQSNasGetSigInfo (nasGetSigInfoResp *pGetSigInfoResp)`



- [ULONG SLQSNasIndicationRegisterExt](#) ([nasIndicationRegisterReq](#) \*pIndicationRegisterReq)
- [ULONG SLQSGetPLMNName](#) ([nasPLMNNameReq](#) \*pPLMNNameReq, [nasPLMNNameResp](#) \*pPLMNNameResp)
- [ULONG SLQSGetOperatorNameData](#) ([nasOperatorNameResp](#) \*pOperatorNameData)
- [ULONG SLQSNasGet3GPP2Subscription](#) ([nasGet3GPP2SubscriptionInfoReq](#) \*pGet3GPP2SubsInfoReq, [nasGet3GPP2SubscriptionInfoResp](#) \*pGet3GPP2SubsInfoResp)
- [ULONG SLQSNasGetCellLocationInfo](#) ([nasCellLocationInfoResp](#) \*pNasCellLocationInfoResp)
- [ULONG SLQSInitiateNetworkRegistration](#) ([nasInitNetworkReg](#) \*pNasInitNetRegistrationReg)
- [ULONG SLQSSwiGetHDRPersonality](#) ([HDRPersonalityResp](#) \*pHDRPersonalityResp)
- [ULONG SLQSSwiGetHDRProtSubtype](#) ([HDRProtSubtypResp](#) \*pHDRProtSubtypResp)
- [ULONG SLQSSwiPSDetach](#) ([PSDetachReq](#) \*pPSDetachReq)
- [ULONG SLQSGetErrorRate](#) ([GetErrRateResp](#) \*pGetErrRateResp)
- [ULONG SLQSSwiGetHRPDStats](#) ([GetHRPDStatsResp](#) \*pGetHRPDStatsResp)
- [ULONG SLQSSwiNetworkDebug](#) ([NetworkDebugResp](#) \*pNetworkDebugResp)
- [ULONG SLQSSwiGetLteCQI](#) ([LteCQIParm](#) \*pLteCQIResp)
- [ULONG SLQSConfigSigInfo](#) ([sigInfo](#) \*pSigInfo)
- [ULONG SLQSNasSwtIndicationRegister](#) ([NasSwtIndReg](#) \*pIndRegReq)
- [ULONG GetHomeNetwork3GPP2](#) ([WORD](#) \*pMCC, [WORD](#) \*pMNC, [BYTE](#) nameSize, [CHAR](#) \*pName, [WORD](#) \*pSID, [WORD](#) \*pNID, [WORD](#) \*pNw2MCC, [WORD](#) \*pNw2MNC, [BYTE](#) \*pNw2DescDisp, [BYTE](#) \*pNw2DescEnc, [BYTE](#) nw2DescLen, [BYTE](#) \*pNw2Name)
- [ULONG SLQSNasConfigSigInfo2](#) ([setSignalStrengthInfo](#) \*pSetSignalStrengthInfo)
- [ULONG SLQSNASGetLTEPHYCaInfo](#) ([nasGetLTECphyCa](#) \*pLTECPhyCa)
- [ULONG SLQSNasIndicationRegisterLTECphyCa](#) ([BYTE](#) \*bStatus)

### 9.13.1 Detailed Description

Network Access Service API function prototypes.

### 9.13.2 Macro Definition Documentation

9.13.2.1 `#define IMSI_M_S1_LENGTH 7`

9.13.2.2 `#define IMSI_M_S2_LENGTH 3`

9.13.2.3 `#define MAX_DATA_SRV_CAPABILITIES 0x20`

9.13.2.4 `#define MAX_DESCRIPTION_LENGTH 255`

9.13.2.5 `#define MAX_PILOT_SETS 0xFF`

9.13.2.6 `#define MAX_SERV_SYSTEM_RADIO_INTERFACES 0x0A`

9.13.2.7 `#define NAM_NAME_LENGTH 12`

9.13.2.8 `#define PLMN_LENGTH 3`

9.13.2.9 `#define SLQS_SS_INFO_LIST_MAX_ELEMENTS 18`

9.13.2.10 `#define SLQS_SYSTEM_ID_SIZE 16`

9.13.2.11 `#define UATISIZE 16`

### 9.13.3 Typedef Documentation

#### 9.13.3.1 `typedef struct _slqsNetworkScanInfo slqsNetworkScanInfo`

Contain the network scan information.

## Parameters

<i>pNetworkInfoInstances</i> [IN/OUT]	<ul style="list-style-type: none"> <li>• Upon input, maximum number of elements that the network info instance array can contain.</li> <li>• Upon successful output, the actual number of elements in the network info instance array.</li> </ul>
<i>pNetworkInfo</i> [OUT]	<ul style="list-style-type: none"> <li>• Network info instance array <ul style="list-style-type: none"> <li>– See <a href="#">SlqsNas3GppNetworkInfo</a> for more information</li> </ul> </li> </ul>
<i>pRATInstances</i> [IN/OUT]	<ul style="list-style-type: none"> <li>• Upon input, maximum number of elements that the RAT info instance array can contain.</li> <li>• Upon successful output, the actual number of elements in the RAT info instance array.</li> </ul>
<i>pRATInfo</i> [OUT]	<ul style="list-style-type: none"> <li>• RAT info instance array <ul style="list-style-type: none"> <li>– See <a href="#">SlqsNas3GppNetworkRAT</a> for more information</li> </ul> </li> </ul>
<i>pPCSDigitInstances</i> [IN/OUT]	<ul style="list-style-type: none"> <li>• Upon input, maximum number of elements that the PCS Digit info instance array can contain.</li> <li>• Upon successful output, the actual number of elements in the PCS Digit info instance array.</li> </ul>
<i>pPCSDigitInfo</i> [OUT]	<ul style="list-style-type: none"> <li>• PCS Digit info instance array <ul style="list-style-type: none"> <li>– See <a href="#">SlqsNasPcsDigit</a> for more information</li> </ul> </li> </ul>
<i>pScanResult</i> [OUT]	<ul style="list-style-type: none"> <li>• status of network scan</li> <li>• 0x00 - scan successful</li> <li>• 0x01 - scan was aborted</li> <li>• 0x02 - scan did not complete due to a radio link failure recovery in progress</li> </ul>

## 9.13.3.2 typedef struct \_sysSelectPrefInfo sysSelectPrefInfo

Structure for storing the current preferred system selection settings for the device.

## Parameters

<i>pEmerMode</i>	<ul style="list-style-type: none"> <li>• Optional parameter specifying the emergency Mode</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0 - OFF (normal)</li> <li>– 1 - ON (Emergency)</li> </ul> </li> <li>• function <a href="#">SLQSGetSysSelectionPref()</a> returns a default value FF if no value is returned by the device.</li> </ul>
<i>pModePref</i>	<ul style="list-style-type: none"> <li>• Optional parameter</li> <li>• Bit Mask indicating the radio technology mode preference</li> <li>• Bit values: <ul style="list-style-type: none"> <li>– Bit 0 - cdma2000 1x</li> <li>– Bit 1 - cdma2000 HRPD(1xEV-DO)</li> <li>– Bit 2 - GSM</li> <li>– Bit 3 - UMTS</li> <li>– Bit 4 - LTE</li> </ul> </li> <li>• function <a href="#">SLQSGetSysSelectionPref()</a> returns a default value FF if no value is returned by the device.</li> </ul>
<i>pBandPref</i>	<ul style="list-style-type: none"> <li>• Optional parameter</li> <li>• Bit mask representing the band preference</li> <li>• Bit values: <ul style="list-style-type: none"> <li>– Bit 0 - Band Class 0, A-System</li> <li>– Bit 1 - Band Class 0, B-System, Band Class 0 AB, GSM 850 Band</li> <li>– Bit 2 - Band Class 1, all blocks</li> <li>– Bit 3 - Band Class 2 place holder</li> <li>– Bit 4 - Band Class 3, A-System</li> <li>– Bit 5 - Band Class 4, all blocks</li> <li>– Bit 6 - Band Class 5, all blocks</li> <li>– Bit 7 - GSM_DCS_1800 band</li> <li>– Bit 8 - GSM Extended GSM (E-GSM) 900 band</li> <li>– Bit 9 - GSM Primary GSM (P-GSM) 900 band</li> <li>– Bit 10 - Band Class 6</li> <li>– Bit 11 - Band Class 7</li> <li>– Bit 12 - Band Class 8</li> <li>– Bit 13 - Band Class 9</li> <li>– Bit 14 - Band Class 10</li> <li>– Bit 15 - Band Class 11</li> <li>– Bit 16 - GSM 450 band</li> <li>– Bit 17 - GSM 480 band</li> <li>– Bit 18 - GSM 750 band</li> <li>– Bit 19 - GSM 850 band</li> </ul> </li> </ul>
	<ul style="list-style-type: none"> <li>– Bit 20 - GSM Railways GSM 900 Band</li> <li>– Bit 21 - GSM PCS 1900 band</li> <li>– Bit 22 - WCDMA Europe, Japan, and China IMT 2100 band</li> <li>– Bit 23 - WCDMA U.S. FCC 1200 band</li> </ul>

<i>pPRLPref</i>	<ul style="list-style-type: none"> <li>Optional parameter indicating the CDMA PRL Preference</li> <li>Values: <ul style="list-style-type: none"> <li>0x0001 - Acquire available system only on the A side</li> <li>0x0002 - Acquire available system only on the B side</li> <li>0x3FFF - Acquire any available systems</li> </ul> </li> <li>function <a href="#">SLQSGetSysSelectionPref()</a> returns a default value FFFF if no value is returned by the device.</li> </ul>
<i>pRoamPref</i>	<ul style="list-style-type: none"> <li>Optional parameter indicating the roaming Preference</li> <li>Values: <ul style="list-style-type: none"> <li>0x01 - Acquire only systems for which the roaming indicator is off</li> <li>0x02 - Acquire a system as long as its roaming indicator is not off</li> <li>0x03 - Acquire only systems for which the roaming indicator is off or solid on, i.e. not flashing; CDMA only</li> <li>0xFF - Acquire systems, regardless of their roaming indicator</li> </ul> </li> <li>function <a href="#">SLQSGetSysSelectionPref()</a> returns a default value FFFF if no value is returned by the device.</li> </ul>
<i>pLTEBandPref</i>	<ul style="list-style-type: none"> <li>Optional parameter</li> <li>Bit mask representing the LTE band preference</li> <li>Bit Values <ul style="list-style-type: none"> <li>Bit 0 - E-UTRA Operating Band 1</li> <li>Bit 1 - E-UTRA Operating Band 2</li> <li>Bit 2 - E-UTRA Operating Band 3</li> <li>Bit 3 - E-UTRA Operating Band 4</li> <li>Bit 4 - E-UTRA Operating Band 5</li> <li>Bit 5 - E-UTRA Operating Band 6</li> <li>Bit 6 - E-UTRA Operating Band 7</li> <li>Bit 7 - E-UTRA Operating Band 8</li> <li>Bit 8 - E-UTRA Operating Band 9</li> <li>Bit 9 - E-UTRA Operating Band 10</li> <li>Bit 10 - E-UTRA Operating Band 11</li> <li>Bit 11 - E-UTRA Operating Band 12</li> <li>Bit 12 - E-UTRA Operating Band 13</li> <li>Bit 13 - E-UTRA Operating Band 14</li> <li>Bit 16 - E-UTRA Operating Band 17</li> <li>Bit 17 - E-UTRA Operating Band 18</li> <li>Bit 18 - E-UTRA Operating Band 19</li> <li>Bit 19 - E-UTRA Operating Band 20</li> <li>Bit 20 - E-UTRA Operating Band 21</li> <li>Bit 32 - E-UTRA Operating Band 33</li> <li>Bit 33 - E-UTRA Operating Band 34</li> <li>Bit 34 - E-UTRA Operating Band 35</li> <li>Bit 35 - E-UTRA Operating Band 36</li> <li>Bit 36 - E-UTRA Operating Band 37</li> </ul> </li> </ul>

<i>pNetSelPref</i>	<ul style="list-style-type: none"> <li>• Optional parameter indicating network selection preference</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0x00 - Automatic network selection</li> <li>– 0x01 - Manual network selection</li> </ul> </li> <li>• function <a href="#">SLQSGetSysSelectionPref()</a> returns a default value FF if no value is returned by the device.</li> </ul>
<i>pSrvDomainPref</i>	<ul style="list-style-type: none"> <li>• Optional parameter indicating Service domain preference</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0x00 - Circuit switched only</li> <li>– 0x01 - Packet switched only</li> <li>– 0x02 - Circuit switched and packet switched</li> <li>– 0x03 - Packet switched attach</li> <li>– 0x04 - Packet switched detach</li> </ul> </li> <li>• function <a href="#">SLQSGetSysSelectionPref()</a> returns a default value FFFFFFFF if no value is returned by the device.</li> </ul>
<i>pGWAcqOrder-Pref</i>	<ul style="list-style-type: none"> <li>• Optional parameter indicating GSM/WCDMA Acquisition order Preference</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0x00 - Automatic</li> <li>– 0x01 - GSM then WCDMA</li> <li>– 0x02 - WCDMA then GSM</li> </ul> </li> <li>• function <a href="#">SLQSGetSysSelectionPref()</a> returns a default value FFFFFFFF if no value is returned by the device.</li> </ul>

**Note**

None

**9.13.3.3 typedef struct \_sysSelectPrefParams sysSelectPrefParams**

Contain the system selection preferences.

## Parameters

<i>pEmerMode</i>	<ul style="list-style-type: none"> <li>Optional parameter specifying the emergency Mode</li> <li>Values: <ul style="list-style-type: none"> <li>0 - OFF (normal)</li> <li>1 - ON (Emergency)</li> </ul> </li> </ul>
<i>pModePref</i>	<ul style="list-style-type: none"> <li>Optional parameter</li> <li>Bit Mask indicating the radio technology mode preference</li> <li>Bit values: <ul style="list-style-type: none"> <li>Bit 0 - cdma2000 1x</li> <li>Bit 1 - cdma2000 HRPD(1xEV-DO)</li> <li>Bit 2 - GSM</li> <li>Bit 3 - UMTS</li> <li>Bit 4 - LTE</li> </ul> </li> </ul>
<i>pBandPref</i>	<ul style="list-style-type: none"> <li>Optional parameter</li> <li>Bit mask representing the band preference</li> <li>Bit values: <ul style="list-style-type: none"> <li>Bit 0 - Band Class 0, A-System</li> <li>Bit 1 - Band Class 0, B-System, Band Class 0 AB, GSM 850 Band</li> <li>Bit 2 - Band Class 1, all blocks</li> <li>Bit 3 - Band Class 2 place holder</li> <li>Bit 4 - Band Class 3, A-System</li> <li>Bit 5 - Band Class 4, all blocks</li> <li>Bit 6 - Band Class 5, all blocks</li> <li>Bit 7 - GSM_DCS_1800 band</li> <li>Bit 8 - GSM Extended GSM (E-GSM) 900 band</li> <li>Bit 9 - GSM Primary GSM (P-GSM) 900 band</li> <li>Bit 10 - Band Class 6</li> <li>Bit 11 - Band Class 7</li> <li>Bit 12 - Band Class 8</li> <li>Bit 13 - Band Class 9</li> <li>Bit 14 - Band Class 10</li> <li>Bit 15 - Band Class 11</li> <li>Bit 16 - GSM 450 band</li> <li>Bit 17 - GSM 480 band</li> <li>Bit 18 - GSM 750 band</li> <li>Bit 19 - GSM 850 band</li> <li>Bit 20 - GSM Railways GSM 900 Band</li> <li>Bit 21 - GSM PCS 1900 band</li> <li>Bit 22 - WCDMA Europe, Japan, and China IMT 2100 band</li> <li>Bit 23 - WCDMA U.S. PCS 1900 band</li> <li>Bit 24 - WCDMA Europe and China DCS 1800 band</li> </ul> </li> </ul>

<i>pPRLPref</i>	<ul style="list-style-type: none"> <li>• Optional parameter indicating the CDMA PRL Preference</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0x0001 - Acquire available system only on the A side</li> <li>– 0x0002 - Acquire available system only on the B side</li> <li>– 0x3FFF - Acquire any available systems</li> </ul> </li> </ul>
<i>pRoamPref</i>	<ul style="list-style-type: none"> <li>• Optional parameter indicating the roaming Preference</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0x01 - Acquire only systems for which the roaming indicator is off</li> <li>– 0x02 - Acquire a system as long as its roaming indicator is not off</li> <li>– 0x03 - Acquire only systems for which the roaming indicator is off or solid on, i.e. not flashing; CDMA only</li> <li>– 0xFF - Acquire systems, regardless of their roaming indicator</li> </ul> </li> </ul>
<i>pLTEBandPref</i>	<ul style="list-style-type: none"> <li>• Optional parameter</li> <li>• Bit mask representing the LTE band preference</li> <li>• Bit Values <ul style="list-style-type: none"> <li>– Bit 0 - E-UTRA Operating Band 1</li> <li>– Bit 1 - E-UTRA Operating Band 2</li> <li>– Bit 2 - E-UTRA Operating Band 3</li> <li>– Bit 3 - E-UTRA Operating Band 4</li> <li>– Bit 4 - E-UTRA Operating Band 5</li> <li>– Bit 5 - E-UTRA Operating Band 6</li> <li>– Bit 6 - E-UTRA Operating Band 7</li> <li>– Bit 7 - E-UTRA Operating Band 8</li> <li>– Bit 8 - E-UTRA Operating Band 9</li> <li>– Bit 9 - E-UTRA Operating Band 10</li> <li>– Bit 10 - E-UTRA Operating Band 11</li> <li>– Bit 11 - E-UTRA Operating Band 12</li> <li>– Bit 12 - E-UTRA Operating Band 13</li> <li>– Bit 13 - E-UTRA Operating Band 14</li> <li>– Bit 16 - E-UTRA Operating Band 17</li> <li>– Bit 17 - E-UTRA Operating Band 18</li> <li>– Bit 18 - E-UTRA Operating Band 19</li> <li>– Bit 19 - E-UTRA Operating Band 20</li> <li>– Bit 20 - E-UTRA Operating Band 21</li> <li>– Bit 32 - E-UTRA Operating Band 33</li> <li>– Bit 33 - E-UTRA Operating Band 34</li> <li>– Bit 34 - E-UTRA Operating Band 35</li> <li>– Bit 35 - E-UTRA Operating Band 36</li> </ul> </li> </ul>
	<ul style="list-style-type: none"> <li>– Bit 36 - E-UTRA Operating Band 37</li> <li>– Bit 37 - E-UTRA Operating Band 38</li> <li>– Bit 38 - E-UTRA Operating Band 39</li> <li>– Bit 39 - E-UTRA Operating Band 40</li> </ul>



<i>pNetSelPref</i>	<ul style="list-style-type: none"> <li>- <a href="#">netSelectionPref</a></li> <li>• Optional parameter for specifying Network Selection Preference</li> <li>• Modem selects networks based on this parameter(if present).</li> <li>• see <a href="#">netSelectionPref</a> for more information</li> </ul>
<i>pChgDuration</i>	<ul style="list-style-type: none"> <li>• Optional parameter specifying the duration of the change</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0x00 - Power cycle - Remains active until the next device power cycle</li> <li>– 0x01 - Permanent - Remains active through power cycles until changed by client</li> <li>– Device will use "0x01 - permanent" as default if this parameter is omitted</li> </ul> </li> </ul>
<i>pMNCIncPCS-DigStat</i>	<ul style="list-style-type: none"> <li>• Optional parameter indicating if MNC includes PCS digit</li> <li>• Values: <ul style="list-style-type: none"> <li>– TRUE - MNC is a 3 digit value; e.g., a reported value of 90 corresponds to an MNC value of 090</li> <li>– FALSE - MNC is a 2-digit value; e.g., a reported value of 90 corresponds to an MNC value of 90</li> </ul> </li> </ul>
<i>pSrvDomainPref</i>	<ul style="list-style-type: none"> <li>• Optional parameter indicating Service domain preference</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0x00 - Circuit switched only</li> <li>– 0x01 - Packet switched only</li> <li>– 0x02 - Circuit switched and packet switched</li> <li>– 0x03 - Packet switched attach</li> <li>– 0x04 - Packet switched detach</li> </ul> </li> </ul>
<i>pGWAcqOrder-Pref</i>	<ul style="list-style-type: none"> <li>• Optional parameter indicating GSM/WCDMA Acquisition order Preference</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0x00 - Automatic</li> <li>– 0x01 - GSM then WCDMA</li> <li>– 0x02 - WCDMA then GSM</li> </ul> </li> </ul>

<i>pRAT</i>	<ul style="list-style-type: none"> <li>• Optional parameter Radio Access Technology order Preference</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0x04 - GSM</li> <li>– 0x05 - UMTS</li> <li>– 0x08 - LTE</li> <li>– 0x09 - TDSCDMA</li> </ul> </li> </ul>
-------------	---

### 9.13.4 Enumeration Type Documentation

#### 9.13.4.1 enum `_NAMS_RADIO_IF_TECHNOLOGY_`

Enumerator

***eNAS\_RADIO\_IF\_GSM***  
***eNAS\_RADIO\_IF\_UMTS***  
***eNAS\_RADIO\_IF\_LTE***  
***eNAS\_RADIO\_IF\_TDSCDMA***

#### 9.13.4.2 enum `NAS_LTE_CPHY_CA_BW_NRB`

Enumerator

***eNAS\_LTE\_CPHY\_CA\_BW\_NRB\_6***  
***eNAS\_LTE\_CPHY\_CA\_BW\_NRB\_15***  
***eNAS\_LTE\_CPHY\_CA\_BW\_NRB\_25***  
***eNAS\_LTE\_CPHY\_CA\_BW\_NRB\_50***  
***eNAS\_LTE\_CPHY\_CA\_BW\_NRB\_75***  
***eNAS\_LTE\_CPHY\_CA\_BW\_NRB\_100***

#### 9.13.4.3 enum `NAS_LTE_CPHY_SCELL_STATE`

Enumerator

***eNAS\_LTE\_CPHY\_SCELL\_STATE\_DECONFIGURED***  
***eNAS\_LTE\_CPHY\_SCELL\_STATE\_CONFIGURED\_DEACTIVATED***  
***eNAS\_LTE\_CPHY\_SCELL\_STATE\_CONFIGURED\_ACTIVATED***

### 9.13.5 Function Documentation

#### 9.13.5.1 `ULONG GetACCOLC ( BYTE * pACCOLC )`

Retrieves information about the access overload class (ACCOLC)

## Parameters

<i>pACCOLC[OUT]</i>	<ul style="list-style-type: none"> <li>• ACCOLC : Valid range is 0 to 15</li> </ul>
---------------------	---

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Technology Supported: CDMA  
Timeout: 2 seconds

9.13.5.2 ULONG GetANAAAAuthenticationStatus ( ULONG \* *pStatus* )

AN-AAA authentication status of the device.

## Parameters

<i>pStatus[OUT]</i>	<ul style="list-style-type: none"> <li>• Status of last AN-AAA authentication attempt <ul style="list-style-type: none"> <li>– 0 - Failure</li> <li>– 1 - Success</li> <li>– 2 - Not Requested</li> </ul> </li> </ul>
---------------------	---

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Technology Supported: CDMA  
Timeout: 2 seconds

9.13.5.3 ULONG GetCDMANetworkParameters ( BYTE \* *pSCI*, BYTE \* *pSCM*, BYTE \* *pRegHomeSID*, BYTE \* *pRegForeignSID*, BYTE \* *pRegForeignNID*, BYTE \* *pForceRev0*, BYTE \* *pCustomSCP*, ULONG \* *pProtocol*, ULONG \* *pBroadcast*, ULONG \* *pApplication*, ULONG \* *pRoaming* )

Gets the current CDMA network parameters

## Parameters

<i>pSCI[OUT]</i>	<ul style="list-style-type: none"> <li>Slot cycle index             <ul style="list-style-type: none"> <li>– 0xFF-Unknown</li> </ul> </li> </ul>
<i>pSCM[OUT]</i>	<ul style="list-style-type: none"> <li>Station class mark             <ul style="list-style-type: none"> <li>– 0xFF-Unknown</li> </ul> </li> </ul>
<i>pRegHomeSID[OUT]</i>	<ul style="list-style-type: none"> <li>Register on home SID             <ul style="list-style-type: none"> <li>– 0 - Disabled</li> <li>– 1 - Enabled</li> <li>– 0xFF - Unknown</li> </ul> </li> </ul>
<i>pRegForeignSID[OUT]</i>	<ul style="list-style-type: none"> <li>Register on foreign SID             <ul style="list-style-type: none"> <li>– 0 - Disabled</li> <li>– 1 - Enabled</li> <li>– 0xFF - Unknown</li> </ul> </li> </ul>
<i>pRegForeignNID[OUT]</i>	<ul style="list-style-type: none"> <li>Register on foreign NID             <ul style="list-style-type: none"> <li>– 0 - Disabled</li> <li>– 1 - Enabled</li> <li>– 0xFF - Unknown</li> </ul> </li> </ul>
<i>pForceRev0[OUT]</i>	<ul style="list-style-type: none"> <li>Force CDMA 1x-EV-DO Rev. 0 mode             <ul style="list-style-type: none"> <li>– 0 - Disabled</li> <li>– 1 - Enabled</li> <li>– 0xFF - Unknown</li> </ul> </li> </ul>
<i>pCustomSCP[OUT]</i>	<ul style="list-style-type: none"> <li>Use a custom config for CDMA 1x-EV-DO SCP             <ul style="list-style-type: none"> <li>– 0 - Disabled</li> <li>– 1 - Enabled</li> <li>– 0xFF - Unknown</li> </ul> </li> </ul>

<i>pProtocol</i> [OUT]	<ul style="list-style-type: none"> <li>• Protocol mask for custom SCP config <ul style="list-style-type: none"> <li>– 0x00000001 - Subtype 2 Physical Layer</li> <li>– 0x00000002 - Enhanced CCMAC</li> <li>– 0x00000004 - Enhanced ACMAC</li> <li>– 0x00000008 - Enhanced FTCMAC</li> <li>– 0x00000010 - Subtype 3 RTCMAC</li> <li>– 0x00000020 - Subsystem 1 RTCMAC</li> <li>– 0x00000040 - Enhanced Idle</li> <li>– 0x00000080 - Generic Multimode Capable Disc <a href="#">Port</a></li> <li>– 0xFFFFFFFF - Unknown</li> </ul> </li> </ul>
<i>pBroadcast</i> [OUT]	<ul style="list-style-type: none"> <li>• Broadcast mask for custom SCP config <ul style="list-style-type: none"> <li>– 0x00000001 - Generic broadcast enabled</li> <li>– 0xFFFFFFFF - Unknown</li> </ul> </li> </ul>
<i>pApplication</i> [OUT]	<ul style="list-style-type: none"> <li>• Application mask for custom SCP config <ul style="list-style-type: none"> <li>– 0x00000001 - SN Multiflow Packet Application</li> <li>– 0x00000002 - SN Enhanced Multiflow Packet Application</li> <li>– 0xFFFFFFFF - Unknown</li> </ul> </li> </ul>
<i>pRoaming</i> [OUT]	<ul style="list-style-type: none"> <li>• Roaming preference <ul style="list-style-type: none"> <li>– 0 - Automatic</li> <li>– 1 - Home Only</li> <li>– 2 - Affiliated Roaming Only</li> <li>– 3 - Home and Affiliated Roaming</li> <li>– 0xFFFFFFFF - Unknown</li> </ul> </li> </ul>

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Technology Supported: CDMA  
Timeout: 5 seconds

9.13.5.4 **ULONG** GetHomeNetwork ( **WORD** \* *pMCC*, **WORD** \* *pMNC*, **BYTE** *nameSize*, **CHAR** \* *pName*, **WORD** \* *pSID*, **WORD** \* *pNID* )

Retrieves information about the home network of the device. For 3GPP2 home network information use GetHomeNetwork3GPP2.

## Parameters

<i>pMCC[OUT]</i>	<ul style="list-style-type: none"><li>• Mobile country code (UMTS only).</li></ul>
<i>pMNC[OUT]</i>	<ul style="list-style-type: none"><li>• Mobile network code (UMTS only).</li></ul>
<i>nameSize</i>	<ul style="list-style-type: none"><li>• Maximum number of characters (including NULL terminator) that 8 network name array can contain (UMTS only).</li></ul>
<i>pName[OUT]</i>	<ul style="list-style-type: none"><li>• Network name or description represented as a NULL terminated string (empty string returned when unknown) (UMTS only).</li></ul>
<i>pSID[OUT]</i>	<ul style="list-style-type: none"><li>• Home network system ID<ul style="list-style-type: none"><li>– 0xFFFF - Unknown.</li><li>– Only applies to cdma2000</li></ul></li></ul>
<i>pNID[OUT]</i>	<ul style="list-style-type: none"><li>• Home network ID<ul style="list-style-type: none"><li>– 0xFFFF - Unknown.</li><li>– Only applies to cdma2000</li></ul></li></ul>

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Timeout: 2 seconds

**9.13.5.5** **ULONG** GetHomeNetwork3GPP2 ( **WORD** \* *pMCC*, **WORD** \* *pMNC*, **BYTE** *nameSize*, **CHAR** \* *pName*,  
**WORD** \* *pSID*, **WORD** \* *pNID*, **WORD** \* *pNw2MCC*, **WORD** \* *pNw2MNC*, **BYTE** \* *pNw2DescDisp*, **BYTE** \*  
*pNw2DescEnc*, **BYTE** *nw2DescLen*, **BYTE** \* *pNw2Name* )

Retrieves information about the home network of the device. It will extract 3GPP2 Network Information also.



## Parameters

<i>pMCC[OUT]</i>	<ul style="list-style-type: none"> <li>• Mobile country code (UMTS only).</li> </ul>
<i>pMNC[OUT]</i>	<ul style="list-style-type: none"> <li>• Mobile network code (UMTS only).</li> </ul>
<i>nameSize</i>	<ul style="list-style-type: none"> <li>• Maximum number of characters (including NULL terminator) that 8 network name array can contain (UMTS only).</li> </ul>
<i>pName[OUT]</i>	<ul style="list-style-type: none"> <li>• Network name or description represented as a NULL terminated string (empty string returned when unknown) (UMTS only).</li> </ul>
<i>pSID[OUT]</i>	<ul style="list-style-type: none"> <li>• Home network system ID <ul style="list-style-type: none"> <li>– 0xFFFF - Unknown.</li> <li>– Only applies to cdma2000</li> </ul> </li> </ul>
<i>pNID[OUT]</i>	<ul style="list-style-type: none"> <li>• Home network ID <ul style="list-style-type: none"> <li>– 0xFFFF - Unknown.</li> <li>– Only applies to cdma2000</li> </ul> </li> </ul>
<i>pNw2MCC[OUT]</i>	<ul style="list-style-type: none"> <li>• Mobile country code (3GPP2 only).</li> <li>• Range : 0 to 999</li> </ul>
<i>pNw2MNC[OUT]</i>	<ul style="list-style-type: none"> <li>• Mobile network code (3GPP2 only).</li> <li>• Range : 0 to 999</li> </ul>
<i>pNw2DescDisp[OUT]</i>	<ul style="list-style-type: none"> <li>• Network Name Display (3GPP2 only). -Valid Value <ul style="list-style-type: none"> <li>– 0x00 - Do not display</li> <li>– 0x01 - Display</li> <li>– 0xFF - Unknown</li> </ul> </li> </ul>

<i>pNw2DescDisp[OUT]</i>	<ul style="list-style-type: none"> <li>• Encoding of the network description (3GPP2 only).</li> <li>• Valid Value <ul style="list-style-type: none"> <li>– 0x00 - Octet, unspecified</li> <li>– 0x02 - 7-bit ASCII</li> <li>– 0x04 - Unicode</li> <li>– 0x09 - GSM 7-bit default</li> </ul> </li> </ul>
<i>nw2DescLen[OUT]</i>	<ul style="list-style-type: none"> <li>• Network Description Length (3GPP2 only).</li> </ul>
<i>pNw2Name[OUT]</i>	<ul style="list-style-type: none"> <li>• Network Name (3GPP2 only).</li> </ul>

#### Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

#### See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

#### Note

Timeout: 2 seconds

#### 9.13.5.6 ULONG GetNetworkPreference ( ULONG \* pTechnologyPref, ULONG \* pDuration, ULONG \* pPersistentTechnologyPref )

Returns the network registration preference. This API is deprecated on MC73xx/EM73xx modules since firmware version SWI9X15C\_05\_xx\_xx\_xx and all EM74xx firmware versions. Please use API [SLQGetSysSelectionPref\(\)](#) for new firmware versions and new modules

## Parameters

<i>pTechnologyPref[OUT]</i>	<ul style="list-style-type: none"> <li>• Bitmask representing the radio technology preference set.</li> <li>• No bits set indicates to the device to automatically determine the technology to use</li> <li>• Values: <ul style="list-style-type: none"> <li>– Bit 0 - Technology is 3GPP2</li> <li>– Bit 1 - Technology is 3GPP</li> </ul> </li> <li>• Any combination of the following may be returned: <ul style="list-style-type: none"> <li>– Bit 2 - Analog - AMPS if 3GPP2, GSM if 3GPP</li> <li>– Bit 3 - Digital - CDMA if 3GPP2, WCDMA if 3GPP</li> <li>– Bit 4 - HDR</li> <li>– Bit 5 - LTE</li> <li>– Bits 6 to 15 - Reserved</li> </ul> </li> </ul>
<i>pDuration[OUT]</i>	<ul style="list-style-type: none"> <li>• Duration of active preference <ul style="list-style-type: none"> <li>– 0 - Permanent</li> <li>– 1 - Power cycle</li> <li>– 2 - Until the end of the next call or a power cycle</li> <li>– 3 - Until the end of the next call, a specified time, or a power cycle</li> <li>– 4 to 6 - Until the end of the next call</li> </ul> </li> </ul>
<i>pPersistentTechnologyPref[OUT]</i>	<ul style="list-style-type: none"> <li>• Bit field representing persistent radio technology preference <ul style="list-style-type: none"> <li>– Same representation as the pTechnologyPref parameter</li> </ul> </li> </ul>

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 2 seconds

#### 9.13.5.7 ULONG GetRFInfo ( BYTE \* pInstanceSize, struct RFBandInfoElements \* pRFBandInfo )

Sets the RFInfoList

## Parameters

<i>pInstanceSize</i> [I- N/OUT]	<ul style="list-style-type: none"> <li>• Upon input, maximum number of elements that the RF info instances array can contain.</li> <li>• Upon successful output, actual number of elements in RF info instances array.</li> </ul>
<i>pInstances</i> [OUT]	<ul style="list-style-type: none"> <li>• RF info instances array <ul style="list-style-type: none"> <li>– See <a href="#">RFBandInfoElements</a> for more information</li> </ul> </li> </ul>

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 2 seconds

**9.13.5.8** **ULONG** GetServingNetwork ( **ULONG** \* *pRegistrationState*, **ULONG** \* *pCSDomain*, **ULONG** \* *pPSDomain*, **ULONG** \* *pRAN*, **BYTE** \* *pRadiolfacesSize*, **BYTE** \* *pRadiolfaces*, **ULONG** \* *pRoaming*, **WORD** \* *pMCC*, **WORD** \* *pMNC*, **BYTE** *nameSize*, **CHAR** \* *pName* )

Provides information about the system that provides service to the device. This API is deprecated on MC73xx/E-M73xx modules since firmware version SWI9X15C\_05\_xx\_xx\_xx and all EM74xx firmware versions. Please use API [SLQSNasGetSysInfo\(\)](#) for new firmware versions and new modules

## Parameters

<i>pRegistration- State</i> [OUT]	<ul style="list-style-type: none"> <li>• Registration state: <ul style="list-style-type: none"> <li>– 0 - Not registered</li> <li>– 1 - Registered</li> <li>– 2 - Searching/Not Registered</li> <li>– 3 - Registration Denied</li> <li>– 4 - Unknown</li> </ul> </li> </ul>
---------------------------------------	---

<i>pCSDomain[OUT]</i>	<ul style="list-style-type: none"> <li>• Circuit switch domain status: <ul style="list-style-type: none"> <li>– 0 - Unknown/Not Applicable</li> <li>– 1 - Attached</li> <li>– 2 - Detached</li> </ul> </li> </ul>
<i>pPSDomain[OUT]</i>	<ul style="list-style-type: none"> <li>• Packet switch domain status <ul style="list-style-type: none"> <li>– 0 - Unknown/Not Applicable</li> <li>– 1 - Attached</li> <li>– 2 - Detached</li> </ul> </li> </ul>
<i>pRAN[OUT]</i>	<ul style="list-style-type: none"> <li>• Type of radio access network on which mobile is registered: <ul style="list-style-type: none"> <li>– 0 - Unknown</li> <li>– 1 - cdma2000 network</li> <li>– 2 - UMTS network</li> </ul> </li> </ul>
<i>pRadioIfaces-Size[IN/OUT]</i>	<ul style="list-style-type: none"> <li>• Upon input, maximum number of elements that the radio interface array contain.</li> <li>• Upon successful output, actual number of elements in the radio interface array.</li> </ul>
<i>pRadioIfaces[OUT]</i>	<ul style="list-style-type: none"> <li>• An array of Radio Interface Technology <ul style="list-style-type: none"> <li>– See <a href="#">qaGobiApiTableRadioInterfaces.h</a> for the Radio Interface Technologies</li> </ul> </li> </ul>
<i>pRoaming[OUT]</i>	<ul style="list-style-type: none"> <li>• Roaming indicator</li> </ul>
<i>pMCC[OUT]</i>	<ul style="list-style-type: none"> <li>• Mobile country code</li> </ul>
<i>pMNC[OUT]</i>	<ul style="list-style-type: none"> <li>• Mobile network code</li> </ul>
<i>nameSize</i>	<ul style="list-style-type: none"> <li>• Maximum number of characters (including NULL terminator) that network name array can contain; applicable only for UMTS networks</li> </ul>
<i>pName[OUT]</i>	<ul style="list-style-type: none"> <li>• Network name or description represented as a NULL terminated string; empty string is returned when unknown; applicable only for UMTS networks</li> </ul>

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Timeout: 2 seconds

### 9.13.5.9 ULONG GetServingNetworkCapabilities ( BYTE \* *pDataCapsSize*, BYTE \* *pDataCaps* )

Returns information regarding the data capabilities of the system that currently provides service to the device.

**Parameters**

<i>pDataCapsSize</i> [IN/OUT]	<ul style="list-style-type: none"> <li>• Upon input, the maximum number of elements the data capabilities array can contain.</li> <li>• Upon output, the actual number of elements in the data capabilities array.</li> </ul>
<i>pDataCaps</i> [OUT]	<ul style="list-style-type: none"> <li>• Data capabilities array of unsigned long type <ul style="list-style-type: none"> <li>– 1 - GPRS</li> <li>– 2 - EDGE</li> <li>– 3 - HSDPA</li> <li>– 4 - HSUPA</li> <li>– 5 - WCDMA</li> <li>– 6 - CDMA 1xRTT</li> <li>– 7 - CDMA 1xEV-DO Rev 0</li> <li>– 8 - CDMA 1xEV-DO Rev. A</li> <li>– 9 - GSM</li> <li>– 10 - EVDO Rev. B</li> <li>– 11 - LTE</li> <li>– 12 - HSDPA Plus</li> <li>– 13 - Dual Carrier HSDPA Plus</li> </ul> </li> </ul>

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Timeout: 2 seconds

**9.13.5.10** `ULONG GetSignalStrengths ( ULONG * pArraySizes, INT8 * pSignalStrength, ULONG * pRadioInterface )`

Returns the available signal strengths ( in dBm ) as measured by the device in an array. The API also provides the corresponding radio radio interfaces in an array.

## Parameters

<i>pArraySizes</i> [IN/-OUT]	<ul style="list-style-type: none"> <li>• Upon input maximum number of elements that each array can contain.</li> <li>• Upon successful output actual number of elements in the array.</li> </ul>
<i>pSignal-Strength</i> [OUT]	<ul style="list-style-type: none"> <li>• Received signal strength array (in dBm)</li> </ul>
<i>pRadio-Interface</i> [OUT]	<ul style="list-style-type: none"> <li>• Radio interface technology array of the signal being measured <ul style="list-style-type: none"> <li>– See <a href="#">qaGobiApiTableRadioInterfaces.h</a> for Radio Interface info</li> </ul> </li> </ul>

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 2 seconds

9.13.5.11 **ULONG** InitiateDomainAttach ( **ULONG** *action* )

Initiates a domain attach/detach of the device. This API is deprecated on MC73xx/EM73xx modules since firmware version SWI9X15C\_05\_xx\_xx\_xx and all EM74xx firmware versions. Please use API [SLQSSetSysSelectionPref\(\)](#) for new firmware versions and new modules

## Parameters

<i>action</i>	<ul style="list-style-type: none"> <li>• <a href="#">Domain</a> action to attempt <ul style="list-style-type: none"> <li>1 - Attach</li> <li>2 - Detach</li> </ul> </li> </ul>
---------------	--

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 30 seconds



**9.13.5.12** `ULONG InitiateNetworkRegistration ( ULONG regType, WORD mcc, WORD mnc, ULONG rat )`

Initiates the network registration process. This API is deprecated on MC73xx/EM73xx modules since firmware version SWI9X15C\_05\_xx\_xx\_xx and all EM74xx firmware versions. Please use API [SLQSSetSysSelectionPref\(\)](#) and [SLQSSetBandPreference\(\)](#) for new firmware versions and new modules

## Parameters

<i>regType</i>	<ul style="list-style-type: none"> <li>• Registration type <ul style="list-style-type: none"> <li>– 1 - Automatic</li> <li>– 2 - Manual</li> </ul> </li> </ul>
<i>mcc</i>	<ul style="list-style-type: none"> <li>• Mobile country code</li> </ul>
<i>mnc</i>	<ul style="list-style-type: none"> <li>• Mobile network code</li> </ul>
<i>rat</i>	<ul style="list-style-type: none"> <li>• Radio access technology <ul style="list-style-type: none"> <li>– 4 - GSM</li> <li>– 5 - UMTS</li> </ul> </li> </ul>

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Technology Supported: UMTS  
Timeout: 30 seconds

### 9.13.5.13 ULONG PerformNetworkScan ( BYTE \* *pInstanceSize*, BYTE \* *pInstances* )

Performs scan for available networks.

## Parameters

<i>pInstanceSize</i> [I- N/OUT]	<ul style="list-style-type: none"> <li>• Upon input, maximum number of elements that the network info instance array can contain.</li> <li>• Upon successful output, the actual number of elements in the network info instance array.</li> </ul>
------------------------------------	---

<i>pInstances[OUT]</i>	<ul style="list-style-type: none"> <li>• Network info instance array <ul style="list-style-type: none"> <li>– See <a href="#">QmiNas3GppNetworkInfo</a></li> </ul> </li> </ul>
------------------------	--

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Technology Supported: UMTS  
Timeout: 5 minutes

**9.13.5.14 ULONG SetACCOLC ( CHAR \* *spc*, BYTE *accolc* )**

Sets the access overload class (ACCOLC)

**Parameters**

<i>spc[IN]</i>	<ul style="list-style-type: none"> <li>• service programming code NULL-terminated string of six digit</li> </ul>
<i>ACCOLC</i>	<ul style="list-style-type: none"> <li>• ACCOLC : Valid range is 0 to 15</li> </ul>

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Technology Supported: CDMA  
Timeout: 5 seconds

**9.13.5.15 ULONG SetCDMANetworkParameters ( CHAR \* *pSPC*, BYTE \* *pForceRev0*, BYTE \* *pCustomSCP*, ULONG \* *pProtocol*, ULONG \* *pBroadcast*, ULONG \* *pApplication*, ULONG \* *pRoaming* )**

Sets the CDMA network parameters. Currently the modified settings will not be utilized until the device has been reset. For this reason, the recommended approach when using SetCDMANetworkParameters is for the application to perform the following steps:

1 - Call [SetCDMANetworkParameters\(\)](#) 2 - Call SetPower( 5 ) 3 - Call [QCWWANDisconnect\(\)](#) 4 - Reconnect after the device power cycles

## Parameters

<i>pSPC</i> [IN]	<ul style="list-style-type: none"> <li>• Six digit service programming code (not necessary when only the roaming field is being set)</li> </ul>
<i>pForceRev0</i> [IN]	<ul style="list-style-type: none"> <li>• (Optional)Force CDMA 1x-EV-DO Rev. 0 mode <ul style="list-style-type: none"> <li>– 0 - Disabled</li> <li>– 1 - Enabled Note: Enabled can only be specified if pCustomSCP state is set to Disabled</li> </ul> </li> </ul>
<i>pCustomSCP</i> [I-N]	<ul style="list-style-type: none"> <li>• (Optional)Use a custom config for CDMA 1x-EV-DO SCP <ul style="list-style-type: none"> <li>– 0 - Disabled</li> <li>– 1 - Enabled Note: Enabled can only be specified if pForceRev0 is set to Disabled</li> </ul> </li> </ul>
<i>pProtocol</i> [IN]	<ul style="list-style-type: none"> <li>• Protocol mask for custom SCP config <ul style="list-style-type: none"> <li>– 0x00000001 - Subtype 2 Physical Layer</li> <li>– 0x00000002 - Enhanced CCMAC</li> <li>– 0x00000004 - Enhanced ACMAC</li> <li>– 0x00000008 - Enhanced FTCMAC</li> <li>– 0x00000010 - Subtype 3 RTCMAC</li> <li>– 0x00000020 - Subsystem 1 RTCMAC</li> <li>– 0x00000040 - Enhanced Idle</li> <li>– 0x00000080 - Generic Multimode Capable Disc <a href="#">Port</a></li> <li>– 0xFFFFFFFF - Unknown</li> </ul> </li> </ul>
<i>pBroadcast</i> [IN]	<ul style="list-style-type: none"> <li>• Broadcast mask for custom SCP config <ul style="list-style-type: none"> <li>– 0x00000001 - Generic broadcast enabled</li> <li>– 0xFFFFFFFF - Unknown</li> </ul> </li> </ul>
<i>pApplication</i> [IN]	<ul style="list-style-type: none"> <li>• Application mask for custom SCP config <ul style="list-style-type: none"> <li>– 0x00000001 - SN Multiflow Packet Application</li> <li>– 0x00000002 - SN Enhanced Multiflow Packet Application</li> <li>– 0xFFFFFFFF - Unknown</li> </ul> </li> </ul>

<i>pRoaming</i> [!N]	<ul style="list-style-type: none"> <li>• Roaming preference <ul style="list-style-type: none"> <li>– 0 - Automatic</li> <li>– 1 - Home Only</li> <li>– 2 - Affiliated Roaming Only</li> <li>– 3 - Home and Affiliated Roaming</li> <li>– 0xFFFFFFFF - Unknown</li> </ul> </li> </ul>
----------------------	--

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Technology Supported: CDMA  
Timeout: 5 seconds

#### 9.13.5.16 ULONG SetNetworkPreference ( ULONG *technologyPref*, ULONG *duration* )

Sets the network registration preference. This API is deprecated on MC73xx/EM73xx modules since firmware version SWI9X15C\_05\_xx\_xx\_xx and all EM74xx firmware versions. Please use API [SLQSSetSysSelectionPref\(\)](#) for new firmware versions and new modules

**Parameters**

<i>technologyPref</i>	<ul style="list-style-type: none"> <li>• 2 Byte Bitmask representing radio technology preference <ul style="list-style-type: none"> <li>– No bits set indicates device to automatically determine the technology to use.</li> <li>– Type of technology <ul style="list-style-type: none"> <li>* Bit 0 - Technology is 3GPP2</li> <li>* Bit 1 - Technology is 3GPP</li> </ul> </li> <li>– Technology-specific protocol bitmask <ul style="list-style-type: none"> <li>* Bit 2 - Analog <ul style="list-style-type: none"> <li>· AMPS if 3GPP2, GSM if 3GPP</li> </ul> </li> <li>* Bit 3 - Digital <ul style="list-style-type: none"> <li>· CDMA if 3GPP2, WCDMA if 3GPP</li> </ul> </li> <li>* Bit 4 - HDR</li> <li>* Bit 5 - LTE</li> <li>* All other bits are reserved.</li> </ul> </li> </ul> </li> </ul>
-----------------------	---

<i>duration</i>	<ul style="list-style-type: none"> <li>• Duration of active preference <ul style="list-style-type: none"> <li>– 0 - Persistent</li> <li>– 1 - Power cycle</li> </ul> </li> </ul>
-----------------	--

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Timeout: 2 seconds

**9.13.5.17 ULONG SLQSConfigSigInfo ( sigInfo \* pSigInfo )**

Sets the signal reporting thresholds. This API is deprecated on MC73xx/EM73xx modules since firmware version SWI9X15C\_05\_xx\_xx\_xx and all EM74xx firmware versions.

**Parameters**

<i>pSigInfo[IN]</i>	<ul style="list-style-type: none"> <li>• See <a href="#">sigInfo</a> for more information</li> </ul>
---------------------	--

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Timeout: 5 seconds

**9.13.5.18 ULONG SLQSGetErrorRate ( GetErrRateResp \* pGetErrRateResp )**

This API retrieves current error rate information

**Parameters**

<i>pGetErrRateResp[OUT]</i>	<ul style="list-style-type: none"> <li>• See <a href="#">GetErrRateResp</a> for more information</li> </ul>
-----------------------------	---

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Timeout: 5 seconds

**9.13.5.19 ULONG SLQSGetOperatorNameData ( nasOperatorNameResp \* pOperatorNameData )**

Get the operator name data from the network. This API is deprecated on MC73xx/EM73xx modules since firmware version SWI9X15C\_05\_xx\_xx\_xx and all EM74xx firmware versions. Please use API [SLQSGetPLMNName\(\)](#) for new firmware versions and new modules

**Parameters**

<i>pOperatorName-Data[OUT]</i>	<ul style="list-style-type: none"><li>• See <a href="#">nasOperatorNameResp</a> for more information</li></ul>
--------------------------------	--

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Timeout: 10 seconds

**9.13.5.20 ULONG SLQSGetPLMNName ( nasPLMNNameReq \* pPLMNNameReq, nasPLMNNameResp \* pPLMNNameResp )**

Get the operator name data from the network.

**Parameters**

<i>pPLMNName-Req[IN]</i>	<ul style="list-style-type: none"><li>• See <a href="#">nasPLMNNameReq</a> for more information</li></ul>
<i>pPLMNName-Resp[OUT]</i>	<ul style="list-style-type: none"><li>• See <a href="#">nasPLMNNameResp</a> for more information</li></ul>

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 10 seconds

#### 9.13.5.21 ULONG SLQSGetServingSystem ( qaQmiServingSystemParam \* pServingSystem )

Provides information about the system that provides service to the device. This API is deprecated on MC73xx/E-M73xx modules since firmware version SWI9X15C\_05\_xx\_xx\_xx and all EM74xx firmware versions. Please use API [SLQSNasGetSysInfo\(\)](#) for new firmware versions and new modules. Also report available radio interface technology. If there are more than one radio interface, please choose the right interface(usually the first pair).

- See [SLQSSetSysSelectionPref](#)

## Parameters

<i>pServing-System</i> [OUT]	<ul style="list-style-type: none"> <li>• serving system parameters obtained from the system</li> </ul>
------------------------------	--

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise.

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values.

## Note

Timeout: 2 seconds

#### 9.13.5.22 ULONG SLQSGetSignalStrength ( struct slqsSignalStrengthInfo \* pSignalInfo )

Queries the current signal strength as measured by the device. This API is deprecated on MC73xx/EM73xx modules since firmware version SWI9X15C\_05\_xx\_xx\_xx and all EM74xx firmware versions. Please use API [SLQSNasGetSigInfo\(\)](#) for new firmware versions and new modules

## Parameters

<i>pSignalInfo</i> [IN/-OUT]	<ul style="list-style-type: none"> <li>• See <a href="#">slqsSignalStrengthInfo</a> for more information</li> </ul>
------------------------------	---

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 2 seconds



9.13.5.23 **ULONG** SLQSGetSysSelectionPref ( **sysSelectPrefInfo** \* *pSysSelectPrefInfo* )

Queries the different system selection preferences of the device.

## Parameters

<i>pSysSelectPref-Info[OUT]</i>	<ul style="list-style-type: none"> <li>• See <a href="#">sysSelectPrefInfo</a> for more information</li> </ul>
---------------------------------	--

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 2 seconds

#### 9.13.5.24 ULONG SLQSIInitiateNetworkRegistration ( nasInitNetworkReg \* pNasInitNetRegistrationReg )

Initiates the network registration process.

## Parameters

<i>pNasInitNet-RegistrationReq</i>	<ul style="list-style-type: none"> <li>• Pointer to structure nasInitNetworkReq             <ul style="list-style-type: none"> <li>– See <a href="#">nasInitNetworkReg</a> for more information</li> </ul> </li> </ul>
------------------------------------	--

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Technology Supported: UMTS  
Timeout: 30 seconds

#### 9.13.5.25 ULONG SLQSNasConfigSigInfo2 ( setSignalStrengthInfo \* pSetSignalStrengthInfo )

Sets the signal strength reporting thresholds

## Parameters

<i>pSetSignal-StrengthInfo[IN]</i>	<ul style="list-style-type: none"> <li>• See <a href="#">setSignalStrengthInfo</a> for more information</li> </ul>
------------------------------------	--

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 5 seconds

Mixture of threshold and delta values can be provided in the request. But for each type and RAT, only one of threshold list or delta value is to be provided.

#### 9.13.5.26 ULONG SLQSNasGet3GPP2Subscription ( nasGet3GPP2SubscriptionInfoReq \* pGet3GPP2SubsInfoReq, nasGet3GPP2SubscriptionInfoResp \* pGet3GPP2SubsInfoResp )

This API retrieves 3GPP2 subscription-related information.

## Parameters

<i>pGet3GPP2-SubsInfoReq[IN]</i>	<ul style="list-style-type: none"> <li>See <a href="#">nasGet3GPP2SubscriptionInfoReq</a> for more information</li> </ul>
<i>pGet3GPP2-SubsInfoResp[OUT]</i>	<ul style="list-style-type: none"> <li>See <a href="#">nasGet3GPP2SubscriptionInfoResp</a> for more information</li> </ul>

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Technology Supported: CDMA

Timeout: 2 seconds

This command retrieves 3GPP2 subscription-related information. The QMI\_ERR\_INTERNAL error is returned when no information can be retrieved from the modem.

#### 9.13.5.27 ULONG SLQSNasGetCellLocationInfo ( nasCellLocationInfoResp \* pNasCellLocationInfoResp )

This API retrieves cell location-related information

## Parameters

<i>pNasCell-LocationInfo-Resp[OUT]</i>	<ul style="list-style-type: none"> <li>See <a href="#">nasCellLocationInfoResp</a> for more information</li> </ul>
--	--

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Timeout: 2 seconds

This API retrieves cell location-related information, depending on current serving system.

#### 9.13.5.28 **ULONG** SLQSNasGetHDRColorCode ( **nasGetHDRColorCodeResp** \* **pGetHDRColorCodeResp** )

This API retrieves the current HDR color code value.

**Parameters**

<i>pGetHDRColorCodeResp</i> [OUT]	<ul style="list-style-type: none"> <li>• See <a href="#">nasGetHDRColorCodeResp</a> for more information</li> </ul>
-----------------------------------	---

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Timeout: 2 seconds

#### 9.13.5.29 **ULONG** SLQSNASGetLTECPHYCaInfo ( **nasGetLTECPHYCa** \* **pLTECPHYCa** )

This API Get LTE CPHY Carrier Info

**Parameters**

<i>pLTECPHYCa</i> [IN]	<ul style="list-style-type: none"> <li>• See <a href="#">nasGetLTECPHYCa</a> for more information.</li> </ul>
------------------------	---

**Returns**

eQCWWAN\_ERR\_sNONE on success, eQCWWAN\_xxx error value otherwise.

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values.

#### 9.13.5.30 **ULONG** SLQSNasGetSigInfo ( **nasGetSigInfoResp** \* **pGetSigInfoResp** )

This API queries information regarding the signal strength.

## Parameters

<i>pGetSigInfoResp</i> [OUT]	<ul style="list-style-type: none"> <li>See <a href="#">nasGetSigInfoResp</a> for more information</li> </ul>
------------------------------	--

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 2 seconds

This command queries the signal strength information for currently active RATs. Information is reported only if the corresponding RATs have signal strength values to be reported. If no signal strength information is available for any RAT, the response message contains only the mandatory response message

#### 9.13.5.31 ULONG SLQSNasGetSysInfo ( nasGetSysInfoResp \* pGetSysInfoResp )

Provides the system information. This API is preferred when trying to get the service status info and serving system info. The API [SLQSGetServingSystem\(\)](#) reports similar NAS information, but it is deprecated. Please refer to the header description of API [SLQSGetServingSystem\(\)](#) for more information.

## Parameters

<i>pGetSysInfoResp</i> [OUT]	<ul style="list-style-type: none"> <li>See <a href="#">nasGetSysInfoResp</a> for more information</li> </ul>
------------------------------	--

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 2 seconds

This API queries current serving system information, including registration information and system property. The registration information for all RATs specified in the mode capability setting are included regardless of registration status. The RAT-specific system property are included only for RATs that are specified in the mode capability setting and which are not in either No Service or Power Save modes.

#### 9.13.5.32 ULONG SLQSNasGetTxRxInfo ( nasGetTxRxInfoReq \* pGetTxRxInfoReq, nasGetTxRxInfoResp \* pGetTxRxInfoResp )

This API retrieves the detailed Tx/Rx information.

## Parameters

<i>pGetTxRxInfo-Req</i> [IN]	<ul style="list-style-type: none"><li>• See <a href="#">nasGetTxRxInfoReq</a> for more information</li></ul>
<i>pGetTxRxInfo-Resp</i> [OUT]	<ul style="list-style-type: none"><li>• See <a href="#">nasGetTxRxInfoResp</a> for more information</li></ul>

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 2 seconds

This command retrieves Tx/Rx information for a radio interface. The Rx chain are included in the response message only if they are enabled.

#### 9.13.5.33 ULONG SLQSNasIndicationRegister ( BYTE *systemSelectionInd*, BYTE *DDTMInd*, BYTE *servingSystemInd* )

Register/De-register from NAS (Network access service) broadcast indications. Some callbacks would not be invoked if the indications are not registered. The details are provided in the parameter description.

## Parameters

<i>system-SelectionInd</i>	<ul style="list-style-type: none"> <li>• system selection preference indication registration. The following callbacks would not be invoked if the indication is disabled.  <a href="#">tFNRoamingIndicator</a> <a href="#">tFNDataCapabilities</a> and <a href="#">tFNServingSystem</a> <ul style="list-style-type: none"> <li>– 0x00 - for disable</li> <li>– 0x01 - for enable</li> <li>– 0xFF - No change - Specifying this parameter indicates that the device will continue to use the existing setting (disable/enable) which has been previously set for the device</li> </ul> </li> </ul>
<i>DDTMInd</i>	<ul style="list-style-type: none"> <li>• DDTM (Data Dedicated Transmission Mode) indication registration. <ul style="list-style-type: none"> <li>– 0x00 - for disable</li> <li>– 0x01 - for enable</li> <li>– 0xFF - No change - Specifying this parameter indicates that the device will continue to use the existing setting (disable/enable) which has been previously set for the device</li> </ul> </li> </ul>
<i>servingSystem-Ind</i>	<ul style="list-style-type: none"> <li>• Serving system indication registration. The following callbacks would not be invoked if the indication is disabled.  <a href="#">tFNBandPreference</a> <ul style="list-style-type: none"> <li>– 0x00 - for disable</li> <li>– 0x01 - for enable</li> <li>– 0xFF - No change - Specifying this parameter indicates that the device will continue to use the existing setting (disable/enable) which has been previously set for the device</li> </ul> </li> </ul>

## Returns

eQCWWAN\_ERR\_sNONE on success, eQCWWAN\_xxx error value otherwise.

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values.

## Note

Timeout: 2 seconds

#### 9.13.5.34 ULONG SLQSNasIndicationRegisterExt ( nasIndicationRegisterReq \* pIndicationRegisterReq )

This API Registers/De-registers for different NAS (Network access service) indications.

## Parameters

<i>pIndication-RegisterReq</i> [IN]	<ul style="list-style-type: none"> <li>• See <a href="#">nasIndicationRegisterReq</a> for more information</li> </ul>
-------------------------------------	---

## Returns

eQCWWAN\_ERR\_SNONE on success, eQCWWAN\_xxx error value otherwise.

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values.

## Note

Technology Supported: UMTS/CDMA

Device Supported: MC83x5, MC7700/50

Timeout: 2 seconds

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

#### 9.13.5.35 ULONG SLQSNasIndicationRegisterLTECphyCa ( BYTE \* *bStatus* )

This API Registers/De-registers for NAS CPHY Carrier Info.

## Parameters

<i>bStatus</i> [IN]	<ul style="list-style-type: none"> <li>• Values <ul style="list-style-type: none"> <li>– 0 - De-register.</li> <li>– 1 - Register.</li> </ul> </li> </ul>
---------------------	---

## Returns

eQCWWAN\_ERR\_SNONE on success, eQCWWAN\_xxx error value otherwise.

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values.

#### 9.13.5.36 ULONG SLQSNasSwiIndicationRegister ( NasSwiIndReg \* *pIndRegReq* )

sets the registration state for different QMI\_NAS SWI indications

## Parameters

---



<i>pIndRegReq[IN]</i>	<ul style="list-style-type: none"> <li>• See <a href="#">NasSwiIndReg</a> for more information</li> </ul>
-----------------------	---

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Timeout: 2 seconds

### 9.13.5.37 **ULONG** SLQSNasSwiModemStatus ( **swiModemStatusResp** \* *pModemStatusResp* )

This API requests the device to return the current status of modem.

**Parameters**

<i>pModemStatusResp[OUT]</i>	<ul style="list-style-type: none"> <li>• See <a href="#">swiModemStatusResp</a> for more information</li> </ul>
------------------------------	---

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Timeout: 2 seconds

### 9.13.5.38 **ULONG** SLQSPerformNetworkScan ( **slqsNetworkScanInfo** \* *pNetworkInfo* )

Performs scan for available networks and scans for RAT info as well.

**Parameters**

<i>pNetworkInfo[I-N/OUT]</i>	<ul style="list-style-type: none"> <li>• See <a href="#">slqsNetworkScanInfo</a> for more information</li> <li>• Valid pointers to the following structure members are mandatory <ul style="list-style-type: none"> <li>– pNetworkInfoInstances</li> <li>– pNetworkInfo</li> </ul> </li> </ul>
------------------------------	--

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Technology Supported: UMTS  
Timeout: 5 minutes

**9.13.5.39 ULONG SLQSSetBandPreference ( ULONGLONG *bandpreference* )**

Provides information about the band preference.

## Parameters

<i>bandpreference</i>	<ul style="list-style-type: none"> <li>• Bit mask representing the band preference to be set.</li> <li>• Bit position meanings: <ul style="list-style-type: none"> <li>– 0 - BC0_A - Band Class 0, A-System</li> <li>– 1 - BC0_B - Band Class 0, B-System, Band Class 0 AB , GSM 850 Band</li> <li>– 2 - BC1 - Band Class 1, all blocks</li> <li>– 3 - BC2 - Band Class 2 place holder</li> <li>– 4 - BC3 - Band Class 3, A-System</li> <li>– 5 - BC4 - Band Class 4, all blocks</li> <li>– 6 - BC5 - Band Class 5, all blocks</li> <li>– 7 - GSM_DCS_1800 - GSM DCS band</li> <li>– 8 - GSM_EGSM_900 - GSM Extended GSM (E-GSM) band</li> <li>– 9 - GSM_PGSM_900 - GSM Primary GSM (P-GSM) band</li> <li>– 10 - BC6 - Band Class 6</li> <li>– 11 - BC7 - Band Class 7</li> <li>– 12 - BC8 - Band Class 8</li> <li>– 13 - BC9 - Band Class 9</li> <li>– 14 - BC10 - Band Class 10</li> <li>– 15 - BC11 - Band Class 11</li> <li>– 16 - GSM_450 - GSM 450 band</li> <li>– 17 - GSM_480 - GSM 480 band</li> <li>– 18 - GSM_750 - GSM 750 band</li> <li>– 19 - GSM_850 - GSM 850 band</li> <li>– 20 - GSM_RGSM_900 - GSM Railways GSM Band</li> <li>– 21 - GSM_PCS_1900 - GSM PCS band</li> <li>– 22 - WCDMA_I_IMT_2000 - WCDMA EUROPE JAPAN and CHINA IMT 2100 band</li> <li>– 23 - WCDMA_II_PCS_1900 - WCDMA US PCS 1900 band</li> <li>– 24 - WCDMA_III_1700 - WCDMA EUROPE and CHINA DCS 1800 band</li> <li>– 25 - WCDMA_IV_1700 - WCDMA US 1700 band</li> <li>– 26 - WCDMA_V_850 - WCDMA US 850 band</li> <li>– 27 - WCDMA_VI_800 - WCDMA JAPAN 800 band</li> <li>– 28 - BC12 - Band Class 12</li> <li>– 29 - BC14 - Band Class 14</li> <li>– 30 - RESERVED_2 - Reserved 2</li> <li>– 31 - BC15 - Band Class 15</li> <li>– 32 - 47 - Reserved</li> <li>– 48 - WCDMA_VII_2600 - WCDMA EUROPE 2600 band</li> <li>– 49 - WCDMA_VIII_900 - WCDMA EUROPE and JAPAN 900 band</li> <li>– 50 - WCDMA_IX_1700 - WCDMA JAPAN 1700 band</li> <li>– 51 to 55 - Reserved</li> <li>– 56 - BBC16 - Band Class 16</li> <li>– 57 - BC17 - Band Class 17</li> <li>– 58 - BC18 - Band Class 18</li> </ul> </li> </ul>
Generated on Wed Oct 28 2015 00:40:12 for LinuxQMI SDK by Doxygen	<ul style="list-style-type: none"> <li>– 59 - BC19 - Band Class 19</li> <li>– 60 to 64 - Reserved</li> </ul>

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise.

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values.

**Note**

Timeout: 2 seconds

#### 9.13.5.40 **ULONG** SLQSSetSysSelectionPref ( **sysSelectPrefParams** \* *pSysSelectPrefParams* )

Sets the different system selection preferences of the device.

**Parameters**

<i>pSysSelectPrefParams</i> [IN]	<ul style="list-style-type: none"> <li>• See <a href="#">sysSelectPrefParams</a> for more information</li> </ul>
----------------------------------	--

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Timeout: 2 seconds

#### 9.13.5.41 **ULONG** SLQSSwiGetHDRPersonality ( **HDRPersonalityResp** \* *pHDRPersonalityResp* )

This API retrieves HDR Personality related information

**Parameters**

<i>pHDRPersonalityResp</i> [OUT]	<ul style="list-style-type: none"> <li>• See <a href="#">HDRPersonalityResp</a> for more information</li> </ul>
----------------------------------	---

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Technology Supported: CDMA  
Timeout: 5 seconds

#### 9.13.5.42 **ULONG** SLQSSwiGetHDRProtSubtype ( **HDRProtSubtypResp** \* *pHDRProtSubtypResp* )

This API retrieves HDR Prototype Subtype related information

## Parameters

<i>pHDRProt-SubtypResp[OUT]</i>	<ul style="list-style-type: none"> <li>See <a href="#">HDRProtSubtypResp</a> for more information</li> </ul>
---------------------------------	--

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Technology Supported: CDMA  
Timeout: 5 seconds

#### 9.13.5.43 ULONG SLQSSwiGetHRPDStats ( GetHRPDStatsResp \* pGetHRPDStatsResp )

This API retrieves currently acquired HRPD system statistics

## Parameters

<i>pGetHRPD-StatsResp[OUT]</i>	<ul style="list-style-type: none"> <li>See <a href="#">GetHRPDStatsResp</a> for more information</li> </ul>
--------------------------------	---

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 5 seconds

#### 9.13.5.44 ULONG SLQSSwiGetLteCQI ( LteCQIParm \* pLteCQIResp )

This API Fetch CQI parameters for LTE data session

## Parameters

<i>pLteCQIParm[OUT]</i>	<ul style="list-style-type: none"> <li>See <a href="#">LteCQIParm</a> for more information</li> </ul>
-------------------------	---

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Timeout: 5 seconds

**9.13.5.45 ULONG SLQSSwiNetworkDebug ( NetworkDebugResp \* pNetworkDebugResp )**

This API retrieves device and network status details

**Parameters**

<i>pNetworkDebugResp</i> [OUT]	<ul style="list-style-type: none"> <li>• See <a href="#">NetworkDebugResp</a> for more information</li> </ul>
--------------------------------	---

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Timeout: 5 seconds

**9.13.5.46 ULONG SLQSSwiPSDetach ( PSDetachReq \* pPSDetachReq )**

This API detaches PS connection.

**Parameters**

<i>pPSDetachReq</i> [IN]	<ul style="list-style-type: none"> <li>• See <a href="#">PSDetachReq</a> for more information</li> </ul>
--------------------------	--

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Technology Supported: CDMA  
Timeout: 5 seconds

**9.14 qaGobiApiOadm.h File Reference**

Open Mobile Alliance Device Management Service API function prototypes.

## Functions

- [ULONG OMADMStartSession](#) ([ULONG](#) sessionType)
- [ULONG OMADMCancelSession](#) ( )
- [ULONG OMADMGetSessionInfo](#) ([ULONG](#) \*pSessionState, [ULONG](#) \*pSessionType, [ULONG](#) \*pFailureReason, [BYTE](#) \*pRetryCount, [WORD](#) \*pSessionPause, [WORD](#) \*pTimeRemaining)
- [ULONG OMADMGetPendingNIA](#) ([ULONG](#) \*pSessionType, [USHORT](#) \*pSessionID)

### 9.14.1 Detailed Description

Open Mobile Alliance Device Management Service API function prototypes.

### 9.14.2 Function Documentation

#### 9.14.2.1 [ULONG OMADMCancelSession](#) ( )

Cancels an ongoing OMA-DM session.

##### Parameters

<i>None</i>	
-------------	--

##### Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

##### See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

##### Note

Technology Supported: CDMA  
Timeout: 2 seconds

#### 9.14.2.2 [ULONG OMADMGetPendingNIA](#) ( [ULONG](#) \* *pSessionType*, [USHORT](#) \* *pSessionID* )

Returns information about the pending network-initiated alert

##### Parameters

<i>SessionType[O-UT]</i>	<ul style="list-style-type: none"> <li>• Session Type <ul style="list-style-type: none"> <li>– 0x04 - Network-initiated PRL update</li> <li>– 0x05 - Network-initiated device configure</li> </ul> </li> </ul>
--------------------------	--

<i>SessionID[OUT]</i>	<ul style="list-style-type: none"> <li>• Session Id <ul style="list-style-type: none"> <li>– Unique session ID for NIA request</li> </ul> </li> </ul>
-----------------------	---

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Technology Supported: CDMA  
Timeout: 2 seconds

**9.14.2.3    `ULONG OMADMGetSessionInfo ( ULONG * pSessionState, ULONG * pSessionType, ULONG * pFailureReason, BYTE * pRetryCount, WORD * pSessionPause, WORD * pTimeRemaining )`**

Returns information related to the current (or previous if no session is active) OMA-DM session.

**Parameters**

<i>SessionState[OUT]</i>	<ul style="list-style-type: none"> <li>• Session state <ul style="list-style-type: none"> <li>– 0x00 - Complete, information was updated</li> <li>– 0x01 - Complete, update information is unavailable</li> <li>– 0x02 - Failed</li> <li>– 0x03 - Retrying</li> <li>– 0x04 - Connecting</li> <li>– 0x05 - Connected</li> <li>– 0x06 - Authenticated</li> <li>– 0x07 - Mobile Directory Number (MDN) downloaded</li> <li>– 0x08 - Mobile Station Identifier (MSID) downloaded</li> <li>– 0x09 - PRL downloaded</li> <li>– 0x0A - Mobile IP (MIP) profile downloaded</li> </ul> </li> </ul>
--------------------------	---



<i>sessionType</i> [O-UT]	<ul style="list-style-type: none"> <li>• Session State <ul style="list-style-type: none"> <li>– 0x00 - Client-initiated device configure</li> <li>– 0x01 - Client-initiated PRL update</li> <li>– 0x02 - Client-initiated hands-free activation</li> <li>– 0x03 - Device-initiated hands-free activation</li> <li>– 0x04 - Network-initiated PRL update</li> <li>– 0x05 - Network-initiated device configure</li> </ul> </li> </ul>
<i>FailureReason</i> [-OUT]	<ul style="list-style-type: none"> <li>• Session failure reason (when state indicates failure) <ul style="list-style-type: none"> <li>– 0x00 - Unknown</li> <li>– 0x01 - Network is unavailable</li> <li>– 0x02 - Server is unavailable</li> <li>– 0x03 - Authentication failed</li> <li>– 0x04 - Maximum retry exceeded</li> <li>– 0x05 - Session is cancelled</li> </ul> </li> </ul>
<i>RetryCount</i> [OUT]	<ul style="list-style-type: none"> <li>• Session retry count (when state indicates retrying)</li> </ul>
<i>SessionPause</i> [-OUT]	<ul style="list-style-type: none"> <li>• Session pause timer (in seconds , when state indicates retrying)</li> </ul>
<i>Time-Remaining</i> [OUT]	<ul style="list-style-type: none"> <li>• Pause time remaining (in seconds , when state indicates retrying)</li> </ul>

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Technology Supported: CDMA  
Timeout: 2 seconds

**9.14.2.4 ULONG OMADMStartSession ( ULONG sessionType )**

Starts an OMA-DM session.

## Parameters

<i>sessionType</i>	<ul style="list-style-type: none"> <li>• Session type <ul style="list-style-type: none"> <li>– 0x00 - Client-initiated device configure</li> <li>– 0x01 - Client-initiated PRL update</li> <li>– 0x02 - Client-initiated hands-free activation</li> </ul> </li> </ul>
--------------------	---

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Technology Supported: CDMA  
Timeout: 2 seconds

## 9.15 qaGobiApiPds.h File Reference

Position Determination Service API function prototypes.

### Data Structures

- struct [PDSPositionData](#)
- struct [GPSSStateInfo](#)
- struct [PDSPosMethodStateReq](#)

### Macros

- #define [DEFAULTBYTEVALUE](#) 0xFF
- #define [DEFAULTWORDVALUE](#) 0xFFFF
- #define [DEFAULTLONGVALUE](#) 0xFFFFFFFF

### Enumerations

- enum {  
[eSetServiceAutomaticTrackingDisable](#) =0,  
[eSetServiceAutomaticTrackingEnable](#) =1 }

### Functions

- [ULONG](#) [GetPDSSState](#) ([ULONG](#) \*pEnabledStatus, [ULONG](#) \*pTrackingStatus)
- [ULONG](#) [SetPDSSState](#) ([ULONG](#) enable)
- [ULONG](#) [StartPDSTrackingSessionExt](#) ([BYTE](#) sessionControl, [BYTE](#) sessionType, [BYTE](#) sessionOperation, [BYTE](#) sessionServerOption, [BYTE](#) fixTimeout, [ULONG](#) fixInterval, [ULONG](#) fixCount, [ULONG](#) fixAccuracy)

- [ULONG StopPDSTrackingSession](#) ()
- [ULONG PDSInjectTimeReference](#) ([ULONGLONG](#) systemTime, [USHORT](#) systemDiscontinuities)
- [ULONG GetPDSDefaults](#) ([ULONG](#) \*pOperation, [BYTE](#) \*pTimeout, [ULONG](#) \*pInterval, [ULONG](#) \*pAccuracy)
- [ULONG SetPDSDefaults](#) ([ULONG](#) operation, [BYTE](#) timeout, [ULONG](#) interval, [ULONG](#) accuracy)
- [ULONG GetXTRAAutomaticDownload](#) ([ULONG](#) \*pbEnabled, [USHORT](#) \*pInterval)
- [ULONG SetXTRAAutomaticDownload](#) ([ULONG](#) bEnabled, [USHORT](#) interval)
- [ULONG GetXTRANetwork](#) ([ULONG](#) \*pPreference)
- [ULONG SetXTRANetwork](#) ([ULONG](#) preference)
- [ULONG GetXTRAValidity](#) ([USHORT](#) \*pGPSWeek, [USHORT](#) \*pGPSWeekOffset, [USHORT](#) \*pDuration)
- [ULONG ForceXTRADownload](#) ()
- [ULONG GetServiceAutomaticTracking](#) ([ULONG](#) \*pbAuto)
- [ULONG SetServiceAutomaticTracking](#) ([ULONG](#) bAuto)
- [ULONG GetPortAutomaticTracking](#) ([ULONG](#) \*pbAuto)
- [ULONG SetPortAutomaticTracking](#) ([ULONG](#) bAuto)
- [ULONG ResetPDSData](#) ([ULONG](#) \*pGPSDataMask, [ULONG](#) \*pCellDataMask)
- [ULONG SLQSSetAGPSConfig](#) ([ULONG](#) \*pServerAddress, [ULONG](#) \*pServerPort, [BYTE](#) \*pServerURL, [BYTE](#) \*pServerURLLength, [BYTE](#) \*pNetworkMode)
- [ULONG SLQSPDSInjectAbsoluteTimeReference](#) ([ULONGLONG](#) timeMsec, [ULONG](#) timeUncMsec, [BYTE](#) timeBase, [BYTE](#) forceFlag)
- [ULONG SLQSGetAGPSConfig](#) ([ULONG](#) \*pServerAddress, [ULONG](#) \*pServerPort, [BYTE](#) \*pServerURL, [BYTE](#) \*pServerURLLength, [BYTE](#) \*pNetworkMode)
- [ULONG SLQSPDSInjectPositionData](#) (struct [PDSPositionData](#) \*pPositionData)
- [ULONG SLQSPDSDeterminePosition](#) ()
- [ULONG SLQSGetGPSSStateInfo](#) ([GPSSStateInfo](#) \*pGPSSStateInfo)
- [ULONG SLQSSetPositionMethodState](#) ([PDSPosMethodStateReq](#) \*pPDSPosMethodStateReq)

### 9.15.1 Detailed Description

Position Determination Service API function prototypes.

### 9.15.2 Macro Definition Documentation

9.15.2.1 `#define DEFAULTBYTEVALUE 0xFF`

9.15.2.2 `#define DEFAULTLONGVALUE 0xFFFFFFFF`

9.15.2.3 `#define DEFAULTWORDVALUE 0xFFFF`

### 9.15.3 Enumeration Type Documentation

9.15.3.1 anonymous enum

Enumerator

***eSetServiceAutomaticTrackingDisable***

***eSetServiceAutomaticTrackingEnable***

### 9.15.4 Function Documentation

9.15.4.1 [ULONG ForceXTRADownload](#) ( )

Forces the XTRA database to be downloaded to the device.

## Parameters

<i>none</i>	
-------------	--

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 2 Seconds

#### 9.15.4.2 **ULONG** GetPDSDefaults ( **ULONG** \* *pOperation*, **BYTE** \* *pTimeout*, **ULONG** \* *pInterval*, **ULONG** \* *pAccuracy* )

Returns the default tracking session configuration. The tracking session configuration is used when a tracking session is automatically started using SetServiceAutomaticTracking or due to the device detecting an application opening the NMEA port.

## Parameters

<i>pOperation</i> [OUT]	<ul style="list-style-type: none"> <li>Current session operating mode             <ul style="list-style-type: none"> <li>0 - Standalone</li> <li>1 - MS based</li> <li>2 - MS assisted</li> </ul> </li> </ul>
<i>pTimeout</i> [OUT]	<ul style="list-style-type: none"> <li>Maximum amount of time (seconds) to work on each fix, maximum is 255</li> </ul>
<i>pInterval</i> [OUT]	<ul style="list-style-type: none"> <li>Interval (seconds) between fix requests</li> </ul>
<i>pAccuracy</i> [OUT]	<ul style="list-style-type: none"> <li>Preferred accuracy threshold (meters)</li> </ul>

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 2 Seconds

#### 9.15.4.3 **ULONG** GetPDSState ( **ULONG** \* *pEnabledStatus*, **ULONG** \* *pTrackingStatus* )

Returns the current PDS state.

## Parameters

<i>pEnabled-Status[OUT]</i>	<ul style="list-style-type: none"><li>• Current PDS state<ul style="list-style-type: none"><li>– 0 - disable</li><li>– 1 - enable</li></ul></li></ul>
<i>pTracking-Status[OUT]</i>	<ul style="list-style-type: none"><li>• Current PDS tracking session state</li><li>• Values:<ul style="list-style-type: none"><li>– 0x00 - Unknown</li><li>– 0x01 - Inactive</li><li>– 0x02 - Active</li></ul></li></ul>

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 2 seconds

#### 9.15.4.4 ULONG GetPortAutomaticTracking ( ULONG \* pbAuto )

Returns the automatic tracking configuration for the NMEA COM port.

## Parameters

<i>pbAuto[OUT]</i>	<ul style="list-style-type: none"><li>• Automatic tracking enabled for NMEA COM port<ul style="list-style-type: none"><li>– 0x00 - Disabled</li><li>– 0x01 - Enabled</li></ul></li></ul>
--------------------	--

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 2 Seconds

#### 9.15.4.5 **ULONG** GetServiceAutomaticTracking ( **ULONG** \* *pbAuto* )

Returns the automatic tracking state for the service.

## Parameters

<i>pbAuto</i> [OUT]	<ul style="list-style-type: none"> <li>Automatic tracking session started for service <ul style="list-style-type: none"> <li>0x00 - Disabled</li> <li>0x01 - Enabled</li> </ul> </li> </ul>
---------------------	---

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 2 Seconds

#### 9.15.4.6 ULONG GetXTRAAutomaticDownload ( ULONG \* *pbEnabled*, USHORT \* *pInterval* )

Returns the XTRA automatic database download configuration.

## Parameters

<i>pbEnabled</i> [OUT]	<ul style="list-style-type: none"> <li>Automatic XTRA download status <ul style="list-style-type: none"> <li>0 - Disabled</li> <li>1 - Enabled</li> </ul> </li> </ul>
<i>pInterval</i> [OUT]	<ul style="list-style-type: none"> <li>Interval (hours) between XTRA downloads</li> </ul>

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 2 Seconds

#### 9.15.4.7 ULONG GetXTRANetwork ( ULONG \* *pPreference* )

Returns the XTRA WWAN network preference. When automatic XTRA database downloading is enabled this preference determines which WWAN networks will be used to perform the XTRA database download.

## Parameters

<i>pPreference</i> [OUT]	<ul style="list-style-type: none"> <li>• XTRA WWAN network preference <ul style="list-style-type: none"> <li>– 0x00 - None (any available network)</li> <li>– 0x01 - Home-only, only when on home systems</li> <li>– 0x02 - Roam-only, only when on non-home systems</li> </ul> </li> </ul>
--------------------------	---

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 2 Seconds

#### 9.15.4.8 ULONG GetXTRAVality ( USHORT \* pGPSWeek, USHORT \* pGPSWeekOffset, USHORT \* pDuration )

Returns the XTRA database validity period. When automatic XTRA database downloading is enabled the validity period determines when the XTRA database will be updated through a new download.

## Parameters

<i>pGPSWeek</i> [OUT]	<ul style="list-style-type: none"> <li>• Starting GPS week of validity period</li> </ul>
<i>pGPSWeekOffset</i> [OUT]	<ul style="list-style-type: none"> <li>• Starting GPS week offset (minutes) of validity period</li> </ul>
<i>pDuration</i> [OUT]	<ul style="list-style-type: none"> <li>• Length of validity period (hours)</li> </ul>

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 2 Seconds

#### 9.15.4.9 ULONG PDSInjectTimeReference ( ULONGLONG systemTime, USHORT systemDiscontinuities )

Injects a system time into the PDS engine.



## Parameters

<i>systemTime</i>	<ul style="list-style-type: none"> <li>• System time( milliseconds )</li> </ul>
<i>system-Discontinuities</i>	<ul style="list-style-type: none"> <li>• Number of system time discontinuities</li> </ul>

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 2 Seconds

#### 9.15.4.10 ULONG ResetPDSData ( ULONG \* pGPSDataMask, ULONG \* pCellDataMask )

Resets the specified PDS data.

## Parameters

<i>pGPSDataMask[IN]</i>	<ul style="list-style-type: none"> <li>• Bitmask of GPS data to clear (optional) <ul style="list-style-type: none"> <li>– 0x00000001 - EPH</li> <li>– 0x00000002 - ALM</li> <li>– 0x00000004 - POS</li> <li>– 0x00000008 - TIME</li> <li>– 0x00000010 - IONO</li> <li>– 0x00000020 - UTC</li> <li>– 0x00000040 - HEALTH</li> <li>– 0x00000080 - SVDIR</li> <li>– 0x00000100 - SVSTEER</li> <li>– 0x00000200 - SADATA</li> <li>– 0x00000400 - RTI</li> <li>– 0x00000800 - ALM_CORR</li> <li>– 0x00001000 - FREQ_BIAS_EST</li> </ul> </li> </ul>
-------------------------	--

<i>pCellDataMask</i> [-IN]	<ul style="list-style-type: none"> <li>• Bitmask of cell data to clear (optional) <ul style="list-style-type: none"> <li>– 0x00000001 - POS</li> <li>– 0x00000002 - LATEST_GPS_POS</li> <li>– 0x00000004 - OTA_POS</li> <li>– 0x00000008 - EXT_REF_POS</li> <li>– 0x00000010 - TIMETAG</li> <li>– 0x00000020 - CELLID</li> <li>– 0x00000040 - CACHED_CELLID</li> <li>– 0x00000080 - LAST_SRV_CELL</li> <li>– 0x00000100 - CUR_SRV_CELL</li> <li>– 0x00000200 - NEIGHBOR_INFO</li> </ul> </li> </ul>
----------------------------	---

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Timeout: 2 Seconds

#### 9.15.4.11 **ULONG** SetPDSDefaults ( **ULONG** operation, **BYTE** timeout, **ULONG** interval, **ULONG** accuracy )

Sets the default tracking session configuration. The tracking session configuration is used when a tracking session is automatically started using SetServiceAutomaticTracking or due to the device detecting an application opening the NMEA port.

**Parameters**

<i>operation</i>	<ul style="list-style-type: none"> <li>• Current session operating mode <ul style="list-style-type: none"> <li>– 0 - Standalone</li> <li>– 1 - MS based</li> <li>– 2 - MS assisted</li> </ul> </li> </ul>
------------------	---

<i>timeout</i>	<ul style="list-style-type: none"><li>• Maximum amount of time (seconds) to work on each fix, maximum is 255</li></ul>
<i>interval</i>	<ul style="list-style-type: none"><li>• Interval (seconds) between fix requests</li></ul>
<i>accuracy</i>	<ul style="list-style-type: none"><li>• Preferred accuracy threshold (meters)</li></ul>

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Timeout: 2 Seconds

**9.15.4.12 ULONG SetPDSSState ( ULONG enable )**

Sets the PDS state.

**Parameters**

<i>enable</i>	<ul style="list-style-type: none"><li>• Desired PDS state<ul style="list-style-type: none"><li>– Zero - disable</li><li>– Non-Zero - enable</li></ul></li></ul>
---------------	---

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Timeout: 5 Seconds

**9.15.4.13 ULONG SetPortAutomaticTracking ( ULONG bAuto )**

Sets the automatic tracking configuration for the NMEA COM port.

## Parameters

<i>bAuto</i>	<ul style="list-style-type: none"> <li>• Enable automatic tracking for NMEA COM port <ul style="list-style-type: none"> <li>– 0x00 - Disabled</li> <li>– 0x01 - Enabled</li> </ul> </li> </ul>
--------------	--

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 2 Seconds

#### 9.15.4.14 ULONG SetServiceAutomaticTracking ( ULONG *bAuto* )

Sets the automatic tracking state for the service. Tracking session being started using the default session configuration. Auto-tracking continues to generate fixes indefinitely until requested to be disabled. In StartPDSTracking-SessionExt a tracking session get started using the specified session control method and input parameters. After completion of requested no. of position fixes or service times out to perform fix, tracking session ends and GPS service deactivates.

## Parameters

<i>bAuto</i>	<ul style="list-style-type: none"> <li>• Automatic tracking session started for service <ul style="list-style-type: none"> <li>– 0x00 - Disabled</li> <li>– 0x01 - Enabled</li> </ul> </li> </ul>
--------------	---

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 2 Seconds

#### 9.15.4.15 ULONG SetXTRAAutomaticDownload ( ULONG *bEnabled*, USHORT *interval* )

Sets the XTRA automatic database download configuration.

## Parameters

<i>bEnabled</i>	<ul style="list-style-type: none"> <li>Automatic XTRA download status <ul style="list-style-type: none"> <li>0 - Disabled</li> <li>1 - Enabled</li> </ul> </li> </ul>
<i>interval</i>	<ul style="list-style-type: none"> <li>Interval (hours) between XTRA downloads</li> </ul>

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 2 Seconds

9.15.4.16 ULONG SetXTRANetwork ( ULONG *preference* )

Sets the XTRA WWAN network preference. When automatic XTRA database downloading is enabled this preference determines which WWAN networks will be used to perform the XTRA database download.

## Parameters

<i>preference</i>	<ul style="list-style-type: none"> <li>XTRA WWAN network preference <ul style="list-style-type: none"> <li>0x00 - None (any available network)</li> <li>0x01 - Home-only, only when on home systems</li> <li>0x02 - Roam-only, only when on non-home systems</li> </ul> </li> </ul>
-------------------	---

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 2 Seconds

9.15.4.17 ULONG SLQSGetAGPSConfig ( ULONG \* *pServerAddress*, ULONG \* *pServerPort*, BYTE \* *pServerURL*, BYTE \* *pServerURLLength*, BYTE \* *pNetworkMode* )

Gets the PDS AGPS (MS-based) configuration.

## Parameters

<i>pServer-Address[OUT]</i>	<ul style="list-style-type: none"> <li>IPv4 address of AGPS server. "0" if not set</li> </ul>
<i>pServerPort[OUT]</i>	<ul style="list-style-type: none"> <li>Port number of AGPS server. "0" if not set</li> </ul>
<i>pServerURL[OUT]</i>	<ul style="list-style-type: none"> <li>URL of the AGPS server. "0" if not set</li> </ul>
<i>pServerURL-Length[OUT]</i>	<ul style="list-style-type: none"> <li>URL length of AGPS server. "0" if not set</li> </ul>
<i>pNetworkMode[IN]</i>	<ul style="list-style-type: none"> <li>Network Mode of AGPS Server [optional - should be present in Multimode Systems] <ul style="list-style-type: none"> <li>0x00 - UMTS</li> <li>0x01 - CDMA</li> </ul> </li> </ul>

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 2 Seconds

#### 9.15.4.18 ULONG SLQSGetGPSStateInfo ( GPSStateInfo \* pGPSStateInfo )

Queries the MSM GPS server for receiver state information

## Parameters

<i>pGPSStateInfo</i>	<ul style="list-style-type: none"> <li>contains the GPS State Info</li> <li>See <a href="#">GPSStateInfo</a> for more information</li> </ul>
----------------------	--

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 5 Seconds

**9.15.4.19    ULONG SLQSPDSDeterminePosition (    )**

Requests the MSM GPS service to obtain the current position for manually controlled tracking sessions.

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Timeout: 5 Seconds

**9.15.4.20    ULONG SLQSPDSInjectAbsoluteTimeReference (    ULONGLONG *timeMsec*,    ULONG *timeUncMsec*,    BYTE *timeBase*,    BYTE *forceFlag* )**

Injects a absolute time reference into the PDS engine.

**Parameters**

<i>timeMsec</i>	<ul style="list-style-type: none"> <li>Represents the number of milliseconds elapsed since either a GPS or UTC time base. If the time base is UTC, this value should NOT include leap seconds</li> </ul>
<i>timeUncMsec</i>	<ul style="list-style-type: none"> <li>Time uncertainty in milliseconds</li> </ul>
<i>timeBase</i>	<ul style="list-style-type: none"> <li>Time base               <ul style="list-style-type: none"> <li>– 0x00 - GPS (midnight, Jan 6, 1980)</li> <li>– 0x01 - UTC (midnight, Jan 1, 1970)</li> </ul> </li> </ul>
<i>forceFlag</i>	<ul style="list-style-type: none"> <li>Force acceptance of data</li> </ul>

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Timeout: 2 Seconds

**9.15.4.21    ULONG SLQSPDSInjectPositionData (    struct PDSPositionData \* *pPositionData* )**

Injects position data into the PDS engine.

## Parameters

<i>pPositionData</i>	<ul style="list-style-type: none"> <li>contains the position data to be injected to the PDS engine</li> </ul>
----------------------	---

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 2 Seconds

#### 9.15.4.22 ULONG SLQSSetAGPSConfig ( ULONG \* pServerAddress, ULONG \* pServerPort, BYTE \* pServerURL, BYTE \* pServerURLLength, BYTE \* pNetworkMode )

Sets the PDS AGPS (MS-based) configuration.

## Parameters

<i>pServerAddress</i> [IN]	<ul style="list-style-type: none"> <li>IPv4 address of AGPS server [optional]</li> </ul>
<i>pServerPort</i> [IN]	<ul style="list-style-type: none"> <li><a href="#">Port</a> number of AGPS server [optional - should be present when pServerAddress is present]</li> </ul>
<i>pServerURL</i> [IN]	<ul style="list-style-type: none"> <li>URL of the AGPS server [optional]</li> </ul>
<i>pServerURLLength</i> [IN]	<ul style="list-style-type: none"> <li>URL length of AGPS server [optional - should be present when pServerURL is present]</li> </ul>
<i>pNetworkMode</i> [IN]	<ul style="list-style-type: none"> <li>Network Mode of AGPS Server [optional - should be present in Multimode Systems] <ul style="list-style-type: none"> <li>0x00 - UMTS</li> <li>0x01 - CDMA</li> </ul> </li> </ul>

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 2 Seconds



9.15.4.23 **ULONG** SLQSSetPositionMethodState ( **PDSPosMethodStateReq** \* *pPDSPosMethodStateReq* )

Sets the state of positioning methods for the device.

## Parameters

<i>pPDSPos-MethodStateReq</i>	<ul style="list-style-type: none"> <li>• See <a href="#">PDSPosMethodStateReq</a> for more information</li> </ul>
-------------------------------	---

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 5 Seconds

**9.15.4.24** **ULONG** StartPDSTrackingSessionExt ( **BYTE** *sessionControl*, **BYTE** *sessionType*, **BYTE** *sessionOperation*, **BYTE** *sessionServerOption*, **BYTE** *fixTimeout*, **ULONG** *fixInterval*, **ULONG** *fixCount*, **ULONG** *fixAccuracy* )

This function starts a PDS tracking session.

## Parameters

<i>sessionControl</i>	<ul style="list-style-type: none"> <li>• Control method: <ul style="list-style-type: none"> <li>– 0x0 - Manual</li> </ul> </li> </ul>
<i>sessionType</i>	<ul style="list-style-type: none"> <li>• Type: <ul style="list-style-type: none"> <li>– 0x0 - New</li> </ul> </li> </ul>
<i>session-Operation</i>	<ul style="list-style-type: none"> <li>• Operating mode: <ul style="list-style-type: none"> <li>– 0x00 - Standalone</li> <li>– 0x01 - MS-based</li> </ul> </li> </ul>
<i>sessionServer-Option</i>	<ul style="list-style-type: none"> <li>• Location server option: <ul style="list-style-type: none"> <li>– 0x0 - Default</li> </ul> </li> </ul>

<i>fixTimeout</i>	<ul style="list-style-type: none"> <li>Maximum time to work on each fix (in seconds, max 255)</li> </ul>
<i>fixCount</i>	<ul style="list-style-type: none"> <li>Count of position fix requests for this session (must be at least 1)</li> </ul>
<i>fixTimeout</i>	<ul style="list-style-type: none"> <li>interval between position fix requests (in seconds)</li> </ul>
<i>fixAccuracy</i>	<ul style="list-style-type: none"> <li>Preferred accuracy threshold(in meters)</li> </ul>

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Timeout: 2 Seconds

**9.15.4.25 ULONG StopPDSTrackingSession ( )**

This function stops a PDS tracking session.

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Timeout: 2 Seconds

**9.16 qaGobiApiQos.h File Reference**

Quality of Service API function prototypes.

**Data Structures**

- struct [dataRate](#)
- struct [tokenBucket](#)
- struct [pktErrRate](#)
- struct [swiQosFlow](#)
- struct [IPv4Addr](#)

- struct [Tos](#)
- struct [IPv6Addr](#)
- struct [IPv6TrafCls](#)
- struct [Port](#)
- struct [swiQosFilter](#)
- struct [swiQosReq](#)
- struct [swiQosIds](#)
- struct [swiQosModifyReq](#)
- struct [swiQosGranted](#)
- struct [NWProfile](#)
- struct [sQosFlowStat](#)
- struct [sQosStat](#)
- struct [sApnExtraParams](#)

## Macros

- `#define MAX_QOS_SPEC_PER_APN (10)`
- `#define MAX_QOS_FILTER_TLV 25`

## Functions

- [ULONG SLQSQosReset](#) ([BYTE](#) instance)
- [ULONG SLQSQosReq](#) ([BYTE](#) instance, [swiQosReq](#) \*pQosReq, [swiQosIds](#) \*pQosResp)
- [ULONG SLQSQosRel](#) ([BYTE](#) instance, [swiQosIds](#) \*pQosIds)
- [ULONG SLQSQosSuspend](#) ([BYTE](#) instance, [swiQosIds](#) \*pQosIds)
- [ULONG SLQSQosResume](#) ([BYTE](#) instance, [swiQosIds](#) \*pQosIds)
- [ULONG SLQSQosModify](#) ([BYTE](#) instance, [swiQosModifyReq](#) \*pReq)
- [ULONG SLQSQosGetGranted](#) ([BYTE](#) instance, [ULONG](#) id, [swiQosGranted](#) \*pGranted)
- [ULONG SLQSQosGetFlowStatus](#) ([BYTE](#) instance, [ULONG](#) id, [BYTE](#) \*pStatus)
- [ULONG SLQSQosGetNetworkStatus](#) ([BYTE](#) instance, [BYTE](#) \*pStatus)
- [ULONG SLQSQosGetNWProf](#) ([BYTE](#) instance, [BYTE](#) \*pSz, [NWProfile](#) \*pProfile)
- [ULONG SLQSQosSwiReadDataStats](#) ([BYTE](#) instance, [ULONG](#) apnId, [sQosStat](#) \*pQosStat)
- [ULONG SLQSQosSwiReadApnExtraParams](#) ([BYTE](#) instance, [ULONG](#) apnId, [sApnExtraParams](#) \*pApnExtraParams)

### 9.16.1 Detailed Description

Quality of Service API function prototypes.

### 9.16.2 Macro Definition Documentation

9.16.2.1 `#define MAX_QOS_FILTER_TLV 25`

9.16.2.2 `#define MAX_QOS_SPEC_PER_APN (10)`

### 9.16.3 Function Documentation

9.16.3.1 `ULONG SLQSQosGetFlowStatus ( BYTE instance, ULONG id, BYTE * pStatus )`

Get the status of a QoS flow.

## Parameters

in	<i>instance</i>	<ul style="list-style-type: none"> <li>• QMI instance</li> </ul>
in	<i>id</i>	Qos identifier Index identifying the QoS flow that has been negotiated
out	<i>pStatus</i>	Qos status Current QoS instance status: <ul style="list-style-type: none"> <li>• 0x01 – QMI_QOS_STATUS_ACTIVATED</li> <li>• 0x02 – QMI_QOS_STATUS_SUSPENDED</li> <li>• 0x03 – QMI_QOS_STATUS_GONE</li> </ul>

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

see [qmerrno.h](#) for eQCWWAN\_xxx error values

### 9.16.3.2 ULONG SLQSQosGetGranted ( BYTE *instance*, ULONG *id*, swiQosGranted \* *pGranted* )

Retrieve the QoS parameters that are in effect for the specified QoS 16 flow as a result of network negotiation

## Parameters

in	<i>instance</i>	<ul style="list-style-type: none"> <li>• QMI instance</li> </ul>
in	<i>id</i>	<ul style="list-style-type: none"> <li>• Qos identifier</li> <li>• Index identifying the QoS flow that has been negotiated</li> </ul>
in	<i>pGranted</i>	<ul style="list-style-type: none"> <li>• Tx/Rx Qos granted flow</li> <li>• See <a href="#">swiQosGranted</a> for more information</li> </ul>

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

see [qmerrno.h](#) for eQCWWAN\_xxx error values

### 9.16.3.3 ULONG SLQSQosGetNetworkStatus ( BYTE *instance*, BYTE \* *pStatus* )

Queries whether the device is currently on a network that supports QoS

## Parameters

	<i>instance[IN]</i>	<ul style="list-style-type: none"> <li>• QMI instance</li> </ul>
out	<i>pStatus</i>	Network QoS support status <ul style="list-style-type: none"> <li>• 0 – No QoS support in network</li> <li>• 1 – Network supports QoS</li> </ul>

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

see [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Technology Supported: CDMA

#### 9.16.3.4 ULONG SLQSQosGetNWProf ( BYTE *instance*, BYTE \* *pSz*, NWProfile \* *pProfile* )

Get network supported QoS profile information

## Parameters

in	<i>instance</i>	<ul style="list-style-type: none"> <li>• QMI instance</li> </ul>
	<i>in/out</i>	pSz Number of network supported QoS profiles for one technology
out	<i>pProfile</i>	Network supported QoS profiles

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

see [qmerrno.h](#) for eQCWWAN\_xxx error values

#### 9.16.3.5 ULONG SLQSQosModify ( BYTE *instance*, swiQosModifyReq \* *pReq* )

Resume one or more existing QoS flows

## Warning

NOT IMPLEMENTED

## Parameters

<i>in</i>	<i>instance</i>	<ul style="list-style-type: none"><li>• QMI instance</li></ul>
	<i>pReq</i>	<ul style="list-style-type: none"><li>• See <a href="#">swiQosModifyReq</a> for more information</li></ul>

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

see [qmerrno.h](#) for eQCWWAN\_xxx error values

**9.16.3.6 ULONG SLQSQosRel ( BYTE *instance*, *swiQosIds* \* *pQosIds* )**

Release one or more existing QoS flows

## Parameters

<i>in</i>	<i>instance</i>	<ul style="list-style-type: none"><li>• QMI instance</li></ul>
	<i>pQosIds</i>	<ul style="list-style-type: none"><li>• See <a href="#">swiQosIds</a> for more information</li></ul>

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

see [qmerrno.h](#) for eQCWWAN\_xxx error values

**9.16.3.7 ULONG SLQSQosReq ( BYTE *instance*, *swiQosReq* \* *pQosReq*, *swiQosIds* \* *pQosResp* )**

Triggers QoS negotiation by providing QoS parameters

## Parameters

<i>instance</i> [IN]	<ul style="list-style-type: none"><li>• QMI instance</li></ul>
<i>pQoSReq</i> [IN]	<ul style="list-style-type: none"><li>• See <a href="#">swiQosReq</a> for more information</li></ul>

<i>pQosResp[OUT]</i>	<ul style="list-style-type: none"> <li>• See <a href="#">swiQosIds</a> for more information</li> </ul>
----------------------	--

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

see [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Technology Supported: UMTS/CDMA

Device Supported: MC77XX

Timeout: 2 seconds

At least one pair of optional TLVs must be present; Tx QoS Flow Request and Tx QoS Filter Request TLVs must both be present if either one is present; Rx QoS Flow Request and Rx QoS Filter Request TLVs must both be present if either one is present

**9.16.3.8 ULONG SLQSQosReset ( BYTE *instance* )**

Reset the QoS service state variables of the requesting control point

**Parameters**

<i>in</i>	<i>instance</i>	<ul style="list-style-type: none"> <li>• QMI instance</li> </ul>
-----------	-----------------	--

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Technology Supported: CDMA, UMTS & LTE

Device Supported: MC7750

Timeout: 2 seconds

**9.16.3.9 ULONG SLQSQosResume ( BYTE *instance*, swiQosIds \* *pQosIds* )**

Resume one or more existing QoS flows

**Parameters**



<code>in</code>	<i>instance</i>	<ul style="list-style-type: none"> <li>• QMI instance</li> </ul>
	<i>pQosIds</i>	<ul style="list-style-type: none"> <li>• See <a href="#">swiQosIds</a> for more information</li> </ul>

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

see [qmerrno.h](#) for eQCWWAN\_xxx error values

#### 9.16.3.10 ULONG SLQSQoSSuspend ( BYTE *instance*, swiQosIds \* *pQosIds* )

Suspend one or more existing QoS flows

**Parameters**

<code>in</code>	<i>instance</i>	<ul style="list-style-type: none"> <li>• QMI instance</li> </ul>
	<i>pQosIds</i>	<ul style="list-style-type: none"> <li>• See <a href="#">swiQosIds</a> for more information</li> </ul>

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

see [qmerrno.h](#) for eQCWWAN\_xxx error values

#### 9.16.3.11 ULONG SLQSQoSSwiReadApnExtraParams ( BYTE *instance*, ULONG *apnId*, sApnExtraParams \* *pApnExtraParams* )

Queries extra APN parameters that are not reported by existing QCT QMI service

**Parameters**

<code>in</code>	<i>instance</i>	<ul style="list-style-type: none"> <li>• QMI instance</li> </ul>
<code>in</code>	<i>apnId</i>	<ul style="list-style-type: none"> <li>• APN id</li> </ul>

out	<i>pApnExtraParams</i>	See <a href="#">sApnExtraParams</a> for more information
-----	------------------------	--

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

### 9.16.3.12 ULONG SLQSQosSwiReadDataStats ( BYTE *instance*, ULONG *apnId*, sQosStat \* *pQosStat* )

Get the current number of packets and bytes sent, dropped and received for each UL, DL bearer and a sum of them for UL and DL direction in the modem.

**Parameters**

in	<i>instance</i>	<ul style="list-style-type: none"> <li>• QMI instance</li> </ul>
in	<i>apnId</i>	<ul style="list-style-type: none"> <li>• APN id</li> </ul>
out	<i>pQosStat</i>	See <a href="#">sQosStat</a> for more information

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## 9.17 qaGobiApiRms.h File Reference

Remote Management Service API function prototypes.

**Functions**

- [ULONG GetSMSWake](#) ( [ULONG](#) \*pEnabled, [ULONG](#) \*pWakeMask )
- [ULONG SetSMSWake](#) ( [ULONG](#) bEnable, [ULONG](#) wakeMask )

### 9.17.1 Detailed Description

Remote Management Service API function prototypes.

### 9.17.2 Function Documentation

#### 9.17.2.1 ULONG GetSMSWake ( ULONG \* *pEnabled*, ULONG \* *pWakeMask* )

Queries the state of the SMS wake functionality. When enabled SMS wake functionality results in incoming messages being searched for the configured mask. Upon detection of the mask the incoming message is deleted (i.e. not stored in memory) and the device attempts to wake the host (requires host platform support).

## Parameters

<i>pEnabled</i> [OUT]	<ul style="list-style-type: none"><li>• SMS wake functionality enabled<ul style="list-style-type: none"><li>– 0 - Disabled</li><li>– 1 - Enabled</li></ul></li></ul>
<i>pWakeMask</i> [OUT]	<ul style="list-style-type: none"><li>• SMS wake mask to search for incoming messages (only relevant when enabled)</li></ul>

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 2 seconds

**9.17.2.2 ULONG SetSMSWake ( ULONG *bEnable*, ULONG *wakeMask* )**

Configures the SMS wake functionality. When enabled SMS wake functionality results in incoming messages being searched for the configured mask. Upon detection of the mask the incoming message is deleted (i.e. not stored in memory) and the device attempts to wake the host (requires host platform support).

## Parameters

<i>bEnable</i>	<ul style="list-style-type: none"><li>• Enable SMS wake functionality<ul style="list-style-type: none"><li>– Zero - Disable</li><li>– Non-Zero - Enable</li></ul></li></ul>
<i>wakeMask</i>	<ul style="list-style-type: none"><li>• SMS wake mask to search for incoming messages (only relevant when enabling)</li></ul>

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 2 seconds

## 9.18 qaGobiApiSar.h File Reference

Specific Absorption Rate API function prototypes.

### Enumerations

- enum [eQMISARRFState](#) {  
[QMI\\_SAR\\_RF\\_STATE\\_DEFAULT](#) = 0,  
[QMI\\_SAR\\_RF\\_STATE\\_1](#),  
[QMI\\_SAR\\_RF\\_STATE\\_2](#),  
[QMI\\_SAR\\_RF\\_STATE\\_3](#),  
[QMI\\_SAR\\_RF\\_STATE\\_4](#),  
[QMI\\_SAR\\_RF\\_STATE\\_5](#),  
[QMI\\_SAR\\_RF\\_STATE\\_6](#),  
[QMI\\_SAR\\_RF\\_STATE\\_7](#),  
[QMI\\_SAR\\_RF\\_STATE\\_8](#) }

### Functions

- [ULONG SLQSGetRfSarState](#) ([ULONG](#) \*pSarRFState)
- [ULONG SLQSSetRfSarState](#) ([ULONG](#) RfSarState)

### 9.18.1 Detailed Description

Specific Absorption Rate API function prototypes.

### 9.18.2 Enumeration Type Documentation

#### 9.18.2.1 enum [eQMISARRFState](#)

This enum contains the SAR RF States

#### Parameters

SAR	RF State
	<ul style="list-style-type: none"> <li>• <a href="#">QMI_SAR_RF_STATE_DEFAULT</a> = 0</li> <li>• <a href="#">QMI_SAR_RF_STATE_1</a></li> <li>• <a href="#">QMI_SAR_RF_STATE_2</a></li> <li>• <a href="#">QMI_SAR_RF_STATE_3</a></li> <li>• <a href="#">QMI_SAR_RF_STATE_4</a></li> <li>• <a href="#">QMI_SAR_RF_STATE_5</a></li> <li>• <a href="#">QMI_SAR_RF_STATE_6</a></li> <li>• <a href="#">QMI_SAR_RF_STATE_7</a></li> <li>• <a href="#">QMI_SAR_RF_STATE_8</a></li> </ul>

#### Enumerator

**[QMI\\_SAR\\_RF\\_STATE\\_DEFAULT](#)**

***QMI\_SAR\_RF\_STATE\_1***  
***QMI\_SAR\_RF\_STATE\_2***  
***QMI\_SAR\_RF\_STATE\_3***  
***QMI\_SAR\_RF\_STATE\_4***  
***QMI\_SAR\_RF\_STATE\_5***  
***QMI\_SAR\_RF\_STATE\_6***  
***QMI\_SAR\_RF\_STATE\_7***  
***QMI\_SAR\_RF\_STATE\_8***

### 9.18.3 Function Documentation

#### 9.18.3.1 `ULONG SLQSGetRfSarState ( ULONG * pSarRFState )`

Gets the specified RF SAR state.

Parameters

<i>pSarRFState</i>	<ul style="list-style-type: none"> <li>• SAR RF State               <ul style="list-style-type: none"> <li>– QMI_SAR_RF_STATE_DEFAULT</li> <li>– QMI_SAR_RF_STATE_1</li> <li>– QMI_SAR_RF_STATE_2</li> <li>– QMI_SAR_RF_STATE_3</li> <li>– QMI_SAR_RF_STATE_4</li> <li>– QMI_SAR_RF_STATE_5</li> <li>– QMI_SAR_RF_STATE_6</li> <li>– QMI_SAR_RF_STATE_7</li> <li>– QMI_SAR_RF_STATE_8</li> </ul> </li> </ul>
--------------------	--

Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

Note

Timeout: 2 seconds

#### 9.18.3.2 `ULONG SLQSSetRfSarState ( ULONG RfSarState )`

Sets the specified RF SAR state.

## Parameters

<i>sar_rf_state</i>	<ul style="list-style-type: none"> <li>• SAR RF State <ul style="list-style-type: none"> <li>– QMI_SAR_RF_STATE_DEFAULT</li> <li>– QMI_SAR_RF_STATE_1</li> <li>– QMI_SAR_RF_STATE_2</li> <li>– QMI_SAR_RF_STATE_3</li> <li>– QMI_SAR_RF_STATE_4</li> <li>– QMI_SAR_RF_STATE_5</li> <li>– QMI_SAR_RF_STATE_6</li> <li>– QMI_SAR_RF_STATE_7</li> <li>– QMI_SAR_RF_STATE_8</li> </ul> </li> </ul>
---------------------	--

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 2 seconds

## 9.19 qaGobiApiSms.h File Reference

Short Message Service API function prototypes.

### Data Structures

- struct [slqssendsmsparams\\_s](#)
- struct [BroadcastConfig](#)
- struct [\\_qaQmi3GPPBroadcastCfgInfo](#)
- struct [CDMABroadcastConfig](#)
- struct [\\_qaQmi3GPP2BroadcastCfgInfo](#)
- struct [cdmaMsgEncodingParams](#)
- struct [cdmaMsgDecodingParams](#)
- struct [wcdmaMsgEncodingParams](#)
- struct [wcdmaMsgDecodingParams](#)
- struct [wcdmaLongMsgDecodingParams](#)
- struct [\\_transLayerInfo](#)
- struct [\\_getTransLayerInfoResp](#)
- struct [\\_getTransNWRegInfoResp](#)
- struct [\\_getIndicationRegResp](#)
- struct [\\_setIndicationRegReq](#)
- struct [smsRouteEntry](#)
- struct [smsSetRoutesReq](#)

- struct [smsMsgprotocolResp](#)
- struct [smsMaxStorageSizeReq](#)
- struct [smsMaxStorageSizeResp](#)
- struct [messageWaitingInfoContent](#)
- struct [getMsgWaitingInfo](#)
- struct [slqssendsyncsmsparams\\_s](#)

## Macros

- #define [CONFIG\\_LEN](#) 0x05
- #define [TIME\\_STAMP\\_BUF](#) 0x08
- #define [ABSOLUTE\\_VALIDITY](#) 0x08
- #define [TIME\\_DATE\\_BUF](#) 0x09
- #define [MAX\\_SMS\\_ROUTES](#) 0x0A
- #define [NUM\\_OF\\_SET](#) 0xFF

## Typedefs

- typedef struct [\\_qaQmi3GPPBroadcastCfgInfo](#) [qaQmi3GPPBroadcastCfgInfo](#)
- typedef struct [\\_qaQmi3GPP2BroadcastCfgInfo](#) [qaQmi3GPP2BroadcastCfgInfo](#)
- typedef struct [\\_transLayerInfo](#) [transLayerInfo](#)
- typedef struct [\\_getTransLayerInfoResp](#) [getTransLayerInfoResp](#)
- typedef struct [\\_getTransNWRegInfoResp](#) [getTransNWRegInfoResp](#)
- typedef struct [\\_getIndicationRegResp](#) [getIndicationRegResp](#)
- typedef struct [\\_setIndicationRegReq](#) [setIndicationRegReq](#)

## Functions

- [ULONG SLQSDelSMS](#) ([ULONG](#) storageType, [ULONG](#) \*pMessageIndex, [ULONG](#) \*pMessageTag, [BYTE](#) \*pMessageMode)
- [ULONG SLQSGetSMS](#) ([ULONG](#) storageType, [ULONG](#) messageIndex, [ULONG](#) \*pMessageTag, [ULONG](#) \*pMessageFormat, [ULONG](#) \*pMessageSize, [BYTE](#) \*pMessage, [BYTE](#) \*pMessageMode)
- [ULONG SendSMS](#) ([ULONG](#) messageFormat, [ULONG](#) messageSize, [BYTE](#) \*pMessage, [ULONG](#) \*pMessageFailureCode)
- [ULONG SLQSSendSMS](#) ([slqssendsmsparams\\_s](#) \*pSendSmsParams)
- [ULONG GetSMSCAddress](#) ([BYTE](#) addressSize, [CHAR](#) \*pSMSCAddress, [BYTE](#) typeSize, [CHAR](#) \*pSMSCType)
- [ULONG SetSMSCAddress](#) ([CHAR](#) \*pSMSCAddress, [CHAR](#) \*pSMSCType)
- [ULONG SaveSMS](#) ([ULONG](#) storageType, [ULONG](#) messageFormat, [ULONG](#) messageSize, [BYTE](#) \*pMessage, [ULONG](#) \*pMessageIndex)
- [ULONG SLQSGetSMSList](#) ([ULONG](#) storageType, [ULONG](#) \*pRequestedTag, [ULONG](#) \*pMessageListSize, [BYTE](#) \*pMessageList, [BYTE](#) \*pMessageMode)
- [ULONG SLQSMModifySMSStatus](#) ([ULONG](#) storageType, [ULONG](#) messageIndex, [ULONG](#) messageTag, [BYTE](#) \*pMessageMode)
- [ULONG SLQSGetSmsBroadcastConfig](#) ([BYTE](#) mode, [qaQmi3GPPBroadcastCfgInfo](#) \*pBroadcastConfig, [qaQmi3GPP2BroadcastCfgInfo](#) \*pCDMABroadcastConfig)
- [ULONG SLQSSetSmsBroadcastConfig](#) ([BYTE](#) mode, [qaQmi3GPPBroadcastCfgInfo](#) \*pBroadcastConfig, [qaQmi3GPP2BroadcastCfgInfo](#) \*pCDMABroadcastConfig)
- [ULONG SLQSSetSmsBroadcastActivation](#) ([BYTE](#) mode, [BYTE](#) broadcastActivate)

- [ULONG SLQSCDMAEncodeMOTextMsg](#) (struct [cdmaMsgEncodingParams](#) \*pCdmaMsgEncodingParams)
- [ULONG SLQSCDMADecodeMTTextMsg](#) (struct [cdmaMsgDecodingParams](#) \*pCdmaMsgDecodingParams)
- [ULONG SLQSWCDMAEncodeMOTextMsg](#) (struct [wcdmaMsgEncodingParams](#) \*pWcdmaMsgEncodingParams)
- [ULONG SLQSWCDMADecodeMTTextMsg](#) (struct [wcdmaMsgDecodingParams](#) \*pWcdmaMsgDecodingParams)
- [ULONG SLQSWCDMADecodeLongTextMsg](#) (struct [wcdmaLongMsgDecodingParams](#) \*pWcdmaLongMsgDecodingParams)
- [ULONG SLQSGetTransLayerInfo](#) ([getTransLayerInfoResp](#) \*pGetTransLayerInfoResp)
- [ULONG SLQSGetTransNWRegInfo](#) ([getTransNWRegInfoResp](#) \*pGetTransNWRegInfoResp)
- [ULONG SLQSGetIndicationRegister](#) ([getIndicationRegResp](#) \*pGetIndicationRegInfo)
- [ULONG SLQSSetIndicationRegister](#) ([setIndicationRegReq](#) \*pSetIndicationRegReq)
- [ULONG SLQSSmsSetRoutes](#) ([smsSetRoutesReq](#) \*pSetRoutesReq)
- [ULONG SLQSSmsGetMessageProtocol](#) ([smsMsgprotocolResp](#) \*pMessageProtocol)
- [ULONG SLQSSmsGetMaxStorageSize](#) ([smsMaxStorageSizeReq](#) \*pMaxStorageSizeReq, [smsMaxStorageSizeResp](#) \*pMaxStorageSizeResp)
- [ULONG SLQSGetMessageWaiting](#) ([getMsgWaitingInfo](#) \*pGetMsgWaitingInfoResp)
- [ULONG SLQSSendAsyncSMS](#) ([slqssendasyncsmsparams\\_s](#) \*pSendSmsParams)
- [ULONG SLQSSetSmsStorage](#) ([BYTE](#) smsStorage)
- [ULONG SLQSSwiGetSMSSStorage](#) ([ULONG](#) \*pSmsStorage)
- [ULONG SLQSSendLongSMS](#) ([ULONG](#) messageFormat, [ULONG](#) messageSize, [CHAR](#) \*pMessage, [BYTE](#) encodingScheme, [ULONG](#) \*pMessageFailureCode, [CHAR](#) \*pMobileNum)

### 9.19.1 Detailed Description

Short Message Service API function prototypes.

### 9.19.2 Macro Definition Documentation

9.19.2.1 `#define ABSOLUTE_VALIDITY 0x08`

9.19.2.2 `#define CONFIG_LEN 0x05`

9.19.2.3 `#define MAX_SMS_ROUTES 0x0A`

9.19.2.4 `#define NUM_OF_SET 0xFF`

9.19.2.5 `#define TIME_DATE_BUF 0x09`

9.19.2.6 `#define TIME_STAMP_BUF 0x08`

### 9.19.3 Typedef Documentation

9.19.3.1 `typedef struct _getIndicationRegResp getIndicationRegResp`

This structure contains Get Indication Register Response parameters



## Parameters

<i>pRegTransLayerInfoEvt</i>	- <ul style="list-style-type: none"> <li>Optional 1 BYTE parameter indicating registration status of transport layer information events</li> <li>Values: <ul style="list-style-type: none"> <li>0x00 - Disabled</li> <li>0x01 - Enabled</li> </ul> </li> <li>function <a href="#">SLQSGetIndicationRegister()</a> returns a default value 0xFF if this parameter is allocated memory in the structure and no response is received from the device.</li> </ul>
<i>pRegTransNWRegInfoEvt</i>	- <ul style="list-style-type: none"> <li>Optional 1 BYTE parameter indicating registration status of transport network registration information events</li> <li>Values: <ul style="list-style-type: none"> <li>0x00 - Disabled</li> <li>0x01 - Enabled</li> </ul> </li> <li>function <a href="#">SLQSGetIndicationRegister()</a> returns a default value 0xFF if this parameter is allocated memory in the structure and no response is received from the device.</li> </ul>
<i>pRegCallStatInfoEvt</i>	- <ul style="list-style-type: none"> <li>Optional 1 BYTE parameter indicating registration status of call status information events</li> <li>Values: <ul style="list-style-type: none"> <li>0x00 - Disabled</li> <li>0x01 - Enabled</li> </ul> </li> <li>function <a href="#">SLQSGetIndicationRegister()</a> returns a default value 0xFF if this parameter is allocated memory in the structure and no response is received from the device.</li> </ul>

## 9.19.3.2 typedef struct \_getTransLayerInfoResp getTransLayerInfoResp

This structure contains Get Transport Layer Info Response parameters

## Parameters

<i>pRegInd</i>	- <ul style="list-style-type: none"> <li>Optional parameter indicating if transport layer is registered</li> <li>Values: <ul style="list-style-type: none"> <li>0x00 - Transport layer is not registered</li> <li>0x01 - Transport layer is registered</li> </ul> </li> <li>function <a href="#">SLQSGetTransLayerInfo()</a> returns a default value 0xFF if no response is received from the device.</li> </ul>
<i>pTransLayerInfo</i>	<ul style="list-style-type: none"> <li>Pointer to structure of transLayerInfo. <ul style="list-style-type: none"> <li>Optional parameter</li> <li>See <a href="#">transLayerInfo</a> for more information</li> </ul> </li> <li>function <a href="#">SLQSGetTransLayerInfo()</a> returns a default value 0xFF for parameter values if no response is received from the device.</li> </ul>

## 9.19.3.3 typedef struct \_getTransNWRegInfoResp getTransNWRegInfoResp

This structure contains transport network registration info parameter

## Parameters

<i>pRegStatus</i>	- <ul style="list-style-type: none"> <li>Optional 1 BYTE parameter indicating transport layer network registration status</li> <li>Values: <ul style="list-style-type: none"> <li>0x00 - No service</li> <li>0x01 - In progress</li> <li>0x02 - Failed</li> <li>0x03 - Limited Service</li> <li>0x04 - Full Service</li> </ul> </li> <li>function <a href="#">SLQSGetTransNWRegInfo()</a> returns a default value 0xFF if no response is received from the device.</li> </ul>
-------------------	---

## 9.19.3.4 typedef struct \_qaQmi3GPP2BroadcastCfgInfo qaQmi3GPP2BroadcastCfgInfo

This structure contains the 3GPP2 Broadcast Configuration Information parameters

## Parameters

<i>activated_ind</i>	<ul style="list-style-type: none"> <li>• Broadcast SMS <ul style="list-style-type: none"> <li>– 0x00 - Deactivated</li> <li>– 0x01 - Activated</li> </ul> </li> </ul>
<i>num_instances</i>	<ul style="list-style-type: none"> <li>• Number of sets (N) of parameters Following each set describes one entry in the broadcast configuration table. <ul style="list-style-type: none"> <li>– serviceCategory</li> <li>– language</li> <li>– selected</li> </ul> </li> </ul>
<i>broadcastConfig</i>	<ul style="list-style-type: none"> <li>• A <a href="#">CDMABroadcastConfig</a> structure array.</li> <li>• Further defined by the structure <a href="#">CDMABroadcastConfig</a></li> </ul>

## 9.19.3.5 typedef struct \_qaQmi3GPPBroadcastCfgInfo qaQmi3GPPBroadcastCfgInfo

This structure contains the 3GPP Broadcast Configuration Information parameters

## Parameters

<i>activated_ind</i>	<ul style="list-style-type: none"> <li>• Broadcast SMS <ul style="list-style-type: none"> <li>– 0x00 - Deactivated</li> <li>– 0x01 - Activated</li> </ul> </li> </ul>
<i>num_instances</i>	<ul style="list-style-type: none"> <li>• Number of sets (N) of parameters Following each set describes one entry in the broadcast configuration table. <ul style="list-style-type: none"> <li>– fromServiceId</li> <li>– toServiceId</li> <li>– selected</li> </ul> </li> </ul>

<i>broadcastConfig</i>	<ul style="list-style-type: none"> <li>• A <a href="#">BroadcastConfig</a> structure array.</li> <li>• Further defined by the structure <a href="#">BroadcastConfig</a></li> </ul>
------------------------	--

#### 9.19.3.6 typedef struct \_setIndicationRegReq setIndicationRegReq

This structure contains Indication Register request parameters

##### Parameters

<i>pRegTransLayerInfoEvt</i>	- <ul style="list-style-type: none"> <li>• Optional 1 BYTE parameter indicating registration status of transport layer information events</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0x00 - Disabled</li> <li>– 0x01 - Enabled</li> <li>– NULL - No change - specifying NULL indicates that the device will continue to use the existing setting (disable/enable) which has been previously set for the device</li> </ul> </li> </ul>
<i>pRegTransNWRegInfoEvt</i>	- <ul style="list-style-type: none"> <li>• Optional 1 BYTE parameter indicating registration status of transport network registration information events</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0x00 - Disabled</li> <li>– 0x01 - Enabled</li> <li>– NULL - No change - specifying NULL indicates that the device will continue to use the existing setting (disable/enable) which has been previously set for the device</li> </ul> </li> </ul>
<i>pRegCallStatInfoEvt</i>	- <ul style="list-style-type: none"> <li>• Optional 1 BYTE parameter indicating registration status of call status information events</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0x00 - Disabled</li> <li>– 0x01 - Enabled</li> <li>– NULL - No change - specifying NULL indicates that the device will continue to use the existing setting (disable/enable) which has been previously set for the device</li> </ul> </li> </ul>

#### 9.19.3.7 typedef struct \_transLayerinfo transLayerInfo

This structure contains Transport Layer Information

## Parameters

<i>TransType</i>	<ul style="list-style-type: none"> <li>• Transport Type <ul style="list-style-type: none"> <li>– 0x00 - IMS</li> </ul> </li> </ul>
<i>TransCap</i>	<ul style="list-style-type: none"> <li>• Transport Capability</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0x00 - CDMA</li> <li>– 0x01 - GW</li> </ul> </li> </ul>

## 9.19.4 Function Documentation

9.19.4.1 **ULONG** GetSMSCAddress ( **BYTE** *addressSize*, **CHAR** \* *pSMSCAddress*, **BYTE** *typeSize*, **CHAR** \* *pSMSCType* )

Gets the SMS center address.

## Parameters

<i>addressSize</i>	<ul style="list-style-type: none"> <li>• The maximum number of characters (including NULL terminator) that the SMS center address array can contain.</li> </ul>
<i>pSMSC-Address[0UT]</i>	<ul style="list-style-type: none"> <li>• The SMS center address represented as a NULL terminated string.</li> </ul>
<i>typeSize</i>	<ul style="list-style-type: none"> <li>• The maximum number of characters (including NULL terminator) that the SMS center address type array can contain.</li> </ul>
<i>pSMSCType[0U-T]</i>	<ul style="list-style-type: none"> <li>• The SMS center address type represented as a NULL terminated string.</li> </ul>

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Technology Supported: ALL  
Timeout: 2 seconds

9.19.4.2 **ULONG** SaveSMS ( **ULONG** *storageType*, **ULONG** *messageFormat*, **ULONG** *messageSize*, **BYTE** \* *pMessage*, **ULONG** \* *pMessageIndex* )

Saves an SMS message to device memory

## Parameters

<i>storageType</i>	<ul style="list-style-type: none"> <li>• SMS message storage type <ul style="list-style-type: none"> <li>– 0 - UIM - Invalid in case of CDMA device that does not require SIM</li> <li>– 1 - NV</li> </ul> </li> </ul>
<i>messageFormat</i>	<ul style="list-style-type: none"> <li>• Message format <ul style="list-style-type: none"> <li>– 0 - CDMA (IS-637B)</li> <li>– 1 - 5 (Reserved)</li> <li>– 6 - GSM/WCDMA PP</li> </ul> </li> </ul>
<i>messageSize</i>	<ul style="list-style-type: none"> <li>• The length of the message contents in bytes</li> </ul>
<i>pMessage[IN]</i>	<ul style="list-style-type: none"> <li>• The message contents</li> </ul>
<i>pMessage-Index[OUT]</i>	<ul style="list-style-type: none"> <li>• The message index assigned by the device</li> </ul>

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Technology Supported: ALL  
Timeout: 10 seconds

#### 9.19.4.3 **ULONG** SendSMS ( **ULONG** *messageFormat*, **ULONG** *messageSize*, **BYTE** \* *pMessage*, **ULONG** \* *pMessageFailureCode* )

Sends an SMS message for immediate over-the-air transmission

## Parameters

<i>messageFormat</i>	<ul style="list-style-type: none"> <li>• Message format <ul style="list-style-type: none"> <li>– 0 - CDMA (IS-637B)</li> <li>– 1 - 5 (Reserved)</li> <li>– 6 - GSM/WCDMA PP</li> </ul> </li> </ul>
<i>messageSize</i>	<ul style="list-style-type: none"> <li>• The length of the message contents in bytes</li> </ul>
<i>pMessage[IN]</i>	<ul style="list-style-type: none"> <li>• The message contents in PDU format contains SMS header and payload message</li> </ul>
<i>pMessage-FailureCode[OUT]</i>	<ul style="list-style-type: none"> <li>• pointer to message failure code. If cause code is not provided, then value will be 0xFF-FFFFFF</li> </ul>

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Technology Supported: ALL  
Timeout: 5 minutes

#### 9.19.4.4 ULONG SetSMSCAddress ( CHAR \* pSMSCAddress, CHAR \* pSMSCType )

Sets the SMS center address.

## Parameters

<i>pSMSC-Address[IN]</i>	<ul style="list-style-type: none"> <li>• The SMS center address represented as a NULL terminated string</li> </ul>
<i>pSMSCType[IN]</i>	<ul style="list-style-type: none"> <li>• The SMS center address type represented as a NULL terminated string (optional).</li> </ul>

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values



## Note

Technology Supported: ALL  
Timeout: 5 seconds

**9.19.4.5   ULONG SLQSCDMADecodeMTTextMsg ( struct cdmaMsgDecodingParams \* pCdmaMsgDecodingParams )**

Decodes text message to CDMA PDU message

## Parameters

<i>pMsgToBe- EncodedCDMA[ IN/OUT]</i>	<ul style="list-style-type: none"><li>• Pointer to structure containing parameters needed for decoding</li></ul>
---	--

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

see [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Technology Supported: CDMA  
Timeout: None

**9.19.4.6   ULONG SLQSCDMAEncodeMOTextMsg ( struct cdmaMsgEncodingParams \* pCdmaMsgEncodingParams )**

Encodes text message to CDMA PDU message.

## Parameters

<i>pMsgToBe- EncodedCDMA[ IN/OUT]</i>	<ul style="list-style-type: none"><li>• SLQS Runtime Settings Information</li></ul>
---	---

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

see [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Technology Supported: CDMA  
Timeout: None

#### 9.19.4.7 **ULONG** SLQSDelSDeleteSMS ( **ULONG** *storageType*, **ULONG** \* *pMessageIndex*, **ULONG** \* *pMessageTag*, **BYTE** \* *pMessageMode* )

Deletes one or more SMSs from device memory. If both of the optional parameters, *messageIndex* and *messageTag*, are NULL, all messages are deleted from the storage location specified in the mandatory *storageType* parameter. The optional index and tag parameters narrow the range of messages being deleted. If an index is specified, the single message with the index from the specified memory store will be deleted. If a tag is specified, all messages in the specified memory store whose tag matches that specified will be deleted.

There are three ways to use this message:

- Specify *storageType* only
  - Deletes all messages from memory storage
- Specify *storageType* and a tag
  - Deletes all messages from memory storage that match the given message tag
- Specify *storageType* and an index
  - Deletes only the message with the given index from memory storage

#### Parameters

<i>storageType</i>	<ul style="list-style-type: none"> <li>• SMS message storage type               <ul style="list-style-type: none"> <li>– 0 - UIM - Invalid in case of CDMA device that does not require SIM</li> <li>– 1 - NV</li> </ul> </li> </ul>
<i>pMessageIndex</i> [IN]	<ul style="list-style-type: none"> <li>• (Optional) message index</li> </ul>
<i>pMessageTag</i> [I-N]	<ul style="list-style-type: none"> <li>• (Optional) message tag               <ul style="list-style-type: none"> <li>– 0 - Read</li> <li>– 1 - Not read</li> <li>– 2 - Mobile originated and sent</li> <li>– 3 - Mobile originated but not yet sent</li> </ul> </li> </ul>
<i>pMessageMode</i> [IN]	<ul style="list-style-type: none"> <li>• 0x00 - CDMA, LTE (if network type is CDMA)</li> <li>• 0x01 - GW, LTE (if network type is UMTS)</li> </ul>

#### Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

#### See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Technology Supported: ALL  
 Timeout: 10 seconds

#### 9.19.4.8 ULONG SLQSGetIndicationRegister ( getIndicationRegResp \* pGetIndicationRegInfo )

This API provides registration state of different WMS indications.

## Parameters

<i>pGetIndication-RegInfo</i>	[OUT] <ul style="list-style-type: none"> <li>• Pointer to structure of getIndicationRegResp             <ul style="list-style-type: none"> <li>– See <a href="#">getIndicationRegResp</a> for more information</li> </ul> </li> </ul>
-------------------------------	---

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 2 Secs

#### 9.19.4.9 ULONG SLQSGetMessageWaiting ( getMsgWaitingInfo \* pGetMsgWaitingInfoResp )

This API provides information about the message waiting information.

## Parameters

<i>pGetMsg-WaitingInfoResp</i>	[OUT] <ul style="list-style-type: none"> <li>• Pointer to structure of getMsgWaitingInfoResp             <ul style="list-style-type: none"> <li>– See <a href="#">getMsgWaitingInfoResp</a> for more information</li> </ul> </li> </ul>
--------------------------------	---

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 2 Secs

#### 9.19.4.10 ULONG SLQSGetSMS ( ULONG storageType, ULONG messageIndex, ULONG \* pMessageTag, ULONG \* pMessageFormat, ULONG \* pMessageSize, BYTE \* pMessage, BYTE \* pMessageMode )

Returns an SMS from device memory.

## Parameters

<i>storageType</i>	<ul style="list-style-type: none"> <li>SMS message storage type <ul style="list-style-type: none"> <li>0 - UIM - Invalid in case of CDMA device that does not require SIM</li> <li>1 - NV</li> </ul> </li> </ul>
<i>messageIndex</i>	<ul style="list-style-type: none"> <li>Message index</li> </ul>
<i>pMessageTag[OUT]</i>	<ul style="list-style-type: none"> <li>Message tag <ul style="list-style-type: none"> <li>0 - Read</li> <li>1 - Not read</li> <li>2 - Mobile originated and sent</li> <li>3 - Mobile originated but not yet sent</li> </ul> </li> </ul>
<i>pMessageFormat[OUT]</i>	<ul style="list-style-type: none"> <li>Message format <ul style="list-style-type: none"> <li>0 - CDMA (IS-637B)</li> <li>1 - 5 (Reserved)</li> <li>6 - GSM/WCDMA PP</li> </ul> </li> </ul>
<i>pMessageSize[IN/OUT]</i>	<ul style="list-style-type: none"> <li>Upon input the maximum number of bytes that can be written to the message array.</li> <li>Upon successful output the actual number of bytes written to the message array.</li> </ul>
<i>pMessage[OUT]</i>	<ul style="list-style-type: none"> <li>The message contents array</li> </ul>
<i>pMessageMode[IN]</i>	<ul style="list-style-type: none"> <li>0x00 - CDMA, LTE (if network type is CDMA)</li> <li>0x01 - GW, LTE (if network type is UMTS)</li> </ul>

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Technology Supported: ALL  
Timeout: 5 seconds

9.19.4.11 **ULONG** SLQSGetSmsBroadcastConfig ( **BYTE** *mode*, **qaQmi3GPPBroadcastCfgInfo** \* *pBroadcastConfig*, **qaQmi3GPP2BroadcastCfgInfo** \* *pCDMABroadcastConfig* )

Provides Information about the SMS BroadcastConfiguration

## Parameters

<i>mode</i>	<ul style="list-style-type: none"> <li>• Mode <ul style="list-style-type: none"> <li>– 0x00 - CDMA, LTE (if network type is CDMA)</li> <li>– 0x01 - GW, LTE (if network type is UMTS)</li> </ul> </li> </ul>
<i>pBroadcast-Config[OUT]</i>	<ul style="list-style-type: none"> <li>• The data for 3GPP Broadcast Information.</li> </ul>
<i>pCDMA-Broadcast-Config[OUT]</i>	<ul style="list-style-type: none"> <li>• The data for 3GPP2 Broadcast Information.</li> </ul>

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Technology Supported: ALL

Timeout: 5 seconds

**9.19.4.12** **ULONG** SLQSGetSMSList ( **ULONG** *storageType*, **ULONG** \* *pRequestedTag*, **ULONG** \* *pMessageListSize*, **BYTE** \* *pMessageList*, **BYTE** \* *pMessageMode* )

Returns the list of SMS messages stored on the device.

## Parameters

<i>storageType</i>	<ul style="list-style-type: none"> <li>• SMS message storage type <ul style="list-style-type: none"> <li>– 0 - UIM - Invalid in case of CDMA device that does not require SIM</li> <li>– 1 - NV</li> </ul> </li> </ul>
<i>pRequested-Tag[IN]</i>	<ul style="list-style-type: none"> <li>• (Optional) Message tag <ul style="list-style-type: none"> <li>– 0 - Read</li> <li>– 1 - Not read</li> <li>– 2 - Mobile originated and sent</li> <li>– 3 - Mobile originated but not yet sent</li> </ul> </li> </ul>

<i>pMessageList-Size</i> [IN/OUT]	<ul style="list-style-type: none"> <li>• Upon input the maximum number of elements that the message list array can contain.</li> <li>• Upon successful output the actual number of elements in the message list array.</li> </ul>
<i>pMessageList</i> [OUT]	<ul style="list-style-type: none"> <li>• The message list array</li> </ul>
<i>pMessage-Mode</i> [IN]	<ul style="list-style-type: none"> <li>• 0x00 - CDMA, LTE (if network type is CDMA)</li> <li>• 0x01 - GW, LTE (if network type is UMTS)</li> </ul>

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Technology Supported: ALL  
Timeout: 5 seconds

#### 9.19.4.13 ULONG SLQSGetTransLayerInfo ( getTransLayerInfoResp \* pGetTransLayerInfoResp )

This API provides information about the transport layer.

**Parameters**

<i>pGetTransLayer-InfoResp</i>	[OUT] <ul style="list-style-type: none"> <li>• Pointer to structure of getTransLayerInfoResp               <ul style="list-style-type: none"> <li>– See <a href="#">getTransLayerInfoResp</a> for more information</li> </ul> </li> </ul>
--------------------------------	---

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Timeout: 2 Secs

#### 9.19.4.14 ULONG SLQSGetTransNWRegInfo ( getTransNWRegInfoResp \* pGetTransNWRegInfoResp )

This API provides transport layer network registration info.

## Parameters

<i>pGetTransNW-RegInfoResp</i>	[OUT] <ul style="list-style-type: none"> <li>• Pointer to structure of getTransNWRegInfoResp               <ul style="list-style-type: none"> <li>– See <a href="#">getTransNWRegInfoResp</a> for more information</li> </ul> </li> </ul>
--------------------------------	---

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 2 Secs

#### 9.19.4.15 **ULONG** SLQSMModifySMSStatus ( **ULONG** *storageType*, **ULONG** *messageIndex*, **ULONG** *messageTag*, **BYTE** \* *pMessageMode* )

Modifies the status of an SMS message saved in storage on the device.

## Parameters

<i>storageType</i>	<ul style="list-style-type: none"> <li>• SMS message storage type               <ul style="list-style-type: none"> <li>– 0 - UIM - Invalid in case of CDMA device that does not require SIM</li> <li>– 1 - NV</li> </ul> </li> </ul>
<i>messageIndex</i>	<ul style="list-style-type: none"> <li>• Message index</li> </ul>
<i>messageTag</i>	<ul style="list-style-type: none"> <li>• Message tag               <ul style="list-style-type: none"> <li>– 0 - Read</li> <li>– 1 - Not read</li> </ul> </li> </ul>
<i>pMessage-Mode</i> [IN]	<ul style="list-style-type: none"> <li>• 0x00 - CDMA, LTE (if network type is CDMA)</li> <li>• 0x01 - GW, LTE (if network type is UMTS)</li> </ul>

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values



## Note

Technology Supported: ALL  
 Timeout: 5 seconds

#### 9.19.4.16 ULONG SLQSSendAsyncSMS ( *slqssendasyncsmsparams\_s* \* *pSendSmsParams* )

Sends an SMS message for immediate over-the-air transmission

## Parameters

<i>pSendSmsParams</i>	<ul style="list-style-type: none"> <li>structure containing the SMS parameters. Refer <a href="#">slqssendasyncsmsparams_s</a></li> </ul>
-----------------------	---

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Technology Supported: ALL  
 Timeout: 5 minutes

#### 9.19.4.17 ULONG SLQSSendLongSMS ( ULONG *messageFormat*, ULONG *messageSize*, CHAR \* *pMessage*, BYTE *encodingScheme*, ULONG \* *pMessageFailureCode*, CHAR \* *pMobileNum* )

Sends a long SMS message for immediate over-the-air transmission, a short SMS can be sent by this API as well, the input message is text string without any encoding

## Parameters

<i>messageFormat</i> [IN]	<ul style="list-style-type: none"> <li>Message format               <ul style="list-style-type: none"> <li>0 - CDMA (IS-637B)</li> <li>1 - 5 (Reserved)</li> <li>6 - GSM/WCDMA PP</li> </ul> </li> </ul>
<i>messageSize</i> [IN]	<ul style="list-style-type: none"> <li>Message size of the input message text</li> </ul>
<i>pMessage</i> [IN]	<ul style="list-style-type: none"> <li>Original message text</li> </ul>

<i>encoding-Scheme</i> [IN]	<ul style="list-style-type: none"> <li>• Encoding method to generate the PDU <ul style="list-style-type: none"> <li>– 0 - 7 bit encoding</li> <li>– 4 - 8 bit encoding</li> <li>– 8 - 16 bit UCS2 encoding</li> <li>– others value will be treated as default 7 bit encoding</li> </ul> </li> </ul>
<i>pMessage-FailureCode</i> [OUT]	<ul style="list-style-type: none"> <li>• message failure code. If cause code is not provided, then value will be 0xFFFFFFFF</li> </ul>
<i>pMobileNum</i> [IN]	<ul style="list-style-type: none"> <li>• Mobile number of the receiver</li> </ul>

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Technology Supported: 3GPP and 3GPP2  
Timeout: 5 minutes

#### 9.19.4.18 **ULONG SLQSSendSMS ( slqssendsmsparams\_s \* pSendSmsParams )**

Sends an SMS message for immediate over-the-air transmission

**Parameters**

<i>pSendSms-Params</i>	<ul style="list-style-type: none"> <li>• structure containing the SMS parameters. Refer <a href="#">slqssendsmsparams_s</a></li> </ul>
------------------------	--

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Technology Supported: ALL  
Timeout: 5 minutes

#### 9.19.4.19 **ULONG SLQSSetIndicationRegister ( setIndicationRegReq \* pSetIndicationRegReq )**

This API sets the registration state of different WMS indications.

## Parameters

<i>pSetIndication-RegReq</i>	[IN] <ul style="list-style-type: none"> <li>• Pointer to structure of indicationRegReqParams             <ul style="list-style-type: none"> <li>– See <a href="#">setIndicationRegReq</a> for more information</li> </ul> </li> </ul>
------------------------------	---

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 2 Secs

#### 9.19.4.20 ULONG SLQSSetSmsBroadcastActivation ( BYTE mode, BYTE broadcastActivate )

Enables or disables the reception of broadcast SMS messages.

## Parameters

<i>Mode</i>	<ul style="list-style-type: none"> <li>• Mode</li> <li>• 0x00 - CDMA, LTE (if network type is CDMA)</li> <li>• 0x01 - GW, LTE (if network type is UMTS)</li> </ul>
<i>broadcast-Activate</i>	<ul style="list-style-type: none"> <li>• 0x00 - Disable broadcast</li> <li>• 0x01 - Activate broadcast</li> </ul>

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Technology Supported: ALL  
Timeout: 5 seconds

#### 9.19.4.21 ULONG SLQSSetSmsBroadcastConfig ( BYTE mode, qaQmi3GPPBroadcastCfgInfo \* pBroadcastConfig, qaQmi3GPP2BroadcastCfgInfo \* pCDMABroadcastConfig )

Sets the information about the SMS BroadcastConfiguration

## Parameters

<i>mode</i>	<ul style="list-style-type: none"> <li>• Mode <ul style="list-style-type: none"> <li>– 0x00 - CDMA, LTE (if network type is CDMA)</li> <li>– 0x01 - GW, LTE (if network type is UMTS)</li> </ul> </li> </ul>
<i>pBroadcast-Config[IN]</i>	<ul style="list-style-type: none"> <li>• The data for 3GPP Broadcast Information.</li> </ul>
<i>pCDMA-Broadcast-Config[IN]</i>	<ul style="list-style-type: none"> <li>• The data for 3GPP2 Broadcast Information.</li> </ul>

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Technology Supported: ALL

Timeout: 5 seconds

#### 9.19.4.22 ULONG SLQSSetSmsStorage ( BYTE *smsStorage* )

Sets the SMS Storage on the device

## Parameters

<i>smsStorage</i>	<ul style="list-style-type: none"> <li>• SMS Storage <ul style="list-style-type: none"> <li>– 0x01 - device's permanent memory</li> <li>– 0x02 - UICC</li> </ul> </li> </ul>
-------------------	--

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Technology Supported: ALL

Timeout: 5 seconds

9.19.4.23 **ULONG** SLQSSmsGetMaxStorageSize ( **smsMaxStorageSizeReq** \* *pMaxStorageSizeReq*,  
**smsMaxStorageSizeResp** \* *pMaxStorageSizeResp* )

This API provides the maximum number of messages that can be stored in the specified memory storage. Also it provides the number of slots currently available

## Parameters

<i>pMaxStorage-SizeReq[IN]</i>	<ul style="list-style-type: none"> <li>Request parameters for SmsSLQSGetMaxStorageSize               <ul style="list-style-type: none"> <li>See <a href="#">smsMaxStorageSizeReq</a> for more information</li> </ul> </li> </ul>
<i>pMaxStorage-SizeResp[OUT]</i>	<ul style="list-style-type: none"> <li>Response parameters for SmsSLQSGetMaxStorageSize               <ul style="list-style-type: none"> <li>See <a href="#">smsMaxStorageSizeResp</a> for more information</li> </ul> </li> </ul>

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 2 Secs

#### 9.19.4.24 ULONG SLQSSmsGetMessageProtocol ( smsMsgprotocolResp \* pMessageProtocol )

This API queries the message protocol currently in use for the WMS client.

## Parameters

<i>pMessage-Protocol</i>	<ul style="list-style-type: none"> <li>Pointer to <a href="#">smsMsgprotocolResp</a> <ul style="list-style-type: none"> <li>See <a href="#">smsMsgprotocolResp</a> for more information</li> </ul> </li> </ul>
--------------------------	--

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 2 Secs

#### 9.19.4.25 ULONG SLQSSmsSetRoutes ( smsSetRoutesReq \* pSetRoutesReq )

This API sets the action performed on SMS message receipt for specified message routes. It also specifies the action performed on SMS receipt of status reports.

## Parameters

<i>pSetRoutesReq</i>	[IN] <ul style="list-style-type: none"> <li>• Pointer to structure of <a href="#">smsSetRoutesReq</a> <ul style="list-style-type: none"> <li>– See <a href="#">smsSetRoutesReq</a> for more information</li> </ul> </li> </ul>
----------------------	--

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 2 Secs

#### 9.19.4.26 **ULONG** SLQSSwiGetSMSStorage ( **ULONG** \* *pSmsStorage* )

This API queries the device to return current SMS configuration that is applied to all incoming and outgoing messages.

## Parameters

<i>pSmsStorage</i>	<ul style="list-style-type: none"> <li>• Values:             <ul style="list-style-type: none"> <li>– 0x01 - device's permanent memory</li> <li>– 0x02 - UICC</li> </ul> </li> </ul>
--------------------	--

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 5 Secs

#### 9.19.4.27 **ULONG** SLQSWCDMADecodeLongTextMsg ( struct **wcdmaLongMsgDecodingParams** \* *pWcdmaLongMsgDecodingParams* )

Decodes WCDMA Long SMS PDU message, returns structure filled with decoded parameters

**Parameters**

<i>pwdmaMsg-Decoding-Params</i> [IN/OUT]	<ul style="list-style-type: none"><li>• Pointer to parameters required for decoding</li></ul>
--	---

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

see [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Technology Supported: UMTS  
Timeout: none

9.19.4.28 **ULONG** SLQSWCDMADecodeMTTextMsg ( struct wcdmaMsgDecodingParams \*  
*pWcdmaMsgDecodingParams* )

Decodes WCDMA PDU message, returns structure filled with decoded parameters

**Parameters**

<i>pwdmaMsg-Decoding-Params</i> [IN/OUT]	<ul style="list-style-type: none"><li>• Pointer to parameters required for decoding</li></ul>
--	---

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

see [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Technology Supported: UMTS  
Timeout: none

9.19.4.29 **ULONG** SLQSWCDMAEncodeMOTextMsg ( struct wcdmaMsgEncodingParams \*  
*pWcdmaMsgEncodingParams* )

Returns the encoded WCDMA PDU message.

**Parameters**

<i>pwdmaMsg-Encoding-Params</i> [IN/OUT]	<ul style="list-style-type: none"><li>• Pointer to parameters Required for encoding</li></ul>
--	---



**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

see [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Technology Supported: UMTS  
Timeout: None

## 9.20 qaGobiApiSwi.h File Reference

SWI API function prototypes.

**Functions**

- [ULONG SLQSGetSdkVersion](#) ([CHAR](#) \*\*sdkversionpp)
- [ULONG SLQSSendRawQMI](#) ([BYTE](#) \*pReqBuf, [USHORT](#) service, [USHORT](#) length, [ULONG](#) timeout, [BYTE](#) \*\*pInParm, [USHORT](#) \*pParamLength)
- [int SLQSPidof](#) ([CHAR](#) \*pProcName)

### 9.20.1 Detailed Description

SWI API function prototypes.

### 9.20.2 Function Documentation

#### 9.20.2.1 int SLQSPidof ( [CHAR](#) \* *pProcName* )

Internal Wrapper function for enabling invocation of SLQS implementation pidof() function

**Parameters**

<i>pProcName</i> [IN]	<ul style="list-style-type: none"> <li>• Process name whose PID is to be retrieved</li> </ul>
-----------------------	---

**Returns**

pid if process exists else 0

**See Also**

NA

**Note**

NA

#### 9.20.2.2 ULONG SLQSGetSdkVersion ( [CHAR](#) \*\* *sdkversionpp* )

Returns the SDK version string

## Parameters

<i>ppString[OUT]</i>	<ul style="list-style-type: none"> <li>• pointer to pointer of NULL terminated string</li> </ul>
----------------------	--

## Returns

eQCWWAN\_ERR\_NONE success eQCWWAN\_ERR\_INVALID\_ARG provided pointer is NULL

## Note

Technology Supported: N/A Timeout: 2 seconds

9.20.2.3 **ULONG** SLQSSendRawQMI ( **BYTE** \* *pReqBuf*, **USHORT** *service*, **USHORT** *length*, **ULONG** *timeout*, **BYTE** \*\* *ppInParm*, **USHORT** \* *pParamLength* )

## 9.21 qaGobiApiSwiAudio.h File Reference

M2M Audio Service API function prototypes.

### Data Structures

- struct [GetM2MAudioProfileReq](#)
- struct [GetM2MAudioProfileResp](#)
- struct [SetM2MAudioProfileReq](#)
- struct [GetM2MAudioVolumeReq](#)
- struct [GetM2MAudioVolumeResp](#)
- struct [SetM2MAudioVolumeReq](#)
- struct [PCMparams](#)
- struct [SetM2MAudioAVCFGReq](#)
- struct [SetM2MAudioLPBKReq](#)
- struct [GetM2MSpkrGainReq](#)
- struct [GetM2MSpkrGainResp](#)
- struct [SetM2MSpkrGainReq](#)
- struct [GetM2MAVMuteReq](#)
- struct [GetM2MAVMuteResp](#)
- struct [SetM2MAVMuteReq](#)

### Macros

- `#define MAX_LEN_IFACE_TABLE 255`

### Functions

- **ULONG** SLQSGetM2MAudioProfile ([GetM2MAudioProfileReq](#) \*pGetM2MAudioProfileReq, [GetM2MAudioProfileResp](#) \*pGetM2MAudioProfileResp)
- **ULONG** SLQSSetM2MAudioProfile ([SetM2MAudioProfileReq](#) \*pSetM2MAudioProfileReq)
- **ULONG** SLQSGetM2MAudioVolume ([GetM2MAudioVolumeReq](#) \*pGetM2MAudioVolumeReq, [GetM2MAudioVolumeResp](#) \*pGetM2MAudioVolumeResp)
- **ULONG** SLQSSetM2MAudioVolume ([SetM2MAudioVolumeReq](#) \*pSetM2MAudioVolumeReq)
- **ULONG** SLQSSetM2MAudioAVCFG ([SetM2MAudioAVCFGReq](#) \*pSetM2MAudioAVCFGReq)

- [ULONG SLQSSetM2MAudioLPBK](#) ([SetM2MAudioLPBKReq](#) \*pSetM2MAudioLPBKReq)
- [ULONG SLQSSetM2MAudioNVDef](#) ()
- [ULONG SLQSGetM2MSpkrGain](#) ([GetM2MSpkrGainReq](#) \*pSpkrGainReq, [GetM2MSpkrGainResp](#) \*pSpkrGainResp)
- [ULONG SLQSSetM2MSpkrGain](#) ([SetM2MSpkrGainReq](#) \*pSpkrGainReq)
- [ULONG SLQSGetM2MAVMute](#) ([GetM2MAVMuteReq](#) \*pGetM2MAVMuteReq, [GetM2MAVMuteResp](#) \*pGetM2MAVMuteResp)
- [ULONG SLQSSetM2MAVMute](#) ([SetM2MAVMuteReq](#) \*pSetM2MAVMuteReq)

### 9.21.1 Detailed Description

M2M Audio Service API function prototypes.

### 9.21.2 Macro Definition Documentation

9.21.2.1 `#define MAX_LEN_IFACE_TABLE 255`

### 9.21.3 Function Documentation

9.21.3.1 **ULONG SLQSGetM2MAudioProfile** ( [GetM2MAudioProfileReq](#) \* *pGetM2MAudioProfileReq*, [GetM2MAudioProfileResp](#) \* *pGetM2MAudioProfileResp* )

This API gets the profile content.

Parameters

<i>pGetM2MAudioProfileReq</i> [OUT]	<ul style="list-style-type: none"> <li>• See <a href="#">GetM2MAudioProfileReq</a> for more information</li> </ul>
<i>pGetM2MAudioProfileResp</i> [OUT]	<ul style="list-style-type: none"> <li>• See <a href="#">GetM2MAudioProfileResp</a> for more information</li> </ul>

Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

Note

Timeout: 5 seconds

9.21.3.2 **ULONG SLQSGetM2MAudioVolume** ( [GetM2MAudioVolumeReq](#) \* *pGetM2MAudioVolumeReq*, [GetM2MAudioVolumeResp](#) \* *pGetM2MAudioVolumeResp* )

This API gets the Volume content.

## Parameters

<i>pGetM2MAudioVolumeReq</i> [IN]	<ul style="list-style-type: none"> <li>• See <a href="#">GetM2MAudioVolumeReq</a> for more information</li> </ul>
<i>pGetM2MAudioVolumeResp</i> [OUT]	<ul style="list-style-type: none"> <li>• See <a href="#">GetM2MAudioVolumeResp</a> for more information</li> </ul>

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 5 seconds

### 9.21.3.3 ULONG SLQSGetM2MAVMute ( GetM2MAVMuteReq \* *pGetM2MAVMuteReq*, GetM2MAVMuteResp \* *pGetM2MAVMuteResp* )

This API Gets the AV Mute content.

## Parameters

<i>pGetM2MAVMuteReq</i> [IN]	<ul style="list-style-type: none"> <li>• See <a href="#">GetM2MAVMuteReq</a> for more information</li> </ul>
<i>pGetM2MAVMuteResp</i> [OUT]	<ul style="list-style-type: none"> <li>• See <a href="#">GetM2MAVMuteResp</a> for more information</li> </ul>

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 5 seconds

### 9.21.3.4 ULONG SLQSGetM2MSpkrGain ( GetM2MSpkrGainReq \* *pSpkrGainReq*, GetM2MSpkrGainResp \* *pSpkrGainResp* )

This API Gets the SPKRGAIN content.

## Parameters

<i>pSpkrGainReq</i> [IN]	<ul style="list-style-type: none"> <li>See <a href="#">GetM2MSpkrGainReq</a> for more information</li> </ul>
<i>pSpkrGainResp</i> [OUT]	<ul style="list-style-type: none"> <li>See <a href="#">GetM2MSpkrGainResp</a> for more information</li> </ul>

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 5 seconds

#### 9.21.3.5 ULONG SLQSSetM2MAudioAVCFG ( SetM2MAudioAVCFGReq \* pSetM2MAudioAVCFGReq )

This API sets the AVCFG content.

## Parameters

<i>pSetM2MAudioAVCFGReq</i> [IN]	<ul style="list-style-type: none"> <li>See <a href="#">SetM2MAudioAVCFGReq</a> for more information</li> </ul>
----------------------------------	--

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 5 seconds

#### 9.21.3.6 ULONG SLQSSetM2MAudioLPBK ( SetM2MAudioLPBKReq \* pSetM2MAudioLPBKReq )

This API sets the LPBK content.

## Parameters

<i>pSetM2MAudioLPBKReq</i> [IN]	<ul style="list-style-type: none"> <li>See <a href="#">SetM2MAudioLPBKReq</a> for more information</li> </ul>
---------------------------------	---

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Timeout: 5 seconds

**9.21.3.7 ULONG SLQSSetM2MAudioNVDef ( )**

This API sets the NVDef content.

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Timeout: 5 seconds

**9.21.3.8 ULONG SLQSSetM2MAudioProfile ( SetM2MAudioProfileReq \* pSetM2MAudioProfileReq )**

This API sets the profile content.

**Parameters**

<i>pSetM2MAudioProfileReq</i> [IN]	<ul style="list-style-type: none"><li>• See <a href="#">SetM2MAudioProfileReq</a> for more information</li></ul>
------------------------------------	--

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Timeout: 5 seconds

**9.21.3.9 ULONG SLQSSetM2MAudioVolume ( SetM2MAudioVolumeReq \* pSetM2MAudioVolumeReq )**

This API sets the Volume content.

## Parameters

<i>pSetM2MAudioVolumeReq</i> [IN]	<ul style="list-style-type: none"><li>• See <a href="#">SetM2MAudioVolumeReq</a> for more information</li></ul>
-----------------------------------	---

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 5 seconds

**9.21.3.10   ULONG SLQSSetM2MAVMute ( SetM2MAVMuteReq \* pSetM2MAVMuteReq )**

This API Sets the AV Mute content.

## Parameters

<i>pSetM2MAVMuteReq</i> [IN]	<ul style="list-style-type: none"><li>• See <a href="#">SetM2MAVMuteReq</a> for more information</li></ul>
------------------------------	--

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 5 seconds

**9.21.3.11   ULONG SLQSSetM2MSpkrGain ( SetM2MSpkrGainReq \* pSpkrGainReq )**

This API Sets the SPKRGAIN content.

## Parameters

<i>pSpkrGainReq</i> [IN]	<ul style="list-style-type: none"><li>• See <a href="#">GetM2MSpkrGainReq</a> for more information</li></ul>
--------------------------	--

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Timeout: 5 seconds

**9.22 qaGobiApiSwiOmadms.h File Reference**

SWI Open Mobile Alliance Device Management Service API function prototypes SWI OMA-DM QMI Service revision 1.6.

**Data Structures**

- struct [\\_SLQSOMADMSessionInfo](#)
- struct [\\_SLQSOMADMSettingsReqParams](#)
- struct [\\_SLQSOMADMSettings](#)
- struct [\\_SLQSOMADMSettingsReqParams3](#)

**Typedefs**

- typedef struct [\\_SLQSOMADMSessionInfo](#) SLQSOMADMSessionInfo
- typedef struct [\\_SLQSOMADMSettingsReqParams](#) SLQSOMADMSettingsReqParams
- typedef struct [\\_SLQSOMADMSettings](#) SLQSOMADMSettings
- typedef struct [\\_SLQSOMADMSettingsReqParams3](#) SLQSOMADMSettingsReqParams3

**Functions**

- [ULONG SLQSOMADMStartSession](#) (ULONG sessionType)
- [ULONG SLQSOMADMCancelSession](#) (ULONG session)
- [ULONG SLQSOMADMGetSessionInfo](#) (ULONG \*pSessionType, [SLQSOMADMSessionInfo](#) \*pResp)
- [ULONG SLQSOMADMSendSelection](#) (ULONG selection)
- [ULONG SLQSOMADMGetSettings](#) (ULONG \*pbOMADMEEnabled, ULONG \*pbFOTAdownload, ULONG \*pbFOTAUpdate)
- [ULONG SLQSOMADMSetSettings](#) (ULONG bFOTAdownload, ULONG bFOTAUpdate)
- [ULONG SLQSOMADMSetSettings2](#) ([SLQSOMADMSettingsReqParams](#) \*pSLQSOMADMSettingsReqParams)
- [ULONG SLQSOMADMGetSettings2](#) ([SLQSOMADMSettings](#) \*pSLQSOMADMSettings)
- [ULONG SLQSOMADMStartSession2](#) (ULONG sessionType, ULONG \*pFwAvailability)
- [ULONG SLQSOMADMSendSelection2](#) (ULONG selection, ULONG \*pDeferTime, ULONG \*pRejectReason)
- [ULONG SLQSOMADMSetSettings3](#) ([SLQSOMADMSettingsReqParams3](#) \*pSLQSOMADMSettingsReqParams3)

**9.22.1 Detailed Description**

SWI Open Mobile Alliance Device Management Service API function prototypes SWI OMA-DM QMI Service revision 1.6.



## 9.22.2 Typedef Documentation

### 9.22.2.1 typedef struct \_SLQSOMADMSessionInfo SLQSOMADMSessionInfo

Structure containing the OMA DM Session Info returned by the device. Also used as input parameter to specify the size of variable parameters. (ref. notes)

## Parameters

<i>pStatus</i>	<ul style="list-style-type: none"> <li>• 1 Byte parameter indicating status <ul style="list-style-type: none"> <li>– 0x01 - No Firmware available</li> <li>– 0x02 - Query Firmware Download</li> <li>– 0x03 - Firmware Downloading</li> <li>– 0x04 - Firmware Downloaded</li> <li>– 0x05 - Query Firmware Update</li> <li>– 0x06 - Firmware Updating</li> <li>– 0x07 - Firmware Updated</li> </ul> </li> </ul>
<i>pUpdate-CompleteStatus</i>	<ul style="list-style-type: none"> <li>• 2 byte parameter indicating Update Complete Status <ul style="list-style-type: none"> <li>– See <a href="#">qaGobiApiTableSwiOMADMUpdateCompleteStatus.h</a> Update Complete Status</li> </ul> </li> </ul>
<i>pSeverity</i>	<ul style="list-style-type: none"> <li>• 1 byte parameter indicating severity <ul style="list-style-type: none"> <li>– 0x01 - Mandatory</li> <li>– 0x02 - Optional</li> </ul> </li> </ul>
<i>pSourceLength</i>	<ul style="list-style-type: none"> <li>• 2 byte parameter indicating Length of Vendor Name String in Bytes.</li> </ul>
<i>pSource</i>	<ul style="list-style-type: none"> <li>• Variable length parameter indicating Vendor Name in ASCII</li> </ul>
<i>pPkgName-Length</i>	<ul style="list-style-type: none"> <li>• 2 byte parameter indicating Length of Package Name String in Bytes.</li> </ul>
<i>pPkgName</i>	<ul style="list-style-type: none"> <li>• Variable length parameter indicating Package Name in ASCII</li> </ul>
<i>pPkgDesc-Length</i>	<ul style="list-style-type: none"> <li>• 2 byte parameter indicating Length of Package Description String in Bytes.</li> </ul>
<i>pPkgDescription</i>	<ul style="list-style-type: none"> <li>• Variable length parameter indicating Package Description in ASCII</li> </ul>
<i>pDateLength</i>	<ul style="list-style-type: none"> <li>• 2 byte parameter indicating Length of Package Description String in Bytes.</li> </ul>
<i>pDate</i>	<ul style="list-style-type: none"> <li>• Variable length parameter indicating Package Description in ASCII</li> </ul>

<i>pTimeLength</i>	<ul style="list-style-type: none"> <li>• 2 byte parameter indicating Length of Time String in Bytes.</li> </ul>
<i>pTime</i>	<ul style="list-style-type: none"> <li>• Variable length parameter indicating Time String in ASCII</li> </ul>
<i>pSessionType</i>	<ul style="list-style-type: none"> <li>• 1 byte parameter reflects the last session started for Sprint <ul style="list-style-type: none"> <li>– 0x00 - No session since boot</li> <li>– 0x01 - Sprint CI-DC Session</li> <li>– 0x02 - Sprint CI-PRL Session</li> <li>– 0x03 - Sprint CI-FUMO Session</li> <li>– 0x04 - Sprint HFA-DC Session</li> <li>– 0x05 - Sprint HFA-PRL Session</li> <li>– 0x06 - Sprint HFA-FUMO Session</li> <li>– 0x07 - Sprint NI Session</li> </ul> </li> </ul>
<i>pSessionState</i>	<ul style="list-style-type: none"> <li>• 1 byte parameter indicating session state <ul style="list-style-type: none"> <li>– 0x01 - idle</li> <li>– 0x02 - active</li> <li>– 0x03 - pending</li> </ul> </li> </ul>
<i>pRetryCount</i>	<ul style="list-style-type: none"> <li>• 1 byte parameter indicating retries left count <ul style="list-style-type: none"> <li>– valid values 0 to 6</li> </ul> </li> </ul>

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

As input parameter the members pSourceLength, pPkgNameLength, pPkgDescLength, pDateLength, pTimeLength have to be specified. These should contain the initialized size of pSource, pPkgName, pPkgDescription, pDate, pTime respectively.

**9.22.2.2 typedef struct \_SLQSOMADMSettings SLQSOMADMSettings**

Structure containing the OMA DM settings retrieved from the device

## Parameters

<i>pOMADM-Enabled[OUT]</i>	<ul style="list-style-type: none"> <li>Optional 4 byte parameter indicating OMADM service enabled <ul style="list-style-type: none"> <li>0x00000001 - Client-initiated device configuration</li> <li>0x00000002 - Network-initiated device configuration</li> <li>0x00000010 - Client-initiated FUMO</li> <li>0x00000020 - Network-initiated FUMO</li> </ul> </li> <li>function <a href="#">SLQSOMADMGetSettings2()</a> returns a default value 0xFFFFFFFF in case this parameter is not returned by the modem.</li> </ul>
<i>pFOTA-Adownload[OUT]</i>	<ul style="list-style-type: none"> <li>Optional 1 Byte parameter indicating support for FOTA Automatic download <ul style="list-style-type: none"> <li>0x00 - Host permission required before downloading</li> <li>0x01 - Automatically start downloading, no host permission required</li> <li>0x02 - Automatically start downloading, while not roaming</li> <li>0x03 - Automatically reject download</li> <li>0x04 - Automatically reject download with "Enterprise Reject Policy"</li> </ul> </li> <li>function <a href="#">SLQSOMADMGetSettings2()</a> returns a default value 0xFF in case this parameter is not returned by the modem.</li> </ul>
<i>pFOTAUpdate[OUT]</i>	<ul style="list-style-type: none"> <li>Optional 1 byte parameter indicating FOTA Automatic update <ul style="list-style-type: none"> <li>0x00 - User permission required before updating firmware</li> <li>0x01 - No user permission required before updating firmware</li> <li>0x02 - User permission required, auto update on power up</li> </ul> </li> <li>function <a href="#">SLQSOMADMGetSettings2()</a> returns a default value 0xFF in case this parameter is not returned by the modem.</li> </ul>
<i>pAutosdm[OUT]</i>	<ul style="list-style-type: none"> <li>Optional 1 byte parameter indicating OMA Automatic UI Alert Response <ul style="list-style-type: none"> <li>0x00 - Disabled</li> <li>0x01 - Enabled Accept</li> <li>0x02 - Enabled Reject</li> </ul> </li> <li>function <a href="#">SLQSOMADMGetSettings2()</a> returns a default value 0xFF in case this parameter is not returned by the modem.</li> </ul>

<i>pFwAutoCheck[OUT]</i>	<ul style="list-style-type: none"> <li>Optional 1 byte parameter indicating OMA Automatic Check for Firmware Update on Power-Up Response <ul style="list-style-type: none"> <li>0x00 - Disabled</li> <li>0x01 - Enabled</li> </ul> </li> <li>function <a href="#">SLQSOMADMGetSettings2()</a> returns a default value 0xFF in case this parameter is not returned by the modem.</li> </ul>
--------------------------	--

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

### 9.22.2.3 typedef struct \_SLQSOMADMSettingsReqParams SLQSOMADMSettingsReqParams

Structure containing the OMA DM settings to be set on the device

**Parameters**

<i>FOTAdownload</i>	<ul style="list-style-type: none"> <li>1 Byte parameter indicating support for FOTA Automatic download <ul style="list-style-type: none"> <li>0x00 - Firmware auto download FALSE</li> <li>0x01 - Firmware auto download TRUE</li> </ul> </li> </ul>
<i>FOTAUpdate</i>	<ul style="list-style-type: none"> <li>1 byte parameter indicating FOTA Automatic update <ul style="list-style-type: none"> <li>0x00 - Firmware auto update FALSE</li> <li>0x01 - Firmware auto update TRUE</li> </ul> </li> </ul>
<i>pAutosdm[IN]</i>	<ul style="list-style-type: none"> <li>Optional 1 byte parameter indicating OMA Automatic UI Alert Response <ul style="list-style-type: none"> <li>0x00 - Disabled</li> <li>0x01 - Enabled Accept</li> <li>0x02 - Enabled Reject</li> </ul> </li> </ul>

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

9.22.2.4 `typedef struct _SLQSOMADMSettingsReqParams3 SLQSOMADMSettingsReqParams3`

Structure containing the OMA DM settings to be set on the device

## Parameters

<i>FOTAdownload</i>	<ul style="list-style-type: none"> <li>1 Byte parameter indicating support for FOTA Automatic download <ul style="list-style-type: none"> <li>0x00 - Firmware auto download FALSE</li> <li>0x01 - Firmware auto download TRUE</li> </ul> </li> </ul>
<i>FOTAUpdate</i>	<ul style="list-style-type: none"> <li>1 byte parameter indicating FOTA Automatic update <ul style="list-style-type: none"> <li>0x00 - Firmware auto update FALSE</li> <li>0x01 - Firmware auto update TRUE</li> </ul> </li> </ul>
<i>pAutosdm[IN]</i>	<ul style="list-style-type: none"> <li>Optional 1 byte parameter indicating OMA Automatic UI Alert Response <ul style="list-style-type: none"> <li>0x00 - Disabled</li> <li>0x01 - Enabled Accept</li> <li>0x02 - Enabled Reject</li> </ul> </li> </ul>
<i>pFwAutoCheck[IN]</i>	<ul style="list-style-type: none"> <li>Optional 1 byte parameter indicating OMA Automatic Check for Firmware Update on Power-Up Response <ul style="list-style-type: none"> <li>0x00 - Disabled</li> <li>0x01 - Enabled</li> </ul> </li> </ul>

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## 9.22.3 Function Documentation

## 9.22.3.1 ULONG SLQSOMADMCancelSession ( ULONG session )

Cancels an ongoing OMA-DM session.

## Parameters

<i>session[IN]</i>	<ul style="list-style-type: none"> <li>Session <ul style="list-style-type: none"> <li>0x01 - FOTA, to check availability of FW Update</li> <li>0xFF - Cancel any active OMADM session</li> </ul> </li> </ul>
--------------------	--

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Timeout: 20 seconds

**9.22.3.2    ULONG SLQSOMADMGetSessionInfo (    ULONG \* *pSessionType*, SLQSOMADMSessionInfo \* *pResp* )**

Returns information related to the current (or previous if no session is active) OMA-DM session.

**Parameters**

<i>SessionType</i> [IN]	<ul style="list-style-type: none"><li>• Session type<ul style="list-style-type: none"><li>– 0x01 - FOTA</li><li>– 0xFF - Any active OMADM session. If none active, then previous OMADM session</li></ul></li></ul>
<i>pResp</i> [IN/OUT]	<ul style="list-style-type: none"><li>• See <a href="#">SLQSOMADMSessionInfo</a> for more information</li></ul>

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Timeout: 20 seconds

**9.22.3.3    ULONG SLQSOMADMGetSettings (    ULONG \* *pbOMADMEEnabled*, ULONG \* *pbFOTAdownload*, ULONG \* *pbFOTAUpdate* )**

Returns the OMA-DM settings.



## Parameters

<i>pbOMADM-Enabled[OUT]</i>	<ul style="list-style-type: none"> <li>• Device OMADM service enabled <ul style="list-style-type: none"> <li>– 0x00000001 - Client-initiated device configuration</li> <li>– 0x00000002 - Network-initiated device configuration</li> <li>– 0x00000010 - Client-initiated FUMO</li> <li>– 0x00000020 - Network-initiated FUMO</li> </ul> </li> </ul>
<i>pbFOTA-Adownload[OUT]</i>	<ul style="list-style-type: none"> <li>• Firmware AutoDownload <ul style="list-style-type: none"> <li>– 0x00 - Firmware auto download FALSE</li> <li>– 0x01 - Firmware autod ownload TRUE</li> <li>– 0x02 - Automatically start downloading, while not roaming</li> <li>– 0x03 - Automatically reject download</li> <li>– 0x04 - Automatically reject download with “Enterprise Reject Policy”</li> </ul> </li> </ul>
<i>pbFOTA-Update[OUT]</i>	<ul style="list-style-type: none"> <li>• Firmware AutoUpdate <ul style="list-style-type: none"> <li>– 0x00 - Firmware auto update FALSE</li> <li>– 0x01 - Firmware auto update TRUE</li> </ul> </li> </ul>

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 20 seconds

## 9.22.3.4 ULONG SLQSOMADMGetSettings2 ( SLQSOMADMSettings \* pSLQSOMADMSettings )

Retrieves the OMA-DM settings from the device.

## Parameters

<i>SLQSOMADM-SettingsReq-Params</i>	<ul style="list-style-type: none"> <li>• See <a href="#">SLQSOMADMSettings</a> for more information</li> </ul>
-------------------------------------	--

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Timeout: 20 seconds

**9.22.3.5 ULONG SLQSOMADMSendSelection ( ULONG selection )**

Sends the specified OMA-DM selection for the current network initiated session.

**Parameters**

<i>selection</i> [IN]	<ul style="list-style-type: none"> <li>• OMA-DM NIA Selection <ul style="list-style-type: none"> <li>– 0x01 - Accept</li> <li>– 0x02 - Reject</li> <li>– 0x03 - Defer</li> </ul> </li> </ul>
-----------------------	--

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Timeout: 20 seconds

**9.22.3.6 ULONG SLQSOMADMSendSelection2 ( ULONG selection, ULONG \* pDeferTime, ULONG \* pRejectReason )**

Sends the specified OMA-DM selection for the current network initiated session.

**Parameters**

<i>selection</i> [IN]	<ul style="list-style-type: none"> <li>• OMA-DM NIA Selection <ul style="list-style-type: none"> <li>– 0x01 - Accept</li> <li>– 0x02 - Reject</li> <li>– 0x03 - Defer</li> </ul> </li> </ul>
<i>pDeferTime</i> [IN]	<ul style="list-style-type: none"> <li>• Defer time in minutes. A value of 0 will cause the prompt to be resent immediately.</li> <li>• This TLV is mandatory if selection is set to 0x03.</li> </ul>
<i>pRejectReason</i> [IN]	<ul style="list-style-type: none"> <li>• Reject Reason</li> <li>• This TLV is processed if selection is set to 0x02. If it is not present, the reject reason 0 is used as default.</li> </ul>

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Timeout: 20 seconds

### 9.22.3.7 ULONG SLQSOMADMSetSettings ( ULONG bFOTAdownload, ULONG bFOTAUpdate )

Sets the OMA-DM settings requested.

**Parameters**

<i>bFOTAdownload</i> [IN]	<ul style="list-style-type: none"> <li>Firmware Auto Download               <ul style="list-style-type: none"> <li>0x00 - Host permission required before downloading</li> <li>0x01 - Automatically start downloading, no host permission required</li> <li>0x02 - Automatically start downloading, while not roaming</li> <li>0x03 - Automatically reject download</li> <li>0x04 - Automatically reject download with "Enterprise Reject Policy"</li> </ul> </li> </ul>
<i>bFOTAUpdate</i> [IN]	<ul style="list-style-type: none"> <li>Firmware Auto Update               <ul style="list-style-type: none"> <li>0x00 - User permission required before updating firmware</li> <li>0x01 - No user permission required before updating firmware</li> <li>0x02 - User permission required, auto update on power up</li> </ul> </li> </ul>

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Timeout: 20 seconds

### 9.22.3.8 ULONG SLQSOMADMSetSettings2 ( SLQSOMADMSettingsReqParams \* pSLQSOMADMSettingsReqParams )

Sets the settings related to OMADM. These settings are saved on the modem across power cycles.

## Parameters

<i>pSLQSOMADM-SettingsReq-Params[IN]</i>	<ul style="list-style-type: none"> <li>• See <a href="#">SLQSOMADMSettingsReqParams</a> for more information</li> </ul>
--	---

## Note

Timeout: 20 seconds

#### 9.22.3.9 **ULONG** SLQSOMADMSetSettings3 ( **SLQSOMADMSettingsReqParams3** \* *pSLQSOMADMSettingsReqParams3* )

Sets the settings related to OMADM. These settings are saved on the modem across power cycles.

## Parameters

<i>SLQSOMADM-SettingsReq-ParamsExt[IN]</i>	<ul style="list-style-type: none"> <li>• See <a href="#">SLQSOMADMSettingsReqParamsExt</a> for more information</li> </ul>
--	--

## Note

Timeout: 20 seconds

#### 9.22.3.10 **ULONG** SLQSOMADMStartSession ( **ULONG** *sessionType* )

Starts an OMA-DM session.

## Parameters

<i>sessionType[IN]</i>	<ul style="list-style-type: none"> <li>• Session type <ul style="list-style-type: none"> <li>– 0x01 - FOTA, to check availability of FW Update</li> <li>– 0x02 - DM, to check availability of DM Update</li> <li>– 0x03 - PRL, to check availability of PRL Update</li> </ul> </li> </ul>
------------------------	---

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 20 seconds

#### 9.22.3.11 **ULONG** SLQSOMADMStartSession2 ( **ULONG** *sessionType*, **ULONG** \* *pFwAvailability* )

Starts an OMA-DM session.

## Parameters

<i>sessionType</i> [IN]	<ul style="list-style-type: none"> <li>Session type <ul style="list-style-type: none"> <li>0x01 - FOTA, to check availability of FW Update</li> <li>0x02 - DM, to check availability of DM Update</li> <li>0x03 - PRL, to check availability of PRL Update</li> </ul> </li> </ul>
<i>pFwAvailability</i> [OUT]	<ul style="list-style-type: none"> <li>OMA-DM CHECK FW Available <ul style="list-style-type: none"> <li>0x00000001 - FW Available. For CIDC and CIPRL, this value will be returned by the modem. CIDC and CIPRL are asynchronous OMADM sessions.</li> <li>0x00000002 - FW Not Available</li> <li>0x00000003 - FW Check Timed Out</li> </ul> </li> </ul>

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 20 seconds

## 9.23 qaGobiApiTableBandClasses.h File Reference

Network Access Service API Band Classes table.

### 9.23.1 Detailed Description

Network Access Service API Band Classes table.

### 9.23.2 Band Classes (Value - Description)

- 0 - CDMA Band Class 0
- 1 - CDMA Band Class 1
- 3 - CDMA Band Class 3
- 4 - CDMA Band Class 4
- 5 - CDMA Band Class 5
- 6 - CDMA Band Class 6
- 7 - CDMA Band Class 7
- 8 - CDMA Band Class 8

- 9 - CDMA Band Class 9
- 10 - CDMA Band Class 10
- 11 - CDMA Band Class 11
- 12 - CDMA Band Class 12
- 13 - CDMA Band Class 13
- 14 - CDMA Band Class 14
- 15 - CDMA Band Class 15
- 16 - CDMA Band Class 16
- 17 - CDMA Band Class 17
- 18 - CDMA Band Class 18
- 19 - CDMA Band Class 19
- 40 - GSM 450
- 41 - GSM 480
- 42 - GSM 750
- 43 - GSM 850
- 44 - GSM 900 (Extended)
- 45 - GSM 900 (Primary)
- 46 - GSM 900 (Railways)
- 47 - GSM 1800
- 48 - GSM 1900
- 80 - WCDMA 2100
- 81 - WCDMA PCS 1900
- 82 - WCDMA DCS 1800
- 83 - WCDMA 1700 (US)
- 84 - WCDMA 850
- 85 - WCDMA 800
- 86 - WCDMA 2600
- 87 - WCDMA 900
- 88 - WCDMA 1700 (Japan)
- 90 - WCDMA 1500 band (Japan)
- 91 - WCDMA 850 band (Japan)
- < Reserved 89, 92-109 for WCDMA band classes >
- 110 - WLAN US 2400 MHz
- 111 - WLAN JAPAN 2400 MHz
- 112 - WLAN EUROPEAN 2400 MHz
- 113 - WLAN FRANCE 2400 MHz

- 114 - WLAN SPAIN 2400 MHz
- 115 - WLAN US 5000 MHz band
- 116 - WLAN JAPAN 5000 MHz
- 117 - WLAN EUROPEAN 5000 MHz
- 118 - WLAN FRANCE 5000 MHz
- 119 - WLAN SPAIN 5000 MHz

#### 9.23.2.1 LTE Bands

- 120 - FDD UL:1920-1980; DL:2110-2170; E-UTRA Operating Band 1
- 121 - FDD UL:1850-1910; DL:1930-1990; E-UTRA Operating Band 2
- 122 - FDD UL:1710-1785; DL:1805-1880; E-UTRA Operating Band 3
- 123 - FDD UL:1710-1755; DL:2110-2155; E-UTRA Operating Band 4
- 124 - FDD UL: 824- 849; DL: 869- 894; E-UTRA Operating Band 5
- 125 - FDD UL: 830- 840; DL: 875- 885; E-UTRA Operating Band 6
- 126 - FDD UL:2500-2570; DL:2620-2690; E-UTRA Operating Band 7
- 127 - FDD UL: 880- 915; DL: 925- 960; E-UTRA Operating Band 8
- 128 - FDD UL:1749.9-1784.9; DL:1844.9-1879.9; E-UTRA Operating Band 9
- 129 - FDD UL:1710-1770; DL:2110-2170; E-UTRA Operating Band 10
- 130 - FDD UL:1427.9-1452.9; DL:1475.9-1500.9; E-UTRA Operating Band 11
- 131 - FDD UL:698-716; DL:728-746; E-UTRA Operating Band 12
- 132 - FDD UL: 777- 787; DL: 746-756; E-UTRA Operating Band 13
- 133 - FDD UL: 788- 798; DL: 758-768; E-UTRA Operating Band 14
- 134 - FDD UL: 704-716; DL: 734-746; E-UTRA Operating Band 17
- 135 - TDD LTE UL: 1900-1920; DL: 1900-1920; E-UTRA Operating Band 33
- 136 - TDD LTE UL: 2010-2025; DL: 2010-2025; E-UTRA Operating Band 34
- 137 - TDD LTE UL: 1850-1910; DL: 1850-1910; E-UTRA Operating Band 35
- 138 - TDD LTE UL: 1930-1990; DL: 1930-1990; E-UTRA Operating Band 36
- 139 - TDD LTE UL: 1910-1930; DL: 1910-1930; E-UTRA Operating Band 37
- 140 - TDD LTE UL: 2570-2620; DL: 2570-2620; E-UTRA Operating Band 38
- 141 - TDD LTE UL: 1880-1920; DL: 1880-1920; E-UTRA Operating Band 39
- 142 - TDD LTE UL: 2300-2400; DL: 2300-2400; E-UTRA Operating Band 40
- 143 - FDD LTE UL: 815-830; DL: 860-875; E-UTRA Operating Band 18
- 144 - FDD LTE UL: 830-845; DL: 875-890; E-UTRA Operating Band 19
- 145 - FDD LTE UL: 832-862; DL: 791-821; E-UTRA Operating Band 20
- 146 - FDD LTE UL: 1447.9-1462.9; DL: 1495.9-1510.9; E-UTRA Operating Band 21
- 147 - FDD LTE UL: 1626.5-1660.5; DL: 1525-1559; E-UTRA Operating Band 24

- 148 - FDD LTE UL: 1850-19195; DL: 1930-1995; E-UTRA Operating Band 25
- 149 - TDD LTE UL: 2496-2690; DL: 2496-2690; E-UTRA Operating Band 41
- 150 - TDD LTE UL: 3400-3600; DL: 3400-3600; E-UTRA Operating Band 42
- 151 - TDD LTE UL: 3600-3800; DL: 3600-3800; E-UTRA Operating Band 43
- 200 - TD-SCDMA Band A
- 201 - TD-SCDMA Band B
- 202 - TD-SCDMA Band C
- 203 - TD-SCDMA Band D
- 204 - TD-SCDMA Band E
- 205 - TD-SCDMA Band F

Copyright: © 2011 Sierra Wireless, Inc. all rights reserved

## 9.24 qaGobiApiTableCallControlReturnReasons.h File Reference

Call Control Return Reasons table.

### 9.24.1 Detailed Description

Call Control Return Reasons table.

### 9.24.2 Call Control Result Reasons (Value - Name - Description)

- 0x01 - QMI\_VOICE\_REASON\_FWD\_UNCONDITIONAL - Unconditional call forwarding
- 0x02 - QMI\_VOICE\_REASON\_FWD\_MOBILEBUSY - Forward when the mobile is busy
- 0x03 - QMI\_VOICE\_REASON\_FWD\_NOREPLY - Forward when there is no reply
- 0x04 - QMI\_VOICE\_REASON\_FWD\_UNREACHABLE - Forward when the call is unreachable
- 0x05 - QMI\_VOICE\_REASON\_FWD\_ALLFORWARDING - All forwarding
- 0x06 - QMI\_VOICE\_REASON\_FWD\_ALLCONDITIONAL - All conditional forwarding
- 0x07 - QMI\_VOICE\_REASON\_BARR\_ALLOUTGOING - All outgoing
- 0x08 - QMI\_VOICE\_REASON\_BARR\_OUTGOINGINT - Outgoing internal
- 0x09 - QMI\_VOICE\_REASON\_BARR\_OUTGOINGINTEXTOHOM - Outgoing external to home
- 0x0A - QMI\_VOICE\_REASON\_BARR\_ALLINCOMING - All incoming
- 0x0B - QMI\_VOICE\_REASON\_BARR\_INCOMINGROAMING - Roaming incoming
- 0x0C - QMI\_VOICE\_REASON\_BARR\_ALLBARRING - All calls are barred
- 0x0D - QMI\_VOICE\_REASON\_BARR\_ALLOUTGOINGBARRING - All outgoing calls are barred
- 0x0E - QMI\_VOICE\_REASON\_BARR\_ALLINCOMINGBARRING - All incoming calls are barred
- 0x0F - QMI\_VOICE\_REASON\_CALLWAITING - Call waiting
- 0x10 - VOICE\_CC\_SUPS\_RESULT\_REASON\_CLIP - CLIP



- 0x11 - VOICE\_CC\_SUPS\_RESULT\_REASON\_CLIR - CLIR
- 0x12 - VOICE\_CC\_SUPS\_RESULT\_REASON\_COLP - COLP
- 0x13 - VOICE\_CC\_SUPS\_RESULT\_REASON\_COLR - COLR
- 0x14 - VOICE\_CC\_SUPS\_RESULT\_REASON\_CNAP - CNAP
- 0xFF - Not Available

Copyright: © 2012 Sierra Wireless, Inc. all rights reserved

## 9.25 qaGobiApiTableCallEndReasons.h File Reference

Wireless Data Service Call End Reasons.

### 9.25.1 Detailed Description

Wireless Data Service Call End Reasons.

### 9.25.2 Call end reason codes (Code - Reason)

#### 9.25.2.1 Technology-agnostic call end reasons

- 1 - Reason unspecified, check the verbose call end reason
- 2 - Client ended the call
- 3 - Device has no service
- 4 - Call ended abnormally
- 5 - Received release from base station; no reason given
- 6 - Access attempt already in progress; SD2.0 only
- 7 - Access failure for reason other than the above
- 8 - Call rejected because of redirection or handoff
- 9 - Call failed because close is in progress
- 10 - Authentication failed, 3GPP equivalent ESM(EPS Session Management) cause code value 29, User authentication failed
- 11 - Call ended because of internal call end. This error code is returned when data call is brought down due to some unknown error, such as not specific to any RAT
- 12 - Call ended because of internal error. This error code is returned when data call is brought down due to some unspecified internal error, such as NULL pointer
- 13 - Internal unknown cause code

#### 9.25.2.2 EVDO CDMA 1xEV-DO

- 500 - Device is CDMA-locked until power cycle
- 501 - Received intercept from base station; origination only
- 502 - Received reorder from base station; origination only
- 503 - Received release from base station; service option reject
- 504 - Received incoming call from base station
- 505 - Received alert stop from base station; incoming only
- 506 - Received end activation; OTASP call only
- 507 - Max access probes transmitted
- 508 - Concurrent service is not supported by base station
- 509 - No response received from base station
- 510 - Call rejected by the base station; CDMA only
- 511 - Concurrent services requested were not compatible; CDMA only
- 512 - Corresponds to CM CALL ORIG ERR ALREADY IN TC
- 513 - Used if Call manager subsystem is ending a GPS call in favor of a user call
- 514 - Used if Call manager subsystem is ending a SMS call in favor of a user call
- 515 - CDMA Only; Device has no service

#### 9.25.2.3 WCDMA/GSM call end reasons

- 1000 - Call origination request failed; WCDMA/GSM Only
- 1001 - Client rejected the incoming call; WCDMA/GSM Only
- 1002 - Device has no UMTS service; WCDMA/GSM Only
- 1003 - Network ended the call, look in cc call; WCDMA/GSM Only
- 1004 - LLC(Logical Link Control) or SNDSCP(Sub Network Dependent Convergence Protocol) failure
- 1005 - Insufficient resources, 3GPP equivalent ESM(EPS Session Management) cause code value 26, Insufficient resources
- 1006 - Service option temporarily out of order, 3GPP equivalent ESM(EPS Session Management) cause code value 34, Service option temporarily out of order
- 1007 - PTI already used, 3GPP equivalent ESM(EPS Session Management) cause code value 35, PTI(-Procedure Transaction Identity) already in use
- 1008 - Regular PDP context deactivation, 3GPP equivalent ESM(EPS Session Management) cause code value 36, Regular deactivation
- 1009 - Network failure, 3GPP equivalent ESM(EPS Session Management) cause code value 38, Network failure
- 1010 - Reactivation requested, 3GPP equivalent ESM(EPS Session Management) cause code value 39, Reactivation requested
- 1011 - Protocol error, unspecified, 3GPP equivalent ESM(EPS Session Management) cause code value 111, Protocol error, unspecified

- 1012 - Operator determined barring, 3GPP equivalent ESM(EPS Session Management) cause code value 8, Operator Determined Barring
- 1013 - Unknown or missing Access Point Name (APN), 3GPP equivalent ESM(EPS Session Management) cause code value 27, Missing or unknown APN
- 1014 - Unknown PDP address or PDP type, 3GPP equivalent ESM(EPS Session Management) cause code value 28, Unknown PDN type
- 1015 - Activation rejected by GGSN, 3GPP equivalent ESM(EPS Session Management) cause code value 30, Requested rejected by Serving GW or PDN GW
- 1016 - Activation rejected, unspecified, 3GPP equivalent ESM(EPS Session Management) cause code value 31, Request rejected, unspecified
- 1017 - Service option not supported, 3GPP equivalent ESM(EPS Session Management) cause code value 32, Service option not supported
- 1018 - Requested service option not subscribed, 3GPP equivalent ESM(EPS Session Management) cause code value 33, Requested service option not subscribed
- 1019 - EPS Quality of Service (QoS) not accepted, 3GPP equivalent ESM(EPS Session Management) cause code value 37, EPS QoS not accepted
- 1020 - Semantic error in the TFT operation, 3GPP equivalent ESM(EPS Session Management) cause code value 41, Semantic error in the TFT operation
- 1021 - Syntactical error in the TFT operation, 3GPP equivalent ESM(EPS Session Management) cause code value 42, Syntactical error in the TFT operation
- 1022 - Unknown PDP context, 3GPP equivalent ESM(EPS Session Management) cause code value 43, Invalid EPS bearer identity
- 1023 - Semantic errors in packet filter(s), 3GPP equivalent ESM(EPS Session Management) cause code value 44, Semantic errors in packet filter(s)
- 1024 - Syntactical error in packet filter(s), 3GPP equivalent ESM(EPS Session Management) cause code value 45, Syntactical errors in packet filter(s)
- 1025 - PDP context without TFT already activated, 3GPP equivalent ESM(EPS Session Management) cause code value 46, Unused
- 1026 - Invalid transaction identifier value, 3GPP equivalent ESM(EPS Session Management) cause code value 81, Invalid PTI value
- 1027 - Semantically incorrect message, 3GPP equivalent ESM(EPS Session Management) cause code value 95, Semantically incorrect message
- 1028 - Invalid mandatory information, 3GPP equivalent ESM(EPS Session Management) cause code value 96, Invalid mandatory information
- 1029 - Message type non-existent or not implemented, 3GPP equivalent ESM(EPS Session Management) cause code value 97, Message type non-existent or not implemented
- 1030 - Message not compatible with state, 3GPP equivalent ESM(EPS Session Management) cause code value 98, Message type not compatible with the protocol state
- 1031 - Information element nonexistent or not implemented, 3GPP equivalent ESM(EPS Session Management) cause code value 99, Information element non-existent or not implemented
- 1032 - Conditional information element error, 3GPP equivalent ESM(EPS Session Management) cause code value 100, Conditional IE error
- 1033 - Message not compatible with protocol state, 3GPP equivalent ESM(EPS Session Management) cause code value 101, Message not compatible with the protocol state

- 1034 - APN restriction value incompatible with active PDP context, 3GPP equivalent ESM(EPS Session Management) cause code value 112, APN restriction value incompatible with
  - active EPS bearer context
- 1035 - No GPRS context present
- 1036 - Requested feature not supported, 3GPP equivalent ESM(EPS Session Management) cause code value 40, Feature not supported
- 1037 - Illegal MS, 3GPP equivalent EMM(EPS Mobility Management) cause code value 3, Illegal UE (MS)
- 1038 - Illegal ME, 3GPP equivalent EMM(EPS Mobility Management) cause code value 6, Illegal ME. This error code is sent to the MS if the ME used is not acceptable
  - to the network, e.g. blacklisted
- 1039 - GPRS and non GPRS services not allowed
- 1040 - GPRS services not allowed
- 1041 - MS identity not derived by the network, 3GPP equivalent EMM(EPS Mobility Management) cause code value 9, UE (MS) Identify cannot be derived by the network
- 1042 - Implicitly detached, 3GPP equivalent EMM(EPS Mobility Management) cause code value 10, Implicitly Detached
- 1043 - PLMN not allowed, 3GPP equivalent EMM(EPS Mobility Management) cause code value 11, PLMN not allowed
- 1044 - LA not allowed, this cause is sent to the MS if it requests location updating in a location area where the HPLMN determines that the MS, by subscription, is not allowed to operate.
- 1045 - GPRS services not allowed in PLMN
- 1046 - PDP duplicate
- 1047 - UE radio access technology change
- 1048 - app preempted
- 1049 - Congestion, This cause is sent if the service request or LOCATION UPDATING REQUEST message cannot be actioned because of congestion (e.g. congestion of the MSC or SGSN or GGSN or PDN Gateway; no channel; facility busy/congested etc.).
- 1050 - No PDP context activated
- 1051 - Access class DSAC rejection

#### 9.25.2.4 EVDO CDMA 1xEV-DO

- 1500 - Abort connection setup due to the reception of a Connection Deny message with deny code set to either general or network busy.
- 1501 - Abort connection setup due to the reception of a Connection Deny message with deny code set to either billing or authentication failure.
- 1502 - Change HDR system due to redirection or PRL not preferred
- 1503 - Exit HDR due to redirection or PRL not preferred
- 1504 - No HDR session
- 1505 - Used if Call manager is ending an HDR call origination in favor of a GPS fix
- 1506 - Connection setup timeout
- 1507 - Call manager released HDR call so 1x call can continue

#### 9.25.2.5 call end reason type

- 1 - Mobile IP
- 2 - Internal
- 3 - Call Manager defined
- 6 - 3GPP specification defined
- 7 - PPP
- 8 - EHRPD
- 9 - IPv6

#### 9.25.2.6 Mobile IP call end reasons (Type=1)

- 64 - MIP(Mobile IP) FA(Foreign Agent) ERR REASON UNSPECIFIED, this error code is returned when the data call bring up fails in MIP setup phase since foreign agent rejected MIP registration with unspecified reason
- 65 - MIP(Mobile IP) FA(Foreign Agent) ERR ADMINISTRATIVELY PROHIBITED, this error code is returned when the data call bring up fails in MIP setup phase since foreign agent administratively prohibited MIP registration
- 66 - MIP(Mobile IP) FA(Foreign Agent) ERR INSUFFICIENT RESOURCES, this error code is returned when the data call bring up fails in MIP setup phase since foreign agent rejected MIP registration due to insufficient resources
- 67 - MIP(Mobile IP) FA(Foreign Agent) ERR MOBILE NODE AUTHENTICATION FAILURE, this error code is returned when the data call bring up fails in MIP setup phase since foreign agent rejected MIP registration because MN-AAA authenticator was wrong
- 68 - MIP(Mobile IP) FA(Foreign Agent) ERR HA AUTHENTICATION FAILURE, this error code is returned when the data call bring up fails in MIP setup phase since foreign agent rejected MIP registration because of home agent authentication failure
- 69 - MIP(Mobile IP) FA(Foreign Agent) ERR REQUESTED LIFETIME TOO LONG, this error code is returned when the data call bring up fails in MIP setup phase since foreign agent rejected MIP registration because requested lifetime is too long
- 70 - MIP(Mobile IP) FA(Foreign Agent) ERR MALFORMED REQUEST, this error code is returned when the data call bring up fails in MIP setup phase since foreign agent rejected MIP registration due to malformed request
- 71 - MIP(Mobile IP) FA(Foreign Agent) ERR MALFORMED REPLY, this error code is returned when the data call bring up fails in MIP setup phase since foreign agent rejected MIP registration due to malformed reply
- 72 - MIP(Mobile IP) FA(Foreign Agent) ERR ENCAPSULATION UNAVAILABLE, this error code is returned when the data call bring up fails in MIP setup phase since foreign agent rejected MIP registration because requested encapsulation is unavailable
- 73 - MIP(Mobile IP) FA(Foreign Agent) ERR VJHC UNAVAILABLE, this error code is returned when the data call bring up fails in MIP setup phase since foreign agent rejected MIP registration because VJ Header Compression is unavailable
- 74 - MIP(Mobile IP) FA(Foreign Agent) ERR REVERSE TUNNEL UNAVAILABLE, this error code is returned when the data call bring up fails in MIP setup phase since foreign agent rejected MIP registration because reverse tunnel is unavailable
- 75 - MIP(Mobile IP) FA(Foreign Agent) ERR REVERSE TUNNEL IS MANDATORY AND T BIT NOT SET, this error code is returned when the data call bring up fails in MIP setup phase since foreign agent rejected MIP registration because reverse tunnel is mandatory but not requested by device

- 79 - MIP(Mobile IP) FA(Foreign Agent) ERR DELIVERY STYLE NOT SUPPORTED, this error code is returned when the data call bring up fails in MIP setup phase since foreign agent rejected MIP registration because delivery style is not supported
- 97 - MIP(Mobile IP) FA(Foreign Agent) ERR MISSING NAI, this error code is returned when the data call bring up fails in MIP setup phase since foreign agent rejected MIP registration due to missing NAI
- 98 - MIP(Mobile IP) FA(Foreign Agent) ERR MISSING HA, this error code is returned when the data call bring up fails in MIP setup phase since foreign agent rejected MIP registration due to missing Home Agent
- 99 - MIP(Mobile IP) FA(Foreign Agent) ERR MISSING HOME ADDR, this error code is returned when the data call bring up fails in MIP setup phase since foreign agent rejected MIP registration due to missing Home Address
- 104 - MIP(Mobile IP) FA(Foreign Agent) ERR UNKNOWN CHALLENGE, this error code is returned when the data call bring up fails in MIP setup phase since foreign agent rejected MIP registration due to unknown challenge
- 105 - MIP(Mobile IP) FA(Foreign Agent) ERR MISSING CHALLENGE, this error code is returned when the data call bring up fails in MIP setup phase since foreign agent rejected MIP registration due to missing challenge
- 106 - MIP(Mobile IP) FA(Foreign Agent) ERR STALE CHALLENGE, this error code is returned when the data call bring up fails in MIP setup phase since foreign agent rejected MIP registration due to stale challenge
- 128 - MIP(Mobile IP) FA(Home Agent) ERR REASON UNSPECIFIED, this error code is returned when the data call bring up fails in MIP setup phase since home agent rejected MIP registration with unspecified reason
- 129 - MIP(Mobile IP) FA(Home Agent) ERR ADMINISTRATIVELY PROHIBITED, this error code is returned when the data call bring up fails in MIP setup phase since home agent administratively prohibited MIP registration
- 130 - MIP(Mobile IP) FA(Home Agent) ERR INSUFFICIENT RESOURCES, this error code is returned when the data call bring up fails in MIP setup phase since home agent rejected MIP registration due to insufficient resources
- 131 - MIP(Mobile IP) FA(Home Agent) ERR MOBILE NODE AUTHENTICATION FAILURE, this error code is returned when the data call bring up fails in MIP setup phase since home agent fails authentication because MN-HA authenticator was wrong
- 132 - MIP(Mobile IP) FA(Home Agent) ERR FA AUTHENTICATION FAILURE, this error code is returned when the data call bring up fails in MIP setup phase since home agent rejected MIP registration due to FA authentication failure
- 133 - MIP(Mobile IP) FA(Home Agent) ERR REGISTRATION ID MISMATCH, this error code is returned when the data call bring up fails in MIP setup phase since home agent rejected MIP registration due to registration id mismatch
- 134 - MIP(Mobile IP) FA(Home Agent) ERR MALFORMED REQUEST, this error code is returned when the data call bring up fails in MIP setup phase since home agent rejected MIP registration due to malformed request
- 136 - MIP(Mobile IP) FA(Home Agent) ERR UNKNOWN HA ADDR, this error code is returned when the data call bring up fails in MIP setup phase since home agent rejected MIP registration due to unknown Home Agent address. This code is returned by a home agent when the mobile node is performing dynamic home agent address resolution as described in RFC 3220 (IP Mobility Support for IPV4) Sections 3.6.1.1 and 3.6.1.2
- 137 - MIP(Mobile IP) FA(Home Agent) ERR REVERSE TUNNEL UNAVAILABLE, this error code is returned when the data call bring up fails in MIP setup phase since home agent rejected MIP registration because reverse tunnel is unavailable
- 138 - MIP(Mobile IP) FA(Home Agent) ERR REVERSE TUNNEL IS MANDATORY AND T BIT NOT SET, this error code is returned when the data call bring up fails in MIP setup phase since home agent rejected MIP registration because reverse tunnel is mandatory but not requested by device

- 139 - MIP(Mobile IP) FA(Home Agent) ERR ENCAPSULATION UNAVAILABLE, this error code is returned when the data call bring up fails in MIP setup phase since home agent rejected MIP registration due to encapsulation unavailable
- 65536 - MIP ERR REASON UNKNOWN

#### 9.25.2.7 Internal call end reasons (Type=2)

- 200 - INTERNAL MIN, internal error table offset value, no meaningful message to the error.
- 201 - INTERNAL ERROR, this error code is returned when data call is brought down due to some unspecified internal error
- 202 - CALL ENDED
- 203 - INTERNAL UNKNOWN CAUSE CODE, this error code is returned when data call is brought down due to some unknown error
- 204 - UNKNOWN CAUSE CODE, this error code is returned when data call is brought down due to some unknown error
- 205 - CLOSE IN PROGRESS
- 206 - NETWORK INITIATED TERMINATION
- 207 - APP PREEMPTED
- 208 - ERR PDN IPV4 CALL DISALLOWED, this error code is returned when V4 PDN is in throttled state due to network providing only V6 address during the previous VSNCP bring up (subs\_limited\_to\_v6). The time for which the IPv4 PDN is throttled is determined by the IPv4 throttling timers maintained in the profile
- 209 - ERR PDN IPV4 CALL THROTTLED, this error code is returned when V4 PDN is in throttled state due to previous VSNCP bring up failure(s). The time for which the IPv4 PDN is throttled is determined by the IPv4 throttling timers maintained in the profile
- 210 - ERR PDN IPV6 CALL DISALLOWED, this error code is returned when V6 PDN is in throttled state due to network providing only V4 address during the previous VSNCP bring up (subs\_limited\_to\_v4). The time for which the IPv6 PDN is throttled is determined by the IPv6 throttling timers maintained in the profile
- 211 - ERR PDN IPV6 CALL THROTTLED, this error code is returned when V6 PDN is in throttled state due to previous VSNCP bring up failure(s). The time for which the IPv6 PDN is throttled is determined by the IPv6 throttling timers maintained in the profile
- 212 - MODEM RESTART
- 213 - PDP PPP NOT SUPPORTED
- 214 - UNPREFERRED RAT, this error code is returned when data call is brought down since the RAT on which the data call is attempted/connected is no longer the preferred RAT
- 215 - PHYS LINK CLOSE IN PROGRESS, this error code is returned when data call bring up is rejected because physical link is in the process of cleanup
- 216 - APN PENDING HANDOVER, this error code is returned when interface bring up is attempted for an APN that is yet to be handed over to target RAT
- 217 - PROFILE BEARER INCOMPATIBLE
- 218 - MMGSDI CARD EVT, this error code is returned when data call is brought down because card got refreshed/removed
- 219 - LPM OR PWR DOWN, this error code is returned when data call is brought down because device is going into lower power mode or powering down
- 220 - APN DISABLED, this error code is returned when APN is disabled in card

- 221 - MPIT EXPIRED, this error code is returned when data call is brought down because maximum PPP inactivity timer expired
- 222 - IPV6 ADDR TRANSFER FAILED
- 223 - TRAT SWAP FAILED
- 224 - EHRPD TO HRPD FALLBACK, this error code is returned when data call is brought down because device falls back from eHRPD to HRPD (not because of OOS on eHRPD but due to operator/spec driven eHRPD to HRPD fallback requirements)
- 225 - MANDATORY APN DISABLED, this error code is returned when any mandatory APN is disabled, and MinApnList Disallow call config item is set to TRUE in device
- 226 - MIP CONFIG FAILURE, this error code is returned when UE is in MIP Only config (QCMIP=2) but MIP config fails on call bring up due to incorrect provisioning

#### 9.25.2.8 Call Manager defined call end reasons (Type=3)

- 500 - CDMA LOCK, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) due to device in CDMA locked state
- 501 - INTERCEPT, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) since it received an intercept order from the base station
- 502 - REORDER, this error code is returned when data call is brought down because traffic channel request got rejected by CM(Call Manager) due to receiving a reorder from base station
- 503 - REL SO REJ, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) due to receiving a release from base station with reason: SO Reject
- 504 - INCOM CALL, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) since it received an incoming call from base station
- 505 - ALERT STOP, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) due to RL/FL fade (or) receiving call release from base stations
- 506 - ACTIVATION, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) due to channel acquisition failures. This indicates that device has failed acquiring all the channels in the PRL
- 507 - MAX ACCESS PROBE, this error code is returned when data call is brought down because traffic channel request got rejected by CM(Call Manager) due maximum access probes transmitted
- 508 - CCS NOT SUPPORTED BY BS, this error code is returned when data call is brought down because traffic channel request got rejected by CM(Call Manager) since concurrent service is not supported by base station
- 509 - NO RESPONSE FROM BS, this error code is returned when data call is brought down because traffic channel request got rejected by CM(Call Manager) since there is no response received from base station
- 510 - REJECTED BY BS, this error code is returned when data call is brought down because traffic channel request got rejected by CM(Call Manager) due to base station rejecting the call
- 511 - INCOMPATIBLE, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) since concurrent services requested were not compatible
- 512 - ALREADY IN TC, this error code is returned when data call is brought down because traffic channel request got rejected by CM(Call Manager) since traffic channel is already up for voice calls
- 513 - USER CAL ORIG DURING GPS
- 514 - USER CAL ORIG DURING SMS, this error code is returned when data call is brought down because traffic channel request got rejected since SMS is ongoing



- 515 - NO CDMA SRV, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) since device does not have CDMA service
- 516 - MC ABORT, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) since MC aborted the origination/conversation
- 517 - PSIST NG, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) due to persistence test failure
- 518 - UIM NOT PRESENT, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) due to RUIM not present
- 519 - RETRY ORDER, this error code is returned when data call is brought down because traffic channel request got rejected by CM(Call Manager) due to receiving a retry order from base station
- 520 - ACCESS BLOCK, this error code is returned when data call is brought down because traffic channel rejected/released due to Access blocked by base station
- 521 - ACCESS BLOCK ALL, this error code is returned when data call is brought down because traffic channel rejected due to Access blocked by the base station for all mobile devices
- 522 - IS707B MAX ACC, this error code is returned when data call is brought down because traffic channel request got rejected by CM(Call Manager) due maximum access probes for IS-707B call
- 523 - THERMAL EMERGENCY, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) to put device in thermal emergency
- 524 - CALL ORIG THROTTLED, this error code is returned when data call is brought down because traffic channel request got rejected by CM(Call Manager) since call origination is throttled by DCTM module
- 525 - USER CALL ORIGINATE DURING VOICE CALL, this error code is returned when data call is brought down because traffic channel got released by CM(Call Manager) in favor of voice call or SMS when concurrent voice and data are not supported
- 1000 - CONFERENCE FAILED
- 1001 - INCOMING REJECTED
- 1002 - NO GATEWAY SERVICE
- 1003 - NO GPRS CONTEXT
- 1004 - ILLEGAL MS, This cause is sent to the MS when the network refuses service to the MS either because an identity of the MS is not acceptable to the network or because the MS does not pass the authentication check, i.e. the SRES received from the MS is different from that generated by the network
- 1005 - ILLEGAL ME, This cause is sent to the MS if the ME used is not acceptable to the network, e.g. blacklisted
- 1006 - GPRS SERVICES AND NON GPRS SERVICES NOT ALLOWED
- 1007 - GPRS SERVICES NOT ALLOWED
- 1008 - MS IDENTITY CANNOT BE DERIVED BY THE NETWORK
- 1009 - IMPLICITLY DETACHED, this error code is sent to the MS either if the network has implicitly detached the MS, e.g. some while after the Mobile reachable timer has expired, or if the GMM context data related to the subscription does not exist in the SGSN e.g. because of a SGSN restart.
- 1010 - PLMN NOT ALLOWED, this error code is sent to the MS if it requests location updating in a PLMN where the MS, by subscription or due to operator determined barring is not allowed to operate
- 1011 - LOCAL AREA NOT ALLOWED
- 1012 - GPRS SERVICES NOT ALLOWED IN THIS PLMN
- 1013 - PDP DUPLICATE

- 1014 - USER EQUIPMENT RADIO ACCESS TECHNOLOGY CHANGE
- 1015 - CONGESTION
- 1016 - NO PDP CONEXT ACTIVATED
- 1017 - ACCESS CLASS DSAC REJECTION
- 1018 - PDP ACTIVATE MAX RETRY FAILED
- 1019 - RAB FAILURE
- 1020 - EPS SERVICE NOT ALLOWED
- 1021 - TRACKING AREA NOT ALLOWED
- 1022 - ROAMING NOT ALLOWED IN THIS TRACKING AREA
- 1023 - NO SUITABLE CELLS IN TRACKING AREA
- 1024 - NOT AUTHORIZED FOR THIS CLOSED SUBSCRIBER GROUP
- 1025 - ESM UNKNOWN EPS BEARER CONTEXT
- 1026 - DRB RELEASED AT RRC
- 1027 - NAS SIG CONN RELEASED
- 1028 - EPS MOBILITY MANAGEMENT DETACHED
- 1029 - EPS MOBILITY MANAGEMENT ATTACH FAILED
- 1030 - EPS MOBILITY MANAGEMENT ATTACH STARTED
- 1031 - LTE NAS SERVICE REQ FAILED
- 1032 - ESM(EPS Session Management) ACTIVE DEDICATED BEARER REACTIVATED BY NW
- 1033 - ESM(EPS Session Management) LOWER LAYER FAILURE
- 1034 - ESM(EPS Session Management) SYNC UP WITH NW
- 1035 - ESM(EPS Session Management) NW ACTIVATED DED BEARER WITH ID OF DEF BEARER
- 1036 - ESM(EPS Session Management) BAD OTA MESSAGE
- 1037 - ESM DS REJECTED THE CALL
- 1038 - ESM(EPS Session Management) CONTEXT TRANSFERRED DUE TO IRAT
- 1039 - DS EXPLICIT DEACT
- 1040 - ESM(EPS Session Management) LOCAL CAUSE NONE
- 1041 - LTE NAS SERVICE REQ FAILED NO THROTTLE
- 1042 - ACL FAILURE, This error code should rarely triggered and reported to the application
- 1043 - LTE NAS SERVICE REQ FAILED DS DISALLOW
- 1044 - EMM(EPS Mobility Management) T3417 EXPIRED
- 1045 - EMM(EPS Mobility Management) T3417 EXT EXPIRED
- 1046 - LRR(LTE Radio Resource Control) UL DATA CNF FAILURE TXN - Light Radio Resource Controller Uplink data confirmation failure
- 1047 - LRR(LTE Radio Resource Control) UL DATA CNF FAILURE HO
- 1048 - LRR(LTE Radio Resource Control) UL DATA CNF FAILURE CONN REL

- 1049 - LRRRC(LTE Radio Resource Control) UL DATA CNF FAILURE RLF
- 1050 - LRRRC(LTE Radio Resource Control) UL DATA CNF FAILURE CTRL NOT CONN
- 1051 - LRRRC(LTE Radio Resource Control) CONN EST FAILURE
- 1052 - LRRRC(LTE Radio Resource Control) CONN EST FAILURE ABORTED
- 1053 - LRRRC(LTE Radio Resource Control) CONN EST FAILURE ACCESS BARRED
- 1054 - LRRRC(LTE Radio Resource Control) CONN EST FAILURE CELL RESEL
- 1055 - LRRRC(LTE Radio Resource Control) CONN EST FAILURE CONFIG FAILURE
- 1056 - LRRRC(LTE Radio Resource Control) CONN EST FAILURE TIMER EXPIRED
- 1057 - LRRRC(LTE Radio Resource Control) CONN EST FAILURE LINK FAILURE
- 1058 - LRRRC(LTE Radio Resource Control) CONN EST FAILURE NOT CAMPED
- 1059 - LRRRC(LTE Radio Resource Control) CONN EST FAILURE SI FAILURE
- 1060 - LRRRC(LTE Radio Resource Control) CONN EST FAILURE CONN REJECT
- 1061 - LRRRC(LTE Radio Resource Control) CONN REL NORMAL
- 1062 - LRRRC(LTE Radio Resource Control) CONN REL RLF
- 1063 - LRRRC(LTE Radio Resource Control) CONN REL CRE FAILURE
- 1064 - LRRRC(LTE Radio Resource Control) CONN REL OOS DURING CRE
- 1065 - LRRRC(LTE Radio Resource Control) CONN REL ABORTED
- 1066 - LRRRC(LTE Radio Resource Control) CONN REL SIB READ ERROR
- 1067 - DETACH WITH REATTACH LTE NW DETACH
- 1068 - DETACH WITH OUT REATTACH LTE NW DETACH
- 1069 - ESM(EPS Session Management) PROC TIME OUT
- 1070 - MESSAGE EXCEED MAX L2 LIMIT
- 1500 - CD GEN OR BUSY, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) due to the reception of a Connection Deny message with a deny code of general or network busy
- 1501 - CD BILL OR AUTH, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) due to the reception of a Connection Deny message with a deny code of billing failure or authentication failure
- 1502 - CHG HDR, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) since there is a change to HDR system due to redirection or PRL not preferred
- 1503 - EXIT HDR, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) since device exited HDR due to redirection or PRL not preferred
- 1504 - HDR NO SESSION, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) since device does not have a HDR session
- 1505 - HDR ORIG DURING GPS FIX, this error code is returned when data call is brought down because traffic channel request got rejected by CM(Call Manager) since it is ending an HDR call origination in favor of a GPS fix
- 1506 - HDR CS TIMEOUT, this error code is returned when data call is brought down because traffic channel request got rejected by CM(Call Manager) since connection setup on HDR system timed out

- 1507 - HDR RELEASED BY CM, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) when it wants to release a HDR call so a 1X call can continue
- 1508 - COLLOC ACQ FAIL, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) when device failed to acquire co-located HDR for origination
- 1509 - OTASP COMMIT IN PROG, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) since an OTASP commit is in progress
- 1510 - NO HYBR HDR SRV, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) since device has no Hybrid HDR service
- 1511 - HDR NO LOCK GRANTED, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) since HDR module could not get the RF lock
- 1512 - HOLD OTHER IN PROG, this error code is returned when data call is brought down by CM(Call Manager) because DBM or SMS is in progress
- 1513 - HDR FADE, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) since HDR module released the call due to fade
- 1514 - HDR ACC FAIL, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) due to HDR system Access Failure
- 2000 - CLIENT END, this error code is returned when client ends the data call
- 2001 - NO SRV, this error code is returned when data call is brought down because traffic channel request got rejected by CM(Call Manager) since device has no service
- 2002 - FADE, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) since device lost the system due to fade
- 2003 - REL NORMAL, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) due to receiving a release from base station with no reason
- 2004 - ACC IN PROG, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) due to Access attempt already in progress
- 2005 - ACC FAIL, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) due to Access Failure
- 2006 - REDIR OR HANDOFF, this error code is returned when data call is brought down because device is in the process of redirecting/handing off to a different target system
- 2500 - OFFLINE, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) since device went offline
- 2501 - EMERGENCY MODE, this error code is returned when data call is brought down because traffic channel request got rejected by CM(Call Manager) since device is operating in Emergency mode
- 2502 - PHONE IN USE, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) since device is in use (e.g voice call)
- 2503 - INVALID MODE, this error code is returned when data call is brought down because traffic channel request got rejected by CM(Call Manager) since the device's operational mode is different from the mode requested in the traffic channel bring up
- 2504 - INVALID SIM STATE, this error code is returned when data call is brought down because traffic channel request got rejected by CM(Call Manager) since the SIM was marked by network as invalid for circuit and/or packet service domain
- 2505 - NO COLLOC HDR, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) since there is no collocated HDR
- 2506 - CALL CONTROL REJECTED, this error code is returned when data call is brought down because traffic channel request got rejected by CM(Call Manager) since Call control module rejected the request

## 9.25.2.9 3GPP specification defined call end reasons (Type=6)

- 8 - OPERATOR DETERMINED BARRING, this reason code is posted by the MME(Mobility Management Entity) to indicate operator has barred the UE
- 25 - LLC SNDSCP FAILURE, PDP context deactivation initiated by the MS or by the Network
- 26 - INSUFFICIENT RESOURCES, this reason is posted to indicate that the network cannot provide the requested service due to insufficient resources
- 27 - MISSING OR UNKNOWN APN, the APN was required and not specified or APN could not be resolved. In LTE mode of operation, this is a PDN throttling cause code, meaning the UE may end up throttling further requests to the same APN
- 28 - UNKNOWN PDN TYPE, the reason is posted by the network to indicate that the PDN type was not recognized
- 29 - AUTH FAILED, the reason is posted when authentication fails. In LTE mode of operation, this is a PDN throttling cause code, meaning the UE may end up throttling further requests to the same APN
- 30 - GGSN REJECT, the reason is posted when the request was rejected by Serving GW or PDN GW. In LTE mode of operation, this is a PDN throttling cause code, meaning the UE may end up throttling further requests to the same APN
- 31 - ACTIVATION REJECT, the reason is posted when the request is rejected by the network due to unspecified reasons
- 32 - OPTION NOT SUPPORTED, the reason is posted when UE requested a service not supported by the PLMN
- 33 - OPTION UNSUBSCRIBED, This cause is sent when the MS requests a service option for which it has no subscription
- 34 - OPTION TEMP OOO, service option temporarily out of order, this reason is posted when the network is temporarily out of resources to service the request
- 35 - PTI ALREADY USED, the reason is posted to indicate that PTI (Procedure Transaction Identifier) used in the request is already active via another UE requested procedure
- 36 - REGULAR DEACTIVATION, this reason is posted by the network to initiate a regular release of bearer resources
- 37 - EPS QOS NOT ACCEPTED, this reason is posted by the network to indicate that the QoS requested by the UE could not be accepted
- 38 - NETWORK FAILURE, this reason is posted when an error occurs in the network
- 39 - UMTS REACTIVATION REQ, this reason is posted by the network to request for bearer reactivation. This code may be posted during network congestion
- 40 - FEATURE NOT SUPPORTED, Unsuccessful MBMS context activation requested by the network
- 41 - TFT SEMANTIC ERROR, the reason is posted by the network to indicate semantic error(s) in specifying TFT operation included in the request
- 42 - TFT SYNTAX ERROR, the reason is posted by the network to indicate syntactic error(s) in specifying TFT operation included in the request
- 43 - UNKNOWN PDP CONTEXT, the reason is posted when the bearer identity (or linked bearer identity) in the request is invalid (or inactive)
- 44 - FILTER SEMANTIC ERROR, the reason is posted by the network to indicate semantic error(s) in specifying packet filter(s) associated with a TFT
- 45 - FILTER SYNTAX ERROR, the reason is posted by the network to indicate syntactic error(s) in specifying packet filter(s) associated with a TFT

- 46 - PDP WITHOUT ACTIVE TFT, the reason is posted by the network when UW requested more than one PDP connection without TFT
- 50 - IPV4 ONLY ALLOWED, 3GPP equivalent ESM(EPS Session Management) cause code value 50, PDN type IPv4 only allowed.
- 51 - IPV6 ONLY ALLOWED, 3GPP equivalent ESM(EPS Session Management) cause code value 51, PDN type IPv6 only allowed
- 52 - SINGLE ADDRESS BEARER ONLY, 3GPP equivalent ESM(EPS Session Management) cause code value 52, Single address bearers only allowed. The reason is posted when the network supports single address bearers only, meaning dual IP bearers are not supported
- 53 - ESM INFORMATION NOT RECEIVED, 3GPP equivalent ESM(EPS Session Management) cause code value 53, ESM information not received. The reason is posted by the network to indicate that the PDN connection request was rejected because ESM information was not received
- 54 - PND CONNECTION DOES NOT EXIST, 3GPP equivalent ESM(EPS Session Management) cause code value 54, PDN connection does not exist The reason is posted by the network during handover from a non-3G-PP network to indicate that the MME does not have any information regarding the requested PDN connection
- 55 - MULTIPLE CONNECTION TO SAME PDN NOT ALLOWED, 3GPP equivalent ESM(EPS Session Management) cause code value 55, Multiple PDN connections for a given APN not allowed. The reason is posted by the network to indicate that the UE is already connected to the requested APN via another PDN/PDN connection
- 81 - INVALID TRANSACTION ID, the reason is posted by the network to indicate that the PTI used in the request is unassigned or reserved
- 95 - MESSAGE INCORRECT SEMANTIC, the reason is posted by the network to indicate receipt of an invalid message
- 96 - INVALID MANDATORY INFO, the reason is posted by the network to indicate receipt of a message with semantic error in a mandatory information element
- 97 - MESSAGE TYPE UNSUPPORTED, the reason is posted by the network to indicate receipt of a message that is either undefined or defined but not implemented by the equipment sending this ESM cause
- 98 - MSG TYPE NONCOMPATIBLE STATE, the reason is posted by the network to indicate receipt of a message type that cannot be handled in the current network protocol state
- 99 - UNKNOWN INFO ELEMENT, the reason is posted by the network to indicate receipt of a message that includes an information element that is either not defined or defined but not implemented by the equipment sending the ESM cause
- 100 - CONDITIONAL IE ERROR, the reason is posted by the network to indicate receipt of a message that includes a syntactically incorrect information element. This message is ignored by the network.
- 101 - MSG AND PROTOCOL STATE UNCOMPATIBLE, the reason is posted by the network to indicate receipt of a message that cannot be handled in the current network protocol state
- 111 - PROTOCOL ERROR, the reason is posted by the network to indicate a protocol error when no other error applies
- 112 - APN TYPE CONFLICT
- 113 - INVALID PROXY-CALL SESSION CONTROL FUNCTION ADDRESS

#### 9.25.2.10 PPP call end reasons (Type=7)

- 1 - TIMEOUT, this error code is returned when the data call bring up fails in PPP setup due to timeout (For e.g: LCP Conf Ack not received from network)
- 2 - AUTH FAILURE, this error code is returned when the data call bring up fails in PPP setup due to authentication failure
- 3 - OPTION MISMATCH, this error code is returned when the data call bring up fails in PPP setup due option mismatch (e.g: Authentication is required, but not negotiated with network during LCP phase)
- 31 - PAP FAILURE, this error code is returned when the data call bring up fails in PPP setup due to PAP failure
- 32 - CHAP FAILURE, this error code is returned when the data call bring up fails in PPP setup due to CHAP failure
- 33 - CLOSE IN PROGRESS, this error code is returned when the data call bring up fails in PPP setup since PPP is in the process of cleaning the previous PPP session
- -1 - UNKNOWN, this error code is unused

#### 9.25.2.11 EHRPD call end reasons (Type=8)

- 1 - SUBS LIMITED TO V4, this error code is returned when the V6 interface bring up fails because network provided only V4 address for the upcoming PDN
- 2 - SUBS LIMITED TO V6, this error code is returned when the V4 interface bring up fails because network provided only V6 address for the upcoming PDN
- 4 - VSNCP(Vendor Specific Network Control Protocol) TIMEOUT, this error code is returned when the data call bring up fails in VSNCP phase due to VSNCP timeout error
- 5 - VSNCP(Vendor Specific Network Control Protocol) FAILURE, this error code is returned when VSNCP configuration failed during call bring up
- 6 - VSNCP(Vendor Specific Network Control Protocol) 3GPP2I GEN ERROR, this error code is returned when the data call bring up fails in VSNCP phase due to general error
- 7 - VSNCP(Vendor Specific Network Control Protocol) 3GPP2I UNAUTH APN, this error code is returned when the data call bring up fails in VSNCP phase since network rejected VSNCP config request with reason requested APN is unauthorized
- 8 - VSNCP(Vendor Specific Network Control Protocol) 3GPP2I PDN LIMIT EXCEED, this error code is returned when the data call bring up fails in VSNCP phase since network rejected VSNCP config request with reason PDN limit exceeded
- 9 - VSNCP(Vendor Specific Network Control Protocol) 3GPP2I NO PDN GW, this error code is returned when the data call bring up fails in VSNCP phase since network rejected VSNCP config request with reason no PDN gateway
- 10 - VSNCP(Vendor Specific Network Control Protocol) 3GPP2I PDN GW UNREACH, this error code is returned when the data call bring up fails in VSNCP phase since network rejected VSNCP config request with reason PDN gateway unreachable
- 11 - VSNCP(Vendor Specific Network Control Protocol) 3GPP2I PDN GW REJ, this error code is returned when the data call bring up fails in VSNCP phase since network rejected VSNCP config request with reason PDN gateway reject
- 12 - VSNCP(Vendor Specific Network Control Protocol) 3GPP2I INSUFF PARAM, this error code is returned when the data call bring up fails in VSNCP phase since network rejected VSNCP config request with reason insufficient parameter

- 13 - VSNCP(Vendor Specific Network Control Protocol) 3GPP2I RESOURCE UNAVAIL, this error code is returned when the data call bring up fails in VSNCP phase since network rejected VSNCP config request with reason resource unavailable
- 14 - VSNCP(Vendor Specific Network Control Protocol) 3GPP2I ADMIN PROHIBIT, this error code is returned when the data call bring up fails in SNCP phase since network rejected VSNCP config request with reason admin prohibited
- 15 - VSNCP(Vendor Specific Network Control Protocol) 3GPP2I PDN ID IN USE, this error code is returned when the data call bring up fails in VSNCP phase because network rejected with reason PDN ID IN USE (or) All existing PDNs are brought down with this end reason because one of the PDN bring up got rejected by network with reason PDN ID IN USE
- 16 - VSNCP(Vendor Specific Network Control Protocol) 3GPP2I SUBSCR LIMITATION, this error code is returned when the data call bring up fails in VSNCP phase since network rejected VSNCP config request with reason subscriber limitation
- 17 - VSNCP(Vendor Specific Network Control Protocol) 3GPP2I PDN EXISTS FOR THIS APN, this error code is returned when the data call bring up fails in VSNCP phase since network rejected VSNCP config request with reason PDN exists for this APN

#### 9.25.2.12 IPV6 call end reasons (Type=9)

- 1 - PREFIX UNAVAILABLE, this error code is returned when V6 data call is brought down because device failed to get the prefix from network
- 2 - IPV6 ERR HRPD IPV6 DISABLED, this error code is returned when V6 data call bring up is rejected because IPV6 is disabled in 1X/HRPD mode
- 3 - IPV6 DISABLED, this error code is returned when IPv6 data call bring up is rejected because NV1896 (IPV6 enable) is disabled

Copyright: © 2011-2013 Sierra Wireless, Inc. all rights reserved

## 9.26 qaGobiApiTableCarrierCodes.h File Reference

Carrier Codes table.

### 9.26.1 Detailed Description

Carrier Codes table.

#### 9.26.2 Carrier Codes (Number - Carrier)

- 0 - no carrier specified
- 1 - Generic
- 2 - Telstra
- 4 - AT&T
- 5 - Verizon
- 11 - Sprint
- 12 - Telefonica



- 101 - Verizon
- 102 - Sprint
- 103 - Alltel
- 104 - Bell Mobility
- 105 - Telus
- 106 - U.S. Cellular
- 107 - Telstra
- 108 - China Unicom
- 109 - Telecom New Zealand
- 110 - SK Telecom
- 111 - Reliance Communications
- 112 - Tata Communications
- 113 - MetroPCS Communications
- 114 - Leap Wireless
- 115 - KDDI
- 116 - Grupo Iusacell
- 117 - China Telecom
- 118 - Open Mobile Handset
- 176 - Rogers
- 177 - NetIndex
- 178 - DNA
- 179 - Big Pond
- 201 - AT&T
- 202 - Vodafone
- 203 - T-Mobile
- 204 - Orange
- 205 - Telefonica
- 206 - Telecom Italia
- 207 - 3
- 208 - O2
- 209 - SFR
- 210 - Swisscom AG
- 211 - China Mobile
- 212 - Telstra
- 213 - Singapore Telecommunications
- 214 - Reliance Telecommunications

- 215 - Bharti Airtel
- 216 - NTT docomo
- 217 - E Mobile
- 218 - Softbank
- 219 - Korea Telecom Freetel
- 220 - SK Telecom
- 221 - Telenor
- 222 - NetCom Norway
- 223 - TeliaSonera
- 224 - América Móvil
- 225 - Brasil Vivo
- 0xFFFFFFFF - Unknown

Copyright: © 2011-2014 Sierra Wireless, Inc. all rights reserved

## 9.27 qaGobiApiTableCodingScheme.h File Reference

Data Coding Scheme.

### Macros

- `#define __GOBI_API_CODING_SCHEME_H__`

### 9.27.1 Detailed Description

Data Coding Scheme.

### 9.27.2 Call Control Result Reasons (Value - Name - Description)

#### 9.27.2.1 Use of bits 3..0

- Language using the GSM 7 bit default alphabet Bits 3..0 indicate the language:
  - 0000 German
  - 0001 English
  - 0010 Italian
  - 0011 French
  - 0100 Spanish
  - 0101 Dutch
  - 0110 Swedish
  - 0111 Danish
  - 1000 Portuguese
  - 1001 Finnish

1010 Norwegian  
1011 Greek  
1100 Turkish  
1101 Hungarian  
1110 Polish  
1111 Language unspecified

### 9.27.3 Coding Group Bits 7..4(0001)

#### 9.27.3.1 use of bits 3..0

- 0000 GSM 7 bit default alphabet; message preceded by language indication.  
The first 3 characters of the message are a two-character representation of the language encoded according to ISO 639 [12], followed by a CR character. The CR character is then followed by 90 characters of text.
- 0001 UCS2; message preceded by language indication  
The message starts with a two GSM 7-bit default alphabet character representation of the language encoded according to ISO 639. This is padded to the octet boundary with two bits set to 0 and then followed by 40 characters of UCS2-encoded message.  
An MS not supporting UCS2 coding will present the two character language identifier followed by improperly interpreted user data.

### 9.27.4 Coding Group Bits 7..4(0010)

#### 9.27.4.1 use of bits 3..0

- 0000 Czech  
0001 Hebrew  
0010 Arabic  
0011 Russian  
0100 Icelandic  
0101..1111 Reserved for other languages using the GSM 7 bit default alphabet, with unspecified handling at the MS

### 9.27.5 Coding Group Bits 7..4(0011)

#### 9.27.5.1 use of bits 3..0

- 0000..1111 Reserved for other languages using the GSM 7 bit default alphabet, with unspecified handling at the MS

### 9.27.6 Coding Group Bits 7..4(01xx)

#### 9.27.6.1 use of bits 3..0

- General Data Coding indication
  - Bits 5..0 indicate the following:
    - Bit 5, if set to 0, indicates the text is uncompressed
    - Bit 5, if set to 1, indicates the text is compressed using the compression algorithm defined in 3GPP TS 23.042
    - Bit 4, if set to 0, indicates that bits 1 to 0 are reserved and have no message class meaning
    - Bit 4, if set to 1, indicates that bits 1 to 0 have a message class meaning: Bit 1 Bit 0 Message Class:
      - 0 0 Class 0
      - 0 1 Class 1 Default meaning: ME-specific.
      - 1 0 Class 2 (U)SIM specific message.
      - 1 1 Class 3 Default meaning: TE-specific (see 3GPP TS 27.005 )
  - Bits 3 and 2 indicate the character set being used, as follows:
    - Bit 3 Bit 2 Character set:
      - 0 0 GSM 7 bit default alphabet 0 1 8 bit data
      - 1 0 UCS2 (16 bit) [10]
      - 1 1 Reserved

### 9.27.7 Coding Group Bits 7..4(1001)

#### 9.27.7.1 Reserved coding groups

- Message with User Data Header (UDH) structure:
  - Bit 1 Bit 0 Message Class:
    - 0 0 Class 0
    - 0 1 Class 1 Default meaning: ME-specific.
    - 1 0 Class 2 (U)SIM specific message.
    - 1 1 Class 3 Default meaning: TE-specific (see 3GPP TS 27.005 [8])
  - Bits 3 and 2 indicate the alphabet being used, as follows:
    - Bit 3 Bit 2 Alphabet:
      - 0 0 GSM 7 bit default alphabet
      - 0 1 8 bit data
      - 1 0 USC2 (16 bit) [10]
      - 1 1 Reserved

### 9.27.8 Coding Group Bits 7..4(1010..1101)

#### 9.27.8.1 Reserved coding groups

### 9.27.9 Coding Group Bits 7..4(1110)

#### 9.27.9.1 Defined by the WAP Forum

### 9.27.10 Coding Group Bits 7..4 (1111)

#### 9.27.10.1 Data coding / message handling

- Bit 3 is reserved, set to 0.

Bit 2 Message coding:

0 GSM 7 bit default alphabet

1 8 bit data

Bit 1 Bit 0 Message Class:

0 0 No message class.

0 1 Class 1 user defined.

1 0 Class 2 user defined.

1 1 Class 3

default meaning: TE specific(3GPP TS 27.005)

Copyright: © 2012 Sierra Wireless, Inc. all rights reserved

#### 9.27.11 Macro Definition Documentation

9.27.11.1 `#define __GOBI_API_CODING_SCHEME_H__`

## 9.28 qaGobiApiTableGpsCapabilityCodes.h File Reference

Position Determination Service API GPS Capability Codes.

### 9.28.1 Detailed Description

Position Determination Service API GPS Capability Codes.

### 9.28.2 GPS capability (Value - Capability)

- 0 - None
- 1 - Standalone
- 2 - Assisted (including XTRA and implying standalone is also supported)
- 3 - Assisted (without XTRA and implying standalone is also supported)
- 0xFFFFFFFF - Unknown

Copyright: © 2011 Sierra Wireless, Inc. all rights reserved

## 9.29 qaGobiApiTablePowerModes.h File Reference

Device Management Service API Power Modes table.

### 9.29.1 Detailed Description

Device Management Service API Power Modes table.

### 9.29.2 Power Modes (Value - Description)

- 0 - Online (default)
- 1 - Low power (airplane) mode
- 2 - Factory test mode
- 3 - Offline
- 4 - Reset
- 5 - Power off
- 6 - Persistent low power (airplane) mode
- 7 - Mode - only low power

Copyright: © 2011 Sierra Wireless, Inc. all rights reserved

## 9.30 qaGobiApiTableRadioInterfaces.h File Reference

Network Access Service API Radio Interfaces table.

### 9.30.1 Detailed Description

Network Access Service API Radio Interfaces table.

### 9.30.2 Radio interface

#### 9.30.2.1 Technology (Value - Radio Interface Technology)

- 0 - No service
- 1 - CDMA 1xRTT
- 2 - CDMA 1xEV-DO
- 3 - AMPS (Unsupported)
- 4 - GSM
- 5 - UMTS
- 6 - WLAN
- 7 - GPS
- 8 - LTE

Copyright: © 2011 Sierra Wireless, Inc. all rights reserved

## 9.31 qaGobiApiTableRegionCodes.h File Reference

Region Codes table.

### 9.31.1 Detailed Description

Region Codes table.

### 9.31.2 Region Codes (Code - Region)

- 0 - North America
- 1 - Latin America
- 2 - Europe
- 3 - Asia
- 4 - Australia
- 5 - Global
- 0xFFFFFFFF - Unknown

Copyright: © 2011 Sierra Wireless, Inc. all rights reserved

## 9.32 qaGobiApiTableServiceOptions.h File Reference

Voice Service Options.

### 9.32.1 Detailed Description

Voice Service Options.

### 9.32.2 Service Option codes (Code - Reason)

#### 9.32.2.1 Description

- 0x0001 - Basic variable rate voice service (8 kbps)
- 0x0002 - Mobile station loopback (8 kbps)
- 0x0003 - Enhanced variable rate voice service (8 kbps)
- 0x0004 - Asynchronous data service (9.6 kbps)
- 0x0005 - Group 3 facsimile (9.6 kbps)
- 0x0006 - Short message service (rate set 1)
- 0x0007 - Packet data service: Internet or ISO Protocol stack (9.6 kbps)
- 0x0008 - Packet data service: CDPD Protocol stack (9.6 kbps)
- 0x0009 - Mobile station loopback (13 kbps)
- 0x000A - transparent service
- 0x000B - III nontransparent service
- 0x000C - Asynchronous data service (14.4 or 9.6 kbps)
- 0x000D - Group 3 facsimile (14.4 or 9.6 kbps)

- 0x000E - Short message service (rate set 2)
- 0x000F - Packet data service: Internet or ISO Protocol stack (14.4 kbps)
- 0x0010 - Packet data service: CDPD Protocol stack (14.4 kbps)
- 0x0011 - High-rate voice service (13 kbps)
- 0x0012 - Over-the-air parameter administration (rate set 1)
- 0x0013 - Over-the-air parameter administration (rate set 2)
- 0x0014 - Group 3 analog facsimile (rate set 1)
- 0x0015 - Group 3 analog facsimile (rate set 2)
- 0x0016 - High-speed packet data service: Internet or ISO Protocol stack (RS1 forward, RS1 reverse)
- 0x0017 - High-speed packet data service: Internet or ISO Protocol stack (RS1 forward, RS2 reverse)
- 0x0018 - High-speed packet data service: Internet or ISO Protocol stack (RS2 forward, RS1 reverse)
- 0x0019 - High-speed packet data service: Internet or ISO Protocol stack (RS2 forward, RS2 reverse)
- 0x001A - High-speed packet data service: CDPD Protocol stack (RS1 forward, RS1 reverse)
- 0x001B - High-speed packet data service: CDPD Protocol stack (RS1 forward, RS2 reverse)
- 0x001C - High-speed packet data service: CDPD Protocol stack (RS2 forward, RS1 reverse)
- 0x001D - High-speed packet data service: CDPD Protocol stack (RS2 forward, RS2 reverse)
- 0x001E - RATE\_SET\_1 Supplemental channel loopback test for rate set 1
- 0x001F - RATE\_SET\_2 Supplemental channel loopback test for rate set 2
- 0x0020 - Test Data Service Option (TDSO)
- 0x0021 - cdma2000 high-speed packet data service, Internet or ISO Protocol stack
- 0x0022 - cdma2000 high-speed packet data service, CDPD Protocol
- 0x0023 - Location services, rate set 1 (9.6 kbps)
- 0x0024 - Location services, rate set 2 (14.4 kbps)
- 0x0025 - ISDN interworking service (64 kbps)
- 0x0026 - GSM voice
- 0x0027 - GSM circuit data
- 0x0028 - GSM packet data
- 0x0029 - GSM short message service
- 0x0036 - Markov Service Option (MSO)
- 0x0037 - Loopback Service Option (LSO)
- 0x0038 - Selectable mode vocoder
- 0x0039 - 32 kbps circuit video conferencing
- 0x003A - CONFERENCING 64 kbps circuit video conferencing
- 0x003B - HRPD packet data service, which when used in paging over the 1X air interface, a page response is not required
- 0x003C - Link Layer Assisted Robust Header Compression (LLA ROHC) - header removal



- 0x003D - LLA ROHC - Header Compression
- 0x003E - Source-controlled Variable-Rate Multimode Wideband (VMR-WB) speech codec rate set 2
- 0x003F - Source-controlled VMR-WB speech codec rate set 1
- 0x0040 - HRPD auxiliary packet data service instance
- 0x0041 - cdma2000/GPRS interworking
- 0x0042 - ISO\_PROTOCOL\_SO\_66 cdma2000 high-speed packet data service, Internet or ISO Protocol stack
- 0x0043 - HRPD packet data IP service where higher layer protocol is IP or ROHC
- 0x0044 - Enhanced variable rate voice service (EVRC-B)
- 0x0045 - HRPD packet data service, which when used in paging over the 1X air interface, a page response is required
- 0x0046 - Enhanced variable rate voice service (EVRC-WB)
- 0x1004 - Asynchronous data service, Revision 1 (9.6 or 14.4 kbps)
- 0x1005 - Group 3 facsimile, Revision 1 (9.6 or 14.4 kbps)
- 0x1007 - Packet data service: Internet or ISO Protocol stack, Revision 1 (9.6 or 14.4 kbps)
- 0x1008 - Packet data service: CDPD Protocol stack, Revision 1 (9.6 or 14.4 kbps)
- 0x7FF8 - Identifies service reference identifier 0
- 0x7FF9 - Identifies service reference identifier 1
- 0x7FFA - Identifies service reference identifier 2
- 0x7FFB - Identifies service reference identifier 3
- 0x7FFC - Identifies service reference identifier 4
- 0x7FFD - Identifies service reference identifier 5
- 0x7FFE - Identifies service reference identifier 6
- 0x7FFF - Identifies service reference identifier 7

Copyright: © 2012 Sierra Wireless, Inc. all rights reserved

## 9.33 qaGobiApiTableSupServiceInfoClasses.h File Reference

Voice Supplementary Service Information Classes.

### 9.33.1 Detailed Description

Voice Supplementary Service Information Classes.

### 9.33.2 Supplementary Service Information Classes (Value - Service Class)

- 0X00 - CLASS\_NONE
- 0X01 - CLASS\_VOICE
- 0X02 - CLASS\_DATA
- 0X04 - CLASS\_FAX
- 0X08 - CLASS\_SMS
- 0X10 - CLASS\_DATACIRCUITSYNC
- 0X20 - CLASS\_DATACIRCUITASYNC
- 0X40 - CLASS\_PACKETACCESS
- 0X80 - CLASS\_PADACCESS

Copyright: © 2012 Sierra Wireless, Inc. all rights reserved

## 9.34 qaGobiApiTableSwiAudio.h File Reference

Swi Audio related tables.

### 9.34.1 Detailed Description

Swi Audio related tables.

### 9.34.2 ACDB Device (Device ID - description)

- 0 - Vehicle HF
- 1 - Handset
- 2 - TTY
- 3 - USB
- 4 - NA

### 9.34.3 Physical Interface (Device ID - description - Interface parameters)

- 0 - PCM - Mode: 0-slave, 1-master, 2-Auxiliary PCM; Rate: 0-8k, 1-16k; Format: 0-linear, 1-u-law, 2-A-law; Padding: 0-disable, 1-enable; Bits-frame: 0-8BPF, 1-16BPF, 2-32BPF, 3-64BPF, 4-128BPF, 5-256BPF;
- 1 - I2S - None
- 2 - Analog(with internal codec) - None
- 3 - USB - None

Copyright: © 2013 Sierra Wireless, Inc. all rights reserved

## 9.35 qaGobiApiTableSwiOMADMUpdateCompleteStatus.h File Reference

Update Complete Status table.

### 9.35.1 Detailed Description

Update Complete Status table.

### 9.35.2 OMA DM Update Complete Status (Update Complete Status - Meaning - Usage)

- 200 - Successful - The request has succeeded
- 250-299 - Successful(vendor specified) - successful operation with vendor specified ResultCode
- 400 - Management Client Error - Management Client error - based on User or Device behavior
- 401 - User Cancelled - User chose not to accept the operation when prompted
- 402 - Corrupted Firmware Update Package - Corrupted firmware update package did not store correctly. Detected for example, by mismatch CRCs between actual and expected
- 403 - Firmware UpdatePackage( Device Mismatch ) - Wrong firmware update package delivered to device based on current device characteristics
- 404 - Failed Firmware Update Package Validation - Failure to positively validate digital signature of firmware update package
- 405 - Firmware Update Package Not acceptable - firmware update package is not acceptable
- 406 - Alternate Download Authentication Failure - authentication required but authentication failure was encountered when downloading firmware update package
- 407 - Alternate Download Request Timeout - client has encountered a timeout when downloading firmware update package
- 408 - Not Implemented - the device does not support the requested operation
- 409 - Undefined Error - indicates failure not defined by any other error code
- 410 - Firmware Update Failed - firmware update operation failed in device
- 411 - Malformed or Bad URL - the URL provided for alternate download is bad
- 412 - Alternate Download Server Unavailable - the alternate download server is unavailable or does not respond
- 450 - Client Error ( OMADM General ) - Vendor defined client error
- 451 - Client Error ( OMADM SyncML ) - Vendor defined client error
- 452 - Client Error ( OMADM Auth ) - Vendor defined client error
- 453 - Client Error ( OMADM Protocol ) - Vendor defined client error
- 454 - Client Error ( OMADM Tree ) - Vendor defined client error
- 455 - Client Error ( OMADM DStore ) - Vendor defined client error
- 456 - Client Error ( OMADM Trigger ) - Vendor defined client error
- 457 - Client Error ( OMADM Fumo ) - Vendor defined client error
- 458 - Client Error ( OMADM Comms ) - Vendor defined client error
- 459 - Client Error ( OMADM Parse ) - Vendor defined client error
- 460 - Client Error ( OMADM TNDS ) - Vendor defined client error
- 461 - Client Error ( OMADM SCM ) - Vendor defined client error
- 462 - Client Error ( OMADM Impl ) - Vendor defined client error

- 463-499 - Client Error ( Vendor Specified ) - client error encountered for operation with vendor specified result code
- 500 - Alternate Download Server Error - Alternate download server error encountered
- 501 - Download fails due to device out of memory - The download fails due to insufficient memory in the device to save the firmware update package
- 502 - Firmware update fails due to device out of memory - The update fails because there isn't sufficient memory to update the device
- 503 - Download fails due to network issues - The download fails due to network/transport level errors
- 550-599 - Alternate Download Server Error (vendor specified)- Alternate download server error encountered for operation with vendor specified result code

Copyright: © 2013 Sierra Wireless, Inc. all rights reserved

## 9.36 qaGobiApiTableVoiceCallEndReasons.h File Reference

Voice Service Call and supplementary services end reasons.

### 9.36.1 Detailed Description

Voice Service Call and supplementary services end reasons.

### 9.36.2 Voice Call and supplementary services end reason codes (Code - Reason)

#### 9.36.2.1 General

- 0 - Phone is offline
- 20 - Phone is CDMA locked until a power cycle; CDMA only
- 21 - Phone has no service, this is for backward compatibility
- 22 - Call has ended abnormally; CDMA only
- 23 - Received intercept from the base station; originating only; CDMA only
- 24 - Received reorder from the base station; originating only; CDMA only
- 25 - Received release from the base station; no reason was given
- 26 - Received release from the base station; SO reject; CDMA only
- 27 - Received incoming call from the base station
- 28 - Received alert stop from the base station; incoming only; CDMA only
- 29 - Client ended the call
- 30 - Received end activation; OTASP call only; CDMA only
- 31 - MC aborted the origination/conversation; CDMA only
- 32 - Maximum access probes were transmitted; CDMA only
- 33 - Persistence test failure; FEATURE\_JCDMA only; CDMA only
- 34 - R-UIM is not present

- 35 - Access attempt is already in progress
- 36 - Access failure for a reason other than the above
- 37 - Received retry order; originating only; IS 2000; CDMA only
- 38 - BYBS Concurrent service is not supported by the base station
- 39 - No response was received from the base station
- 40 - Call was rejected by the base station; CDMA only
- 41 - Concurrent services requested were not compatible; CDMA only
- 42 - Access is blocked by the base station; CDMA only
- 43 - Corresponds to CM\_CALL\_ORIG\_ERR\_ALREADY\_IN\_TC
- 44 - Call is ended because an emergency call is flashed over this call; CDMA only
- 45 - Used if CM is ending a GPS call in preference of a user call
- 46 - Used if CM is ending an SMS call in preference of a user call
- 47 - Used if CM is ending a data call in preference of an emergency call
- 48 - Call was rejected because of a redirection or handoff
- 49 - Access is blocked by the base station for all mobiles; KDDI-specific; CDMA only
- 50 - To support OTASP SPC Error indication
- 51 - Maximum access probes for an IS-707B call; CDMA only
- 52 - Base station reject order
- 53 - Base station retry order
- 54 - Timer T42 is expired
- 55 - Timer T40 is expired
- 56 - Service initialization failure - Traffic Channel Initialization
- 57 - Timer T50m is expired - Traffic Channel Initialization
- 58 - Timer T51m is expired - Traffic Channel Initialization
- 59 - Acknowledgement timeout due to 12 retransmissions
- 60 - Bad forward link or timer T5M is expired
- 61 - Transceiver Resource Manager request failed
- 62 - Timer T41 is expired
- 100 - WCDMA/GSM/TDS only; call end LL cause, Received a reason for ending the call from the lower layer
- 101 - WCDMA/GSM only; Call origination request failed
- 102 - WCDMA/GSM only; client rejected an incoming call
- 103 - WCDMA/GSM only; client rejected a setup indication
- 104 - WCDMA/GSM only; network ended the call
- 105 - WCDMA/GSM only
- 106 - GWM/WCDMA only; phone has no service
- 107 - 1X only; phone has no service
- 108 - Full service is unavailable
- 109 - Indicates resources are not available to handle a new MO/MT PS call

#### 9.36.2.2 service Errors

- 110 - Unknown subscriber
- 111 - Illegal subscriber
- 112 - Bearer service not provisioned
- 113 - Tele service not provisioned
- 114 - Illegal equipment
- 115 - Call barred
- 116 - Illegal ss operation
- 117 - Ss error status
- 118 - Ss not available
- 119 - Ss subscription violation
- 120 - Ss incompatibility
- 121 - Facility not supported
- 122 - Absent subscriber
- 123 - Short term denial
- 124 - Long term denial
- 125 - System failure
- 126 - Data missing
- 127 - Unexpected data value
- 128 - Pwd registration failure
- 129 - Negative pwd check
- 130 - Num of pwd attempts violation
- 131 - Position method failure
- 132 - Unknown alphabet
- 133 - Ussd busy
- 134 - Rejected by user
- 135 - Rejected by network
- 136 - Deflection to served subscriber
- 137 - Special service code
- 138 - Invalid deflected to number
- 139 - Mpty participants exceeded
- 140 - Resources not available

## 9.36.2.3 control cause values

- 141 - Unassigned number
- 142 - No route to destination
- 143 - Channel unacceptable
- 144 - Operator determined barring
- 145 - Normal call clearing
- 146 - User busy sEE [s3, aNNEX h]
- 147 - No user responding sEE [s3, aNNEX h]
- 148 - User alerting no answer
- 149 - Call rejected sEE [s3, aNNEX h]
- 150 - Number changed sEE [s3, aNNEX h]
- 151 - Preemption sEE [s3, aNNEX h]
- 152 - Destination out of order
- 153 - Invalid number format
- 154 - Facility rejected
- 155 - Resp to status enquiry
- 156 - Normal unspecified
- 157 - No circuit or channel available
- 158 - Network out of order
- 159 - Temporary failure
- 160 - Switching equipment congestion
- 161 - Access information discarded
- 162 - Requested circuit or channel not available
- 163 - Resources unavailable or unspecified
- 164 - Qos unavailable
- 165 - Requested facility not subscribed
- 166 - Incoming calls barred within cug
- 167 - Bearer capability not auth
- 168 - Bearer capability unavailable
- 169 - Service option not available
- 170 - Acn limit exceeded
- 171 - Bearer service not implemented
- 172 - Requested facility not implemented
- 173 - Only digital information bearer available
- 174 - Service or option not implemented

- 175 - Invalid transaction identifier
- 176 - USER NOT MEMBER OF CUG
- 177 - Incompatible destination
- 178 - Invalid transit nw selection
- 179 - Semantically incorrect message
- 180 - Invalid mandatory information
- 181 - Message type non implemented
- 182 - Message type not compatible with protocol state
- 183 - Information element non existent
- 184 - Conditional ie error
- 185 - Message not compatible with protocol state
- 186 - Recovery on timer expired
- 187 - Protocol error unspecified
- 188 - Interworking unspecified
- 189 - Outgoing calls barred within cug
- 190 - No cug selection
- 191 - Unknown cug index
- 192 - Cug index incompatible
- 193 - Cug call failure unspecified
- 194 - Clir not subscribed
- 195 - Ccbs possible sEE
- 196 - Ccbs not possible

#### 9.36.2.4 reject causes

- 197 - Imsi unknown in hlr
- 198 - Illegal ms
- 199 - Imsi unknown in vlr
- 200 - Imei not accepted
- 201 - Illegal me sEE
- 202 - Plmn not allowed
- 203 - Location area not allowed
- 204 - Roaming not allowed in this location area
- 205 - No suitable cells in location area
- 206 - Network failure sEE
- 207 - mac failure sEE
- 208 - Synch failure



- 209 - Network congestion
- 210 - GSM authentication unacceptable
- 211 - Service not subscribed
- 212 - Service temporarily out of order
- 213 - Call cannot be identified
- 214 - Incorrect semantics in message
- 215 - Mandatory information invalid
- 216 - Call failed due to other access stratum failures
- 217 - SIM is invalid
- 218 - Invalid call state
- 219 - Access class is blocked
- 220 - No resources are in the protocol stack to allow the call
- 221 - Invalid user data was received

#### 9.36.2.5 reject causes

- 222 - Timer T3230 is expired
- 223 - No cell is available
- 224 - Abort message was received
- 225 - Radio link was lost due to other lower layer causes

#### 9.36.2.6 reject causes

- 226 - Timer T303 is expired
- 227 - CNM MM release is pending

#### 9.36.2.7 stratum reject causes

- 228 - Access stratum RR release indication
- 229 - Access stratum random access failure
- 230 - RRC\_REL\_IND Access stratum RRC release indication
- 231 - Access stratum close session indication
- 232 - Access stratum open session failure
- 233 - Access stratum low level failure
- 234 - Access stratum low level failure redial is not allowed
- 235 - Access stratum low level immediate retry
- 236 - Access stratum abort radio is unavailable

#### 9.36.2.8 reject causes

- 237 - Service option is not supported

#### 9.36.2.9 IP end reasons

- 300 - Received SIP 400 bad request;waiting for INVITE response
- 301 - Received SIP 400 bad request;waiting for INVITE response
- 302 - Received SIP 404 not found; call failed; called party does not exist
- 303 - Received SIP 415 unsupported media type; call failed; called party does not support media
- 304 - Received SIP 480 temporarily unavailable; call failed; called party is not in the LTE area
- 305 - No network response; call failed
- 306 - No network response; unable to put call on hold
- 307 - Moved to eHRPD; call failed or dropped; not in the LTE area
- 308 - Upgrade/downgrade rejected (200 OK with the current call SDP)
- 309 - Received 403 call forbidden; waiting for INVITE response
- 310 - Generic timeout; did not receive a response from the server or other end
- 311 - Reported on the MO side for generic internal software errors; user can try again if the call still exists
- 312 - Reported on the MT side if the upgrade timer has been cancelled or cannot complete the request for some reason after notifying the user of a re-invite request
- 313 - Call origination is rejected due to a Service-Specific Access Control (SSAC) barring
- 314 - Phone was put in thermal emergency
- 315 - 1XCSFB call ended because of a soft failure
- 316 - 1XCSFB call ended because of a hard failure

Copyright: © 2012 Sierra Wireless, Inc. all rights reserved

## 9.37 qaGobiApiUim.h File Reference

Uim Service API function prototypes.

### Data Structures

- struct [UIMPowerDownReq](#)
- struct [fileInfo](#)
- struct [UIMRefreshEvent](#)
- struct [appStatus](#)
- struct [slotInfo](#)
- struct [cardStatus](#)
- struct [hotSwapStatus](#)
- struct [UIMGetCardStatusResp](#)
- struct [UIMSessionInformation](#)
- struct [setPINProtection](#)
- struct [UIMSetPinProtectionReq](#)

- struct [remainingRetries](#)
- struct [encryptedPIN1](#)
- struct [UIMPinResp](#)
- struct [verifyUIMPIN](#)
- struct [UIMVerifyPinReq](#)
- struct [changeUIMPIN](#)
- struct [UIMChangePinReq](#)
- struct [unlockUIMPIN](#)
- struct [UIMUnlockPinReq](#)
- struct [UIMEventRegisterReqResp](#)
- struct [UIMRefreshOKReq](#)
- struct [registerRefresh](#)
- struct [UIMRefreshRegisterReq](#)
- struct [UIMRefreshCompleteReq](#)
- struct [UIMRefreshGetLastEventResp](#)
- struct [UIMRefreshGetLastEventReq](#)
- struct [UIMGetFileAttributesReq](#)
- struct [cardResult](#)
- struct [fileAttributes](#)
- struct [UIMGetFileAttributesResp](#)
- struct [depersonalizationInformation](#)
- struct [UIMDepersonalizationReq](#)
- struct [UIMDepersonalizationResp](#)
- struct [authenticationData](#)
- struct [UIMAuthenticateReq](#)
- struct [authenticateResult](#)
- struct [UIMAuthenticateResp](#)
- struct [readResult](#)
- struct [readTransparentInfo](#)
- struct [UIMReadTransparentReq](#)
- struct [UIMReadTransparentResp](#)
- struct [UIMPowerUpReq](#)

## Macros

- `#define MAX_DESCRIPTION_LENGTH 255`
- `#define MAX_CONTENT_LENGTH 1024`
- `#define MAX_NO_OF_APPLICATIONS 10`
- `#define MAX_NO_OF_SLOTS 5`
- `#define MAX_PUK_LENGTH 8`
- `#define MAX_PATH_LENGTH 255`

## Functions

- [ULONG SLQSUIReset \(\)](#)
- [ULONG SLQSUIPowerDown \(UIMPowerDownReq \\*pUIMPowerDownReq\)](#)
- [ULONG SLQSUIGetCardStatus \(UIMGetCardStatusResp \\*pUIMGetCardStatusResp\)](#)
- [ULONG SLQSUISetPinProtection \(UIMSetPinProtectionReq \\*pUIMSetPinProtectionReq, UIMPinResp \\*pUIMSetPinProtectionResp\)](#)
- [ULONG SLQSUIVerifyPin \(UIMVerifyPinReq \\*pUIMVerifyPinReq, UIMPinResp \\*pUIMVerifyPinResp\)](#)
- [ULONG SLQSUIChangePin \(UIMChangePinReq \\*pUIMChangePinReq, UIMPinResp \\*pUIMChangePinResp\)](#)
- [ULONG SLQSUIUnlockPin \(UIMUnlockPinReq \\*pUIMUnlockPinReq, UIMPinResp \\*pUIMUnlockPinResp\)](#)

- [ULONG SLQSUIEventRegister](#) ([UIMEventRegisterReqResp](#) \*pUIMEventRegisterReqResp)
- [ULONG SLQSUIRefreshOK](#) ([UIMRefreshOKReq](#) \*pUIMRefreshOKReq)
- [ULONG SLQSUIRefreshRegister](#) ([UIMRefreshRegisterReq](#) \*pUIMRefreshRegisterReq)
- [ULONG SLQSUIRefreshComplete](#) ([UIMRefreshCompleteReq](#) \*pUIMRefreshCompleteReq)
- [ULONG SLQSUIRefreshGetLastEvent](#) ([UIMRefreshGetLastEventReq](#) \*pUIMRefreshGetLastEventReq, [UIMRefreshGetLastEventResp](#) \*pUIMRefreshGetLastEventResp)
- [ULONG SLQSUIGetFileAttributes](#) ([UIMGetFileAttributesReq](#) \*pUIMGetFileAttributesReq, [UIMGetFileAttributesResp](#) \*pUIMGetFileAttributesResp)
- [ULONG SLQSUIDepersonalization](#) ([UIMDepersonalizationReq](#) \*pUIMDepersonalizationReq, [UIMDepersonalizationResp](#) \*pUIMDepersonalizationResp)
- [ULONG SLQSUIAuthenticate](#) ([UIMAuthenticateReq](#) \*pUIMAuthenticateReq, [UIMAuthenticateResp](#) \*pUIMAuthenticateResp)
- [ULONG SLQSUIReadTransparent](#) ([UIMReadTransparentReq](#) \*pUIMReadTransparentReq, [UIMReadTransparentResp](#) \*pUIMReadTransparentResp)
- [ULONG SLQSUIPowerUp](#) ([UIMPowerUpReq](#) \*pUIMPowerUpReq)

### 9.37.1 Detailed Description

Uim Service API function prototypes.

### 9.37.2 Macro Definition Documentation

9.37.2.1 `#define MAX_CONTENT_LENGTH 1024`

9.37.2.2 `#define MAX_DESCRIPTION_LENGTH 255`

9.37.2.3 `#define MAX_NO_OF_APPLICATIONS 10`

9.37.2.4 `#define MAX_NO_OF_SLOTS 5`

9.37.2.5 `#define MAX_PATH_LENGTH 255`

9.37.2.6 `#define MAX_PUK_LENGTH 8`

### 9.37.3 Function Documentation

9.37.3.1 **ULONG SLQSUIAuthenticate** ( [UIMAuthenticateReq](#) \* *pUIMAuthenticateReq*, [UIMAuthenticateResp](#) \* *pUIMAuthenticateResp* )

This API executes the authentication algorithm on the card.

Parameters

<i>pUIM-Authenticate-Req[IN]</i>	<ul style="list-style-type: none"> <li>• See <a href="#">UIMAuthenticateReq</a> for more information.</li> </ul>
<i>pUIM-Authenticate-Resp[OUT]</i>	<ul style="list-style-type: none"> <li>• See <a href="#">UIMAuthenticateResp</a> for more information.</li> </ul>

Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Timeout: 30 Secs

This API executes a security command on the card that depends on the card type.  
 The response contains the status code received from the card (SW1 and SW2) when the card responded to the read request.  
 The client can pass a token in the request to receive the result in a subsequent SLQSUIMAuthenticateCallback

### 9.37.3.2 ULONG SLQSUIMChangePin ( UIMChangePinReq \* *pUIMChangePinReq*, UIMPinResp \* *pUIMChangePinResp* )

This API changes the value of the specified PIN.

**Parameters**

<i>pUIMChangePinReq</i> [IN]	<ul style="list-style-type: none"> <li>See <a href="#">UIMChangePinReq</a> for more information.</li> </ul>
<i>pUIMChangePinResp</i> [OUT]	<ul style="list-style-type: none"> <li>See <a href="#">UIMPinResp</a> for more information.</li> </ul>

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Timeout: 30 Secs

This API changes the value of the specified PIN.  
 The application must pass both the new and the old values of the PIN to complete the operation  
 The same PIN can be used by multiple sessions (i.e., the PIN is shared between GSM and RUIM in an ICC card).  
 The PIN is automatically set for all the sessions when the API is executed.  
 The client can pass a token in the request to receive the result in a subsequent SLQSUIMChangePinCallback.

### 9.37.3.3 ULONG SLQSUIMDepersonalization ( UIMDepersonalizationReq \* *pUIMDepersonalizationReq*, UIMDepersonalizationResp \* *pUIMDepersonalizationResp* )

This API de-activates or unblocks the personalization on the phone.

## Parameters

<i>pUIM-Depersonilization-Req[IN]</i>	<ul style="list-style-type: none"> <li>See <a href="#">UIMDepersonalizationReq</a> for more information.</li> </ul>
<i>pUIM-Depersonilization-Resp[OUT]</i>	<ul style="list-style-type: none"> <li>See <a href="#">UIMDepersonalizationResp</a> for more information.</li> </ul>

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 30 Secs

This API deactivates or unblocks the personalization on the phone. Each feature can be deactivated/unblocked independently of the other features.

#### 9.37.3.4 ULONG SLQSUIMEventRegister ( UIMEventRegisterReqResp \* *pUIMEventRegisterReqResp* )

This API Registers for event notifications from the card.

## Parameters

<i>pUIMEvent-RegisterReq-Resp[IN/OUT]</i>	<ul style="list-style-type: none"> <li>See <a href="#">UIMEventRegisterReqResp</a> for more information.</li> </ul>
---	---

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 30 Secs

This function registers for event notifications from the card. The client must verify the mask in the response to determine which events were registered successfully. Events not supported correctly are not registered. The client can deregister from all event notifications by passing "0x00000000" bitmask in the request.

#### 9.37.3.5 ULONG SLQSUIGetCardStatus ( UIMGetCardStatusResp \* *pUIMGetCardStatusResp* )

This API retrieves the current status of the card.

## Parameters

<i>pUIMGetCardStatusResp[OUT]</i>	<ul style="list-style-type: none"> <li>See <a href="#">UIMGetCardStatusResp</a> for more information.</li> </ul>
-----------------------------------	--

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 30 Secs

This function retrieves the current status of the card and the status of all applications available on the card. The function also returns support information for the hot-swap feature and the status of the switch used to detect a card removal/insertion.

### 9.37.3.6 ULONG SLQSUIMGetFileAttributes ( UIMGetFileAttributesReq \* pUIMGetFileAttributesReq, UIMGetFileAttributesResp \* pUIMGetFileAttributesResp )

This API retrieves the file attributes for any Elementary File(EF) or Dedicated File(DF) in the card and provides access by the path.

## Parameters

<i>pUIMGetFileAttributesReq[IN]</i>	<ul style="list-style-type: none"> <li>See <a href="#">UIMGetFileAttributesReq</a> for more information.</li> </ul>
<i>pUIMGetFileAttributesResp[OUT]</i>	<ul style="list-style-type: none"> <li>See <a href="#">UIMGetFileAttributesResp</a> for more information.</li> </ul>

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 30 Secs

This API retrieves the file attributes for any Elementary File(EF) or Dedicated File(DF) in the card and provides access by the path. The response contains the status code received from the card (SW1 and SW2) when the card responded to the select request. The client can pass a token in the request to receive the result in a subsequent SLQSUIMGetFileAttributesCallback.

### 9.37.3.7 ULONG SLQSUIMPowerDown ( UIMPowerDownReq \* pUIMPowerDownReq )

This API powers down the SIM card.

## Parameters

<i>pUIMPowerDownReq</i> [IN]	<ul style="list-style-type: none"> <li>• See <a href="#">UIMPowerDownReq</a> for more information.</li> </ul>
------------------------------	---

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 30 Secs

This function powers down the card.  
This is usually performed when the phone is switched off or when it is set to Airplane mode.

### 9.37.3.8 ULONG SLQSUIMPowerUp ( UIMPowerUpReq \* *pUIMPowerUpReq* )

This API powers up the SIM card.

## Parameters

<i>pUIMPowerUpReq</i> [IN]	<ul style="list-style-type: none"> <li>• See <a href="#">UIMPowerUpReq</a> for more information.</li> </ul>
----------------------------	---

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 30 Secs

This function powers up the card.  
This is usually performed when the phone is switched off or when it is set to Airplane mode.

### 9.37.3.9 ULONG SLQSUIMReadTransparent ( UIMReadTransparentReq \* *pUIMReadTransparentReq*, UIMReadTransparentResp \* *pUIMReadTransparentResp* )

This API executes the Read Transparent algorithm on the card.



## Parameters

<i>pUIMRead-Transparent-Req</i> [IN]	<ul style="list-style-type: none"> <li>See <a href="#">UIMReadTransparentReq</a> for more information.</li> </ul>
<i>pUIMRead-Transparent-Resp</i> [OUT]	<ul style="list-style-type: none"> <li>See <a href="#">UIMReadTransparentResp</a> for more information.</li> </ul>

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 30 Secs

This API Provides read access to any transparent file in the card and provides access by the path.  
The response contains the status code received from the card (SW1 and SW2) when the card responded to the read request.  
The client can pass a token in the request to receive the result in a subsequent QMI\_UIM\_READ\_TRANSPARENT\_IND indication.

### 9.37.3.10 ULONG SLQSUIMRefreshComplete ( UIMRefreshCompleteReq \* pUIMRefreshCompleteReq )

This API invoked when the client has finished the Refresh procedure.

## Parameters

<i>pUIMRefresh-CompleteReq</i> [IN]	<ul style="list-style-type: none"> <li>See <a href="#">UIMRefreshCompleteReq</a> for more information.</li> </ul>
-------------------------------------	---

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 30 Secs

This function is invoked when the client has finished the Refresh procedure (has reread all the cached files) and communicates this to the modem. This function enables the terminal response to be sent to the card

### 9.37.3.11 ULONG SLQSUIMRefreshGetLastEvent ( UIMRefreshGetLastEventReq \* pUIMRefreshGetLastEventReq, UIMRefreshGetLastEventResp \* pUIMRefreshGetLastEventResp )

This API provides the ability to retrieve the last refresh event.

## Parameters

<i>pUIMRefresh-GetLastEvent-Req</i> [IN]	<ul style="list-style-type: none"> <li>See <a href="#">UIMRefreshGetLastEventReq</a> for more information.</li> </ul>
<i>pUIMRefresh-GetLastEvent-Resp</i> [OUT]	<ul style="list-style-type: none"> <li>See <a href="#">UIMRefreshGetLastEventResp</a> for more information.</li> </ul>

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 30 Secs

This function provides the ability to retrieve the last refresh event. The event information is usually passed in as an indication from the QMI to the application and is saved by the application at that time. If the event information is not saved, the client can retrieve the last refresh event.  
Details regarding the Refresh procedure (i.e., the stages and actions that an application must complete) are described in document: 80-VM566-1 (NAA Refresh High Level Guide )

### 9.37.3.12 ULONG SLQSUIRefreshOK ( UIMRefreshOKReq \* pUIMRefreshOKReq )

This API Enables the client to indicate whether it is OK to start the Refresh procedure.

## Parameters

<i>pUIMRefreshOKReq</i>	<ul style="list-style-type: none"> <li>Consist of parameters for SLQSUIRefreshOK. Please see /ref <a href="#">UIMRefreshOKReq</a> for details.</li> </ul>
-------------------------	---

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 30 Secs

This function enables the client to indicate whether it is OK to start the Refresh procedure. This command is used only after a refresh event is received, which indicates the need to vote.

### 9.37.3.13 ULONG SLQSUIRefreshRegister ( UIMRefreshRegisterReq \* pUIMRefreshRegisterReq )

This API Registers for file change notifications triggered by the card.

## Parameters

<i>pUIMRefreshRegisterReq</i> [IN]	<ul style="list-style-type: none"> <li>See <a href="#">UIMRefreshRegisterReq</a> for more information.</li> </ul>
------------------------------------	---

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 30 Secs

This function registers for file change notifications triggered by the card. The client can specify a list of files. The client is notified only when one of the files is modified by the Refresh procedure. This function can be invoked multiple times for each session type. If the function is invoked twice with the same session type, the new values overwrite the previous values. The client can also use this function to stop receiving indications of the refresh. This API should be invoked prior to the invocation of the SLQSUIMSetRefreshCallBack for the events to be registered.

## 9.37.3.14 ULONG SLQSUIMReset ( )

This API resets the issuing control points state kept by the service.

## Parameters

<i>None</i>	
-------------	--

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 30 Secs

The list of events to which the client is registered is emptied. The client must register again using the SLQSUIMEventReg API to start receiving the events again. This would mean that the callback registrations would be reset after this API.

## 9.37.3.15 ULONG SLQSUIMSetPinProtection ( UIMSetPinProtectionReq \* pUIMSetPinProtectionReq, UIMPinResp \* pUIMSetPinProtectionResp )

This API enables or disables the protection of the UIM contents by a specific PIN.

## Parameters

<i>pUIMSetPinProtectionReq</i> [IN]	<ul style="list-style-type: none"> <li>See <a href="#">UIMSetPinProtectionReq</a> for more information.</li> </ul>
<i>pUIMSetPinProtectionResp</i> [OUT]	<ul style="list-style-type: none"> <li>See <a href="#">UIMPinResp</a> for more information.</li> </ul>

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 30 Secs

This API enables or disables the protection of UIM contents by a specific PIN.

The same PIN can be used by multiple sessions (i.e., the PIN is shared between GSM and RUIM in an ICC card).

The PIN is automatically set for all the sessions when the API is executed.

The client can pass a token in the request to receive the result in a subsequent SLQSUIMSetPinProtectionCallback indication.

**9.37.3.16** **ULONG SLQSUIMUnblockPin ( UIMUnblockPinReq \* *pUIMUnblockPinReq*, UIMPinResp \* *pUIMUnblockPinResp* )**

This API unblocks a blocked PIN using the PUK code.

## Parameters

<i>pUIMUnblockPinReq</i> [IN]	<ul style="list-style-type: none"> <li>See <a href="#">UIMUnblockPinReq</a> for more information.</li> </ul>
<i>pUIMUnblockPinResp</i> [OUT]	<ul style="list-style-type: none"> <li>See <a href="#">UIMPinResp</a> for more information.</li> </ul>

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Timeout: 30 Secs

This API unblocks a blocked PIN using the PUK code.  
 The client must pass PUK1 to unblock PIN1 or PUK2 to unblock PIN2.  
 The same PIN can be used by multiple sessions (i.e., the PIN is shared between GSM and RUIM in an ICC card).  
 The PIN is automatically set for all the sessions when the API is executed.  
 The client can pass a token in the request to receive the result in a subsequent SLQSUIMUnlockPinCallback.

### 9.37.3.17 ULONG SLQSUIMVerifyPin ( UIMVerifyPinReq \* pUIMVerifyPinReq, UIMPinResp \* pUIMVerifyPinResp )

This API verifies the PIN before the card content is accessed.

**Parameters**

<i>pUIMVerifyPinReq</i> [IN]	<ul style="list-style-type: none"> <li>See <a href="#">UIMVerifyPinReq</a> for more information.</li> </ul>
<i>pUIMVerifyPinResp</i> [OUT]	<ul style="list-style-type: none"> <li>See <a href="#">UIMPinResp</a> for more information.</li> </ul>

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Timeout: 30 Secs

This API verifies the PIN before the card content is accessed.  
 The same PIN can be used by multiple sessions (i.e., the PIN is shared between GSM and RUIM in an ICC card).  
 The PIN is automatically set for all the sessions when the API is executed.  
 The client can pass a token in the request to receive the result in a subsequent SLQSUIMVerifyPinCallback.

## 9.38 qaGobiApiVoice.h File Reference

Voice Service API function prototypes.

**Data Structures**

- struct [USSInfo](#)
- struct [UUSInfo](#)
- struct [CUGInfo](#)
- struct [calledPartySubAdd](#)
- struct [voiceCallRequestParams](#)
- struct [alphaIDInfo](#)

- struct [ccSUPSType](#)
- struct [voiceCallResponseParams](#)
- struct [callFwdTypeAndPlan](#)
- struct [voiceSetSUPSServiceReq](#)
- struct [voiceSetSUPSServiceResp](#)
- struct [airTimer](#)
- struct [roamTimer](#)
- struct [prefVoiceSO](#)
- struct [voiceSetConfigReq](#)
- struct [voiceSetConfigResp](#)
- struct [voiceAnswerCall](#)
- struct [CLIRResp](#)
- struct [voiceGetCLIRResp](#)
- struct [CLIPResp](#)
- struct [voiceGetCLIPResp](#)
- struct [voiceGetCallWaitInfo](#)
- struct [voiceGetCallBarringReq](#)
- struct [voiceGetCallBarringResp](#)
- struct [voiceGetCallFWReq](#)
- struct [callFWInfo](#)
- struct [callFWExtInfo](#)
- struct [getCallFWInfo](#)
- struct [getCallFWExtInfo](#)
- struct [voiceGetCallFWResp](#)
- struct [voiceSetCallBarringPwdInfo](#)
- struct [voiceSetCallBarringPwdResp](#)
- struct [callInfo](#)
- struct [remotePartyNum](#)
- struct [remotePartyName](#)
- struct [connectNumInfo](#)
- struct [diagInfo](#)
- struct [voiceCallInfoReq](#)
- struct [voiceCallInfoResp](#)
- struct [getAllCallInformation](#)
- struct [getAllCallRmtPtyNum](#)
- struct [getAllCallRmtPtyName](#)
- struct [allCallsUUSInfo](#)
- struct [allCallsAlphaIDInfo](#)
- struct [allCallsDiagInfo](#)
- struct [peerNumberInfo](#)
- struct [arrCallInfo](#)
- struct [arrRemotePartyNum](#)
- struct [arrRemotePartyName](#)
- struct [arrAlertingType](#)
- struct [arrUUSInfo](#)
- struct [arrSvcOption](#)
- struct [arrCallEndReason](#)
- struct [arrAlphaID](#)
- struct [arrConnectPartyNum](#)
- struct [arrDiagInfo](#)
- struct [arrCalledPartyNum](#)
- struct [arrRedirPartyNum](#)
- struct [arrAlertingPattern](#)
- struct [voiceGetAllCallInfo](#)
- struct [voiceManageCallsReq](#)

- struct [voiceManageCallsResp](#)
- struct [burstDTMFInfo](#)
- struct [DTMFLengths](#)
- struct [voiceBurstDTMFInfo](#)
- struct [voiceContDTMFInfo](#)
- struct [voiceStopContDTMFInfo](#)
- struct [voiceFlashInfo](#)
- struct [voiceSetPrefPrivacy](#)
- struct [voiceIndicationRegisterInfo](#)
- struct [DTMFInfo](#)
- struct [SUPSInfo](#)
- struct [newPwdData](#)
- struct [COLPResp](#)
- struct [COLRResp](#)
- struct [CNAPResp](#)
- struct [voiceGetConfigReq](#)
- struct [curAMRConfig](#)
- struct [voiceGetConfigResp](#)
- struct [voiceOrigUSSDNoWaitInfo](#)
- struct [voiceBindSubscriptionInfo](#)
- struct [voiceALSSetLineSwitchInfo](#)
- struct [voiceALSSelectLineInfo](#)
- struct [voiceGetCOLPResp](#)
- struct [voiceGetCOLRResp](#)
- struct [voiceGetCNAPResp](#)
- struct [USSResp](#)
- struct [USSDRespFNetwork](#)

## Macros

- #define [MAXUSSDLENGTH](#) 182
- #define [MAX\\_CALL\\_NO\\_LEN](#) 81
- #define [MAX\\_DESCRIPTION\\_LENGTH](#) 255
- #define [PASSWORD\\_LENGTH](#) 4
- #define [MAX\\_NO\\_OF\\_CALLS](#) 20

## Enumerations

- enum [serviceClassInformation](#) {  
    [VOICE\\_SUPS\\_SRV\\_CLASS\\_NONE](#) = 0x00,  
    [VOICE\\_SUPS\\_SRV\\_CLASS\\_VOICE](#) = 0x01,  
    [VOICE\\_SUPS\\_SRV\\_CLASS\\_DATA](#) = 0x02,  
    [VOICE\\_SUPS\\_SRV\\_CLASS\\_FAX](#) = 0x04,  
    [VOICE\\_SUPS\\_SRV\\_CLASS\\_SMS](#) = 0x08,  
    [VOICE\\_SUPS\\_SRV\\_CLASS\\_DATACIRCUITSYNC](#) = 0x10,  
    [VOICE\\_SUPS\\_SRV\\_CLASS\\_DATACIRCUITASYNC](#) = 0x20,  
    [VOICE\\_SUPS\\_SRV\\_CLASS\\_PACKETACCESS](#) = 0x40,  
    [VOICE\\_SUPS\\_SRV\\_CLASS\\_PADACCESS](#) = 0x80 }

## Functions

- [ULONG OriginateUSSD](#) ([BYTE](#) \*pInfo)
- [ULONG AnswerUSSD](#) ([BYTE](#) \*pInfo)
- [ULONG CancelUSSD](#) ()
- [ULONG SLQSVoiceDialCall](#) ([voiceCallRequestParams](#) \*pCallRequestParams, [voiceCallResponseParams](#) \*pCallResponseParams)
- [ULONG SLQSVoiceEndCall](#) ([BYTE](#) \*pCallId)
- [ULONG SLQSVoiceSetSUPSService](#) ([voiceSetSUPSServiceReq](#) \*pVoiceSetSUPSServiceReq, [voiceSetSUPSServiceResp](#) \*pVoiceSetSUPSServiceResp)
- [ULONG SLQSVoiceSetConfig](#) ([voiceSetConfigReq](#) \*pVoiceSetConfigReq, [voiceSetConfigResp](#) \*pVoiceSetConfigResp)
- [ULONG SLQSVoiceAnswerCall](#) ([voiceAnswerCall](#) \*pVoiceAnswerCall)
- [ULONG SLQSVoiceGetCLIR](#) ([voiceGetCLIRResp](#) \*pVoiceGetCLIRResp)
- [ULONG SLQSVoiceGetCLIP](#) ([voiceGetCLIPResp](#) \*pVoiceGetCLIPResp)
- [ULONG SLQSVoiceGetCallWaiting](#) ([voiceGetCallWaitInfo](#) \*pVoiceGetCallWaitInfo)
- [ULONG SLQSVoiceGetCallBarring](#) ([voiceGetCallBarringReq](#) \*pVoiceGetCallBarringReq, [voiceGetCallBarringResp](#) \*pVoiceGetCallBarringResp)
- [ULONG SLQSVoiceGetCallForwardingStatus](#) ([voiceGetCallFWReq](#) \*pVoiceGetCallFWReq, [voiceGetCallFWResp](#) \*pVoiceGetCallFWResp)
- [ULONG SLQSVoiceSetCallBarringPassword](#) ([voiceSetCallBarringPwdInfo](#) \*pVoiceSetCallBarringPwdInfo, [voiceSetCallBarringPwdResp](#) \*pSetCallBarringPwdResp)
- [ULONG SLQSVoiceGetCallInfo](#) ([voiceCallInfoReq](#) \*pGetCallInfoReq, [voiceCallInfoResp](#) \*pGetCallInfoResp)
- [ULONG SLQSVoiceGetAllCallInfo](#) ([voiceGetAllCallInfo](#) \*pGetAllCallInfo)
- [ULONG SLQSVoiceManageCalls](#) ([voiceManageCallsReq](#) \*pVoiceManageCallsReq, [voiceManageCallsResp](#) \*pVoiceManageCallsResp)
- [ULONG SLQSVoiceBurstDTMF](#) ([voiceBurstDTMFInfo](#) \*pBurstDTMFInfo)
- [ULONG SLQSVoiceStartContDTMF](#) ([voiceContDTMFInfo](#) \*pContDTMFInfo)
- [ULONG SLQSVoiceStopContDTMF](#) ([voiceStopContDTMFInfo](#) \*pVoiceStopContDTMFInfo)
- [ULONG SLQSVoiceSendFlash](#) ([voiceFlashInfo](#) \*pFlashInfo)
- [ULONG SLQSVoiceSetPreferredPrivacy](#) ([voiceSetPrefPrivacy](#) \*pSetPrefPrivacy)
- [ULONG SLQSVoiceIndicationRegister](#) ([voiceIndicationRegisterInfo](#) \*pVoiceIndicationRegisterInfo)
- [ULONG SLQSVoiceGetConfig](#) ([voiceGetConfigReq](#) \*pVoiceGetConfigReq, [voiceGetConfigResp](#) \*pVoiceGetConfigResp)
- [ULONG SLQSVoiceOrigUSSDNoWait](#) ([voiceOrigUSSDNoWaitInfo](#) \*pVoiceOrigUSSDNoWaitInfo)
- [ULONG SLQSVoiceBindSubscription](#) ([voiceBindSubscriptionInfo](#) \*pVoiceBindSubscriptionInfo)
- [ULONG SLQSVoiceALSSetLineSwitching](#) ([voiceALSSetLineSwitchInfo](#) \*pVoiceALSSetLineSwitchInfo)
- [ULONG SLQSVoiceALSSelectLine](#) ([voiceALSSelectLineInfo](#) \*pVoiceALSSelectLineInfo)
- [ULONG SLQSVoiceGetCOLP](#) ([voiceGetCOLPResp](#) \*pVoiceGetCOLPResp)
- [ULONG SLQSVoiceGetCOLR](#) ([voiceGetCOLRResp](#) \*pVoiceGetCOLRResp)
- [ULONG SLQSVoiceGetCNAP](#) ([voiceGetCNAPResp](#) \*pVoiceGetCNAPResp)
- [ULONG SLQSVoiceOriginateUSSD](#) (struct [USSInfo](#) \*pReq, struct [USSResp](#) \*pResp)

### 9.38.1 Detailed Description

Voice Service API function prototypes.

### 9.38.2 Macro Definition Documentation

9.38.2.1 `#define MAX_CALL_NO_LEN 81`

9.38.2.2 `#define MAX_DESCRIPTION_LENGTH 255`

9.38.2.3 `#define MAX_NO_OF_CALLS 20`



9.38.2.4 `#define MAXUSSDLENGTH 182`

9.38.2.5 `#define PASSWORD_LENGTH 4`

### 9.38.3 Enumeration Type Documentation

9.38.3.1 `enum serviceClassInformation`

Service Class information

Enumerator

**`VOICE_SUPS_SRV_CLASS_NONE`**  
**`VOICE_SUPS_SRV_CLASS_VOICE`**  
**`VOICE_SUPS_SRV_CLASS_DATA`**  
**`VOICE_SUPS_SRV_CLASS_FAX`**  
**`VOICE_SUPS_SRV_CLASS_SMS`**  
**`VOICE_SUPS_SRV_CLASS_DATACIRCUITSYNC`**  
**`VOICE_SUPS_SRV_CLASS_DATACIRCUITASYNC`**  
**`VOICE_SUPS_SRV_CLASS_PACKETACCESS`**  
**`VOICE_SUPS_SRV_CLASS_PADACCESS`**

### 9.38.4 Function Documentation

9.38.4.1 `ULONG AnswerUSSD ( BYTE * pInfo )`

Responds to a USSD request from the network.

Parameters

<i>pInfo</i> [IN]	<ul style="list-style-type: none"><li>• USS information</li></ul>
-------------------	---

Returns

`eQCWWAN_ERR_NONE` on success, `eQCWWAN_xxx` error value otherwise

See Also

See [qmerrno.h](#) for `eQCWWAN_xxx` error values

Note

Technology Supported: UMTS  
Timeout: 5 mins

9.38.4.2 `ULONG CancelUSSD ( )`

Cancels an in-progress USSD operation.

## Parameters

<i>None</i>
-------------

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Technology Supported: UMTS  
Timeout: 30 Secs

9.38.4.3 ULONG OriginateUSSD ( BYTE \* *pInfo* )

Initiates a USSD operation.

## Parameters

<i>pInfo</i> [IN]	<ul style="list-style-type: none"> <li>• USS information</li> <li>• See <a href="#">USSInfo</a> for more details</li> </ul>
-------------------	---

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Technology Supported: UMTS  
Timeout: 5 mins

9.38.4.4 ULONG SLQSOOriginateUSSD ( struct USSInfo \* *pReq*, struct USSResp \* *pResp* )

Initiates a USSD session.

## Parameters

<i>pReq</i>	[IN]	<ul style="list-style-type: none"> <li>• USS information</li> <li>• See <a href="#">USSInfo</a> for more details</li> </ul>
<i>pResp</i>	[OUT]	<ul style="list-style-type: none"> <li>• USS information</li> </ul>

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Technology Supported: UMTS  
Device Supported: MC83x5  
Timeout: 5 mins

**9.38.4.5 ULONG SLQSVoiceALSSelectLine ( voiceALSSelectLineInfo \* pVoiceALSSelectLineInfo )**

This API allows the user to select the preferred line.

**Parameters**

<i>pVoiceALS- SelectLineInfo</i>	<ul style="list-style-type: none"><li>See <a href="#">voiceALSSelectLineInfo</a> for more information.</li></ul>
--------------------------------------	--

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Technology Supported: UMTS  
Timeout: 30 Secs  
This API allows the user to select the preferred line, and the status is updated on the card. The API is supported only for specific SIM/USIMs that support alternate line service. This command is applicable only in 3GPP devices. A No Effect error is returned if the update on the card fails.

**9.38.4.6 ULONG SLQSVoiceALSSetLineSwitching ( voiceALSSetLineSwitchInfo \* pVoiceALSSetLineSwitchInfo )**

This API sets the line switch setting on the card.

**Parameters**

<i>pVoiceALSSet- LineSwitchInfo</i>	<ul style="list-style-type: none"><li>See <a href="#">voiceALSSetLineSwitchInfo</a> for more information.</li></ul>
---	---

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Technology Supported: UMTS

Timeout: 30 Secs

This API sets a line to be switchable or unswitchable, and the switch status is updated on the card. The API is supported only for specific SIM/USIMs that support alternate line service. This command is applicable only in 3GPP devices. A No Effect error is returned if the update on the card fails.

**9.38.4.7 ULONG SLQSVoiceAnswerCall ( voiceAnswerCall \* pVoiceAnswerCall )**

Answers an incoming voice call.

**Parameters**

<i>pVoiceAnswer- Call</i> [IN/OUT]	<ul style="list-style-type: none"><li>• Pointer to structure of <a href="#">voiceAnswerCall</a><ul style="list-style-type: none"><li>– See <a href="#">voiceAnswerCall</a> for more information</li></ul></li></ul>
--	---

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Timeout: 10 Secs

This API is used to answer an incoming voice call when the incoming voice call is the only call present at that time. If there are other calls while an incoming call (waiting call) is received, API "SLQSVoiceSendFlash" can be used case of 3GPP2(CDMA) and API "SLQSVoiceManageCalls" in the case of 3GPP(UMTS). If the result indicates success, the device has started the requested operation and it does not mean that the call has been answered. "SLQSVoiceSetAllCallStatusCallback" can be subscribed to check the call Information/State.

**9.38.4.8 ULONG SLQSVoiceBindSubscription ( voiceBindSubscriptionInfo \* pVoiceBindSubscriptionInfo )**

This API binds a subscription type to a specific voice client ID.

**Parameters**

<i>pVoiceBind- SubscriptionInfo</i>	<ul style="list-style-type: none"><li>• See <a href="#">voiceBindSubscriptionInfo</a> for more information.</li></ul>
---	---

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

#### See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

#### Note

Timeout: 30 Secs

Some versions of the modem support the Dual SIM feature. With this feature the modem can register with two different cellular networks simultaneously. Each network registration is associated with a different subscription, e.g., phone number, such that the modem appears to the network to be two different users. By default, the Voice client is bound to the primary subscription. This command allows the Voice client to change this binding. After receiving a successful response to this command, all future commands sent by the client will affect the newly bound subscription only.

#### 9.38.4.9 ULONG SLQSVoiceBurstDTMF ( voiceBurstDTMFInfo \* pBurstDTMFInfo )

Sends a burst Dual-Tone Multi frequency (DTMF) (applicable only for 3GPP2)

##### Parameters

<i>pBurstDTMFInfo</i>	<ul style="list-style-type: none"><li>• Structure containing parameters of burst DTMF.</li><li>• See <a href="#">voiceBurstDTMFInfo</a> for more information</li></ul>
-----------------------	--

#### Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

#### See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

#### Note

Technology Supported: CDMA

Timeout: 30 Secs

Applicable only for 3GPP2. This API sends a burst DTMF. If API result indicates success, this means the device has started the requested operation. It does not mean that the burst DTMF request has been sent to the network. A burst DTMF request is sent to the current active/alerting call when CallId is set to 0xFF. This API is applicable only in 3GPP2.

#### 9.38.4.10 ULONG SLQSVoiceDialCall ( voiceCallRequestParams \* pCallRequestParams, voiceCallResponseParams \* pCallResponseParams )

Originates a voice call (MO call).

## Parameters

<i>pCallRequestParams</i> [IN]	<ul style="list-style-type: none"> <li>• Pointer to structure of <a href="#">voiceCallRequestParams</a> <ul style="list-style-type: none"> <li>– See <a href="#">voiceCallRequestParams</a> for more information</li> </ul> </li> </ul>
<i>pCallResponseParams</i> [OUT]	<ul style="list-style-type: none"> <li>• Pointer to structure of <a href="#">voiceCallResponseParams</a> <ul style="list-style-type: none"> <li>– See <a href="#">voiceCallResponseParams</a> for more information</li> </ul> </li> </ul>

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 30 Secs

This API originates a voice call (MO). If the function returns success with a call\_id, the device has started the requested operation. It does not mean that the call has been connected. SLQSVoiceSetAllCallStatus-Callback() callback can be subscribed to learn if the call was successful.

#### 9.38.4.11 ULONG SLQSVoiceEndCall ( BYTE \* pCallId )

This message ends a voice call

## Parameters

<i>pCallId</i>	[IN/OUT]
	<ul style="list-style-type: none"> <li>• Unique call identifier for the call that must be ended</li> </ul>

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 30 Secs

If the function returns success, the device has started the requested operation. It does not mean that the call has been ended. The application should always process the SLQSVoiceSetAllCallStatusCallback() callback to learn if the call was ended.

#### 9.38.4.12 ULONG SLQSVoiceGetAllCallInfo ( voiceGetAllCallInfo \* pGetAllCallInfo )

This API queries the information associated with all the calls originating or terminating from a particular device.

## Parameters

<i>pGetAllCallInfo</i> [OUT]	<ul style="list-style-type: none"> <li>See <a href="#">voiceGetAllCallInfo</a> for more information.</li> </ul>
------------------------------	---

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 30 Secs

This command is used by the control point to get information of all the calls to and fro from the device in progress. The information keeps on updating constantly, as the state of a call changes example, from incoming to conversation to terminated.

This API requires a firmware with at least voice 2.0 support.

#### 9.38.4.13 ULONG SLQSVoiceGetCallBarring ( voiceGetCallBarringReq \* pVoiceGetCallBarringReq, voiceGetCallBarringResp \* pVoiceGetCallBarringResp )

Queries the status of Call Barring Supplementary Service (applicable only for 3GPP).

## Parameters

<i>pVoiceGetCallBarringReq</i> [IN]	<ul style="list-style-type: none"> <li>Pointer to structure of <a href="#">voiceGetCallBarringReq</a> <ul style="list-style-type: none"> <li>See <a href="#">voiceGetCallBarringReq</a> for more information</li> </ul> </li> </ul>
<i>pVoiceGetCallBarringResp</i> [OUT]	<ul style="list-style-type: none"> <li>Pointer to structure of <a href="#">voiceGetCallBarringResp</a> <ul style="list-style-type: none"> <li>See <a href="#">voiceGetCallBarringResp</a> for more information</li> </ul> </li> </ul>

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Technology Supported: UMTS

Timeout: 30 Secs

This API queries the status of the call barring supplementary service, i.e., to find whether the call barring supplementary service is active and, if active, for which service classes it is active. The request is a blocking request, i.e., the response is sent only after confirmation is received from the network. The maximum time it takes for the response to be sent is approximately 30 sec. This API is applicable only in 3GPP devices.

9.38.4.14 **ULONG** SLQSVoiceGetCallForwardingStatus ( **voiceGetCallFWReq** \* *pVoiceGetCallFWReq*,  
**voiceGetCallFWResp** \* *pVoiceGetCallFWResp* )

Queries the status of Call Forwarding Supplementary Service.



## Parameters

<i>pVoiceGetCallFWReq</i> [IN]	<ul style="list-style-type: none"> <li>• Pointer to structure of <a href="#">voiceGetCallFWReq</a> <ul style="list-style-type: none"> <li>– See <a href="#">voiceGetCallFWReq</a> for more information</li> </ul> </li> </ul>
<i>pVoiceGetCallFWResp</i> [OUT]	<ul style="list-style-type: none"> <li>• Pointer to structure of <a href="#">voiceGetCallFWResp</a> <ul style="list-style-type: none"> <li>– See <a href="#">voiceGetCallFWResp</a> for more information</li> </ul> </li> </ul>

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Technology Supported: UMTS

Timeout: 30 Secs

This API queries the status of the call forwarding supplementary service, i.e., to find whether the call forwarding supplementary service is active and, if active, for which service classes and call forwarding number it is active. The request is a blocking request, i.e., the response is sent only after confirmation is received from the network. The maximum time it takes for the response to be sent is approximately 30 sec. This API is applicable only in 3GPP devices.

**9.38.4.15** `ULONG SLQSVoiceGetCallInfo ( voiceCallInfoReq * pGetCallInfoReq, voiceCallInfoResp * pGetCallInfoResp )`

This API queries the information associated with a call and gives information about a particular call whose call Id is sent in as request.

## Parameters

<i>pGetCallInfoReq</i> [IN]	<ul style="list-style-type: none"> <li>• See <a href="#">voiceCallInfoReq</a> for more information.</li> </ul>
<i>pGetCallInfoResp</i> [OUT]	<ul style="list-style-type: none"> <li>• See <a href="#">voiceCallInfoResp</a> for more information.</li> </ul>

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Timeout: 30 Secs

If no voice call is in progress or an invalid `call_id` is sent in the request, an error is returned as the response.

This API requires a firmware with atleast voice 2.0 support.

**9.38.4.16 ULONG SLQSVoiceGetCallWaiting ( voiceGetCallWaitInfo \* pVoiceGetCallWaitInfo )**

Queries the status of Call Waiting Supplementary Service (applicable only for 3GPP).

**Parameters**

<i>pVoiceGetCallWaitInfo[IN/OUT]</i>	<ul style="list-style-type: none"> <li>• Pointer to structure of <a href="#">voiceGetCallWaitInfo</a> <ul style="list-style-type: none"> <li>– See <a href="#">voiceGetCallWaitInfo</a> for more information</li> </ul> </li> </ul>
--------------------------------------	---

**Returns**

`eQCWWAN_ERR_NONE` on success, `eQCWWAN_xxx` error value otherwise

**See Also**

See [qmerrno.h](#) for `eQCWWAN_xxx` error values

**Note**

Technology Supported: UMTS

Timeout: 30 Secs

This API queries the status of the call waiting supplementary service, i.e., to find whether the call waiting supplementary service is active. The request is a blocking request, i.e., the response is sent only after confirmation is received from the network. The maximum time it takes for the response to be sent is approximately 30 sec. This API is applicable only in 3GPP devices.

**9.38.4.17 ULONG SLQSVoiceGetCLIP ( voiceGetCLIPResp \* pVoiceGetCLIPResp )**

Queries the status of the Calling Line Identification Presentation (CLIP) supplementary service (applicable only for 3GPP).

**Parameters**

<i>pVoiceGetCLIPResp[OUT]</i>	<ul style="list-style-type: none"> <li>• Pointer to structure of <a href="#">voiceGetCLIPResp</a> <ul style="list-style-type: none"> <li>– See <a href="#">voiceGetCLIPResp</a> for more information</li> </ul> </li> </ul>
-------------------------------	---

**Returns**

`eQCWWAN_ERR_NONE` on success, `eQCWWAN_xxx` error value otherwise

**See Also**

See [qmerrno.h](#) for `eQCWWAN_xxx` error values

**Note**

Technology Supported: UMTS

Timeout: 30 Secs

This API queries the status of the CLIP supplementary service. The `active_status` field is only applicable when `provision_status` is `PROVISIONED`, i.e., there is not any case where `provision_status` is `NOT_PROVISIONED` and `active_status` is `ACTIVE`. The request is a blocking request, i.e., the response is sent only after confirmation is received from the network. The maximum time it takes for the response to be sent is approximately 30 sec. This API is applicable only in 3GPP devices.

**9.38.4.18 ULONG SLQSVoiceGetCLIR ( voiceGetCLIRResp \* pVoiceGetCLIRResp )**

Queries the status of the Calling Line Identification Restriction (CLIR) supplementary service (applicable only for 3GPP).

**Parameters**

<i>pVoiceGetCLIR-Resp[OUT]</i>	<ul style="list-style-type: none"> <li>• Pointer to structure of <a href="#">voiceGetCLIRResp</a> <ul style="list-style-type: none"> <li>– See <a href="#">voiceGetCLIRResp</a> for more information</li> </ul> </li> </ul>
--------------------------------	---

**Returns**

`eQCWWAN_ERR_NONE` on success, `eQCWWAN_xxx` error value otherwise

**See Also**

See [qmerrno.h](#) for `eQCWWAN_xxx` error values

**Note**

Technology Supported: UMTS

Timeout: 30 Secs

This API queries the status of the CLIR supplementary service. The `active_status` field is only applicable when `provision_status` is `PROVISIONED`, i.e., there is not any case where `provision_status` is `NOT_PROVISIONED` and `active_status` is `ACTIVE`. The request is a blocking request, i.e., the response is sent only after confirmation is received from the network. The maximum time it takes for the response to be sent is approximately 30 sec. This API is applicable only in 3GPP devices.

**9.38.4.19 ULONG SLQSVoiceGetCNAP ( voiceGetCNAPResp \* pVoiceGetCNAPResp )**

Queries the status of the Calling Name Presentation(CNAP) supplementary service (applicable only for 3GPP).

**Parameters**

<i>pVoiceGetCNA-PResp[OUT]</i>	<ul style="list-style-type: none"> <li>• Pointer to structure of <a href="#">voiceGetCNAPResp</a> <ul style="list-style-type: none"> <li>– See <a href="#">voiceGetCNAPResp</a> for more information</li> </ul> </li> </ul>
--------------------------------	---

**Returns**

`eQCWWAN_ERR_NONE` on success, `eQCWWAN_xxx` error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Technology Supported: UMTS  
Timeout: 30 Secs

This API queries the status of the CNAP supplementary service. A response indicates whether CNAP is active/inactive and provisioned/not provisioned in the network. The active\_status field is only applicable when provision\_status is PROVISIONED, i.e., there is not any case where provision\_status is NOT\_PROVISIONED and active\_status is ACTIVE. This API is applicable only in 3GPP devices.

**9.38.4.20 ULONG SLQSVoiceGetCOLP ( voiceGetCOLPResp \* pVoiceGetCOLPResp )**

Queries the status of the Connected Line Identification Presentation (COLP) supplementary service (applicable only for 3GPP).

**Parameters**

<i>pVoiceGetCOLPResp[OUT]</i>	<ul style="list-style-type: none"> <li>• Pointer to structure of <a href="#">voiceGetCOLPResp</a> <ul style="list-style-type: none"> <li>– See <a href="#">voiceGetCOLPResp</a> for more information</li> </ul> </li> </ul>
-------------------------------	---

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Technology Supported: UMTS  
Timeout: 30 Secs

This API queries the status of the COLP supplementary service. A response indicates whether COLP is active/inactive and provisioned/not provisioned in the network. The active\_status field is only applicable when provision\_status is PROVISIONED, i.e., there is not any case where provision\_status is NOT\_PROVISIONED and active\_status is ACTIVE. This API is applicable only in 3GPP devices.

**9.38.4.21 ULONG SLQSVoiceGetCOLR ( voiceGetCOLRResp \* pVoiceGetCOLRResp )**

Queries the status of the Connected Line Identification Restriction (COLR) supplementary service (applicable only for 3GPP).

**Parameters**

<i>pVoiceGetCOLRResp[OUT]</i>	<ul style="list-style-type: none"> <li>• Pointer to structure of <a href="#">voiceGetCOLRResp</a> <ul style="list-style-type: none"> <li>– See <a href="#">voiceGetCOLRResp</a> for more information</li> </ul> </li> </ul>
-------------------------------	---

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Technology Supported: UMTS  
Timeout: 30 Secs

This API queries the status of the COLR supplementary service. A response indicates whether COLR is active/inactive and provisioned/not provisioned in the network. The active\_status field is only applicable when provision\_status is PROVISIONED, i.e., there is not any case where provision\_status is NOT\_PROVISIONED and active\_status is ACTIVE. This API is applicable only in 3GPP devices.

#### 9.38.4.22 ULONG SLQSVoiceGetConfig ( voiceGetConfigReq \* pVoiceGetConfigReq, voiceGetConfigResp \* pVoiceGetConfigResp )

This API retrieves various configuration parameters that control the modem behavior related to circuit switched services.

**Parameters**

<i>pVoiceGet-ConfigReq</i>	<ul style="list-style-type: none"> <li>Structure containing Get Config request parameters. <ul style="list-style-type: none"> <li>See <a href="#">voiceGetConfigReq</a> for more information.</li> </ul> </li> </ul>
<i>pVoiceGet-ConfigResp</i>	<ul style="list-style-type: none"> <li>Structure containing Get Config response parameters. <ul style="list-style-type: none"> <li>See <a href="#">voiceGetConfigResp</a> for more information.</li> </ul> </li> </ul>

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Timeout: 30 Secs  
Any invalid value in a request message causes the service point to reject the message without retrieving any configuration information.

#### 9.38.4.23 ULONG SLQSVoiceIndicationRegister ( voiceIndicationRegisterInfo \* pVoiceIndicationRegisterInfo )

Sets the registration state for different QMI\_VOICE indications for the requesting control point

## Parameters

<i>pVoice-Indication-RegisterInfo</i> [IN]	<ul style="list-style-type: none"> <li>Structure containing Indication Register Information. <ul style="list-style-type: none"> <li>See <a href="#">voiceIndicationRegisterInfo</a> for more information.</li> </ul> </li> </ul>
--	--

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 10 Secs

This API is used by a device to register/deregister for different QMI VOICE indications.  
The device's registration state variables that control registration for indications will be modified to reflect the settings indicated in the request message.  
At least one optional parameter must be present in the request.

#### 9.38.4.24 ULONG SLQSVoiceManageCalls ( voiceManageCallsReq \* pVoiceManageCallsReq, voiceManageCallsResp \* pVoiceManageCallsResp )

Manages the calls by using the supplementary service applicable during the call. In cases of successful API completion if the state of any call is changed, it is indicated using Callback SLQSVoiceSetAllCallStatusCallBack. If there are other calls while an incoming voice call (waiting call) is received, this API is used to answer the call. This API is applicable only in "3GPP devices".

## Parameters

<i>pVoiceManageCallsReq</i> [IN]	<ul style="list-style-type: none"> <li>Request structure of to manage calls.</li> </ul>
<i>pVoiceManageCallsResp</i> [OUT]	<ul style="list-style-type: none"> <li>Response Structure to manage Calls</li> </ul>

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Technology Supported: UMTS  
Timeout: 10 Secs  
Applicable only for "3GPP"

#### 9.38.4.25 `ULONG SLQSVoiceOrigUSSDNoWait ( voiceOrigUSSDNoWaitInfo * pVoiceOrigUSSDNoWaitInfo )`

This API initiates a USSD operation such that the response for this request is returned immediately and the data is returned via an indication.

## Parameters

<i>pVoiceOrigUSSD-NoWaitInfo</i>	<ul style="list-style-type: none"> <li>See <a href="#">voiceOrigUSSDNoWaitInfo</a> for more information.</li> </ul>
----------------------------------	---

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Technology Supported: UMTS

Timeout: 30 Secs

This API starts a new USSD operation. The response to the request is sent immediately. The response result is sent to the client via the SLQSVoiceOrigUSSDNoWaitCallback. This command is applicable only in 3GPP devices.

#### 9.38.4.26 ULONG SLQSVoiceSendFlash ( voiceFlashInfo \* pFlashInfo )

This API sends a simple flash message. Applicable only for 3GPP2 devices.

## Parameters

<i>pFlashInfo[IN/OUT]</i>	<ul style="list-style-type: none"> <li>See <a href="#">voiceFlashInfo</a> for more information.</li> </ul>
---------------------------	--

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Technology Supported: CDMA

Timeout: 10 Secs

If success, it only means the device has started the requested operation and not that the Flash has been sent. If the optional parameter Flash Type is not set, the default flash type is assumed to be a simple flash. If the parameter Flash Type is set to 1 the call ID corresponding to it is either an incoming or waiting call's call ID. If the parameter Flash Type is set to 2 the call ID corresponding to it is a held call's call ID. A Flash request is sent to the appropriate call when call\_id is set to 0xFF.

#### 9.38.4.27 ULONG SLQSVoiceSetCallBarringPassword ( voiceSetCallBarringPwdInfo \* pVoiceSetCallBarringPwdInfo, voiceSetCallBarringPwdResp \* pSetCallBarringPwdResp )

Sets a Call Barring Password (applicable only for 3GPP).



## Parameters

<i>pVoiceSetCallBarringPwdInfo</i> [IN]	<ul style="list-style-type: none"> <li>• Pointer to structure of <a href="#">voiceSetCallBarringPwdInfo</a> <ul style="list-style-type: none"> <li>– See <a href="#">voiceSetCallBarringPwdInfo</a> for more information</li> </ul> </li> </ul>
<i>pSetCallBarringPwdResp</i> [OUT]	<ul style="list-style-type: none"> <li>• Pointer to structure of <a href="#">voiceSetCallBarringPwdResp</a> <ul style="list-style-type: none"> <li>– See <a href="#">voiceSetCallBarringPwdResp</a> for more information</li> </ul> </li> </ul>

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Technology Supported: UMTS

Timeout: 30 Secs

This API changes the call barring supplementary service password. The request is a blocking request, i.e., the response is sent only after confirmation is received from the network. The maximum time it takes for the response to be sent is approximately 30 sec. This API is applicable only in 3GPP devices.

#### 9.38.4.28 ULONG SLQSVoiceSetConfig ( voiceSetConfigReq \* pVoiceSetConfigReq, voiceSetConfigResp \* pVoiceSetConfigResp )

This message sets various configuration parameters that control the modem behavior related to circuit-switched services.

## Parameters

<i>pVoiceSetConfigReq</i> [IN]	<ul style="list-style-type: none"> <li>• Pointer to structure of <a href="#">voiceSetConfigReq</a> <ul style="list-style-type: none"> <li>– See <a href="#">voiceSetConfigReq</a> for more information</li> </ul> </li> </ul>
<i>pVoiceSetConfigResp</i> [OUT]	<ul style="list-style-type: none"> <li>• Pointer to structure of <a href="#">voiceSetConfigResp</a> <ul style="list-style-type: none"> <li>– See <a href="#">voiceSetConfigResp</a> for more information</li> </ul> </li> </ul>

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Timeout: 30 Secs

Any invalid value in a request message causes the device to reject the message without updating any configuration information. In the case of a successful update of all requested information, a QMI\_ERR\_NONE error is returned. In the case where a subset of information failed to be written, a QMI\_ERR\_INTERNAL error is returned with corresponding optional information requested in the request message.

#### 9.38.4.29 **ULONG SLQSVoiceSetPreferredPrivacy ( voiceSetPrefPrivacy \* pSetPrefPrivacy )**

This API sets the voice privacy preference. Applicable only for 3GPP2 devices.

**Parameters**

<i>pSetPrefPrivacy</i> [IN]	<ul style="list-style-type: none"> <li>See <a href="#">voiceSetPrefPrivacy</a> for more information.</li> </ul>
-----------------------------	---

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Technology Supported: CDMA

Device Supported: SL9090

Timeout: 10 Secs

#### 9.38.4.30 **ULONG SLQSVoiceSetSUPSService ( voiceSetSUPSServiceReq \* pVoiceSetSUPSServiceReq, voiceSetSUPSServiceResp \* pVoiceSetSUPSServiceResp )**

This API manages call-independent supplementary services, e.g., activation of call forwarding (to forward incoming calls to a third party), activation of call barring (to request the network to block some of the call attempts), and activation of call waiting (to be notified of an incoming call even when the user is engaged in an active or held call).

**Parameters**

<i>pVoiceSetSUPSServiceReq</i>	[IN] <ul style="list-style-type: none"> <li>Pointer to structure of <a href="#">voiceSetSUPSServiceReq</a> <ul style="list-style-type: none"> <li>See <a href="#">voiceSetSUPSServiceReq</a> for more information</li> </ul> </li> </ul>
<i>pVoiceSetSUPSServiceResp</i>	[OUT] <ul style="list-style-type: none"> <li>Pointer to structure of <a href="#">voiceSetSUPSServiceResp</a> <ul style="list-style-type: none"> <li>See <a href="#">voiceSetSUPSServiceResp</a> for more information</li> </ul> </li> </ul>

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Technology Supported: UMTS

Timeout: 30 Secs

Some of the call-independent services are provided by the network operator as part of the service agreement. If they are not provided by default, the user has to explicitly request them. This API provides the facility to the control point for sending the explicit request to the network for enabling/disabling

#### 9.38.4.31 ULONG SLQSVoiceStartContDTMF ( voiceContDTMFInfo \* pContDTMFInfo )

Starts a continuous DTMF.

## Parameters

<i>pContDTMFInfo</i>	<ul style="list-style-type: none"> <li>• Structure containing Continuous DTMF Information. <ul style="list-style-type: none"> <li>– See <a href="#">voiceContDTMFInfo</a> for more Information.</li> </ul> </li> </ul>
----------------------	--

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 30 Secs

This API starts a continuous DTMF. If the API results indicates success, it means that the device has started the requested operation. It does not mean that the start continuous DTMF request has been sent to the network. A start continuous DTMF request is sent to the current active/alerting call when CallId is set to 0xFF.

#### 9.38.4.32 ULONG SLQSVoiceStopContDTMF ( voiceStopContDTMFInfo \* pVoiceStopContDTMFInfo )

Stops a continuous DTMF.

## Parameters

<i>pVoiceStopCont-DTMFInfo</i>	<ul style="list-style-type: none"> <li>• Structure containing Continuous Stop DTMF Information. <ul style="list-style-type: none"> <li>– See <a href="#">voiceStopContDTMFInfo</a> for more information.</li> </ul> </li> <li>• Start continuous DTMF request is sent to the current active/alerting call when CallId is set to 0xFF.</li> <li>• This is IN/OUT params, value passed by user will packed in request and before unpacking response this will be assigned with an invalid callID value "0". It change to a valid value if received as part of response otherwise Invalid value will be present.</li> </ul>
--------------------------------	--

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Timeout: 30 Secs

This API starts a continuous DTMF. If the API results indicates success, it means that the device has started the requested operation. It does not mean that the start continuous DTMF request has been sent to the network. A stop continuous DTMF request is sent to the current active/alerting call when CallId is set to 0xFF.

## 9.39 qaGobiApiWds.h File Reference

Wireless Data Service API function prototypes.

**Data Structures**

- struct [dataBearerTechnology](#)
- struct [dataBearers](#)
- struct [QmiWdsIpAddressInfo](#)
- struct [WdsIpAddressInfoReq](#)
- struct [UMTSQoS](#)
- struct [ProfileIdentifier](#)
- struct [GPRSQoS](#)
- struct [PCSCFIPv4ServerAddressList](#)
- struct [PCSCFFQDNAddress](#)
- struct [PCSCFFQDNAddressList](#)
- struct [Domain](#)
- struct [DomainNameList](#)
- struct [IPv6AddressInfo](#)
- struct [IPv6GWAddressInfo](#)
- struct [qmiWdsRunTimeSettings](#)
- struct [WdsRunTimeSettings](#)
- struct [ssdatasession\\_params](#)
- struct [SLQSDeleteProfileParams](#)
- struct [Profile3GPP](#)
- struct [Profile3GPP2](#)
- union [WdsProfileParam](#)
- struct [CreateProfileIn](#)
- struct [UMTSMinQoS](#)
- struct [GPRSRequestedQoS](#)
- struct [TFTIDParams](#)
- struct [UMTSReqQoSsigInd](#)
- struct [QosClassID](#)
- struct [CreateProfileOut](#)
- struct [slqsautoconnect](#)
- struct [ModifyProfileIn](#)
- struct [ModifyProfileOut](#)
- struct [\\_slqs3GPPConfigItem](#)
- struct [\\_GetProfileSettingIn](#)

- struct [\\_GetProfileSettingOut](#)
- struct [TrStatInd](#)
- struct [wdsSetEventReportReq](#)
- struct [swiPDPRuntimeSettingsReq](#)
- struct [swiPDPRuntimeSettingsResp](#)
- struct [WdsConnectionRateElmnts](#)
- struct [WdsConnectionRate](#)
- struct [WdsByteTotalsElmnts](#)
- struct [WdsByteTotals](#)
- struct [GetInstIDResp](#)
- struct [GetSessionIDResp](#)
- struct [TransferStatInd](#)
- struct [getDUNCallInfoReq](#)
- struct [ConnectionStatus](#)
- struct [ChannelRate](#)
- struct [getDUNCallInfoResp](#)
- struct [WdsPktStatisticsElmnts](#)
- struct [WdsPktStatisticsResp](#)
- struct [WdsPktStatisticsReq](#)
- struct [CurrNetworkInfo](#)
- struct [CurrDataSysStat](#)
- struct [swiRMTrasnferStaticsReq](#)
- struct [DataULongTlv](#)
- struct [DataULongLongTlv](#)
- struct [QmiCbkWdsStatisticsIndState](#)
- struct [DataBearerTech](#)
- struct [DataBearerTechExt](#)
- struct [WDSSWICurrentChannelRates](#)
- struct [WDSSetLoopbackData](#)
- struct [WDSGetLoopbackData](#)

## Macros

- [#define IPV6\\_ADDRESS\\_ARRAY\\_SIZE](#) 8

## Typedefs

- typedef struct [dataBearerTechnology](#) [QmiWDSDataBearerTechnology](#)
- typedef struct [dataBearers](#) [QmiWDSDataBearers](#)
- typedef union [WdsProfileParam](#) [QmiProfileInfo](#)
- typedef struct [\\_slqs3GPPConfigItem](#) [slqs3GPPConfigItem](#)
- typedef struct [\\_GetProfileSettingIn](#) [GetProfileSettingIn](#)
- typedef struct [\\_GetProfileSettingOut](#) [GetProfileSettingOut](#)

## Enumerations

- enum [qmiDataBearerMasks](#) {  
[QMI\\_WDS\\_CURRENT\\_CALL\\_DB\\_MASK](#) = 0x01,  
[QMI\\_WDS\\_LAST\\_CALL\\_DB\\_MASK](#) = 0x02 }

## Functions

- [ULONG SetMobileIP](#) (ULONG mode)
- [ULONG GetMobileIP](#) (ULONG \*pMode)
- [ULONG SetMobileIPParameters](#) (CHAR \*pSPC, ULONG \*pMode, BYTE \*pRetryLimit, BYTE \*pRetryInterval, BYTE \*pReRegPeriod, BYTE \*pReRegTraffic, BYTE \*pHAAAuthenticator, BYTE \*pHA2002bis)
- [ULONG SetAutoconnect](#) (ULONG setting)
- [ULONG GetAutoconnect](#) (ULONG \*pSetting)
- [ULONG SetDefaultProfile](#) (ULONG profileType, ULONG \*pPDPTType, ULONG \*pIPAddress, ULONG \*pPrimaryDNS, ULONG \*pSecondaryDNS, ULONG \*pAuthentication, CHAR \*pName, CHAR \*pAPNName, CHAR \*pUsername, CHAR \*pPassword)
- [ULONG SetDefaultProfileLTE](#) (ULONG profileType, ULONG \*pPDPTType, ULONG \*pIPv4Address, ULONG \*pPrimaryDNSv4, ULONG \*pSecondaryDNSv4, USHORT \*pIPv6Address, USHORT \*pPrimaryDNSv6, USHORT \*pSecondaryDNSv6, ULONG \*pAuthentication, CHAR \*pName, CHAR \*pAPNName, CHAR \*pUsername, CHAR \*pPassword)
- [ULONG SetDefaultProfileLTEV2](#) (ULONG profileType, ULONG \*pPDPTType, ULONG \*pIPv4Address, ULONG \*pPrimaryDNSv4, ULONG \*pSecondaryDNSv4, USHORT \*pIPv6Address, USHORT \*pPrimaryDNSv6, USHORT \*pSecondaryDNSv6, ULONG \*pAuthentication, CHAR \*pName, CHAR \*pAPNName, CHAR \*pUsername, CHAR \*pPassword)
- [ULONG GetDefaultProfile](#) (ULONG profileType, ULONG \*pPDPTType, ULONG \*pIPAddress, ULONG \*pPrimaryDNS, ULONG \*pSecondaryDNS, ULONG \*pAuthentication, BYTE nameSize, CHAR \*pName, BYTE apnSize, CHAR \*pAPNName, BYTE userSize, CHAR \*pUsername)
- [ULONG GetDefaultProfileLTE](#) (ULONG profileType, ULONG \*pPDPTType, ULONG \*pIPv4Address, ULONG \*pPrimaryDNSv4, ULONG \*pSecondaryDNSv4, USHORT \*pIPv6Address, USHORT \*pPrimaryDNSv6, USHORT \*pSecondaryDNSv6, ULONG \*pAuthentication, BYTE nameSize, CHAR \*pName, BYTE apnSize, CHAR \*pAPNName, BYTE userSize, CHAR \*pUsername)
- [ULONG GetSessionState](#) (ULONG \*pState, BYTE instance)
- [ULONG GetPacketStatus](#) (ULONG \*pTXPacketSuccesses, ULONG \*pRXPacketSuccesses, ULONG \*pTXPacketErrors, ULONG \*pRXPacketErrors, ULONG \*pTXPacketOverflows, ULONG \*pRXPacketOverflows, BYTE instance)
- [ULONG GetByteTotals](#) (ULONGLONG \*pTXTotalBytes, ULONGLONG \*pRXTotalBytes, BYTE instance)
- [ULONG GetDormancyState](#) (ULONG \*pDormancyState, BYTE instance)
- [ULONG GetDataBearerTechnology](#) (ULONG \*pDataBearer, BYTE instance)
- [ULONG SLQSGetDataBearerTechnology](#) (QmiWDSDataBearers \*pDataBearers, BYTE instance)
- [ULONG GetSessionDuration](#) (ULONGLONG \*pDuration, BYTE instance)
- [ULONG GetIPAddressLTE](#) (WdsIpAddressInfoReq \*)
- [ULONG GetConnectionRate](#) (ULONG \*pCurrentChannelTXRate, ULONG \*pCurrentChannelRXRate, ULONG \*pMaxChannelTXRate, ULONG \*pMaxChannelRXRate, BYTE instance)
- [ULONG GetMobileIPProfile](#) (BYTE index, BYTE \*pEnabled, ULONG \*pAddress, ULONG \*pPrimaryHA, ULONG \*pSecondaryHA, BYTE \*pRevTunneling, BYTE naiSize, CHAR \*pNAI, ULONG \*pHASPI, ULONG \*pAAASPI, ULONG \*pHASState, ULONG \*pAAASState)
- [ULONG GetLastMobileIPError](#) (ULONG \*pError)
- [ULONG iSLQSMISetIPFamilyPreference](#) (BYTE IPFamilyPreference, BYTE instance)
- [BOOL WDS\\_IsGobiDevice](#) ()
- [ULONG SetActiveMobileIPProfile](#) (CHAR \*pSPC, BYTE index)
- [ULONG SetMobileIPProfile](#) (CHAR \*pSPC, BYTE index, BYTE \*pEnabled, ULONG \*pAddress, ULONG \*pPrimaryHA, ULONG \*pSecondaryHA, BYTE \*pRevTunneling, CHAR \*pNAI, ULONG \*pHASPI, ULONG \*pAAASPI, CHAR \*pMNHA, CHAR \*pMNAAS)
- [ULONG SLQSGetRuntimeSettings](#) (struct WdsRunTimeSettings \*pRunTimeSettings)
- [ULONG SLQSSetProfile](#) (ULONG profileType, BYTE profileId, ULONG \*pPDPTType, ULONG \*pIPAddress, ULONG \*pPrimaryDNS, ULONG \*pSecondaryDNS, ULONG \*pAuthentication, CHAR \*pName, CHAR \*pAPNName, CHAR \*pUsername, CHAR \*pPassword)
- [ULONG SLQSGetProfile](#) (ULONG profileType, BYTE profileId, ULONG \*pPDPTType, ULONG \*pIPAddress, ULONG \*pPrimaryDNS, ULONG \*pSecondaryDNS, ULONG \*pAuthentication, BYTE nameSize, CHAR \*pName, BYTE apnSize, CHAR \*pAPNName, BYTE userSize, CHAR \*pUsername, WORD \*pExtendedErrorCode)

- [ULONG SLQSStartStopDataSession](#) (struct [ssdatasession\\_params](#) \*pin)
- [ULONG SLQSDeleteProfile](#) (struct [SLQSDeleteProfileParams](#) \*pProfileToDelete, [WORD](#) \*pExtendedErrorCode)
- [ULONG SLQSCreateProfile](#) (struct [CreateProfileIn](#) \*pReq, struct [CreateProfileOut](#) \*pResp)
- [ULONG SLQSAutoConnect](#) (struct [slqsautoconnect](#) \*pacreq)
- [ULONG SLQSModifyProfile](#) (struct [ModifyProfileIn](#) \*pReq, struct [ModifyProfileOut](#) \*pResp)
- [ULONG SLQSSet3GPPConfigItem](#) ([slqs3GPPConfigItem](#) \*pSLQS3GPPConfigItem)
- [ULONG SLQSGet3GPPConfigItem](#) ([slqs3GPPConfigItem](#) \*pSLQS3GPPConfigItem)
- [ULONG SLQSGetProfileSettings](#) ([GetProfileSettingIn](#) \*pReq, [GetProfileSettingOut](#) \*pResp)
- [ULONG SLQSWdsSetEventReport](#) ([wdsSetEventReportReq](#) \*pSetEventReportReq)
- [ULONG SLQSWdsSwiPDPRuntimeSettings](#) ([swiPDPRuntimeSettingsReq](#) \*pPDPRuntimeSettingsReq, [swiPDPRuntimeSettingsResp](#) \*pPDPRuntimeSettingsResp)
- [ULONG iGetConnectionRate](#) ([ULONG](#) \*pv4sessionId, [ULONG](#) \*pv6sessionId, struct [WdsConnectionRateElmnts](#) \*pConnectionRateElmnt)
- [ULONG SLQSGetConnectionRate](#) (struct [WdsConnectionRate](#) \*pConnectionRate)
- [ULONG iGetByteTotals](#) ([ULONG](#) \*pv4sessionId, [ULONG](#) \*pv6sessionId, struct [WdsByteTotalsElmnts](#) \*pByteTotalsElmnt)
- [ULONG SLQSGetByteTotals](#) (struct [WdsByteTotals](#) \*pByteTotals)
- [ULONG SLQSWdsGoDormant](#) (void)
- [ULONG SLQSWdsGoActive](#) (void)
- [ULONG SLQSGetSessionState](#) ([ULONG](#) \*pStateV4, [ULONG](#) \*pStateV6, [BYTE](#) instance)
- [ULONG SLQSGetDUNCallInfo](#) ([getDUNCallInfoReq](#) \*pGetDUNCallInfoReq, [getDUNCallInfoResp](#) \*pGetDUNCallInfoResp)
- [ULONG GetPacketStatistics](#) (struct [WdsPktStatisticsReq](#) \*pStatMask, struct [WdsPktStatisticsElmnts](#) \*pPktStatisticsElmnt, [BYTE](#) instance)
- [ULONG iGetPacketStatistics](#) ([ULONG](#) \*pV4sessionId, [ULONG](#) \*pV6sessionId, struct [WdsPktStatisticsReq](#) \*pStatMask, struct [WdsPktStatisticsElmnts](#) \*pPktStatisticsElmnt)
- [ULONG SLQSGetPacketStatistics](#) (struct [WdsPktStatisticsReq](#) \*pStatMask, struct [WdsPktStatisticsResp](#) \*pPktStatistics)
- [ULONG SLQSGetCurrDataSystemStat](#) ([CurrDataSysStat](#) \*pCurrDataSysStat)
- [ULONG RMSetTransferStatistics](#) ([swiRMTrasferStaticsReq](#) \*pSwiRMTrasferStaticsReq)
- [ULONG SLQSResetPacketStatics](#) ()
- [ULONG SLQSSetHostMTU](#) (int \*mtu, [BYTE](#) instance)
- [ULONG SLQSGetDataBearerTechnologyExt](#) ([DataBearerTechExt](#) \*pDataBearerTech, [BYTE](#) instance)
- [ULONG SLQSGetCurrentChannelRate](#) ([WDSSWICurrentChannelRates](#) \*pRates, [BYTE](#) instance)
- [ULONG SLQSSetLoopback](#) ([WDSSetLoopbackData](#) \*pReq)
- [ULONG SLQSGetLoopback](#) ([WDSGetLoopbackData](#) \*data)

### 9.39.1 Detailed Description

Wireless Data Service API function prototypes.

### 9.39.2 Macro Definition Documentation

9.39.2.1 `#define IPV6_ADDRESS_ARRAY_SIZE 8`

### 9.39.3 Typedef Documentation

9.39.3.1 `typedef struct _GetProfileSettingIn GetProfileSettingIn`

This structure contains the input parameters for `SLQSGetProfileSettings`

## Parameters

<i>ProfileType</i>	<ul style="list-style-type: none"> <li>Identifies the technology type of the profile <ul style="list-style-type: none"> <li>0x00 - 3GPP</li> <li>0x01 - 3GPP2</li> </ul> </li> </ul>
<i>ProfileID</i>	<ul style="list-style-type: none"> <li>index identifying the profile</li> </ul>

## 9.39.3.2 typedef struct \_GetProfileSettingOut GetProfileSettingOut

This structure contains the profile settings retrieved by the API SLQSGetProfileSettings

## Parameters

<i>curProfile</i>	<ul style="list-style-type: none"> <li>Structure containing details of the profile</li> <li>See <a href="#">QmiProfileInfo</a> for more details</li> </ul>
<i>pExtErrCode</i>	<ul style="list-style-type: none"> <li>pointer to a 2 byte extended error code</li> <li>Error code will only be present if error code eQCWWAN_ERR_QMI_EXTENDED_INTERNAL is returned by device.</li> <li>See <a href="#">qm_wds_ds_profile_extended_err_codes</a> enum in <a href="#">qmerrno.h</a> for received error description.</li> </ul>

## 9.39.3.3 typedef union WdsProfileParam QmiProfileInfo

This union [WdsProfileParam](#) consist of [Profile3GPP](#) and [Profile3GPP2](#) out of which one will be used to create profile.

## 9.39.3.4 typedef struct dataBearers QmiWSDDataBearers

Structure to hold the data bearer technology values



## Parameters

<i>dataBearerMask</i> [OUT]	<ul style="list-style-type: none"> <li>This bit mask indicates if data bearer information for the current and/or last call has been received from the device. If a bit is set, then the information is available in the corresponding structure i.e. the one provided by the caller. Refer to <a href="#">qmiDataBearerMasks</a> for bit-mask positions.</li> </ul>
<i>pCurDataBearerTechnology</i> [OUT]	<ul style="list-style-type: none"> <li>current data bearer technology value. <ul style="list-style-type: none"> <li>– NULL if the parameter is not required</li> </ul> </li> </ul>
<i>pLastCallDataBearerTechnology</i> [OUT]	<ul style="list-style-type: none"> <li>last call data bearer technology value. <ul style="list-style-type: none"> <li>– NULL if the parameter is not required</li> </ul> </li> </ul>

## 9.39.3.5 typedef struct dataBearerTechnology QmiWDSDataBearerTechnology

Structure to hold the current data bearer technology values

## Parameters

<i>pCurrent-Network[OUT]</i>	<ul style="list-style-type: none"> <li>current selected network <ul style="list-style-type: none"> <li>0 - UNKNOWN</li> <li>1 - 3GPP2</li> <li>2 - 3GPP</li> </ul> </li> </ul>
<i>pRatMask[OUT]</i>	<ul style="list-style-type: none"> <li>Radio Access Technology (RAT) mask to indicate the type of technology (RAT mask value of zero indicates that this field is ignored) <ul style="list-style-type: none"> <li>0x8000 - NULL Bearer</li> <li>0x0000 - DO_NOT_CARE CDMA RAT mask</li> <li>0x01 - CDMA_1X</li> <li>0x02 - EVDO_REV0</li> <li>0x04 - EVDO_REVA UMTS RAT mask</li> <li>0x01 - WCDMA</li> <li>0x02 - GPRS</li> <li>0x04 - HSDPA</li> <li>0x08 - HSUPA</li> <li>0x10 - EDGE</li> <li>0x20 - LTE</li> <li>0x40 - HSDPA+</li> <li>0x80 - DC_HSDPA+</li> </ul> </li> </ul>
<i>pSoMask[OUT]</i>	<ul style="list-style-type: none"> <li>Service Option (SO) mask to indicate the SO or type of application (SO mask value of zero indicates that this field is ignored) <ul style="list-style-type: none"> <li>0x00 - DO_NOT_CARE CDMA 1X SO mask</li> <li>0x01 - CDMA_1X_IS95</li> <li>0x02 - CDMA_1X_IS2000</li> <li>0x04 - CDMA_1X_IS2000_REL_A CDMA EV-DO Rev A SO mask</li> <li>0x01 - EVDO_REVA_DPA</li> <li>0x02 - EVDO_REVA_MFPA</li> <li>0x04 - EVDO_REVA_EMPA</li> <li>0x08 - EVDO_REVA_EMPA_EHRPD</li> </ul> </li> </ul>

## 9.39.3.6 typedef struct \_slqs3GPPConfigItem slqs3GPPConfigItem

This structure contains the 3gpp Configuration Item information.

## Parameters

<i>pLTEAttach-Profile</i>	<ul style="list-style-type: none"> <li>• LTE Attach Profile <ul style="list-style-type: none"> <li>– points to a single WORD Value indicating the attached LTE Profile</li> <li>– Optional parameter with possible values 1-16</li> <li>– function <a href="#">SLQSGet3GPPConfigItem()</a> returns a default value 255 if no LTE Attach Profile is configured</li> </ul> </li> <li>• This setting is deprecated on MC/EM74xx</li> </ul>
<i>pProfileList</i>	<ul style="list-style-type: none"> <li>• Profile List <ul style="list-style-type: none"> <li>– an array of 4 profile configurations</li> <li>– Each element points to a single WORD value indicating profile</li> <li>– Optional parameter with possible values <ul style="list-style-type: none"> <li>* 1 - 16 (MC/EM73xx and before)</li> <li>* 1 - 24 (MC/EM74xx and onwards)</li> </ul> </li> <li>– function <a href="#">SLQSGet3GPPConfigItem()</a> returns a default value 255 if no 3gpp configuration is present</li> </ul> </li> </ul>
<i>pDefaultPDN-Enabled</i>	<ul style="list-style-type: none"> <li>• Always Connect Default PDN <ul style="list-style-type: none"> <li>– A single BYTE value indicating the status of Always connect default PDN <ul style="list-style-type: none"> <li>* 0 - disabled</li> <li>* 1 - enabled</li> </ul> </li> <li>– Optional parameter</li> <li>– function <a href="#">SLQSGet3GPPConfigItem()</a> returns a default value 255 if no 3gpp configuration is present</li> </ul> </li> </ul>
<i>p3gppRelease</i>	<ul style="list-style-type: none"> <li>• 3gpp release <ul style="list-style-type: none"> <li>– A single BYTE value indicating the 3gpp release <ul style="list-style-type: none"> <li>* 0 - Release 99</li> <li>* 1 - Release 5</li> <li>* 2 - Release 6</li> <li>* 3 - Release 7</li> <li>* 4 - Release 8</li> </ul> </li> <li>– Optional parameter</li> <li>– function <a href="#">SLQSGet3GPPConfigItem()</a> returns a default value 255 if no 3gpp configuration is present</li> </ul> </li> </ul>
<i>pLTEAttach-ProfileList</i>	<ul style="list-style-type: none"> <li>• pointer to WORD array indicating LTE Attach Profile List <ul style="list-style-type: none"> <li>– Optional parameter</li> <li>– possible values: 1-24</li> <li>– This setting is only supported for MC/EM74xx onwards</li> </ul> </li> </ul>
<p>Generated on Wed Oct 28 2015 00:40:12 for LinuxQMI SDK by Doxygen</p> <p>– The new equivalent option for "pLTEAttachProfile" on 74xx modems is "pLTEAttachProfileList". Please provide attach profiles in order of decreasing priority in this list.</p>	

<i>LTEAttach-ProfileListLen</i>	<ul style="list-style-type: none"> <li>• Number of element in pLTEAttachProfileList <ul style="list-style-type: none"> <li>– valid range: 0-2</li> <li>* This setting is only supported for MC/EM74xx onwards</li> </ul> </li> </ul>
---------------------------------	--

## 9.39.4 Enumeration Type Documentation

### 9.39.4.1 enum qmiDataBearerMasks

Bit mask values to indicate the presence of data bearer information for the current and last data calls

Enumerator

***QMI\_WDS\_CURRENT\_CALL\_DB\_MASK***

***QMI\_WDS\_LAST\_CALL\_DB\_MASK***

## 9.39.5 Function Documentation

### 9.39.5.1 ULONG GetAutoconnect ( ULONG \* pSetting )

Returns the current auto connect data session setting.

Parameters

<i>pSetting[OUT]</i>	<ul style="list-style-type: none"> <li>• NDIS auto connect setting <ul style="list-style-type: none"> <li>– 0 - Disabled</li> <li>– 1 - Enabled</li> </ul> </li> </ul>
----------------------	--

Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

Note

Timeout: 2 seconds

### 9.39.5.2 ULONG GetByteTotals ( ULONGLONG \* pTXTotalBytes, ULONGLONG \* pRXTotalBytes, BYTE instance )

Returns the Rx/Tx byte counts since the start of the last packet data session for IPV4 session only.

## Parameters

<i>pTXTotalBytes[OUT]</i>	<ul style="list-style-type: none"> <li>Bytes transmitted without error</li> </ul>
<i>pRXTotalBytes[OUT]</i>	<ul style="list-style-type: none"> <li>Bytes received without error</li> </ul>
<i>instance</i>	<ul style="list-style-type: none"> <li>PDP instance</li> </ul>

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 2 seconds, Rx/Tx byte counts for IPV4 only

**9.39.5.3** **ULONG** GetConnectionRate ( **ULONG** \* *pCurrentChannelTXRate*, **ULONG** \* *pCurrentChannelRXRate*, **ULONG** \* *pMaxChannelTXRate*, **ULONG** \* *pMaxChannelRXRate*, **BYTE** *instance* )

Returns connection rate information for the packet data connection. This API is not applicable when multiple data session is up. For multiple PDN, please use API [SLQSGetConnectionRate\(\)](#)

## Parameters

<i>pCurrent-ChannelTX-Rate[OUT]</i>	<ul style="list-style-type: none"> <li>Current channel Tx rate (in bps)</li> </ul>
<i>pCurrent-ChannelRX-Rate[OUT]</i>	<ul style="list-style-type: none"> <li>Current channel Rx rate (in bps)</li> </ul>
<i>pMaxChannelTXRate[OUT]</i>	<ul style="list-style-type: none"> <li>Maximum Tx rate (bps) that may be assigned to device by serving system.</li> </ul>
<i>pMaxChannelRXRate[OUT]</i>	<ul style="list-style-type: none"> <li>Maximum Rx rate (bps) that may be assigned to device by serving system.</li> </ul>
<i>instance</i>	<ul style="list-style-type: none"> <li>PDP instance</li> </ul>

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 2 seconds

#### 9.39.5.4 ULONG GetDataBearerTechnology ( ULONG \* pDataBearer, BYTE instance )

Retrieves the current data bearer technology (only valid when connected). This API is deprecated on MC73xx/E-M73xx modules since firmware version SWI9X15C\_05\_xx\_xx\_xx and all EM74xx firmware versions. Please use API [SLQSGetDataBearerTechnologyExt\(\)](#) for new firmware versions and new modules.

## Parameters

<i>pDataBearer[O-UT]</i>	<ul style="list-style-type: none"> <li>• Data bearer technology <ul style="list-style-type: none"> <li>– 0x01 - CDMA2000 1x</li> <li>– 0x02 - CDMA 1xEV-DO Rev 0</li> <li>– 0x03 - GSM</li> <li>– 0x04 - UMTS</li> <li>– 0x05 - CDMA2000 HRPD (1xEV-DO Rev A)</li> <li>– 0x06 - EDGE</li> <li>– 0x07 - HSDPA AND WCDMA</li> <li>– 0x08 - WCDMA AND HSUPA</li> <li>– 0x09 - HSDPA AND HSUPA</li> <li>– 0x0A - LTE</li> <li>– 0x0B - CDMA2000 EHRPD</li> <li>– 0x0C - HSDPA+ and WCDMA</li> <li>– 0x0D - HSDPA+ and HSUPA</li> <li>– 0x0E - DC_HSDPA+ and WCDMA</li> <li>– 0x0F - DC_HSDPA+ and HSUPA</li> <li>– 0x10 - HSDPA+ and 64QAM</li> <li>– 0x11 - HSDPA+, 64QAM and HSUPA</li> <li>– 0x12 - TDSCDMA</li> <li>– 0x13 - TDSCDMA and HSDPA</li> <li>– 0xFF - Unknown</li> </ul> </li> </ul>
--------------------------	---

<i>instance</i>	<ul style="list-style-type: none"><li>• PDP instance</li></ul>
-----------------	--

#### Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

#### See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

#### Note

The QMI command of this API is deprecated. Use [SLQSGetDataBearerTechnologyExt\(\)](#) for new modules (chipset 9x15, 9x30). Version Introduced: Major-1, Minor-12 Version Deprecated: Major-1, Minor-40 Timeout: 2 seconds

**9.39.5.5** `ULONG GetDefaultProfile ( ULONG profileType, ULONG * pPDPTType, ULONG * pIPAddress, ULONG * pPrimaryDNS, ULONG * pSecondaryDNS, ULONG * pAuthentication, BYTE nameSize, CHAR * pName, BYTE apnSize, CHAR * pAPNName, BYTE userSize, CHAR * pUsername )`

Reads the default profile settings from the device. The default profile is used to establish an auto connect data session.

## Parameters

<i>profileType</i>	<ul style="list-style-type: none"> <li>Type of profile <ul style="list-style-type: none"> <li>0 - UMTS</li> </ul> </li> </ul>
<i>pPDPTType[OUT]</i>	<ul style="list-style-type: none"> <li>Packet Data Protocol (PDP) type specifies the type of data payload exchanged over the air link when the packet data session is established with this profile <ul style="list-style-type: none"> <li>0 - PDP-IP (IPv4)</li> </ul> </li> </ul>
<i>pIPAddress[OUT]</i>	<ul style="list-style-type: none"> <li>Preferred IPv4 address to be assigned to device</li> </ul>
<i>pPrimaryDNS[OUT]</i>	<ul style="list-style-type: none"> <li>Primary DNS Ipv4 address preference</li> </ul>
<i>pSecondaryDNS[OUT]</i>	<ul style="list-style-type: none"> <li>Secondary DNS Ipv4 address preference</li> </ul>
<i>pAuthentication[OUT]</i>	<ul style="list-style-type: none"> <li>Bitmap that indicates authentication algorithm preference <ul style="list-style-type: none"> <li>0x00000001 - PAP preference <ul style="list-style-type: none"> <li>0 - Never performed</li> <li>1 - May be performed</li> </ul> </li> <li>0x00000002 - CHAP preference <ul style="list-style-type: none"> <li>0 - Never performed</li> <li>1 - May be performed</li> </ul> </li> <li>All other bits are reserved and must be set to 0</li> <li>If more than 1 bit is set, then device decides which authentication procedure is performed while setting up data session e.g. the device may have a policy to select the most secure authentication mechanism.</li> </ul> </li> </ul>
<i>nameSize</i>	<ul style="list-style-type: none"> <li>Maximum number of characters (including NULL terminator) that profile name array can contain.</li> </ul>
<i>pName[OUT]</i>	<ul style="list-style-type: none"> <li>Profile name</li> </ul>
<i>apnSize</i>	<ul style="list-style-type: none"> <li>Maximum number of characters (including NULL terminator) that APN name array can contain</li> </ul>



<i>pAPNName</i> [OUT]	<ul style="list-style-type: none"> <li>Access point name. NULL-terminated string parameter that is a logical name used to select GGSN and external packet data network.</li> <li>If value is NULL or omitted, then subscription default value will be requested.</li> </ul>
<i>userSize</i>	<ul style="list-style-type: none"> <li>Maximum number of characters (including NULL terminator) that username array can contain.</li> </ul>
<i>pUsername</i> [OUT]	<ul style="list-style-type: none"> <li>Username used during network authentication</li> </ul>

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Timeout: 2 seconds

**9.39.5.6** **ULONG** GetDefaultProfileLTE ( **ULONG** *profileType*, **ULONG** \* *pPDPTType*, **ULONG** \* *pIPAddressv4*, **ULONG** \* *pPrimaryDNSv4*, **ULONG** \* *pSecondaryDNSv4*, **USHORT** \* *pIPAddressv6*, **USHORT** \* *pPrimaryDNSv6*, **USHORT** \* *pSecondaryDNSv6*, **ULONG** \* *pAuthentication*, **BYTE** *nameSize*, **CHAR** \* *pName*, **BYTE** *apnSize*, **CHAR** \* *pAPNName*, **BYTE** *userSize*, **CHAR** \* *pUsername* )

Reads the default profile settings from the device. The default profile is used to establish an auto connect data session.

**Parameters**

<i>profileType</i>	<ul style="list-style-type: none"> <li>Type of profile <ul style="list-style-type: none"> <li>0 - UMTS</li> </ul> </li> </ul>
<i>pPDPTType</i> [OUT]	<ul style="list-style-type: none"> <li>Packet Data Protocol (PDP) type specifies the type of data payload exchanged over the air link when the packet data session is established with this profile <ul style="list-style-type: none"> <li>0 - PDP-IP (IPv4)</li> </ul> </li> </ul>

<i>pIPvAddressv4[-OUT]</i>	<ul style="list-style-type: none"> <li>Preferred IPv4 addr to be assigned to device</li> </ul>
<i>pPrimaryDN-Sv4[OUT]</i>	<ul style="list-style-type: none"> <li>Primary DNS Ipv4 address preference</li> </ul>
<i>pSecondaryDN-Sv4[OUT]</i>	<ul style="list-style-type: none"> <li>Secondary DNS Ipv4 address preference</li> </ul>
<i>pIPvAddressv6[-OUT]</i>	<ul style="list-style-type: none"> <li>Preferred IPv6 addr to be assigned to device Space for storing 8 element array for the IPv6 addresses is allocated by the application. The IP Address will be retrieved in the big endian format. For example User buffer contents: [&lt;U0&gt;..&lt;&lt;U7&gt;]</li> </ul> <p>IPv6 address:  1234:2A01:.....:5678  U0 corresponds to 1234  U1 corresponds to 2A01  -----  -----  U7 corresponds to 5678</p>
<i>pPrimaryDN-Sv6[OUT]</i>	<ul style="list-style-type: none"> <li>Primary DNS Ipv6 address preference</li> </ul>
<i>pSecondaryDN-Sv6[OUT]</i>	<ul style="list-style-type: none"> <li>Secondary DNS Ipv6 address preference</li> </ul>
<i>pAuthentication[-OUT]</i>	<ul style="list-style-type: none"> <li>Bitmap that indicates authentication algorithm preference <ul style="list-style-type: none"> <li>0x00000001 - PAP preference <ul style="list-style-type: none"> <li>0 - Never performed</li> <li>1 - May be performed</li> </ul> </li> <li>0x00000002 - CHAP preference <ul style="list-style-type: none"> <li>0 - Never performed</li> <li>1 - May be performed</li> </ul> </li> <li>All other bits are reserved and must be set to 0</li> <li>If more than 1 bit is set, then device decides which authentication procedure is performed while setting up data session e.g. the device may have a policy to select the most secure authentication mechanism.</li> </ul> </li> </ul>

<i>nameSize</i>	<ul style="list-style-type: none"> <li>Maximum number of characters (including NULL terminator) that Profile name array can contain</li> </ul>
<i>pName[OUT]</i>	<ul style="list-style-type: none"> <li>Profile name</li> </ul>
<i>apnSize</i>	<ul style="list-style-type: none"> <li>Maximum number of characters (including NULL terminator) that APN name array can contain</li> </ul>
<i>pAPNName[IN]</i>	<ul style="list-style-type: none"> <li>Access point name. NULL-terminated string parameter that is a logical name used to select GGSN and external packet data network.</li> <li>If value is NULL or omitted, then subscription default value will be requested.</li> </ul>
<i>userSize</i>	<ul style="list-style-type: none"> <li>Maximum number of characters including NULL terminator) that username array can contain.</li> </ul>
<i>pUsername[OUT]</i>	<ul style="list-style-type: none"> <li>Username used during network authentication</li> </ul>

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Technology Supported: LTE  
Timeout: 2 seconds

**9.39.5.7 ULONG GetDormancyState ( ULONG \* pDormancyState, BYTE instance )**

Returns the dormancy state of the current packet data session when connected.

**Parameters**

<i>pDormancy-State[OUT]</i>	<ul style="list-style-type: none"> <li>Dormancy state of current packet data session <ul style="list-style-type: none"> <li>1 - Traffic channel dormant</li> <li>2 - Traffic channel active</li> </ul> </li> </ul>
-----------------------------	--

<i>instance</i>	<ul style="list-style-type: none"> <li>• PDP instance</li> </ul>
-----------------	--

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Timeout: 2 seconds

**9.39.5.8 ULONG GetIPAddressLTE ( WdslpAddressInfoReq \* )**

Returns the current packet data session IP address(es)

- Parameter values default to their data type's maximum unsigned value unless explicitly stated otherwise.

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Technology Supported: LTE  
Timeout: 2 seconds.

**9.39.5.9 ULONG GetLastMobileIPError ( ULONG \* pError )**

Returns the last mobile IP error.

**Parameters**

<i>pError[OUT]</i>	<ul style="list-style-type: none"> <li>• Status of last MIP call (or attempt) <ul style="list-style-type: none"> <li>– Zero - Success</li> <li>– NonZero - Error code</li> </ul> </li> </ul> <p>See <a href="#">qaGobiApiTableCallEndReasons.h</a> for Mobile IP Error codes</p>
--------------------	--

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Technology Supported: CDMA  
 Device Supported: MC83x5  
 Timeout: 2 seconds

#### 9.39.5.10 ULONG GetMobileIP ( ULONG \* pMode )

Returns the current mobile IP setting.

## Parameters

<i>mode[OUT]</i>	<ul style="list-style-type: none"> <li>• Mobile IP setting               <ul style="list-style-type: none"> <li>– 0 - Mobile IP off (simple IP only)</li> <li>– 1 - Mobile IP preferred</li> <li>– 2 - Mobile IP only</li> </ul> </li> </ul>
------------------	--

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Technology Supported: CDMA  
 Timeout: 2 seconds

#### 9.39.5.11 ULONG GetMobileIPProfile ( BYTE index, BYTE \* pEnabled, ULONG \* pAddress, ULONG \* pPrimaryHA, ULONG \* pSecondaryHA, BYTE \* pRevTunneling, BYTE naiSize, CHAR \* pNAI, ULONG \* pHASPI, ULONG \* pAAASPI, ULONG \* pHASState, ULONG \* pAAASState )

Returns the specified mobile IP profile settings.

## Parameters

<i>index</i>	<ul style="list-style-type: none"> <li>• Mobile IP profile ID</li> </ul>
<i>pEnabled[OUT]</i>	<ul style="list-style-type: none"> <li>• Profile enabled: <ul style="list-style-type: none"> <li>– 0 - Disabled</li> <li>– 1 - Enabled</li> <li>– 0xFF - Unknown</li> </ul> </li> </ul>
<i>pAddress[OUT]</i>	<ul style="list-style-type: none"> <li>• Home IPv4 address: <ul style="list-style-type: none"> <li>– 0xFFFFFFFF - Unknown</li> </ul> </li> </ul>
<i>pPrimaryHA[OUT]</i>	<ul style="list-style-type: none"> <li>• Primary home agent IPv4 address <ul style="list-style-type: none"> <li>– 0xFFFFFFFF - Unknown</li> </ul> </li> </ul>
<i>pSecondaryHA[OUT]</i>	<ul style="list-style-type: none"> <li>• Secondary home agent IPv4 address <ul style="list-style-type: none"> <li>– 0xFFFFFFFF - Unknown</li> </ul> </li> </ul>
<i>pRevTunneling[OUT]</i>	<ul style="list-style-type: none"> <li>• Reverse tunneling enabled <ul style="list-style-type: none"> <li>– 0 - Disabled</li> <li>– 1 - Enabled</li> <li>– 0xFF - Unknown</li> </ul> </li> </ul>
<i>naiSize</i>	<ul style="list-style-type: none"> <li>• The maximum number of characters (including NULL terminator) that the NAI array can contain.</li> </ul>
<i>pNAI[OUT]</i>	<ul style="list-style-type: none"> <li>• Network access identifier string</li> </ul>
<i>pHASPI[OUT]</i>	<ul style="list-style-type: none"> <li>• Home agent security parameter index</li> </ul>
<i>pAAASPI[OUT]</i>	<ul style="list-style-type: none"> <li>• AAA server security parameter index <ul style="list-style-type: none"> <li>– 0xFFFFFFFF - Unknown</li> </ul> </li> </ul>

<i>pHState[OUT]</i>	<ul style="list-style-type: none"> <li>• Home agent key state <ul style="list-style-type: none"> <li>– 0 - Unset</li> <li>– 1 - Set, default value</li> <li>– 2 - Set, modified from default</li> <li>– 3 - 0xFFFFFFFF - Unknown</li> </ul> </li> </ul>
<i>pAAASState[OUT]</i>	<ul style="list-style-type: none"> <li>• AAA key state <ul style="list-style-type: none"> <li>– 0 - Unset</li> <li>– 1 - Set, default value</li> <li>– 2 - Set, modified from default</li> <li>– 3 - 0xFFFFFFFF - Unknown</li> </ul> </li> </ul>

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Technology Supported: CDMA  
Timeout: 2 seconds

#### 9.39.5.12 **ULONG** GetPacketStatistics ( struct WdsPktStatisticsReq \* *pStatMask*, struct WdsPktStatisticsElmnts \* *pPktStatisticsElmnt*, **BYTE** *instance* )

Returns the current packet transfer counter values from the device. Since the start of the last packet data session.

**Parameters**

<i>pStatMask[IN]</i>	<ul style="list-style-type: none"> <li>• See <a href="#">WdsPktStatisticsReq</a> for more information</li> </ul>
<i>pPktStatistics-Elmnt[OUT]</i>	<ul style="list-style-type: none"> <li>• See <a href="#">WdsPktStatisticsElmnts</a> for more information</li> </ul>
<i>instance</i>	<ul style="list-style-type: none"> <li>• PDP instance</li> </ul>

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 5 seconds

**9.39.5.13** **ULONG** GetPacketStatus ( **ULONG** \* *pTXPacketSuccesses*, **ULONG** \* *pRXPacketSuccesses*, **ULONG** \* *pTXPacketErrors*, **ULONG** \* *pRXPacketErrors*, **ULONG** \* *pTXPacketOverflows*, **ULONG** \* *pRXPacketOverflows*, **BYTE** *instance* )

Returns the packet data transfer statistics since the start of the current packet data.

## Parameters

<i>pTXPacketSuccesses</i> [OUT]	<ul style="list-style-type: none"> <li>No. of packets transmitted without error</li> </ul>
<i>pRXPacketSuccesses</i> [OUT]	<ul style="list-style-type: none"> <li>No. of packets received without error</li> </ul>
<i>pTXPacketErrors</i> [OUT]	<ul style="list-style-type: none"> <li>No. of outgoing packets with framing errors</li> </ul>
<i>pRXPacketErrors</i> [OUT]	<ul style="list-style-type: none"> <li>No. of incoming packets with framing errors</li> </ul>
<i>pTXPacketOverflows</i> [OUT]	<ul style="list-style-type: none"> <li>Number of packets dropped because Tx buffer overflowed</li> </ul>
<i>pRXPacketOverflows</i> [OUT]	<ul style="list-style-type: none"> <li>Number of packets dropped because Rx buffer overflowed</li> </ul>
<i>instance</i>	<ul style="list-style-type: none"> <li>PDP instance</li> </ul>

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 2 seconds

**9.39.5.14** **ULONG** GetSessionDuration ( **ULONGLONG** \* *pDuration*, **BYTE** *instance* )

Returns the duration of the current packet data session.



## Parameters

<i>pDuration[OUT]</i>	<ul style="list-style-type: none"><li>• Duration of the current packet session in milliseconds</li></ul>
<i>instance</i>	<ul style="list-style-type: none"><li>• PDP instance</li></ul>

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Technology Supported: UMTS/CDMA  
Device Supported: MC83x5, MC7700/50  
Timeout: 2 seconds

**9.39.5.15 ULONG GetSessionState ( ULONG \* pState, BYTE instance )**

Returns the state of the current packet data session.

## Parameters

<i>pState[OUT]</i>	<ul style="list-style-type: none"><li>• Current link status<ul style="list-style-type: none"><li>– 1 - DISCONNECTED</li><li>– 2 - CONNECTED</li><li>– 3 - SUSPENDED (not supported)</li><li>– 4 - AUTHENTICATING</li></ul></li></ul>
<i>instance</i>	<ul style="list-style-type: none"><li>• PDP instance</li></ul>

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 2 seconds

- 9.39.5.16 **ULONG** iGetByteTotals ( **ULONG** \* *pv4sessionId*, **ULONG** \* *pv6sessionId*, struct WdsByteTotalsElmnts \* *pByteTotalsElmnt* )
- 9.39.5.17 **ULONG** iGetConnectionRate ( **ULONG** \* *pv4sessionId*, **ULONG** \* *pv6sessionId*, struct WdsConnectionRateElmnts \* *pConnectionRateElmnt* )
- 9.39.5.18 **ULONG** iGetPacketStatistics ( **ULONG** \* *pV4sessionId*, **ULONG** \* *pV6sessionId*, struct WdsPktStatisticsReq \* *pStatMask*, struct WdsPktStatisticsElmnts \* *pPktStatisticsElmnt* )
- 9.39.5.19 **ULONG** iSLQSMISetIPFamilyPreference ( **BYTE** *IPFamilyPreference*, **BYTE** *instance* )
- 9.39.5.20 **ULONG** RMSetTransferStatistics ( swiRMTrasnferStaticsReq \* *pSwiRMTrasnferStaticsReq* )

This API request the device to fetch current data system transfer Statistics.

#### Parameters

<i>pSwiRMTrasnferStaticsReq</i> [IN/OUT]	<ul style="list-style-type: none"> <li>See <a href="#">swiRMTrasnferStaticsReq</a> for more information</li> </ul>
--	--

#### Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_XXX error value otherwise

#### See Also

see [qmerrno.h](#) for eQCWWAN\_XXX error values

Timeout: 2 seconds\n

- 9.39.5.21 **ULONG** SetActiveMobileIPProfile ( **CHAR** \* *pSPC*, **BYTE** *index* )

Sets active mobile IP profile.

#### Parameters

<i>pSPC</i> [IN]	<ul style="list-style-type: none"> <li>NULL-terminated string representing six digit service programming code</li> </ul>
<i>index</i> [IN]	<ul style="list-style-type: none"> <li>Index of the profile to be set as the active profile</li> </ul>

#### Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_XXX error value otherwise

#### See Also

see [qmerrno.h](#) for eQCWWAN\_XXX error values

#### Note

Technology Supported: CDMA  
Timeout: 2 seconds

#### 9.39.5.22 ULONG SetAutoconnect ( ULONG *setting* )

Sets the auto connect data session setting.

## Parameters

<i>setting</i>	<ul style="list-style-type: none"> <li>• NDIS autoconnect setting <ul style="list-style-type: none"> <li>– 0 - Disabled</li> <li>– 1 - Enabled</li> </ul> </li> </ul>
----------------	---

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

When enabling, timeout is 5 minutes,  
When disabling, timeout is 5 seconds

**9.39.5.23** **ULONG** SetDefaultProfile ( **ULONG** *profileType*, **ULONG** \* *pPDPTType*, **ULONG** \* *pIPAddress*, **ULONG** \* *pPrimaryDNS*, **ULONG** \* *pSecondaryDNS*, **ULONG** \* *pAuthentication*, **CHAR** \* *pName*, **CHAR** \* *pAPNName*, **CHAR** \* *pUsername*, **CHAR** \* *pPassword* )

Writes the default profile settings to the device. The default profile is used to establish an autoconnect data session.

## Parameters

<i>profileType</i>	<ul style="list-style-type: none"> <li>• Type of profile <ul style="list-style-type: none"> <li>– 0 - UMTS</li> </ul> </li> </ul>
<i>pPDPTType[IN]</i>	<ul style="list-style-type: none"> <li>• Packet Data Protocol (PDP) type specifies the type of data payload exchanged over the air link when the packet data session is established with this profile (optional) <ul style="list-style-type: none"> <li>– 0 - PDP-IP (IPv4)</li> </ul> </li> </ul>
<i>pIPAddress[IN]</i>	<ul style="list-style-type: none"> <li>• Preferred IPv4 addr to be assigned to device (optional)</li> </ul>
<i>pPrimaryDNS[IN]</i>	<ul style="list-style-type: none"> <li>• Primary DNS Ipv4 address preference (optional)</li> </ul>
<i>pSecondaryDNS[IN]</i>	<ul style="list-style-type: none"> <li>• Secondary DNS Ipv4 address preference (optional)</li> </ul>

<i>pAuthentication</i> [IN]	<ul style="list-style-type: none"> <li>• Bitmap that indicates authentication algorithm preference (optional) <ul style="list-style-type: none"> <li>– 0x00000001 - PAP preference <ul style="list-style-type: none"> <li>* 0 - Never performed</li> <li>* 1 - May be performed</li> </ul> </li> <li>– 0x00000002 - CHAP preference <ul style="list-style-type: none"> <li>* 0 - Never performed</li> <li>* 1 - May be performed</li> </ul> </li> <li>– All other bits are reserved and must be set to 0</li> <li>– If more than 1 bit is set, then device decides which authentication procedure is performed while setting up data session e.g. the device may have a policy to select the most secure authentication mechanism.</li> </ul> </li> </ul>
<i>pName</i> [IN]	<ul style="list-style-type: none"> <li>• profile Name (optional)</li> </ul>
<i>pAPNName</i> [IN]	<ul style="list-style-type: none"> <li>• Access point name. NULL-terminated string parameter that is a logical name used to select GGSN and external packet data network (optional)</li> <li>• If value is NULL or omitted, then subscription default value will be requested.</li> </ul>
<i>pUsername</i> [IN]	<ul style="list-style-type: none"> <li>• Username used during network authentication (optional)</li> </ul>
<i>pPassword</i> [IN]	<ul style="list-style-type: none"> <li>• Password used during network authentication (optional)</li> </ul>

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Timeout is 2 seconds.

**9.39.5.24** **ULONG** SetDefaultProfileLTE ( **ULONG** *profileType*, **ULONG** \* *pPDPTType*, **ULONG** \* *pIPAddressv4*, **ULONG** \* *pPrimaryDNSv4*, **ULONG** \* *pSecondaryDNSv4*, **USHORT** \* *pIPAddressv6*, **USHORT** \* *PrimaryDNSv6*, **USHORT** \* *pSecondaryDNSv6*, **ULONG** \* *pAuthentication*, **CHAR** \* *pName*, **CHAR** \* *pAPNName*, **CHAR** \* *pUsername*, **CHAR** \* *pPassword* )

Writes the default profile settings to the device. The default profile is used to establish an auto connect data session.

## Parameters

<i>profileType</i>	<ul style="list-style-type: none"> <li>Type of profile <ul style="list-style-type: none"> <li>0 - UMTS</li> </ul> </li> </ul>
<i>pPDPTType[IN]</i>	<ul style="list-style-type: none"> <li>Packet Data Protocol (PDP) type specifies the type of data payload exchanged over the air link when the packet data session is established with this profile (optional) <ul style="list-style-type: none"> <li>0 - PDP-IP (IPv4)</li> </ul> </li> </ul>
<i>pIPAddressv4[IN]</i>	<ul style="list-style-type: none"> <li>Preferred IPv4 address to be assigned to device (optional)</li> </ul>
<i>pPrimaryDN-Sv4[IN]</i>	<ul style="list-style-type: none"> <li>Primary DNS Ipv4 address preference (optional)</li> </ul>
<i>pSecondaryDN-Sv4[IN]</i>	<ul style="list-style-type: none"> <li>Secondary DNS Ipv4 address preference (optional)</li> </ul>
<i>pIPAddressv6[IN]</i>	<ul style="list-style-type: none"> <li>Preferred IPv6 address to be assigned to device (optional)</li> </ul>
<i>pPrimaryDN-Sv6[IN]</i>	<ul style="list-style-type: none"> <li>Primary DNS Ipv6 address preference (optional)</li> </ul>
<i>pSecondaryDN-Sv6[IN]</i>	<ul style="list-style-type: none"> <li>Secondary DNS Ipv6 address preference (optional)</li> </ul>
<i>pAuthentication[IN]</i>	<ul style="list-style-type: none"> <li>Bitmap that indicates authentication algorithm preference (optional) <ul style="list-style-type: none"> <li>0x00000001 - PAP preference <ul style="list-style-type: none"> <li>0 - Never performed</li> <li>1 - May be performed</li> </ul> </li> <li>0x00000002 - CHAP preference <ul style="list-style-type: none"> <li>0 - Never performed</li> <li>1 - May be performed</li> </ul> </li> <li>All other bits are reserved and must be set to 0</li> <li>If more than 1 bit is set, then device decides which authentication procedure is performed while setting up data session e.g.the device may have a policy to select the most secure authentication mechanism.</li> </ul> </li> </ul>

<i>pName</i> [IN]	<ul style="list-style-type: none"> <li>• profile Name (optional)</li> </ul>
<i>pAPNName</i> [IN]	<ul style="list-style-type: none"> <li>• Access point name. NULL-terminated string parameter that is a logical name used to select GGSN and external packet data network (optional)</li> <li>• If value is NULL or omitted, then subscription default value will be requested</li> </ul>
<i>pUsername</i> [IN]	<ul style="list-style-type: none"> <li>• Username used during network authentication (optional)</li> </ul>
<i>pPassword</i> [IN]	<ul style="list-style-type: none"> <li>• Password used during network authentication (optional)</li> </ul>

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Deprecated, please use SetDefaultProfileLTEV2 instead  
 Technology Supported: LTE  
 Timeout: 2 seconds

**9.39.5.25** **ULONG** SetDefaultProfileLTEV2 ( **ULONG** *profileType*, **ULONG** \* *pPDPTType*, **ULONG** \* *plIPv4Address*, **ULONG** \* *pPrimaryDNSv4*, **ULONG** \* *pSecondaryDNSv4*, **USHORT** \* *plIPv6Address*, **USHORT** \* *pPrimaryDNSv6*, **USHORT** \* *pSecondaryDNSv6*, **ULONG** \* *pAuthentication*, **CHAR** \* *pName*, **CHAR** \* *pAPNName*, **CHAR** \* *pUsername*, **CHAR** \* *pPassword* )

Writes the default profile settings to the device. The default profile is used to establish an auto connect data session.

**Parameters**

<i>profileType</i>	<ul style="list-style-type: none"> <li>• Type of profile           <ul style="list-style-type: none"> <li>– 0 - UMTS</li> </ul> </li> </ul>
<i>pPDPTType</i> [IN]	<ul style="list-style-type: none"> <li>• Packet Data Protocol (PDP) type specifies the type of data payload exchanged over the air link when the packet data session is established with this profile (optional)           <ul style="list-style-type: none"> <li>– 0 - PDP-IP (IPv4)</li> </ul> </li> </ul>

<i>pIPvAddressv4</i> [I-N]	<ul style="list-style-type: none"> <li>Preferred IPv4 address to be assigned to device (optional)</li> </ul>
<i>pPrimaryDN-Sv4</i> [IN]	<ul style="list-style-type: none"> <li>Primary DNS Ipv4 address preference (optional)</li> </ul>
<i>pSecondaryDN-Sv4</i> [IN]	<ul style="list-style-type: none"> <li>Secondary DNS Ipv4 address preference (optional)</li> </ul>
<i>pIPvAddressv6</i> [I-N]	<ul style="list-style-type: none"> <li>Preferred IPv6 addr to be assigned to device (optional)</li> </ul>
<i>pPrimaryDN-Sv6</i> [IN]	<ul style="list-style-type: none"> <li>Primary DNS Ipv6 address preference (optional)</li> </ul>
<i>pSecondaryDN-Sv6</i> [IN]	<ul style="list-style-type: none"> <li>Secondary DNS Ipv6 address preference (optional)</li> </ul>
<i>pAuthentication</i> [I-N]	<ul style="list-style-type: none"> <li>Bitmap that indicates authentication algorithm preference (optional) <ul style="list-style-type: none"> <li>0x00000001 - PAP preference <ul style="list-style-type: none"> <li>0 - Never performed</li> <li>1 - May be performed</li> </ul> </li> <li>0x00000002 - CHAP preference <ul style="list-style-type: none"> <li>0 - Never performed</li> <li>1 - May be performed</li> </ul> </li> <li>All other bits are reserved and must be set to 0</li> <li>If more than 1 bit is set, then device decides which authentication procedure is performed while setting up data session e.g.the device may have a policy to select the most secure authentication mechanism.</li> </ul> </li> </ul>
<i>pName</i> [IN]	<ul style="list-style-type: none"> <li>profile Name (optional)</li> </ul>
<i>pAPNName</i> [IN]	<ul style="list-style-type: none"> <li>Access point name. NULL-terminated string parameter that is a logical name used to select GGSN and external packet data network (optional)</li> <li>If value is NULL or omitted, then subscription default value will be requested</li> </ul>
<i>pUsername</i> [IN]	<ul style="list-style-type: none"> <li>Username used during network authentication (optional)</li> </ul>
<i>pPassword</i> [IN]	<ul style="list-style-type: none"> <li>Password used during network authentication (optional)</li> </ul>

#### Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise



#### See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

#### Note

Technology Supported: LTE

Timeout: 2 seconds

Replaces deprecated Function SetDefaultProfileLTE

#### 9.39.5.26 ULONG SetMobileIP ( ULONG mode )

Sets the current mobile IP setting.

##### Parameters

<i>mode</i>	
	<ul style="list-style-type: none"><li>• Mobile IP setting<ul style="list-style-type: none"><li>– 0 - Mobile IP off (simple IP only)</li><li>– 1 - Mobile IP preferred</li><li>– 2 - Mobile IP only</li></ul></li></ul>

#### Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

#### See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Technology Supported: CDMA

Timeout: 2 seconds

**9.39.5.27** **ULONG** SetMobileIPParameters ( **CHAR** \* *pSPC*, **ULONG** \* *pMode*, **BYTE** \* *pRetryLimit*, **BYTE** \* *pRetryInterval*, **BYTE** \* *pReRegPeriod*, **BYTE** \* *pReRegTraffic*, **BYTE** \* *pHAAAuthenticator*, **BYTE** \* *pHA2002bis* )

Sets the specified mobile IP parameters.

## Parameters

<i>pSPC</i> [IN]	<ul style="list-style-type: none"> <li>• NULL-terminated string representing six digit service programming code.</li> </ul>
<i>pMode</i> [IN]	<ul style="list-style-type: none"> <li>• Mode to be set (optional) <ul style="list-style-type: none"> <li>– 0 - Mobile IP off (simple IP only)</li> <li>– 1 - Mobile IP preferred</li> <li>– 2 - Mobile IP only</li> </ul> </li> </ul>
<i>pRetryLimit</i> [IN]	<ul style="list-style-type: none"> <li>• Registration retry attempt limit (optional)</li> </ul>
<i>pRetryInterval</i> [IN]	<ul style="list-style-type: none"> <li>• Registration retry attempt interval used to determine the time between registration attempts (optional)</li> </ul>
<i>pReRegPeriod</i> [IN]	<ul style="list-style-type: none"> <li>• Period (in minutes) to attempt re-registration before current registration expires (optional)</li> </ul>
<i>pReRegTraffic</i> [IN]	<ul style="list-style-type: none"> <li>• Re-registration only if traffic since last attempt (optional) <ul style="list-style-type: none"> <li>– Zero - Disabled</li> <li>– NonZero - Enabled</li> </ul> </li> </ul>
<i>pHAAAuthenticator</i> [IN]	<ul style="list-style-type: none"> <li>• MH-HA authenticator calculator (optional) <ul style="list-style-type: none"> <li>– Zero - Disabled</li> <li>– NonZero - Enabled</li> </ul> </li> </ul>
<i>pHA2002bis</i> [IN]	<ul style="list-style-type: none"> <li>• MH-HA RFC 2002bis authentication instead of RFC2002 (optional) <ul style="list-style-type: none"> <li>– Zero - Disabled</li> <li>– NonZero - Enabled</li> </ul> </li> </ul>

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Technology Supported: CDMA

Device Supported: None

Timeout: 2 seconds

**9.39.5.28** **ULONG** SetMobileIPProfile ( **CHAR** \* *pSPC*, **BYTE** *index*, **BYTE** \* *pEnabled*, **ULONG** \* *pAddress*, **ULONG** \* *pPrimaryHA*, **ULONG** \* *pSecondaryHA*, **BYTE** \* *pRevTunneling*, **CHAR** \* *pNAI*, **ULONG** \* *pHASPI*, **ULONG** \* *pAAASPI*, **CHAR** \* *pMNHA*, **CHAR** \* *pMNAAA* )

Sets the mobile IP parameters.

## Parameters

<i>pSPC</i> [IN]	<ul style="list-style-type: none"> <li>Six digit service programming code string</li> </ul>
<i>index</i> [IN]	<ul style="list-style-type: none"> <li>Index of the profile to modify</li> </ul>
<i>pEnabled</i> [IN]	<ul style="list-style-type: none"> <li>(Optional) Enable profile? 0 - Disabled Nonzero - Enabled</li> </ul>
<i>pAddress</i> [IN]	<ul style="list-style-type: none"> <li>(Optional) Home IPv4 address</li> </ul>
<i>pPrimaryHA</i> [IN]	<ul style="list-style-type: none"> <li>(Optional) Primary home agent IPv4 address</li> </ul>
<i>pSecondaryHA</i> [IN]	<ul style="list-style-type: none"> <li>(Optional) Secondary home agent IPv4 address</li> </ul>
<i>pRevTunneling</i> [IN]	<ul style="list-style-type: none"> <li>(Optional) Enable reverse tunneling? 0 - Disabled Nonzero - Enabled</li> </ul>
<i>pNAI</i> [IN]	<ul style="list-style-type: none"> <li>(Optional) Network access identifier string</li> </ul>
<i>pHASPI</i> [IN]	<ul style="list-style-type: none"> <li>(Optional) Home agent security parameter index</li> </ul>
<i>pAAASPI</i> [IN]	<ul style="list-style-type: none"> <li>(Optional) AAA server security parameter index</li> </ul>

<i>pMNHA</i> [IN]	<ul style="list-style-type: none"> <li>• (Optional) MN-HA key string</li> </ul>
<i>pMNA</i> [IN]	<ul style="list-style-type: none"> <li>• (Optional) MN-AAA key string</li> </ul>

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

see [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Technology Supported: CDMA  
Timeout: 2 seconds

**9.39.5.29 ULONG SLQSAutoConnect ( struct slqsautoconnect \* pcreq )**

Returns auto connect settings

**Parameters**

<i>slqsautoconnect</i>	<ul style="list-style-type: none"> <li>• SLQS auto connect settings</li> </ul>
------------------------	--

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

see [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Technology Supported: CDMA/UMTS  
Device Supported: MC83x5, MC7700  
Timeout: 2 seconds

**9.39.5.30 ULONG SLQSCreateProfile ( struct CreateProfileIn \* pReq, struct CreateProfileOut \* pResp )**

Create a new profile with the specified parameters. Note that some firmware versions do not support the optional Profile ID parameter. In this case an error will be returned and the caller can subsequently create a profile by specifying a NULL pointer for the Profile ID parameter. The Profile ID pertaining to the newly created profile is returned in the response structure (pResp).

## Parameters

<i>pReq[IN]</i>	<ul style="list-style-type: none"> <li>• SLQS Create profile Information</li> </ul>
<i>pResp[OUT]</i>	<ul style="list-style-type: none"> <li>• SLQS profile identifier information</li> </ul>

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

see [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Technology Supported: UMTS  
 Device Supported: MC83x5, MC7700  
 Timeout: 2 seconds

**9.39.5.31** **ULONG** SLQSDeleteProfile ( struct SLQSDeleteProfileParams \* *pProfileToDelete*, **WORD** \* *pExtendedErrorCode* )

Deletes a configured profile stored on the device. The deletion of a profile does not affect profile index assignments.

## Parameters

<i>pProfileToDelete</i>	<ul style="list-style-type: none"> <li>• Information about the profile to be deleted.</li> <li>• See <a href="#">SLQSDeleteProfileParams</a> for more details.</li> </ul>
<i>pExtendedError-Code</i>	<ul style="list-style-type: none"> <li>• The extended error code received from DS Profile subsystem of type eWDS_ERR_PROFILE_REG_xxx.</li> <li>• Error code will only will be present if error code eQCWWAN_ERR_QMI_EXTENDED_INTERNAL is returned by device.</li> <li>• See <a href="#">qm_wds_ds_profile_extended_err_codes</a> enum in <a href="#">qmerrno.h</a> for received error description.</li> </ul>

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

see [qmerrno.h](#) for eQCWWAN\_xxx error values.

## Note

Timeout: 2 seconds

9.39.5.32 **ULONG** SLQSGet3GPPConfigItem ( **slqs3GPPConfigItem** \* *pSLQS3GPPConfigItem* )

Reads the 3gpp configuration item.

## Parameters

<i>pSLQS3GPP-ConfigItem</i> [OUT]	<ul style="list-style-type: none"> <li>See <a href="#">slqs3GPPConfigItem</a> for more information</li> </ul>
-----------------------------------	---

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Technology Supported: UMTS/LTE  
Timeout: 2 seconds

### 9.39.5.33 ULONG SLQSGetByteTotals ( struct WdsByteTotals \* pByteTotals )

This API request the device to fetch ByteTotals for IPV4 and IPV6.

## Parameters

<i>pByteTotals</i> [IN/-OUT]	<ul style="list-style-type: none"> <li>See <a href="#">WdsByteTotals</a> for more information</li> </ul>
------------------------------	--

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

see [qmerrno.h](#) for eQCWWAN\_xxx error values

Timeout: 2 seconds\n

### 9.39.5.34 ULONG SLQSGetConnectionRate ( struct WdsConnectionRate \* pConnectionRate )

This API request the device to fetch ConnectionRate. It can be used for both mono and multiple PDN use case.

## Parameters

<i>pConnection-Rate</i> [IN/OUT]	<ul style="list-style-type: none"> <li>See <a href="#">WdsConnectionRate</a> for more information</li> </ul>
----------------------------------	--

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

see [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Technology Supported: UMTS/CDMA  
 Device Supported: MC77XX  
 Timeout: 2 seconds

**9.39.5.35    ULONG SLQSGetCurrDataSystemStat ( CurrDataSysStat \* pCurrDataSysStat )**

This API request the device to fetch current data system status.

**Parameters**

<i>pCurrDataSysStat</i> [IN/OUT]	<ul style="list-style-type: none"> <li>• See <a href="#">CurrDataSysStat</a> for more information</li> </ul>
----------------------------------	--

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

see [qmerrno.h](#) for eQCWWAN\_xxx error values

Timeout: 5 seconds\n

**9.39.5.36    ULONG SLQSGetCurrentChannelRate ( WDSSWICurrentChannelRates \* pRates, BYTE instance )**

This API Queries current bitrate of a packet data connection.

**Parameters**

<i>pRates</i>	[IN]	<ul style="list-style-type: none"> <li>• See <a href="#">WDSSWICurrentChannelRates</a> for more information</li> </ul>
---------------	------	--

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

see [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

This feature depends on custom feature setting IPCHANNELRATEEN which can be set via SLQSSetCust-Features

Timeout: 2 seconds

**9.39.5.37    ULONG SLQSGetDataBearerTechnology ( QmiWDSDataBearers \* pDataBearers, BYTE instance )**

Retrieves the data bearer technology values for current and/or last data calls. The device must be in a data call for this function to execute successfully.



## Parameters

<i>qmiWDSData-Bearers[OUT]</i>	<ul style="list-style-type: none"> <li>Indicates the current and the last call data bearer technology. Should not be NULL, on input.</li> </ul>
<i>instance</i>	<ul style="list-style-type: none"> <li>PDP instance</li> </ul>

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Technology Supported: UMTS/CDMA  
 Device Supported: MC7750,GOBI,MC7700  
 Timeout: 2 seconds

#### 9.39.5.38 ULONG SLQSGetDataBearerTechnologyExt ( DataBearerTechExt \* pDataBearerTech, BYTE instance )

This API Get Data Bearer Technology. This is a new API to replace API [GetDataBearerTechnology\(\)](#). see the description of [GetDataBearerTechnology\(\)](#) to get more information

## Parameters

<i>pDataBearer-Tech</i>	[IN] <ul style="list-style-type: none"> <li>See <a href="#">DataBearerTechExt</a> for more information</li> </ul>
<i>instance</i>	[IN] <ul style="list-style-type: none"> <li>PDP instance</li> </ul>

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

see [qmerrno.h](#) for eQCWWAN\_xxx error values

Timeout: 2 seconds\n

#### 9.39.5.39 ULONG SLQSGetDUNCallInfo ( getDUNCallInfoReq \* pGetDUNCallInfoReq, getDUNCallInfoResp \* pGetDUNCallInfoResp )

This API queries the current modem connection status.

## Parameters

<i>pGetDUNCall-InfoReq[IN]</i>	<ul style="list-style-type: none"> <li>• See <a href="#">getDUNCallInfoReq</a> for more information</li> </ul>
<i>pGetDUNCall-InfoResp[OUT]</i>	<ul style="list-style-type: none"> <li>• See <a href="#">getDUNCallInfoResp</a> for more information</li> </ul>

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

see [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 5 seconds

#### 9.39.5.40 **ULONG** SLQSGetPacketStatistics ( struct WdsPktStatisticsReq \* *pStatMask*, struct WdsPktStatisticsResp \* *pPktStatistics* )

This API request the device to fetch current packet transfer counter values from the device

## Parameters

• <i>pStatMask[IN]</i>	– See <a href="#">WdsPktStatisticsReq</a> for more information
<i>pPktStatistics[OUT]</i>	– See <a href="#">WdsPktStatisticsResp</a> for more information

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

see [qmerrno.h](#) for eQCWWAN\_xxx error values

Timeout: 5 seconds\n

#### 9.39.5.41 **ULONG** SLQSGetProfile ( **ULONG** *profileType*, **BYTE** *profileId*, **ULONG** \* *pPDPTType*, **ULONG** \* *piPAddress*, **ULONG** \* *pPrimaryDNS*, **ULONG** \* *pSecondaryDNS*, **ULONG** \* *pAuthentication*, **BYTE** *nameSize*, **CHAR** \* *pName*, **BYTE** *apnSize*, **CHAR** \* *pAPNName*, **BYTE** *userSize*, **CHAR** \* *pUsername*, **WORD** \* *pExtendedErrorCode* )

Reads the profile settings from the device for the specified profile id.

## Parameters

<i>profileType</i>	<ul style="list-style-type: none"> <li>Type of profile <ul style="list-style-type: none"> <li>0 - UMTS</li> </ul> </li> </ul>
<i>profileId</i>	<ul style="list-style-type: none"> <li>Index of the configured profile for which settings are read <ul style="list-style-type: none"> <li>Value between 1 - 16</li> </ul> </li> </ul>
<i>pPDPTType[OUT]</i>	<ul style="list-style-type: none"> <li>Packet Data Protocol (PDP) type specifies the type of data payload exchanged over the air link when the packet data session is established with this profile <ul style="list-style-type: none"> <li>0 - PDP-IP (IPv4)</li> </ul> </li> </ul>
<i>pIPAddress[OUT]</i>	<ul style="list-style-type: none"> <li>Preferred IPv4 address to be assigned to device</li> </ul>
<i>pPrimaryDNS[OUT]</i>	<ul style="list-style-type: none"> <li>Primary DNS Ipv4 address preference</li> </ul>
<i>pSecondaryDNS[OUT]</i>	<ul style="list-style-type: none"> <li>Secondary DNS Ipv4 address preference</li> </ul>
<i>pAuthentication[OUT]</i>	<ul style="list-style-type: none"> <li>Bitmap that indicates authentication algorithm preference <ul style="list-style-type: none"> <li>0x00000001 - PAP preference <ul style="list-style-type: none"> <li>0 - Never performed</li> <li>1 - May be performed</li> </ul> </li> <li>0x00000002 - CHAP preference <ul style="list-style-type: none"> <li>0 - Never performed</li> <li>1 - May be performed</li> </ul> </li> <li>All other bits are reserved and must be set to 0</li> <li>If more than 1 bit is set, then device decides which authentication procedure is performed while setting up data session e.g. the device may have a policy to select the most secure authentication mechanism.</li> </ul> </li> </ul>

<i>nameSize</i>	<ul style="list-style-type: none"> <li>Maximum number of characters (including NULL terminator) that profile name array can contain.</li> </ul>
<i>pName[OUT]</i>	<ul style="list-style-type: none"> <li>Profile name</li> </ul>
<i>apnSize</i>	<ul style="list-style-type: none"> <li>Maximum number of characters (including NULL terminator) that APN name array can contain</li> </ul>
<i>pAPNName[OUT]</i>	<ul style="list-style-type: none"> <li>Access point name. NULL-terminated string parameter that is a logical name used to select GGSN and external packet data network.</li> <li>If value is NULL or omitted, then subscription default value will be requested.</li> </ul>
<i>userSize</i>	<ul style="list-style-type: none"> <li>Maximum number of characters (including NULL terminator) that username array can contain.</li> </ul>
<i>pUsername[OUT]</i>	<ul style="list-style-type: none"> <li>Username used during network authentication</li> </ul>
<i>pExtendedErrorCode</i>	<ul style="list-style-type: none"> <li>The extended error code received from DS Profile subsystem of type eWDS_ERR_PROFILE_REG_xxx.</li> <li>Error code will only be present if error code eQCWWAN_ERR_QMI_EXTENDED_INTERNAL is returned by device.</li> <li>See <a href="#">qm_wds_ds_profile_extended_err_codes</a> enum in <a href="#">qmerrno.h</a> for received error description.</li> </ul>

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Timeout: 2 seconds

#### 9.39.5.42 ULONG SLQSGetProfileSettings ( GetProfileSettingIn \* pReq, GetProfileSettingOut \* pResp )

Retrieves a profile(3GPP/3GPP2) with the specified parameters.

## Parameters

<i>pReq[IN]</i>	<ul style="list-style-type: none"> <li>• details of the profile to be fetched</li> <li>• See <a href="#">GetProfileSettingIn</a> for more information</li> </ul>
<i>pResp[OUT]</i>	<ul style="list-style-type: none"> <li>• The profile settings and/or extended error code returned by the device based on input parameters.</li> <li>• See <a href="#">GetProfileSettingOut</a> for more information</li> </ul>

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

see [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 2 seconds

#### 9.39.5.43 ULONG SLQSGetRuntimeSettings ( struct WdsRunTimeSettings \* *pRunTimeSettings* )

Returns the packet data session settings currently in use.

## Parameters

<i>pRunTime-Settings[OUT]</i>	<ul style="list-style-type: none"> <li>• SLQS Runtime Settings Information</li> </ul>
-------------------------------	---

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

see [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 2 seconds

#### 9.39.5.44 ULONG SLQSGetSessionState ( ULONG \* *pStateV4*, ULONG \* *pStateV6*, BYTE *instance* )

Returns the state of the current packet data session.

## Parameters

<i>pStateV4[OUT]</i>	<ul style="list-style-type: none"> <li>• Current link status for IPV4 Session <ul style="list-style-type: none"> <li>– 1 - DISCONNECTED</li> <li>– 2 - CONNECTED</li> <li>– 3 - SUSPENDED (not supported)</li> <li>– 4 - AUTHENTICATING</li> </ul> </li> </ul>
<i>pStateV6[OUT]</i>	<ul style="list-style-type: none"> <li>• Current link status for IPV6 Session <ul style="list-style-type: none"> <li>– 1 - DISCONNECTED</li> <li>– 2 - CONNECTED</li> <li>– 3 - SUSPENDED (not supported)</li> <li>– 4 - AUTHENTICATING</li> </ul> </li> </ul>
<i>instance</i>	<ul style="list-style-type: none"> <li>• PDP instance</li> </ul>

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 2 seconds

#### 9.39.5.45 ULONG SLQSMModifyProfile ( struct ModifyProfileIn \* *pReq*, struct ModifyProfileOut \* *pResp* )

Modify a profile(3GPP/3GPP2) with the specified parameters.

## Parameters

<i>pReq[IN]</i>	<ul style="list-style-type: none"> <li>• Contains parameters which can be modified</li> </ul>
<i>pResp[OUT]</i>	<ul style="list-style-type: none"> <li>• Contains parameters which indicates modification success or failure</li> </ul>

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

see [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Timeout: 2 seconds

**9.39.5.46    ULONG SLQSResetPacketStatics (    )**

This API request the device to reset packet data transfer statistics.

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

see [qmerrno.h](#) for eQCWWAN\_xxx error values

Timeout: 2 seconds\n

**9.39.5.47    ULONG SLQSSet3GPPConfigItem ( slqs3GPPConfigItem \* pSLQS3GPPConfigItem )**

Sets the 3gpp configuration item.

**Parameters**

<i>pSLQS3GPP-ConfigItem</i> [IN]	<ul style="list-style-type: none"> <li>See <a href="#">slqs3GPPConfigItem</a> for more information</li> </ul>
----------------------------------	---

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Technology Supported: UMTS/LTE  
Timeout: 2 seconds

**9.39.5.48    ULONG SLQSSetHostMTU ( int \* mtu, BYTE instance )**

This API Set Network Adaptor MTU.

**Parameters**

<i>mtu</i> [IN]	<ul style="list-style-type: none"> <li>• MTU value</li> </ul>
<i>instance</i>	[IN] <ul style="list-style-type: none"> <li>• PDP instance</li> </ul>

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

see [qmerrno.h](#) for eQCWWAN\_xxx error values

Timeout: 2 seconds\n

**9.39.5.49** **ULONG** SLQSSetProfile ( **ULONG** *profileType*, **BYTE** *profileId*, **ULONG** \* *pPDPTType*, **ULONG** \* *pIPAddress*, **ULONG** \* *pPrimaryDNS*, **ULONG** \* *pSecondaryDNS*, **ULONG** \* *pAuthentication*, **CHAR** \* *pName*, **CHAR** \* *pAPNName*, **CHAR** \* *pUsername*, **CHAR** \* *pPassword* )

Writes the profile settings for the specified profile Id.

**Parameters**

<i>profileType</i>	<ul style="list-style-type: none"> <li>• Type of profile               <ul style="list-style-type: none"> <li>– 0 - UMTS</li> </ul> </li> </ul>
<i>profileId</i>	<ul style="list-style-type: none"> <li>• Profile number to be modified               <ul style="list-style-type: none"> <li>– Value between 1 - 16</li> </ul> </li> </ul>
<i>pPDPTType</i> [IN]	<ul style="list-style-type: none"> <li>• Packet Data Protocol (PDP) type specifies the type of data payload exchanged over the air link when the packet data session is established with this profile (optional)               <ul style="list-style-type: none"> <li>– 0 - PDP-IP (IPv4)</li> </ul> </li> </ul>
<i>pIPAddress</i> [IN]	<ul style="list-style-type: none"> <li>• Preferred IPv4 address to be assigned to device (optional)</li> </ul>
<i>pPrimaryDNS</i> [I-N]	<ul style="list-style-type: none"> <li>• Primary DNS Ipv4 address preference (optional)</li> </ul>
<i>pSecondaryDNS</i> [IN]	<ul style="list-style-type: none"> <li>• Secondary DNS Ipv4 address preference (optional)</li> </ul>



<i>pAuthentication</i> [IN]	<ul style="list-style-type: none"> <li>• Bitmap that indicates authentication algorithm preference (optional) <ul style="list-style-type: none"> <li>– 0x00000001 - PAP preference <ul style="list-style-type: none"> <li>* 0 - Never performed</li> <li>* 1 - May be performed</li> </ul> </li> <li>– 0x00000002 - CHAP preference <ul style="list-style-type: none"> <li>* 0 - Never performed</li> <li>* 1 - May be performed</li> </ul> </li> <li>– All other bits are reserved and must be set to 0</li> <li>– If more than 1 bit is set, then device decides which authentication procedure is performed while setting up data session e.g. the device may have a policy to select the most secure authentication mechanism.</li> </ul> </li> </ul>
<i>pName</i> [IN]	<ul style="list-style-type: none"> <li>• profile Name (optional)</li> </ul>
<i>pAPNName</i> [IN]	<ul style="list-style-type: none"> <li>• Access point name. NULL-terminated string parameter that is a logical name used to select GGSN and external packet data network (optional)</li> <li>• If value is NULL or omitted, then subscription default value will be requested.</li> </ul>
<i>pUsername</i> [IN]	<ul style="list-style-type: none"> <li>• Username used during network authentication (optional)</li> </ul>
<i>pPassword</i> [IN]	<ul style="list-style-type: none"> <li>• Password used during network authentication (optional)</li> </ul>

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Timeout is 2 seconds.

**9.39.5.50 ULONG SLQSSGetLoopback ( WDSGetLoopbackData \* data )**

This API to Get the value of loopback mode and multiplier.

**Parameters**

<i>pReq</i>	[IN]
	<ul style="list-style-type: none"> <li>• See <a href="#">WDSGetLoopbackData</a> for more information</li> </ul>

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

see [qmerrno.h](#) for eQCWWAN\_xxx error values

Timeout: 2 seconds\n

**9.39.5.51 ULONG SLQSSetLoopback ( WDSsetLoopbackData \* pReq )**

This API to Enable/disable Data Loopback Mode and set the value of loopback multiplier.

**Parameters**

<i>pReq</i>	[IN]
<ul style="list-style-type: none"> <li>See <a href="#">WDSsetLoopbackData</a> for more information</li> </ul>	

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

see [qmerrno.h](#) for eQCWWAN\_xxx error values

Timeout: 2 seconds\n

**9.39.5.52 ULONG SLQSStartStopDataSession ( struct ssdatasession\_params \* pin )**

Starts or stops a 3GPP/3GPP2 data session on a preconfigured profile. To set the IP family for the data session, execute SLQSSetIPFamilyPreference prior to calling this API.

**Parameters**

<i>pin</i> [IN]	
<ul style="list-style-type: none"> <li><a href="#">ssdatasession_params</a> structure</li> <li>See <a href="#">ssdatasession_params</a> for more details</li> </ul>	

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See Also**

see [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Timeout: 5 Minutes

Use [SLQSSetProfile](#) to configure 3GPP profiles

**9.39.5.53 ULONG SLQSWdsGoActive ( void )**

Forces the device to immediately reestablish the traffic channel on the serving radio interface

## Parameters

<i>None</i>	
-------------	--

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

This channel can go dormant any time after it has been reactivated. There is no assurance that the channel remains active for any guaranteed period. Timeout: 5 seconds

**9.39.5.54   ULONG SLQSWdsGoDormant ( void )**

Forces the device to immediately drop the traffic channel on the serving radio interface

## Parameters

<i>None</i>	
-------------	--

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

This channel can be reactivated as soon as data is sent over the network interface. There is no assurance that the channel remains dormant for any guaranteed period. Timeout: 5 seconds

**9.39.5.55   ULONG SLQSWdsSetEventReport ( wdsSetEventReportReq \* pSetEventReportReq )**

This API sets the wireless data connection state reporting conditions for the requesting control point.

## Parameters

<i>pSetEvent- ReportReq[IN]</i>	<ul style="list-style-type: none"><li>See <a href="#">wdsSetEventReportReq</a> for more information.</li></ul>
-------------------------------------	--

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See Also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Timeout: 2 seconds

The control point event reporting state variables are modified to reflect the settings indicated in the request message. The service maintains a set of state variables for each control point. Relevant wireless data connection state changes are communicated to the registered WDS control point via the SLQSWdsSetEventReport-CallBack. The AT command equivalents to this command are AT+CMER, AT+CIND, and AT+CIEV

**9.39.5.56** `ULONG SLQSWdsSwtPDPRuntimeSettings ( swtPDPRuntimeSettingsReq * pPDPRuntimeSettingsReq, swtPDPRuntimeSettingsResp * pPDPRuntimeSettingsResp )`

This API requests the device to return the active PDP context associated with a context id.

**Parameters**

<i>pPDPRuntimeSettingsReq</i> [IN]	<ul style="list-style-type: none"> <li>See <a href="#">swtPDPRuntimeSettingsReq</a> for more information</li> </ul>
<i>pPDPRuntimeSettingsResp</i> [OUT]	<ul style="list-style-type: none"> <li>See <a href="#">swtPDPRuntimeSettingsResp</a> for more information</li> </ul>

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_XXX error value otherwise

**See Also**

see [qmerrno.h](#) for eQCWWAN\_XXX error values

**Note**

Technology Supported: UMTS/CDMA

Device Supported: MC77XX

Timeout: 2 seconds

The AT command equivalent to this command is AT+CGCONTRDP

**9.39.5.57** `BOOL WDS_IsGobiDevice ( )`

## 9.40 qaNasGetRFBandInfo.h File Reference

**Data Structures**

- struct [QmiNasGetRFBandInfoResp](#)

**Enumerations**

- enum [eQMI\\_NAS\\_GET\\_RF\\_INFO\\_RESP](#) { [eTLV\\_RF\\_BAND\\_INFO](#) = 0x01 }

**Functions**

- enum [eQCWWANError](#) [PkQmiNasGetRFBandInfo](#) (WORD \*pMlength, BYTE \*pBuffer)
- enum [eQCWWANError](#) [UpkQmiNasGetRFBandInfo](#) (BYTE \*pMdmResp, struct [QmiNasGetRFBandInfoResp](#) \*pApiResp)

### 9.40.1 Enumeration Type Documentation

#### 9.40.1.1 enum eQMI\_NAS\_GET\_RF\_INFO\_RESP

Enumerator

**eTLV\_RF\_BAND\_INFO**

### 9.40.2 Function Documentation

#### 9.40.2.1 enum eQCWWANError PkQmiNasGetRFBandInfo ( WORD \* pMlength, BYTE \* pBuffer )

#### 9.40.2.2 enum eQCWWANError UpkQmiNasGetRFBandInfo ( BYTE \* pMdmResp, struct QmiNasGetRFBandInfoResp \* pApiResp )

## 9.41 qaNasPerformNetworkScan.h File Reference

### Data Structures

- struct [QmiNas3GppNetworkInfo](#)
- struct [QmiNasPerformNetworkScanResp](#)

### Macros

- #define [QMI\\_NAS\\_NETSTATUS\\_MASK](#) 0x03
- #define [QMI\\_NAS\\_MAX\\_INSTANCES](#) 20
- #define [INDEX\\_ZERO](#) 0
- #define [ROAMING\\_INDEX](#) 2
- #define [FORBIDDEN\\_INDEX](#) 4
- #define [PREFERRED\\_INDEX](#) 6
- #define [MAX\\_DESCRIPTION\\_LENGTH](#) 255

### Enumerations

- enum [eQMI\\_NAS\\_PERFORM\\_NETWORK\\_SCAN\\_RESP](#) { [eTLV\\_3GPP\\_NETWORK\\_INFO](#) = 0x10 }

### Functions

- enum [eQCWWANError](#) PkQmiNasPerformNetworkScan (WORD \*pMlength, [BYTE](#) \*pParamField)
- enum [eQCWWANError](#) UpkQmiNasPerformNetworkScan ([BYTE](#) \*pMdmResp, struct [QmiNasPerformNetworkScanResp](#) \*pAipResp)

### 9.41.1 Macro Definition Documentation

#### 9.41.1.1 #define FORBIDDEN\_INDEX 4

#### 9.41.1.2 #define INDEX\_ZERO 0

#### 9.41.1.3 #define MAX\_DESCRIPTION\_LENGTH 255

#### 9.41.1.4 #define PREFERRED\_INDEX 6

9.41.1.5 `#define QMI_NAS_MAX_INSTANCES 20`

9.41.1.6 `#define QMI_NAS_NETSTATUS_MASK 0x03`

9.41.1.7 `#define ROAMING_INDEX 2`

## 9.41.2 Enumeration Type Documentation

9.41.2.1 `enum eQMI_NAS_PERFORM_NETWORK_SCAN_RESP`

Enumerator

***eTLV\_3GPP\_NETWORK\_INFO***

## 9.41.3 Function Documentation

9.41.3.1 `enum eQCWWANError PkQmiNasPerformNetworkScan ( WORD * pMlength, BYTE * pParamField )`

9.41.3.2 `enum eQCWWANError UpkQmiNasPerformNetworkScan ( BYTE * pMdmResp, struct QmiNasPerformNetworkScanResp * pAipResp )`

## 9.42 qmerrno.h File Reference

## Enumerations

- enum eQCWWANError {
  - eQCWWAN\_ERR\_ENUM\_BEGIN = -1,
  - eQCWWAN\_ERR\_NONE,
  - eQCWWAN\_ERR\_GENERAL,
  - eQCWWAN\_ERR\_INTERNAL,
  - eQCWWAN\_ERR\_MEMORY,
  - eQCWWAN\_ERR\_INVALID\_ARG,
  - eQCWWAN\_ERR\_BUFFER\_SZ,
  - eQCWWAN\_ERR\_NO\_DEVICE,
  - eQCWWAN\_ERR\_INVALID\_DEVID,
  - eQCWWAN\_ERR\_NO\_CONNECTION,
  - eQCWWAN\_ERR\_QMI\_IFACE,
  - eQCWWAN\_ERR\_QMI\_CONNECT,
  - eQCWWAN\_ERR\_QMI\_REQ\_SCH,
  - eQCWWAN\_ERR\_QMI\_REQ,
  - eQCWWAN\_ERR\_QMI\_RSP,
  - eQCWWAN\_ERR\_QMI\_REQ\_TO,
  - eQCWWAN\_ERR\_QMI\_RSP\_TO,
  - eQCWWAN\_ERR\_MALFORMED\_QMI\_RSP,
  - eQCWWAN\_ERR\_INVALID\_QMI\_RSP,
  - eQCWWAN\_ERR\_INVALID\_FILE,
  - eQCWWAN\_ERR\_FILE\_OPEN,
  - eQCWWAN\_ERR\_FILE\_COPY,
  - eQCWWAN\_ERR\_OFFLINE = 27,
  - eQCWWAN\_ERR\_RESET,
  - eQCWWAN\_ERR\_NO\_SIGNAL,
  - eQCWWAN\_ERR\_MULTIPLE\_DEVICES,
  - eQCWWAN\_ERR\_DRIVER,
  - eQCWWAN\_ERR\_NO\_CANCELABLE\_OP,
  - eQCWWAN\_ERR\_CANCEL\_OP,
  - eQCWWAN\_ERR\_API\_MUTEX\_TIMEOUT,
  - eQCWWAN\_ERR\_PDU\_GENERATION,
  - eQCWWAN\_ERR\_ENUM\_END,
  - eQCWWAN\_ERR\_QMI\_OFFSET = 1000,
  - eQCWWAN\_ERR\_QMI\_MALFORMED\_MSG = 1001,
  - eQCWWAN\_ERR\_QMI\_NO\_MEMORY,
  - eQCWWAN\_ERR\_QMI\_INTERNAL,
  - eQCWWAN\_ERR\_QMI\_ABORTED,
  - eQCWWAN\_ERR\_QMI\_CLIENT\_IDS\_EXHAUSTED,
  - eQCWWAN\_ERR\_QMI\_UNABORTABLE\_TRANSACTION,
  - eQCWWAN\_ERR\_QMI\_INVALID\_CLIENT\_ID,
  - eQCWWAN\_ERR\_QMI\_NO\_THRESHOLDS,
  - eQCWWAN\_ERR\_QMI\_INVALID\_HANDLE,
  - eQCWWAN\_ERR\_QMI\_INVALID\_PROFILE,
  - eQCWWAN\_ERR\_QMI\_INVALID\_PINID,
  - eQCWWAN\_ERR\_QMI\_INCORRECT\_PIN,
  - eQCWWAN\_ERR\_QMI\_NO\_NETWORK\_FOUND,
  - eQCWWAN\_ERR\_QMI\_CALL\_FAILED,
  - eQCWWAN\_ERR\_QMI\_OUT\_OF\_CALL,
  - eQCWWAN\_ERR\_QMI\_NOT\_PROVISIONED,
  - eQCWWAN\_ERR\_QMI\_MISSING\_ARG,
  - eQCWWAN\_ERR\_QMI\_ARG\_TOO\_LONG = 1019,
  - eQCWWAN\_ERR\_QMI\_INVALID\_TX\_ID = 1022,
  - eQCWWAN\_ERR\_QMI\_DEVICE\_IN\_USE,
  - eQCWWAN\_ERR\_QMI\_OP\_NETWORK\_UNSUPPORTED,
  - eQCWWAN\_ERR\_QMI\_OP\_DEVICE\_UNSUPPORTED,
  - eQCWWAN\_ERR\_QMI\_NO\_EFFECT,
  - eQCWWAN\_ERR\_QMI\_NO\_FREE\_PROFILE,
  - eQCWWAN\_ERR\_QMI\_INVALID\_SDP\_TYPE,
  - eQCWWAN\_ERR\_QMI\_INVALID\_TECH\_PREF,
  - eQCWWAN\_ERR\_QMI\_INVALID\_PROFILE\_TYPE,
  - eQCWWAN\_ERR\_QMI\_INVALID\_SERVICE\_TYPE,
  - eQCWWAN\_ERR\_QMI\_INVALID\_REGISTER\_ACTION,

```

eQCWWAN_ERR_QMI_WIDTH = 0xFFFF }
• enum qm_wds_ds_profile_extended_err_codes {
eWDS_ERR_PROFILE_REG_RESULT_FAIL = 1,
eWDS_ERR_PROFILE_REG_RESULT_ERR_INVALID_HNDL,
eWDS_ERR_PROFILE_REG_RESULT_ERR_INVALID_OP,
eWDS_ERR_PROFILE_REG_RESULT_ERR_INVALID_PROFILE_TYPE,
eWDS_ERR_PROFILE_REG_RESULT_ERR_INVALID_PROFILE_NUM,
eWDS_ERR_PROFILE_REG_RESULT_ERR_INVALID_IDENT,
eWDS_ERR_PROFILE_REG_RESULT_ERR_INVALID,
eWDS_ERR_PROFILE_REG_RESULT_ERR_LIB_NOT_INITED,
eWDS_ERR_PROFILE_REG_RESULT_ERR_LEN_INVALID,
eWDS_ERR_PROFILE_REG_RESULT_LIST_END,
eWDS_ERR_PROFILE_REG_RESULT_ERR_INVALID_SUBS_ID,
eWDS_ERR_PROFILE_REG_INVALID_PROFILE_FAMILY,
eWDS_ERR_PROFILE_REG_3GPP_INVALID_PROFILE_FAMILY = 1001,
eWDS_ERR_PROFILE_REG_3GPP_ACCESS_ERR,
eWDS_ERR_PROFILE_REG_3GPP_CONTEXT_NOT_DEFINED,
eWDS_ERR_PROFILE_REG_3GPP_VALID_FLAG_NOT_SET,
eWDS_ERR_PROFILE_REG_3GPP_READ_ONLY_FLAG_SET,
eWDS_ERR_PROFILE_REG_3GPP_ERR_OUT_OF_PROFILES,
eWDS_ERR_PROFILE_REG_3GPP2_ERR_INVALID_IDENT_FOR_PROFILE = 1101,
eWDS_ERR_PROFILE_REG_END }

```

## 9.42.1 Enumeration Type Documentation

### 9.42.1.1 enum eQCWWANError

#### QMI Error Code Enumeration

Enumerator

```

eQCWWAN_ERR_ENUM_BEGIN
eQCWWAN_ERR_NONE 00 - Success
eQCWWAN_ERR_GENERAL 01 - General error
eQCWWAN_ERR_INTERNAL 02 - Internal error
eQCWWAN_ERR_MEMORY 03 - Memory error
eQCWWAN_ERR_INVALID_ARG 04 - Invalid argument
eQCWWAN_ERR_BUFFER_SZ 05 - Buffer too small
eQCWWAN_ERR_NO_DEVICE 06 - Unable to detect WWAN device
eQCWWAN_ERR_INVALID_DEVID 07 - Invalid WWAN device ID
eQCWWAN_ERR_NO_CONNECTION 08 - No connection to WWAN device
eQCWWAN_ERR_QMI_IFACE 09 - Unable to obtain QMI interface
eQCWWAN_ERR_QMI_CONNECT 10 - Unable to connect to QMI interface
eQCWWAN_ERR_QMI_REQ_SCH 11 - Unable to schedule QMI request
eQCWWAN_ERR_QMI_REQ 12 - Error sending QMI request
eQCWWAN_ERR_QMI_RSP 13 - Error receiving QMI response
eQCWWAN_ERR_QMI_REQ_TO 14 - Timeout while sending QMI request
eQCWWAN_ERR_QMI_RSP_TO 15 - Timeout while receiving QMI response
eQCWWAN_ERR_MALFORMED_QMI_RSP 16 - Malformed QMI response received
eQCWWAN_ERR_INVALID_QMI_RSP 17 - Invalid QMI response received
eQCWWAN_ERR_INVALID_FILE 18 - Invalid file path
eQCWWAN_ERR_FILE_OPEN 19 - Unable to open file

```



**eQCWWAN\_ERR\_FILE\_COPY** 20 - Unable to copy file

**eQCWWAN\_ERR\_OFFLINE** 27 - Unable to set WWAN device offline

**eQCWWAN\_ERR\_RESET** 28 - Unable to reset WWAN device

**eQCWWAN\_ERR\_NO\_SIGNAL** 29 - No available signal

**eQCWWAN\_ERR\_MULTIPLE\_DEVICES** 30 - Multiple WWAN devices detected

**eQCWWAN\_ERR\_DRIVER** 31 - Error interfacing to driver

**eQCWWAN\_ERR\_NO\_CANCELABLE\_OP** 32 - No cancelable operation is pending

**eQCWWAN\_ERR\_CANCEL\_OP** 33 - Error canceling outstanding operation

**eQCWWAN\_ERR\_API\_MUTEX\_TIMEOUT** 34 - api mutex lock timeout

**eQCWWAN\_ERR\_PDU\_GENERATION** 35 - PDU generation error

**eQCWWAN\_ERR\_ENUM\_END** End of SLQS SDK specific error codes

**eQCWWAN\_ERR\_QMI\_OFFSET** 1000 - This is not an error code but the offset from which mapped QMI error codes start from

**eQCWWAN\_ERR\_QMI\_MALFORMED\_MSG** 1001 - Malformed or Corrupted QMI msg

**eQCWWAN\_ERR\_QMI\_NO\_MEMORY** 1002 - Device could not allocate memory for QMI Resp

**eQCWWAN\_ERR\_QMI\_INTERNAL** 1003 - Unexpected error occurred during processing

**eQCWWAN\_ERR\_QMI\_ABORTED** 1004 - Processing aborted

**eQCWWAN\_ERR\_QMI\_CLIENT\_IDS\_EXHAUSTED** 1005 - QMI client IDs have been exhausted

**eQCWWAN\_ERR\_QMI\_UNABORTABLE\_TRANSACTION** 1006 - Unable to abort QMI transaction

**eQCWWAN\_ERR\_QMI\_INVALID\_CLIENT\_ID** 1007 - Invalid QMI client ID

**eQCWWAN\_ERR\_QMI\_NO\_THRESHOLDS** 1008 - No thresholds were provided

**eQCWWAN\_ERR\_QMI\_INVALID\_HANDLE** 1009 - Invalid Handle provided in the QMI request

**eQCWWAN\_ERR\_QMI\_INVALID\_PROFILE** 1010 - Profile specified is invalid

**eQCWWAN\_ERR\_QMI\_INVALID\_PINID** 1011 - Invalid PIN ID specified

**eQCWWAN\_ERR\_QMI\_INCORRECT\_PIN** 1012 - Incorrect PIN ID specified

**eQCWWAN\_ERR\_QMI\_NO\_NETWORK\_FOUND** 1013 - No network found

**eQCWWAN\_ERR\_QMI\_CALL\_FAILED** 1014 - Call failed

**eQCWWAN\_ERR\_QMI\_OUT\_OF\_CALL** 1015 - Device is not in a call

**eQCWWAN\_ERR\_QMI\_NOT\_PROVISIONED** 1016 - Requested information element not provisioned on device

**eQCWWAN\_ERR\_QMI\_MISSING\_ARG** 1017 - Mandatory QMI TLV not provided

**eQCWWAN\_ERR\_QMI\_ARG\_TOO\_LONG** 1019 - Arg passed in QMI TLV larger than available storage in device

**eQCWWAN\_ERR\_QMI\_INVALID\_TX\_ID** 1022 - Invalid TX ID specified

**eQCWWAN\_ERR\_QMI\_DEVICE\_IN\_USE** 1023 - Device currently in a call

**eQCWWAN\_ERR\_QMI\_OP\_NETWORK\_UNSUPPORTED** 1024 - The selected operation is not supported by the network

**eQCWWAN\_ERR\_QMI\_OP\_DEVICE\_UNSUPPORTED** 1025 - The selected operation is not supported by the device

**eQCWWAN\_ERR\_QMI\_NO\_EFFECT** 1026 - Requested operation would have no effect

**eQCWWAN\_ERR\_QMI\_NO\_FREE\_PROFILE** 1027 - No space for a profile is available

**eQCWWAN\_ERR\_QMI\_INVALID\_PDP\_TYPE** 1028 - Invalid PDP type specified

**eQCWWAN\_ERR\_QMI\_INVALID\_TECH\_PREF** 1029 - Invalid technology preference specified

**eQCWWAN\_ERR\_QMI\_INVALID\_PROFILE\_TYPE** 1030 - Invalid profile type specified

**eQCWWAN\_ERR\_QMI\_INVALID\_SERVICE\_TYPE** 1031 - Invalid service type specified

**eQCWWAN\_ERR\_QMI\_INVALID\_REGISTER\_ACTION** 1032 - Invalid register action specified

**eQCWWAN\_ERR\_QMI\_INVALID\_PS\_ATTACH\_ACTION** 1033 - Invalid PS attach/detach action specified

**eQCWWAN\_ERR\_QMI\_AUTHENTICATION\_FAILED** 1034 - Authentication of supplied information element failed

**eQCWWAN\_ERR\_QMI\_PIN\_BLOCKED** 1035 - PIN is blocked; an unblock operation needs to be issued

**eQCWWAN\_ERR\_QMI\_PIN\_PERM\_BLOCKED** 1036 - PIN is permanently blocked; the UIM is unusable

**eQCWWAN\_ERR\_QMI\_SIM\_NOT\_INITIALIZED** 1037 - UIM initialization has not completed

**eQCWWAN\_ERR\_QMI\_MAX\_QOS\_REQUESTS\_IN\_USE** 1038 - Max QOS requests are used

**eQCWWAN\_ERR\_QMI\_INCORRECT\_FLOW\_FILTER** 1039 - The Flow filter is incorrect

**eQCWWAN\_ERR\_QMI\_NETWORK\_QOS\_UNAWARE** 1040 - Network unaware of the QOS requested

**eQCWWAN\_ERR\_QMI\_INVALID\_ID** 1041 - Invalid QOS ID

**eQCWWAN\_ERR\_QMI\_INVALID\_QOS\_ID** 1041 - Invalid QOS ID

**eQCWWAN\_ERR\_QMI\_REQUESTED\_NUM\_UNSUPPORTED** 1042 - The request number is not supported

**eQCWWAN\_ERR\_QMI\_INTERFACE\_NOT\_FOUND** 1043 - Unable to find the interface

**eQCWWAN\_ERR\_QMI\_FLOW\_SUSPENDED** 1044 - Flow suspended

**eQCWWAN\_ERR\_QMI\_INVALID\_DATA\_FORMAT** 1045 - Data format is invalid

**eQCWWAN\_ERR\_QMI\_GENERAL** 1046 - General error

**eQCWWAN\_ERR\_QMI\_UNKNOWN** 1047 - Unknown error

**eQCWWAN\_ERR\_QMI\_INVALID\_ARG** 1048 - A specified argument is invalid

**eQCWWAN\_ERR\_QMI\_INVALID\_INDEX** 1049 - A specified index is invalid

**eQCWWAN\_ERR\_QMI\_NO\_ENTRY** 1050 - No information element exists at specified memory designation

**eQCWWAN\_ERR\_QMI\_DEVICE\_STORAGE\_FULL** 1051 - The memory storage specified in the request is full

**eQCWWAN\_ERR\_QMI\_DEVICE\_NOT\_READY** 1052 - Device not in a ready state

**eQCWWAN\_ERR\_QMI\_NETWORK\_NOT\_READY** 1053 - Network not in a ready state

**eQCWWAN\_ERR\_QMI\_CAUSE\_CODE** 1054 - Error provided in SMS cause code

**eQCWWAN\_ERR\_QMI\_MESSAGE\_NOT\_SENT** 1055 - The message could not be sent

**eQCWWAN\_ERR\_QMI\_MESSAGE\_DELIVERY\_FAILURE** 1056 - The message could not be delivered

**eQCWWAN\_ERR\_QMI\_INVALID\_MESSAGE\_ID** 1057 - The message ID specified for the message is invalid

**eQCWWAN\_ERR\_QMI\_ENCODING** 1058 - The message is not encoded properly

**eQCWWAN\_ERR\_QMI\_AUTHENTICATION\_LOCK** 1059 - Maximum number of authentication failures has been reached

**eQCWWAN\_ERR\_QMI\_INVALID\_TRANSITION** 1060 - Operating mode transition from the current mode is invalid

**eQCWWAN\_ERR\_QMI\_NOT\_A\_MCAST\_IFACE** 1061 - The interface is not multicast

**eQCWWAN\_ERR\_QMI\_MAX\_MCAST\_REQUESTS\_IN\_USE** 1062 - Maximum requests in use

**eQCWWAN\_ERR\_QMI\_INVALID\_MCAST\_HANDLE** 1063 - Invalid multicast handle

**eQCWWAN\_ERR\_QMI\_INVALID\_IP\_FAMILY\_PREF** 1064 - Invalid IP family preference

**eQCWWAN\_ERR\_QMI\_SESSION\_INACTIVE** 1065 - No tracking session has been started

**eQCWWAN\_ERR\_QMI\_SESSION\_INVALID** 1066 - Current session does not allow this operation

**eQCWWAN\_ERR\_QMI\_SESSION\_OWNERSHIP** 1067 - Current tracking session not started by this QMI control point

**eQCWWAN\_ERR\_QMI\_INSUFFICIENT\_RESOURCES** 1068 - Device GPS service resources insufficient for request

**eQCWWAN\_ERR\_QMI\_DISABLED** 1069 - Device GPS service disabled

**eQCWWAN\_ERR\_QMI\_INVALID\_OPERATION** 1070 - Invalid operation specified

**eQCWWAN\_ERR\_QMI\_INVALID\_QMI\_CMD** 1071 - Invalid/unknown QMI command specified

**eQCWWAN\_ERR\_QMI\_TPDU\_TYPE** 1072 - Message contains TPDU type that cannot be read as raw message

**eQCWWAN\_ERR\_QMI\_SMSC\_ADDR** 1073 - The SMSC address specified is invalid

**eQCWWAN\_ERR\_QMI\_INFO\_UNAVAILABLE** 1074 - Information element is unavailable at this point

**eQCWWAN\_ERR\_QMI\_SEGMENT\_TOO\_LONG** 1075 - Segment size too large

**eQCWWAN\_ERR\_QMI\_SEGMENT\_ORDER** 1076 - Segment order is incorrect

**eQCWWAN\_ERR\_QMI\_BUNDLING\_NOT\_SUPPORTED** 1077 - Bundling not supported

**eQCWWAN\_ERR\_QMI\_OP\_PARTIAL\_FAILURE** 1078 - The operation failed partially

**eQCWWAN\_ERR\_QMI\_POLICY\_MISMATCH** 1079 - Policy mismatch

**eQCWWAN\_ERR\_QMI\_SIM\_FILE\_NOT\_FOUND** 1080 - SIM file not found

**eQCWWAN\_ERR\_QMI\_EXTENDED\_INTERNAL** 1081 - Extended internal error

**eQCWWAN\_ERR\_QMI\_ACCESS\_DENIED** 1082 - Access to a required entity is not available

**eQCWWAN\_ERR\_QMI\_HARDWARE\_RESTRICTED** 1083 - Selected operating mode is invalid with current hardware setting

**eQCWWAN\_ERR\_QMI\_ACK\_NOT\_SENT** 1084 - ACK not sent

**eQCWWAN\_ERR\_QMI\_INJECT\_TIMEOUT** 1084 - Inject a timeout for the request

**eQCWWAN\_ERR\_QMI\_INCOMPATIBLE\_STATE** 1090 - Incompatible state

**eQCWWAN\_ERR\_QMI\_FDN\_RESTRICT** 1091 - FDN Restrict

**eQCWWAN\_ERR\_QMI\_SUPS\_FAILURE\_CAUSE** 1092 - SUPS failure cause

**eQCWWAN\_ERR\_QMI\_NO\_RADIO** 1093 - No Radio

**eQCWWAN\_ERR\_QMI\_NOT\_SUPPORTED** 1094 - Not Supported

**eQCWWAN\_ERR\_QMI\_NO\_SUBSCRIPTION** 1095 - No Subscription

**eQCWWAN\_ERR\_QMI\_CARD\_CALL\_CONTROL\_FAILED** 1096 - Card call control failed

**eQCWWAN\_ERR\_QMI\_NETWORK\_ABORTED** 1097 - Network Aborted

**eQCWWAN\_ERR\_QMI\_MSG\_BLOCKED** 1098 - Open Error

**eQCWWAN\_ERR\_QMI\_MAX** Error - End of QMI specific defines

**eQCWWAN\_ERR\_SWICM\_START** Vendor defines - **Connection Manager error codes**

**eQCWWAN\_ERR\_SWICM\_NOT\_IMPLEMENTED** 0xE001 - The API is yet to be implemented

**eQCWWAN\_ERR\_SWICM\_QMI\_SVC\_NOT\_SUPPORTED** 0xE002 - The service is not supported

**eQCWWAN\_ERR\_SWICM\_QMI\_CLNT\_NOT\_SUPPORTED** 0xE003 - The client is not supported

**eQCWWAN\_ERR\_SWICM\_TIMEOUT** 0xE004 - API Timeout

**eQCWWAN\_ERR\_SWICM\_SOCKET\_IN\_USE** 0xE005 - The communication socket is in use

**eQCWWAN\_ERR\_SWICM\_AM\_VERS\_ERROR** 0xE006 - SLQS API and SDK version mismatch

**eQCWWAN\_ERR\_SWICM\_FAILED\_TO\_KILL\_SDK\_PROCESS** 0xE007 - Failed to kill SDK process

**eQCWWAN\_ERR\_SWICM\_CALL\_IN\_PROGRESS** 0xE008 - Call in progress

**eQCWWAN\_ERR\_SWICM\_V4DWN\_V6DWN** 0xE009 - IPV4 and IPV6 is down

**eQCWWAN\_ERR\_SWICM\_V4DWN\_V6UP** 0xE00A - IPV4 is down and IPV6 is up

**eQCWWAN\_ERR\_SWICM\_V4UP\_V6DWN** 0xE00B - IPV4 is up and IPV6 is down

**eQCWWAN\_ERR\_SWICM\_V4UP\_V6UP** 0xE00C - IPV4 and IPV6 is up

**eQCWWAN\_ERR\_SWICM\_INVALID\_SESSION\_ID** 0xE00D - Invalid V4 Session ID

**eQCWWAN\_ERR\_SWICM\_INVALID\_V4\_SESSION\_ID** 0xE00E - Invalid V4 Session ID

**eQCWWAN\_ERR\_SWICM\_INVALID\_V6\_SESSION\_ID** 0xE00F - Invalid V6 Session ID

**eQCWWAN\_ERR\_SWICM\_SM\_NO\_AVAILABLE\_SESSIONS** 0xE010 - No available Session Manager slots for additional data sessions

**eQCWWAN\_ERR\_SWICM\_END** 0xE011 - End of connection manager specific codes

**eQCWWAN\_ERR\_SWISMS\_START** Vendor defines - SMS Error codes

**eQCWWAN\_ERR\_SWISMS\_MSG\_LEN\_TOO\_LONG** 0xE101 - SMS message length is long

**eQCWWAN\_ERR\_SWISMS\_MSG\_CORRUPTED** 0xE102 - The SMS message is corrupted (encoding wrong)

**eQCWWAN\_ERR\_SWISMS\_SMSC\_NUM\_CORRUPTED** 0xE103 - The SMS number is corrupted (incorrect number)

**eQCWWAN\_ERR\_SWISMS\_BEARER\_DATA\_NOT\_FOUND** 0xE104 - The SMS bearer data is not available

**eQCWWAN\_ERR\_SWISM\_END**

**eQCWWAN\_ERR\_SWIIM\_START** Vendor defines - Image Management error codes

**eQCWWAN\_ERR\_SWIIM\_INVALID\_PATH** 0xE801 - Invalid directory path

**eQCWWAN\_ERR\_SWIIM\_OPENING\_DIR** 0xE802 - Unable to open the directory

**eQCWWAN\_ERR\_SWIIM\_FILE\_NOT\_FOUND** 0xE803 - No Firmware image present in the path

**eQCWWAN\_ERR\_SWIIM\_OPENING\_FILE** 0xE804 - Unable to open the file

**eQCWWAN\_ERR\_SWIIM\_CORRUPTED\_FW\_IMAGE** 0xE805 - Firmware image is corrupted

**eQCWWAN\_ERR\_SWIIM\_FIRMWARE\_NOT\_DOWNLOADED** 0xE806 - No Firmware image download needed

**eQCWWAN\_ERR\_SWIIM\_FW\_UPDATE\_FAIL** 0xE807 - Firmware update failed

**eQCWWAN\_ERR\_SWIIM\_FW\_PREFERENCE\_MISMATCH** 0xE808 - Update success but pri/fw preference mismatch

**eQCWWAN\_ERR\_SWIIM\_FW\_UPDATE\_SUCCESS** 0xE809 - Update successful

**eQCWWAN\_ERR\_SWIIM\_END**

**eQCWWAN\_ERR\_SWIDCS\_START** Vendor defines - Device Connectivity error codes

**eQCWWAN\_ERR\_SWIDCS\_IOCTL\_ERR** 0xE901 - IO Control error

**eQCWWAN\_ERR\_SWIDCS\_FILEIO\_ERR** 0xE902 - file open/read/write error

**eQCWWAN\_ERR\_SWIDCS\_DEVNODE\_NOT\_FOUND** 0xE903 - The device is not found

**eQCWWAN\_ERR\_SWIDCS\_APP\_DISCONNECTED** 0xE904 - Application is disconnected from SDK

**eQCWWAN\_ERR\_SWIDCS\_END**

**eQCWWAN\_ERR\_QMI\_CAT\_START** QMI errors related to CAT

**eQCWWAN\_ERR\_QMI\_EVENT\_REG\_FAILED** 62441 - CAT event registration failed

**eQCWWAN\_ERR\_QMI\_INVALID\_TERMINAL\_RSP** 62442 - Invalid terminal response

**eQCWWAN\_ERR\_QMI\_INVALID\_ENVELOPE\_CMD** 62443 - Invalid envelope command

**eQCWWAN\_ERR\_QMI\_CARD\_BUSY\_RSP** 62444 - Card busy response for envelope command

**eQCWWAN\_ERR\_QMI\_ENVELOPE\_CMD\_FAILURE** 62445 - Envelope command failure

**eQCWWAN\_ERR\_QMI\_CAT\_END**

**eQCWWAN\_ERR\_NULL\_TLV**

**eQCWWAN\_ERR\_QMI\_WIDTH** 0xFFFF - Not an error, represent the end of QMI errors

#### 9.42.1.2 enum qm\_wds\_ds\_profile\_extended\_err\_codes

##### WDS DS profile extented error codes

###### Enumerator

**eWDS\_ERR\_PROFILE\_REG\_RESULT\_FAIL** 1 - General Failure

**eWDS\_ERR\_PROFILE\_REG\_RESULT\_ERR\_INVALID\_HNDL** 2 - The request contains an invalid profile handle

- eWDS\_ERR\_PROFILE\_REG\_RESULT\_ERR\_INVAL\_OP*** 3 - An invalid operation was requested.
- eWDS\_ERR\_PROFILE\_REG\_RESULT\_ERR\_INVAL\_PROFILE\_TYPE*** 4 - The request contains an invalid technology type
- eWDS\_ERR\_PROFILE\_REG\_RESULT\_ERR\_INVAL\_PROFILE\_NUM*** 5 - The request contains an invalid profile number
- eWDS\_ERR\_PROFILE\_REG\_RESULT\_ERR\_INVAL\_IDENT*** 6 - The request contains an invalid profile identifier
- eWDS\_ERR\_PROFILE\_REG\_RESULT\_ERR\_INVAL*** 7 - The request contains an invalid argument other than profile number and profile identifier received.
- eWDS\_ERR\_PROFILE\_REG\_RESULT\_ERR\_LIB\_NOT\_INITED*** 8 - Profile registry has not been initialized yet
- eWDS\_ERR\_PROFILE\_REG\_RESULT\_ERR\_LEN\_INVALID*** 9 - The request contains a parameter with invalid length.
- eWDS\_ERR\_PROFILE\_REG\_RESULT\_LIST\_END*** 10 - End of the profile list was reached while searching for the requested profile.
- eWDS\_ERR\_PROFILE\_REG\_RESULT\_ERR\_INVAL\_SUBS\_ID*** 11 - The request contains an invalid subscription identifier.
- eWDS\_ERR\_PROFILE\_REG\_INVAL\_PROFILE\_FAMILY*** 12 - The request contains an invalid profile family.
- eWDS\_ERR\_PROFILE\_REG\_3GPP\_INVAL\_PROFILE\_FAMILY*** 1001 - The request contains an invalid 3GPP profile family.
- eWDS\_ERR\_PROFILE\_REG\_3GPP\_ACCESS\_ERR*** 1002 - An error was encountered while accessing the 3GPP profiles.
- eWDS\_ERR\_PROFILE\_REG\_3GPP\_CONTEXT\_NOT\_DEFINED*** 1003 - The given 3GPP profile doesn't have a valid context.
- eWDS\_ERR\_PROFILE\_REG\_3GPP\_VALID\_FLAG\_NOT\_SET*** 1004 - The given 3GPP profile is marked invalid.
- eWDS\_ERR\_PROFILE\_REG\_3GPP\_READ\_ONLY\_FLAG\_SET*** 1005 - The given 3GPP profile is marked read-only.
- eWDS\_ERR\_PROFILE\_REG\_3GPP\_ERR\_OUT\_OF\_PROFILES*** 1006 - Creation of a new 3GPP profile failed because the limit of 16 profiles has already been reached.
- eWDS\_ERR\_PROFILE\_REG\_3GPP2\_ERR\_INVALID\_IDENT\_FOR\_PROFILE*** 1101 - An invalid profile identifier was received as part of the 3GPP2 profile modification request.
- eWDS\_ERR\_PROFILE\_REG\_END***

## 9.43 SwiDataTypes.h File Reference

SWI data types.

### Macros

- #define [SWI\\_API](#)
- #define [QMI\\_NO\\_LTE\\_FW\\_SUPPORT](#) 0
- #define [QMI\\_TLV\\_PLACEHOLDER](#) 0x8F
- #define [UNUSEDPARAM\(x\)](#) (void)x

## Typedefs

- typedef unsigned long [ULONG](#)
- typedef unsigned long long [ULONGLONG](#)
- typedef signed char [INT8](#)
- typedef unsigned char [BYTE](#)
- typedef char [CHAR](#)
- typedef unsigned short [WORD](#)
- typedef unsigned short [USHORT](#)
- typedef const char \* [LPCSTR](#)
- typedef int [BOOL](#)
- typedef signed short [SHORT](#)
- typedef signed int [INT32](#)
- typedef float [FLOAT](#)

### 9.43.1 Detailed Description

SWI data types.

### 9.43.2 Macro Definition Documentation

9.43.2.1 `#define QMI_NO_LTE_FW_SUPPORT 0`

9.43.2.2 `#define QMI_TLV_PLACEHOLDER 0x8F`

9.43.2.3 `#define SWI_API`

9.43.2.4 `#define UNUSEDPARAM( x ) (void)x`

Macro used to avoid “unused variable” compiler warnings generated due to the inclusion of the “-Wextra” flag in our make files.

### 9.43.3 Typedef Documentation

9.43.3.1 typedef int [BOOL](#)

9.43.3.2 typedef unsigned char [BYTE](#)

9.43.3.3 typedef char [CHAR](#)

9.43.3.4 typedef float [FLOAT](#)

9.43.3.5 typedef signed int [INT32](#)

9.43.3.6 typedef signed char [INT8](#)

9.43.3.7 typedef const char\* [LPCSTR](#)

9.43.3.8 typedef signed short [SHORT](#)

9.43.3.9 typedef unsigned long [ULONG](#)

9.43.3.10 typedef unsigned long long [ULONGLONG](#)

9.43.3.11 typedef unsigned short USHORT

9.43.3.12 typedef unsigned short WORD

## 9.44 SWIWWANCMAPI.h File Reference

# Index

- [\\_GetProfileSettingIn, 46](#)
  - [ProfileID, 47](#)
  - [ProfileType, 47](#)
- [\\_GetProfileSettingOut, 47](#)
  - [curProfile, 47](#)
  - [pExtErrCode, 47](#)
- [\\_SLQSOMADMSessionInfo, 61](#)
  - [pDate, 63](#)
  - [pDateLength, 63](#)
  - [pPkgDescLength, 63](#)
  - [pPkgDescription, 63](#)
  - [pPkgName, 63](#)
  - [pPkgNameLength, 63](#)
  - [pRetryCount, 63](#)
  - [pSessionState, 63](#)
  - [pSessionType, 63](#)
  - [pSeverity, 63](#)
  - [pSource, 63](#)
  - [pSourceLength, 63](#)
  - [pStatus, 63](#)
  - [pTime, 63](#)
  - [pTimeLength, 64](#)
  - [pUpdateCompleteStatus, 64](#)
- [\\_SLQSOMADMSettings, 64](#)
  - [pAutosdm, 65](#)
  - [pFOTAUpdate, 65](#)
  - [pFOTAdownload, 65](#)
  - [pFwAutoCheck, 65](#)
  - [pOMADMEEnabled, 65](#)
- [\\_SLQSOMADMSettingsReqParams, 65](#)
  - [FOTAUpdate, 66](#)
  - [FOTAdownload, 66](#)
  - [pAutosdm, 66](#)
- [\\_SLQSOMADMSettingsReqParams3, 66](#)
  - [FOTAUpdate, 67](#)
  - [FOTAdownload, 67](#)
  - [pAutosdm, 67](#)
  - [pFwAutoCheck, 67](#)
- [\\_SLQSSwiGetHostDevInfoParams, 67](#)
  - [bManSize, 68](#)
  - [bModelSize, 68](#)
  - [bPlasmaIDSize, 68](#)
  - [bSWVerSize, 68](#)
  - [pManString, 68](#)
  - [pModelString, 68](#)
  - [pPlasmaIDString, 68](#)
  - [pSWVerString, 68](#)
- [\\_SLQSSwiGetOSInfoParams, 68](#)
  - [bNameSize, 68](#)
  - [bVersionSize, 68](#)
  - [pNameString, 68](#)
  - [pVersionString, 68](#)
- [\\_SLQSSwiGetSerialNoExtParams, 68](#)
  - [meidLength, 69](#)
  - [pMeidString, 69](#)
- [\\_SLQSSwiSetHostDevInfoParams, 69](#)
  - [bManSize, 69](#)
  - [bModelSize, 69](#)
  - [bPlasmaIDSize, 69](#)
  - [bSWVerSize, 69](#)
  - [pManString, 69](#)
  - [pModelString, 69](#)
  - [pPlasmaIDString, 69](#)
  - [pSWVerString, 69](#)
- [\\_SLQSSwiSetOSInfoParams, 69](#)
  - [bNameSize, 70](#)
  - [bVersionSize, 70](#)
  - [pNameString, 70](#)
  - [pVersionString, 70](#)
- [\\_getIndicationRegResp, 45](#)
  - [pRegCallStatInfoEvt, 46](#)
  - [pRegTransLayerInfoEvt, 46](#)
  - [pRegTransNWRegInfoEvt, 46](#)
- [\\_getTransLayerInfoResp, 47](#)
  - [pRegInd, 49](#)
  - [pTransLayerInfo, 49](#)
- [\\_getTransNWRegInfoResp, 49](#)
  - [pRegStatus, 50](#)
- [\\_modemTempNotification, 50](#)
  - [ModemTempState, 50](#)
  - [ModemTemperature, 50](#)
- [\\_packetSrvStatus, 50](#)
  - [bearerID, 52](#)
  - [connStatus, 52](#)
  - [ipFamily, 52](#)
  - [pQmiInterfaceInfo, 52](#)
  - [reconfigReqd, 52](#)
  - [sessionEndReason, 52](#)
  - [techName, 52](#)
  - [verboseSessnEndReason, 52](#)
  - [verboseSessnEndReasonType, 52](#)
- [\\_qaQmi3GPP2BroadcastCfgInfo, 53](#)
  - [activated\\_ind, 53](#)
  - [CDMABroadcastConfig, 53](#)
  - [num\\_instances, 53](#)
- [\\_qaQmi3GPPBroadcastCfgInfo, 53](#)
  - [activated\\_ind, 55](#)
  - [broadcastConfig, 55](#)



- num\_instances, 55
- \_setIndicationRegReq, 55
  - pRegCallStatInfoEvt, 56
  - pRegTransLayerInfoEvt, 56
  - pRegTransNWRegInfoEvt, 56
- \_slqs3GPPConfigItem, 56
  - LTEAttachProfileListLen, 59
  - p3gppRelease, 59
  - pDefaultPDNEnabled, 59
  - pLTEAttachProfile, 59
  - pLTEAttachProfileList, 59
  - pProfileList, 59
- \_slqsNetworkScanInfo, 59
  - pNetworkInfo, 60
  - pNetworkInfoInstances, 60
  - pPCSDigitInfo, 60
  - pPCSDigitInstances, 60
  - pRATInfo, 60
  - pRATInstances, 60
  - pScanResult, 60
- \_sysSelectPrefInfo, 70
  - pBandPref, 73
  - pEmerMode, 73
  - pGWAcqOrderPref, 73
  - pLTEBandPref, 73
  - pModePref, 73
  - pNetSelPref, 73
  - pPRLPref, 73
  - pRoamPref, 74
  - pSrvDomainPref, 74
- \_sysSelectPrefParams, 74
  - pAcqOrderPref, 78
  - pBandPref, 78
  - pCSGID, 78
  - pChgDuration, 78
  - pEmerMode, 78
  - pGWAcqOrderPref, 78
  - pLTEBandPref, 78
  - pMNCIncPCSDigStat, 78
  - pModePref, 78
  - pNetSelPref, 78
  - pPRLPref, 78
  - pRAT, 78
  - pRoamPref, 78
  - pSrvDomainPref, 78
  - pSrvRegRestriction, 78
  - pTdsdmaBandPref, 78
- \_transLayerInfoNotification, 80
  - pTransLayerInfo, 81
  - regInd, 81
- \_transLayerinfo, 78
  - TransCap, 80
  - TransType, 80
- \_transNWRegInfoNotification, 81
  - NWRegStat, 81
- ABSOLUTE\_VALIDITY
  - qaGobiApiSms.h, 988
- ALS
  - getAllCallInformation, 217
- absoluteValidity
  - cdmaMsgDecodingParams, 132
- accelAcceptReady
  - qaGobiApiCbk.h, 743
- accelAcceptReady\_s, 81
  - batchPerSec, 82
  - injectEnable, 82
  - samplesPerBatch, 82
- accelTempAcceptReady
  - qaGobiApiCbk.h, 744
- accelTempAcceptReady\_s, 82
  - batchPerSec, 83
  - injectEnable, 83
  - samplesPerBatch, 83
- AccessMac
  - protocolSubtypeElement, 421
- ackIndicator
  - SMSTransferRouteMTMessage, 554
- acqOrdeLen
  - acqOrderPref, 84
- acqOrderPref, 83
  - acqOrdeLen, 84
  - pAcqOrder, 84
- acroamsetting
  - slqsautoconnect, 519
- acsetting
  - slqsautoconnect, 519
- ActPilotPNElement, 84
  - ActSetPilotPN, 84
  - ActSetPilotPNStrength, 84
- ActSetCnt
  - NetworkStat1x, 369
- ActSetPilotPN
  - ActPilotPNElement, 84
- ActSetPilotPNStrength
  - ActPilotPNElement, 84
- action
  - slqsautoconnect, 519
  - ssdatasession\_params, 559
- ActivateAutomatic
  - qaGobiApiDms.h, 844
- activated\_ind
  - \_qaQmi3GPP2BroadcastCfgInfo, 53
  - \_qaQmi3GPPBroadcastCfgInfo, 55
- activeBandClass
  - RFBandInfoElements, 463
- activeChannel
  - RFBandInfoElements, 463
- activeInd
  - messageWaitingInfoContent, 332
- ActiveStatus
  - CLIPResp, 146
  - CLIRResp, 146
  - CNAPResp, 149
  - COLPResp, 150
  - COLRResp, 151
- AddCDMASysInfo, 84

- geoSysIdx, 85
- regPrd, 85
- AddSysInfo, 85
  - cellBroadcastCap, 86
  - geoSysIdx, 86
- addr
  - IPv4Addr, 300
  - IPv6Addr, 302
- aid
  - UIMRefreshEvent, 609
  - UIMSessionInformation, 612
- aidLength
  - appStatus, 92
  - UIMRefreshEvent, 609
  - UIMSessionInformation, 612
- aidVal
  - appStatus, 92
- aidingIndicatorMask
  - sensorDataUsage\_s, 479
- airTimer, 86
  - airTimerValue, 86
  - namID, 86
- airTimerValue
  - airTimer, 86
- alertPitch
  - signalInfo, 516
- alertingPattern
  - arrAlertingPattern, 93
- AlertingType
  - arrAlertingType, 94
- alertmsg
  - omaDmConfigTlv, 375
  - omaDmConfigTlvExt, 378
- alertmsglength
  - omaDmConfigTlv, 375
  - omaDmConfigTlvExt, 378
- allCallsAlphaIDInfo, 87
  - AlphaIDInfo, 87
  - callID, 87
- allCallsAlphaIDInfoArr
  - arrAlphaID, 95
- allCallsDiagInfo, 87
  - callID, 87
  - DiagInfo, 87
- AllCallsUUSInfo
  - arrUUSInfo, 101
- allCallsUUSInfo, 88
  - callID, 88
  - uusInfo, 88
- alphaDcs
  - alphaIDInfo, 89
- AlphaID
  - CatAlphaIdentifierTlv, 123
- AlphaIDInfo
  - allCallsAlphaIDInfo, 87
- alphaIDInfo, 88
  - alphaDcs, 89
  - alphaLen, 89
  - alphaText, 89
- alphaIDLen
  - SMSAsyncRawSend\_s, 542
- AlphaIDLength
  - CatAlphaIdentifierTlv, 123
- alphaLen
  - alphaIDInfo, 89
- alphaText
  - alphaIDInfo, 89
- alphabet
  - wcdmaMsgEncodingParams, 705
- Altitude
  - GPSSStateInfo, 255
- ambr\_dl
  - sApnExtraParams, 475
- ambr\_dl\_ext
  - sApnExtraParams, 475
- ambr\_dl\_ext2
  - sApnExtraParams, 475
- ambr\_ul
  - sApnExtraParams, 475
- ambr\_ul\_ext
  - sApnExtraParams, 476
- ambr\_ul\_ext2
  - sApnExtraParams, 476
- AnswerUSSD
  - qaGobiApiVoice.h, 1080
- apdoxypages.c, 727
- apnId
  - sApnExtraParams, 476
  - sQosStat, 556
- appNameLength
  - LocApplicationInfo, 305
- appProviderLength
  - LocApplicationInfo, 305
- appState
  - appStatus, 92
- AppStatus
  - slotInfo, 518
- appStatus, 89
  - aidLength, 92
  - aidVal, 92
  - appState, 92
  - appType, 92
  - persoFeature, 92
  - persoRetries, 92
  - persoState, 92
  - persoUnblockRetries, 92
  - pin1Retries, 92
  - pin1State, 92
  - pin2Retries, 92
  - pin2State, 92
  - puk1Retries, 92
  - puk2Retries, 92
  - univPin, 93
- appType
  - appStatus, 92
- appVersionLength

- LocApplicationInfo, [305](#)
- appVersionValid
  - LocApplicationInfo, [305](#)
- appversion\_str
  - slqsfwinfo\_s, [521](#)
- arfcn
  - GERANInfo, [214](#)
  - gsmCellInfo, [258](#)
- arrAlertingPattern, [93](#)
  - alertingPattern, [93](#)
  - callID, [93](#)
  - numInstances, [93](#)
- arrAlertingType, [93](#)
  - AlertingType, [94](#)
  - callID, [94](#)
  - numInstances, [94](#)
- arrAlphaID, [94](#)
  - allCallsAlphaIDInfoArr, [95](#)
  - numInstances, [95](#)
- arrCallEndReason, [95](#)
  - callEndReason, [96](#)
  - callID, [96](#)
  - numInstances, [96](#)
- arrCallInfo, [96](#)
  - getAllCallInfo, [96](#)
  - numInstances, [96](#)
- arrCallInformation
  - voiceSetAllCallStatusCbInfo, [682](#)
- arrCalledPartyNum, [95](#)
  - CalledPartyNum, [95](#)
  - numInstances, [95](#)
- arrConnectPartyNum, [97](#)
  - ConnectedPartyNum, [97](#)
  - numInstances, [97](#)
- arrDiagInfo, [97](#)
  - DiagInfo, [98](#)
  - numInstances, [98](#)
- arrRedirPartyNum, [98](#)
  - numInstances, [98](#)
  - RedirPartyNum, [98](#)
- arrRemotePartyName, [98](#)
  - GetAllCallRmtPtyName, [99](#)
  - numInstances, [99](#)
- arrRemotePartyNum, [99](#)
  - numInstances, [99](#)
  - RmtPtyNum, [99](#)
- arrSvcOption, [99](#)
  - callID, [100](#)
  - numInstances, [100](#)
  - srvOption, [100](#)
- arrUUSInfo, [100](#)
  - AllCallsUUSInfo, [101](#)
  - numInstances, [101](#)
- arrfileInfo
  - registerRefresh, [458](#)
  - UIMRefreshEvent, [609](#)
- AtCmdPort
  - DcsUsbPortNames, [184](#)

- Audio Service (AUDIO), [37](#)
- authData
  - UIMAuthenticateReq, [597](#)
- AuthProt
  - protocolSubtypeElement, [421](#)
- authenticateResult, [101](#)
  - content, [101](#)
  - contentLen, [101](#)
- authenticationData, [101](#)
  - context, [102](#)
  - data, [102](#)
  - dataLen, [102](#)
- avgPeriod
  - LTESigRptCfg, [323](#)
  - LTESigRptConfig, [324](#)
- azimuth
  - satelliteInfo, [478](#)
- bAltitudeAssumed
  - gnssSvInfoNotification, [250](#)
- bManSize
  - \_SLQSSwiGetHostDevInfoParams, [68](#)
  - \_SLQSSwiSetHostDevInfoParams, [69](#)
- bModelSize
  - \_SLQSSwiGetHostDevInfoParams, [68](#)
  - \_SLQSSwiSetHostDevInfoParams, [69](#)
- bNameSize
  - \_SLQSSwiGetOSInfoParams, [68](#)
  - \_SLQSSwiSetOSInfoParams, [70](#)
- BOOL
  - SwiDataTypes.h, [1159](#)
- bPlasmaIDSize
  - \_SLQSSwiGetHostDevInfoParams, [68](#)
  - \_SLQSSwiSetHostDevInfoParams, [69](#)
- bResetStatistics
  - swiRMTrasferStaticsReq, [581](#)
- bSWVerSize
  - \_SLQSSwiGetHostDevInfoParams, [68](#)
  - \_SLQSSwiSetHostDevInfoParams, [69](#)
- bVersionSize
  - \_SLQSSwiGetOSInfoParams, [68](#)
  - \_SLQSSwiSetOSInfoParams, [70](#)
- BYTE
  - SwiDataTypes.h, [1159](#)
- band
  - LTEInfo, [315](#)
- band1900
  - gsmCellInfo, [258](#)
- bandwidth
  - LTEInfo, [315](#)
- baseId
  - CDMAInfo, [129](#)
  - CDMASysInfo, [139](#)
- baseLat
  - CDMAInfo, [129](#)
  - CDMASysInfo, [139](#)
- baseLong
  - CDMAInfo, [129](#)
  - CDMASysInfo, [140](#)

- BasestationID
  - qaQmiServingSystemParam, 426
- BasestationLatitude
  - qaQmiServingSystemParam, 426
- BasestationLongitude
  - qaQmiServingSystemParam, 426
- batchPerSec
  - accelAcceptReady\_s, 82
  - accelTempAcceptReady\_s, 83
  - gyroAcceptReady\_s, 264
  - gyroTempAcceptReady\_s, 265
- BdsSV, 102
  - id, 103
  - mask, 103
- BdsSVInfo, 103
  - len, 104
  - pSV, 104
- bearerID
  - \_packetSrvStatus, 52
- bearerId
  - sQosFlowStat, 555
- bootversion\_str
  - slqsfwinfo\_s, 521
- BroadcastConfig, 104
  - fromServiceId, 104
  - selected, 104
  - toServiceId, 104
- broadcastConfig
  - \_qaQmi3GPPBroadcastCfgInfo, 55
- bsInfoValid
  - CDMASysInfo, 140
- bsPRev
  - CDMASysInfo, 140
- bsPRevValid
  - CDMASysInfo, 140
- bsic
  - GERANInfo, 214
- bsicId
  - gsmCellInfo, 258
- bucketSz
  - tokenBucket, 588
- buildID
  - CurrImageInfo, 167
  - ImageIdElement, 277
- buildIDLen
  - CurrImageInfo, 167
- buildIDLength
  - ImageIdElement, 277
- buildId
  - ImageElement, 276
- buildIdLength
  - ImageElement, 277
- BurstDTMFInfo
  - voiceBurstDTMFInfo, 640
- burstDTMFInfo, 104
  - digitCnt, 105
  - pCallID, 105
  - pDigitBuff, 105
- ByteLoopbackMode
  - WDSGetLoopbackData, 715
- ByteLoopbackMultiplier
  - WDSGetLoopbackData, 715
- ByteTotalsElmntsV4
  - WdsByteTotals, 712
- ByteTotalsElmntsV6
  - WdsByteTotals, 712
- CATEventDataType, 124
  - eventMask, 124
  - pErrorMask, 124
- CATSendEnvelopeCommand
  - qaGobiApiCat.h, 734
- CATSendTerminalResponse
  - qaGobiApiCat.h, 734
- CBK\_DISABLE\_EVENT
  - qaGobiApiCbk.h, 742
- CBK\_ENABLE\_EVENT
  - qaGobiApiCbk.h, 742
- CBK\_NOCHANGE
  - qaGobiApiCbk.h, 742
- CCETlv
  - QmiCbkCatEventStatusReportInd, 427
- CDMA\_P\_Rev
  - qaQmiServingSystemParam, 426
- CDMABroadcastConfig, 126
  - \_qaQmi3GPP2BroadcastCfgInfo, 53
  - language, 127
  - selected, 127
  - serviceCategory, 127
- CDMAChannel, 127
  - priChA, 128
  - priChB, 128
  - secChA, 128
  - secChB, 128
- CDMAECIOThresh, 128
  - CDMAECIOThreshListLen, 128
  - pCDMAECIOThreshList, 128
- CDMAECIOThreshListLen
  - CDMAECIOThresh, 128
- CDMAInfo, 128
  - baseId, 129
  - baseLat, 129
  - baseLong, 129
  - nid, 129
  - refpn, 129
  - sid, 129
- CDMARSSIThresh, 135
  - CDMARSSIThreshListLen, 135
  - pCDMARSSIThreshList, 135
- CDMARSSIThreshListLen
  - CDMARSSIThresh, 135
- CDMASSInfo, 135
  - ecio, 136
  - rsi, 136
- CDMASysInfo, 136
  - baseId, 139
  - baseLat, 139

- baseLong, [140](#)
- bsInfoValid, [140](#)
- bsPRev, [140](#)
- bsPRevValid, [140](#)
- ccsSupported, [140](#)
- ccsSupportedValid, [140](#)
- cdmaSysIdValid, [140](#)
- isSysPrIMatch, [140](#)
- isSysPrIMatchValid, [140](#)
- MCC, [140](#)
- MNC, [140](#)
- networkID, [140](#)
- networkIdValid, [140](#)
- pRevInUse, [140](#)
- pRevInUseValid, [140](#)
- packetZone, [140](#)
- packetZoneValid, [140](#)
- sysInfoCDMA, [140](#)
- systemID, [140](#)
- CDMASysInfoExt, [140](#)
  - imsi\_11\_12, [141](#)
  - MCC, [141](#)
- CDMASystemInfoExt
  - qaQmiServingSystemParam, [427](#)
- CHAR
  - SwiDataTypes.h, [1159](#)
- CLIPResp, [145](#)
  - ActiveStatus, [146](#)
  - ProvisionStatus, [146](#)
- CLIRResp, [146](#)
  - ActiveStatus, [146](#)
  - ProvisionStatus, [146](#)
- CNAPResp, [149](#)
  - ActiveStatus, [149](#)
  - ProvisionStatus, [149](#)
- COLPResp, [149](#)
  - ActiveStatus, [150](#)
  - ProvisionStatus, [150](#)
- COLRResp, [150](#)
  - ActiveStatus, [151](#)
  - ProvisionStatus, [151](#)
- CONFIG\_LEN
  - qaGobiApiSms.h, [988](#)
- CQIValueCW0
  - LteCQIParm, [311](#)
- CQIValueCW1
  - LteCQIParm, [311](#)
- CSGID, [160](#)
  - id, [161](#)
  - mcc, [161](#)
  - mnc, [161](#)
  - mncPcsDigits, [161](#)
  - rat, [161](#)
- CUGIndex
  - CUGInfo, [162](#)
- CUGInfo, [161](#)
  - CUGIndex, [162](#)
  - SuppOA, [162](#)
  - SuppPrefCUG, [162](#)
- CallBackK registration (CBK), [25](#)
- CallBarStatus
  - qaQmiServingSystemParam, [426](#)
- callBarStatus, [106](#)
  - csBarStatus, [107](#)
  - psBarStatus, [107](#)
- CallBarringSysInfo, [105](#)
  - csBarStatus, [106](#)
  - psBarStatus, [106](#)
- CallEndReason
  - DUNCallInfoInd, [195](#)
- callEndReason
  - arrCallEndReason, [96](#)
- CallFWExtInfo
  - getCallFWExtInfo, [225](#)
- callFWExtInfo, [112](#)
  - noReplyTimer, [115](#)
  - numLen, [115](#)
  - numPlan, [115](#)
  - numType, [115](#)
  - number, [115](#)
  - PI, [115](#)
  - SI, [115](#)
  - SvcClass, [115](#)
  - SvcStatus, [115](#)
- CallFWInfo
  - getCallFWInfo, [226](#)
- callFWInfo, [115](#)
  - noReplyTimer, [116](#)
  - numLen, [116](#)
  - number, [116](#)
  - SvcClass, [116](#)
  - SvcStatus, [116](#)
- callFwdTypeAndPlan, [111](#)
  - numberPlan, [112](#)
  - numberType, [112](#)
- callID
  - allCallsAlphaIDInfo, [87](#)
  - allCallsDiagInfo, [87](#)
  - allCallsUUSInfo, [88](#)
  - arrAlertingPattern, [93](#)
  - arrAlertingType, [94](#)
  - arrCallEndReason, [96](#)
  - arrSvcOption, [100](#)
  - callInfo, [118](#)
  - DTMFInfo, [194](#)
  - getAllCallRmtPtyName, [217](#)
  - getAllCallRmtPtyNum, [219](#)
  - peerNumberInfo, [393](#)
  - voiceCallInfoReq, [640](#)
  - voiceInfoRec, [676](#)
  - voiceOTASPStatusInfo, [680](#)
  - voicePrivacyInfo, [680](#)
  - voiceStopContDTMFInfo, [694](#)
  - voiceSUPSNotification, [699](#)
- callInfo, [116](#)
  - callID, [118](#)

- callState, [118](#)
  - callType, [118](#)
  - direction, [118](#)
  - mode, [118](#)
- callNumber
  - voiceCallRequestParams, [646](#)
- callState
  - callInfo, [118](#)
- callType
  - callInfo, [118](#)
- calledPartyInfo, [107](#)
  - numLen, [109](#)
  - numPlan, [109](#)
  - numType, [109](#)
  - number, [109](#)
  - PI, [109](#)
  - SI, [109](#)
- CalledPartyNum
  - arrCalledPartyNum, [95](#)
- calledPartySubAdd, [109](#)
  - extBit, [110](#)
  - oddEvenInd, [110](#)
  - subAddr, [110](#)
  - subAddrLen, [110](#)
  - subAddrType, [110](#)
- callerID
  - callerIDInfo, [111](#)
  - connectNumInfo, [156](#)
- callerIDInfo, [110](#)
  - callerID, [111](#)
  - callerIDLen, [111](#)
  - PI, [111](#)
- callerIDLen
  - callerIDInfo, [111](#)
  - connectNumInfo, [156](#)
- callerName
  - remotePartyName, [460](#)
- Callinfo
  - getAllCallInformation, [217](#)
- callingPartyInfo, [118](#)
  - numLen, [120](#)
  - numPlan, [120](#)
  - numType, [120](#)
  - number, [120](#)
  - PI, [120](#)
  - SI, [120](#)
- CancelUSSD
  - qaGobiApiVoice.h, [1080](#)
- Card Application Toolkit (CAT), [28](#)
- cardResult, [120](#)
  - sw1, [121](#)
  - sw2, [121](#)
- cardState
  - slotInfo, [518](#)
- cardStatus, [121](#)
  - index1xPri, [122](#)
  - index1xSec, [122](#)
  - indexGwPri, [122](#)
  - indexGwSec, [122](#)
  - numSlot, [122](#)
  - SlotInfo, [122](#)
- Carrier
  - fwinfo\_s, [212](#)
- carrier
  - CurrentImgList, [165](#)
- carrier\_str
  - slqsfwinfo\_s, [521](#)
- CatAlPhalIdentifierTlv, [122](#)
  - AlphaID, [123](#)
  - AlphaIDLength, [123](#)
  - ReferenceID, [123](#)
- CatAlphaIdtfr
  - currentCatEvent, [165](#)
- CatCommonEventTlv, [123](#)
  - CatEvent, [123](#)
  - EventID, [123](#)
  - EventLength, [123](#)
  - TlvPresent, [123](#)
- CatEndPS
  - currentCatEvent, [165](#)
- CatEndProactiveSessionTlv, [123](#)
  - EndProactiveSession, [124](#)
- CatEvIDData
  - currentCatEvent, [165](#)
- CatEvent
  - CatCommonEventTlv, [123](#)
- CatEventIDDataTlv, [124](#)
  - Data, [124](#)
  - DataLength, [124](#)
  - ReferenceID, [124](#)
- CatEventListTlv, [124](#)
  - SetupEventList, [125](#)
- CatEventLst
  - currentCatEvent, [165](#)
- CatRefresh
  - currentCatEvent, [165](#)
- CatRefreshTlv, [125](#)
  - RefreshMode, [125](#)
  - RefreshStage, [125](#)
- causeCode
  - SMSAsyncRawSend\_s, [542](#)
- ccSUPSType, [125](#)
  - reason, [126](#)
  - svcType, [126](#)
- ccsSupported
  - CDMASysInfo, [140](#)
- ccsSupportedValid
  - CDMASysInfo, [140](#)
- cdmaMsgDecodingParams, [130](#)
  - absoluteValidity, [132](#)
  - mcTimeStamp, [132](#)
  - messageLength, [132](#)
  - pAlertPriority, [132](#)
  - pCallbkAddr, [132](#)
  - pCallbkAddrLength, [132](#)
  - pDisplayMode, [132](#)

- pLanguage, [132](#)
- pMessage, [132](#)
- pMessageID, [132](#)
- pPriority, [132](#)
- pPrivacy, [132](#)
- pReadAcknowledgementReq, [133](#)
- pRelativeValidity, [133](#)
- pSenderAddr, [133](#)
- pSenderAddrLength, [133](#)
- pTextMsg, [133](#)
- pTextMsgLength, [133](#)
- pUserAcknowledgementReq, [133](#)
- cdmaMsgEncodingParams, [133](#)
  - messageId, [134](#)
  - pCallbackAddr, [134](#)
  - pDestAddr, [134](#)
  - pEncodingAlphabet, [134](#)
  - pMessage, [134](#)
  - pMessageSize, [134](#)
  - pPriority, [134](#)
  - pRelValidity, [134](#)
  - pTextMsg, [134](#)
  - textMsgLength, [135](#)
- cdmaSysIdValid
  - CDMASysInfo, [140](#)
- cell\_resel\_priority
  - infoInterFreq, [299](#)
- cellBroadcastCap
  - AddSysInfo, [86](#)
- CellDb, [141](#)
  - mask, [141](#)
- CellID
  - qaQmiServingSystemParam, [427](#)
- cellID
  - GERANInfo, [214](#)
  - UMTSInfo, [617](#)
- cellId
  - GSMSysInfo, [263](#)
  - LTESysInfo, [330](#)
  - WCDMASysInfo, [710](#)
- cellIdValid
  - gsmCellInfo, [258](#)
  - GSMSysInfo, [263](#)
  - LTESysInfo, [330](#)
  - WCDMASysInfo, [710](#)
- cellInterFreqParams
  - infoInterFreq, [299](#)
- cellsTDD
  - umtsLTENbrCell, [619](#)
- CellParams
  - LTEInfoIntraFreq, [318](#)
- cellParams, [141](#)
  - pci, [142](#)
  - rsrp, [142](#)
  - rsrq, [142](#)
  - rssi, [142](#)
  - srxlev, [142](#)
- cellReselPriority
  - lteGsmCellInfo, [312](#)
  - LTEInfoIntraFreq, [318](#)
  - lteWcdmaCellInfo, [332](#)
- cells\_len
  - infoInterFreq, [299](#)
  - lteGsmCellInfo, [312](#)
- cellsLen
  - LTEInfoIntraFreq, [318](#)
  - lteWcdmaCellInfo, [332](#)
- changePIN
  - UIMChangePinReq, [598](#)
- changeUIMPIN, [142](#)
  - oldPINLen, [143](#)
  - oldPINVal, [143](#)
  - pinID, [143](#)
  - pinLen, [143](#)
  - pinVal, [143](#)
- ChannelRate, [144](#)
  - CurrChanRxRate, [145](#)
  - CurrChanTxRate, [145](#)
  - DUNCallInfoInd, [195](#)
  - MaxChanRxRate, [145](#)
  - MaxChanTxRate, [145](#)
- channelRate, [143](#)
  - CurrChanRxRate, [144](#)
  - CurrChanTxRate, [144](#)
- Chipset
  - DeviceConfigDetail, [190](#)
- ckLen
  - depersonalizationInformation, [187](#)
- ckVal
  - depersonalizationInformation, [187](#)
- ClkInfo, [147](#)
  - mask, [148](#)
- codingScheme
  - PLMNNetworkNameData, [402](#)
  - remotePartyName, [460](#)
- CommInfo, [151](#)
  - imsRegState, [153](#)
  - modemMode, [153](#)
  - psState, [153](#)
  - systemMode, [153](#)
  - temperature, [153](#)
- commonInfo
  - swiModemStatusResp, [566](#)
- concSvcInfo
  - qaQmiServingSystemParam, [427](#)
- ConnRateElmntsV4
  - WdsConnectionRate, [713](#)
- ConnRateElmntsV6
  - WdsConnectionRate, [713](#)
- connStatus
  - \_packetSrvStatus, [52](#)
- connectNumInfo, [154](#)
  - callerID, [156](#)
  - callerIDLen, [156](#)
  - numPlan, [156](#)
  - numPresInd, [156](#)

- numType, 156
- screeningInd, 156
- ConnectedPartyNum
  - arrConnectPartyNum, 97
- ConnectionStatus, 153
  - MDMCallDuration, 154
  - MDMConnStatus, 154
- connetionState
  - imsaPdpStatusInfo, 281
- content
  - authenticateResult, 101
  - readResult, 454
- contentLen
  - authenticateResult, 101
  - readResult, 454
- context
  - authenticationData, 102
- contextId
  - swiPDPRuntimeSettingsReq, 568
- contextType
  - swiPDPRuntimeSettingsReq, 568
- ControlMac
  - protocolSubtypeElement, 421
- countryInitials
  - PLMNNetworkNameData, 402
- cpich\_ecno
  - wcdmaCellInfo, 700
- cpich\_rscp
  - wcdmaCellInfo, 700
- cradleMountConfigStatus
  - QmiCbkLocCradleMountInd, 428
- crashData
  - CrashInfo, 158
- crashId
  - CrashInfo, 158
- CrashInfo, 156
  - crashData, 158
  - crashId, 158
  - crashStrLen, 158
  - gcDumpStrLen, 158
  - numCrashes, 158
  - pCrashString, 158
  - pGCDumpString, 158
- CrashInfoParams, 158
  - pCrashInfo, 159
  - pDevCrashStatus, 159
- crashStrLen
  - CrashInfo, 158
- CreateProfileIn, 159
  - curProfile, 160
  - pProfileID, 160
  - pProfileType, 160
- CreateProfileOut, 160
  - pExtErrorCode, 160
  - pProfileIndex, 160
  - pProfileType, 160
- csAttachState
  - ServingSystemInfo, 482
- servSystem, 484
- csBarStatus
  - CallBarringSysInfo, 106
  - callBarStatus, 107
- cur\_carr\_name
  - slqsfwinfno\_s, 521
- cur\_carr\_rev
  - slqsfwinfno\_s, 521
- curAMRConfig, 162
  - gsmAmrStat, 163
  - wcdmaAmrStat, 163
- curProfile
  - \_GetProfileSettingOut, 47
  - CreateProfileIn, 160
  - ModifyProfileIn, 336
- CurrChanRxRate
  - ChannelRate, 145
  - channelRate, 144
- CurrChanTxRate
  - ChannelRate, 145
  - channelRate, 144
- CurrDataSysStat, 163
  - pCurrNetworkInfo, 164
  - pNetworkInfoLen, 164
  - pPrefNetwork, 164
- CurrImageInfo, 167
  - buildID, 167
  - buildIDLen, 167
  - imageType, 167
  - uniqueID, 167
- CurrNetworkInfo, 168
  - NetworkType, 170
  - RATMask, 170
  - SOMask, 170
- current\_channel\_rx\_rate
  - WDSSWICurrentChannelRates, 725
- current\_channel\_tx\_rate
  - WDSSWICurrentChannelRates, 725
- currentCatEvent, 164
  - CatAlphaldtfr, 165
  - CatEndPS, 165
  - CatEvIDData, 165
  - CatEventLst, 165
  - CatRefresh, 165
- CurrentImgList, 165
  - carrier, 165
  - fwvers, 165
  - numEntries, 166
  - pCurrImgInfo, 166
  - pkgver, 166
  - priver, 166
- currentNetwork
  - dataBearerTechnology, 181
- CurrentPLMN
  - qaQmiServingSystemParam, 427
- currentPLMN, 166
  - MCC, 166
  - MNC, 167



- netDescr, [167](#)
- netDescrLength, [167](#)
- cust\_attr
  - custSettingInfo, [175](#)
- cust\_id
  - custSettingInfo, [175](#)
  - getCustomInput, [227](#)
  - setCustomSettingV2, [493](#)
- cust\_value
  - custSettingInfo, [175](#)
  - setCustomSettingV2, [493](#)
- custFeaturesInfo, [170](#)
  - GpsEnable, [172](#)
  - pDHCPRelayEnabled, [172](#)
  - pDisableIMSI, [172](#)
  - pGPSLPM, [172](#)
  - pGPSSel, [172](#)
  - plPFamSupport, [172](#)
  - plsVoiceEnabled, [172](#)
  - pRMAutoConnect, [172](#)
  - pSMSSupport, [172](#)
  - qaGobiApiDms.h, [837](#)
- custFeaturesSetting, [172](#)
  - pDHCPRelayEnabled, [174](#)
  - pGPSEnable, [174](#)
  - pGPSLPM, [174](#)
  - pGPSSel, [174](#)
  - plsVoiceEnabled, [174](#)
  - qaGobiApiDms.h, [840](#)
- custSetting
  - custSettingList, [176](#)
- custSettingInfo, [174](#)
  - cust\_attr, [175](#)
  - cust\_id, [175](#)
  - cust\_value, [175](#)
  - id\_length, [175](#)
  - value\_length, [175](#)
- custSettingList, [175](#)
  - custSetting, [176](#)
  - list\_type, [176](#)
  - num\_instances, [176](#)
- CwtMute
  - GetM2MAudioProfileResp, [240](#)
  - GetM2MAVMuteResp, [243](#)
- DEVICE\_STATE\_BOOT
  - qaGobiApiCbK.h, [782](#)
- DEVICE\_STATE\_DISCONNECTED
  - qaGobiApiCbK.h, [782](#)
- DEVICE\_STATE\_READY
  - qaGobiApiCbK.h, [782](#)
- DEFAULTBYTEVALUE
  - qaGobiApiPds.h, [959](#)
- DEFAULTLONGVALUE
  - qaGobiApiPds.h, [959](#)
- DEFAULTWORDVALUE
  - qaGobiApiPds.h, [959](#)
- DEREGISTER\_EVENT
  - qaGobiApiCbK.h, [742](#)
- DEREGISTER\_SRV
  - qaGobiApiCbK.h, [742](#)
- DEVICE\_RESET
  - qaGobiApiFms.h, [878](#)
- DEVICE\_SHUTDOWN
  - qaGobiApiFms.h, [878](#)
- DRCCover
  - DRCParams, [193](#)
- DRCParams, [193](#)
  - DRCCover, [193](#)
  - DRCValue, [193](#)
- DRCValue
  - DRCParams, [193](#)
- DTMFEvent
  - DTMFInfo, [194](#)
- DTMFInfo, [194](#)
  - callID, [194](#)
  - DTMFEvent, [194](#)
  - digitBuff, [194](#)
  - digitCnt, [194](#)
- DTMFInformation
  - voiceDTMFEventInfo, [649](#)
- DTMFInterdigitInterval
  - DTMFLengths, [195](#)
- DTMFLengths, [194](#)
  - DTMFInterdigitInterval, [195](#)
  - DTMFPulseWidth, [195](#)
- DTMFPulseWidth
  - DTMFLengths, [195](#)
- DTMFdigit
  - voiceContDTMFInfo, [648](#)
- DTMInd
  - qaQmiServingSystemParam, [427](#)
- DUNCallInfoInd, [195](#)
  - CallEndReason, [195](#)
  - ChannelRate, [195](#)
  - DataBearerTech, [196](#)
  - DormancyStatus, [196](#)
  - MdmConnStatus, [196](#)
  - RXOKBytesCount, [196](#)
  - TXOKBytesCount, [196](#)
- Data
  - CatEventIDDDataTlv, [124](#)
- data
  - authenticationData, [102](#)
  - SMSCAddress, [543](#)
  - SMSEtwSMessage, [543](#)
  - SMSTransferRouteMTMessage, [554](#)
  - SwiOTAMsg\_s, [567](#)
- data\_len
  - SwiOTAMsg\_s, [567](#)
- dataBearerMask
  - dataBearers, [177](#)
- DataBearerTech, [177](#)
  - DUNCallInfoInd, [196](#)
  - ratValue, [179](#)
  - soMask, [179](#)
  - techType, [179](#)

- DataBearerTechExt, [179](#)
  - pBearerTech, [179](#)
  - pLastBearerTech, [179](#)
- dataBearerTechnology, [179](#)
  - currentNetwork, [181](#)
  - ratMask, [181](#)
  - soMask, [181](#)
- dataBearers, [176](#)
  - dataBearerMask, [177](#)
  - pCurDataBearerTechnology, [177](#)
  - pLastCallDataBearerTechnology, [177](#)
- dataCapabilities
  - dataSrvCapabilities, [182](#)
- dataCapabilitiesLen
  - dataSrvCapabilities, [182](#)
- dataLen
  - authenticationData, [102](#)
- DataLength
  - CatEventIDDDataTlv, [124](#)
- dataRate, [181](#)
  - dataRateMax, [181](#)
  - guaranteedRate, [181](#)
- dataRateMax
  - dataRate, [181](#)
- DataSrvCapabilities
  - qaQmiServingSystemParam, [427](#)
- dataSrvCapabilities, [181](#)
  - dataCapabilities, [182](#)
  - dataCapabilitiesLen, [182](#)
- DataStatusDetail, [182](#)
  - IPAddress, [184](#)
  - LastErrCode, [184](#)
- DataULongLongTlv, [184](#)
  - TlvPresent, [184](#)
  - ulldata, [184](#)
- DataULongTlv, [184](#)
  - TlvPresent, [184](#)
  - ulldata, [184](#)
- Date
  - wcdmaLongMsgDecodingParams, [703](#)
  - wcdmaMsgDecodingParams, [704](#)
- day
  - UniversalTime, [628](#)
- dayOfWeek
  - UniversalTime, [628](#)
- daylightSavings
  - qaQmi3Gpp2TimeZone, [422](#)
- DcsUsbPortNames, [184](#)
  - AtCmdPort, [184](#)
  - DmPort, [184](#)
  - NmeaPort, [184](#)
- defaultRoamInd
  - qaQmiServingSystemParam, [427](#)
- delAssistDataStatus, [184](#)
  - status, [185](#)
- delayClass
  - GPRSQoS, [251](#)
  - GPRSRequestedQoS, [252](#)
- DeleteStoredImage
  - qaGobiApiFms.h, [881](#)
- deliveryErrSDU
  - UMTSMinQoS, [622](#)
  - UMTSQoS, [625](#)
- depersonalizationInformation, [185](#)
  - ckLen, [187](#)
  - ckVal, [187](#)
  - feature, [187](#)
  - operation, [187](#)
- depersonalisationInfo
  - UIMDepersonalizationReq, [599](#)
- Description
  - SlqsNas3GppNetworkInfo, [523](#)
- description
  - omaDmFotaTlv, [380](#)
  - omaDmFotaTlvExt, [382](#)
- descriptionlength
  - omaDmFotaTlv, [380](#)
  - omaDmFotaTlvExt, [382](#)
- destPortRangeEnd
  - TFTIDParams, [587](#)
- destPortRangeStart
  - TFTIDParams, [588](#)
- detailSvcInfo, [187](#)
  - hdrHybrid, [189](#)
  - hdrSrvStatus, [189](#)
  - isSysForbidden, [189](#)
  - srvCapability, [189](#)
  - srvStatus, [189](#)
- DetailedSvcInfo
  - qaQmiServingSystemParam, [427](#)
- dev
  - qmifwinfo\_s, [441](#)
- Device
  - SetM2MAudioAVCFGRReq, [500](#)
- Device Connectivity Service (DCS), [21](#)
- Device Management Service (DMS), [23](#)
- device\_state\_enum
  - qaGobiApiCbk.h, [782](#)
- DeviceConfigDetail, [189](#)
  - Chipset, [190](#)
  - HWVersion, [190](#)
  - QLIC, [190](#)
  - Technology, [190](#)
- DiagInfo
  - allCallsDiagInfo, [87](#)
  - arrDiagInfo, [98](#)
- diagInfo, [190](#)
  - diagInfoLen, [191](#)
  - diagnosticInfo, [191](#)
- diagInfoLen
  - diagInfo, [191](#)
- diagnosticInfo
  - diagInfo, [191](#)
- digitBuff
  - DTMFInfo, [194](#)
- digitCnt

- burstDTMFInfo, [105](#)
- DTMFInfo, [194](#)
- dirNum, [191](#)
  - dirNum, [191](#)
  - dirNumLen, [191](#)
  - dirNum, [191](#)
- dirNumLen
  - dirNum, [191](#)
- direction
  - callInfo, [118](#)
- dispType
  - extDispRecInfo, [203](#)
- displayCondition
  - serviceProviderName, [481](#)
- dl\_bw\_value
  - PhyCaAggPcellInfo, [394](#)
  - PhyCaAggScellIDBw, [395](#)
  - PhyCaAggScellInfo, [397](#)
- DmPort
  - DcsUsbPortNames, [184](#)
- dmsCurrentPRLInfo, [191](#)
  - pPRLPreference, [192](#)
  - pPRLVersion, [192](#)
  - qaGobiApiDms.h, [841](#)
- Domain, [192](#)
  - domainLen, [192](#)
  - domainName, [192](#)
- domain
  - DomainNameList, [193](#)
- domainLen
  - Domain, [192](#)
- domainName
  - Domain, [192](#)
- DomainNameList, [193](#)
  - domain, [193](#)
  - numInstances, [193](#)
- DormancyStatus
  - DUNCallInfoInd, [196](#)
- downLink
  - NSSAudioCtrl, [374](#)
- dscp
  - QosMap, [453](#)
- dtmSupp
  - GSMSysInfo, [263](#)
- dtmSuppValid
  - GSMSysInfo, [263](#)
- eGOBI\_DEV\_SERIES\_9X15
  - qaGobiApiFms.h, [879](#)
- eGOBI\_DEV\_SERIES\_9X30
  - qaGobiApiFms.h, [879](#)
- eGOBI\_DEV\_SERIES\_G3K
  - qaGobiApiFms.h, [879](#)
- eGOBI\_DEV\_SERIES\_NON\_GOBI
  - qaGobiApiFms.h, [879](#)
- eGOBI\_DEV\_SERIES\_SIERRA\_GOBI
  - qaGobiApiFms.h, [879](#)
- eGOBI\_DEV\_SERIES\_UNKNOWN
  - qaGobiApiFms.h, [879](#)
- eGOBI\_IMG\_CAR\_3
  - qaGobiApiFms.h, [880](#)
- eGOBI\_IMG\_CAR\_AERIS
  - qaGobiApiFms.h, [880](#)
- eGOBI\_IMG\_CAR\_ALLTEL
  - qaGobiApiFms.h, [879](#)
- eGOBI\_IMG\_CAR\_AMX\_TELCEL
  - qaGobiApiFms.h, [880](#)
- eGOBI\_IMG\_CAR\_ATT
  - qaGobiApiFms.h, [880](#)
- eGOBI\_IMG\_CAR\_BELL
  - qaGobiApiFms.h, [879](#)
- eGOBI\_IMG\_CAR\_BHARTI
  - qaGobiApiFms.h, [880](#)
- eGOBI\_IMG\_CAR\_BRASIL\_VIVO
  - qaGobiApiFms.h, [880](#)
- eGOBI\_IMG\_CAR\_CHINA\_MOBILE
  - qaGobiApiFms.h, [880](#)
- eGOBI\_IMG\_CAR\_CHINA\_TELECOM
  - qaGobiApiFms.h, [880](#)
- eGOBI\_IMG\_CAR\_CHINA\_UNICOM
  - qaGobiApiFms.h, [880](#)
- eGOBI\_IMG\_CAR\_EMOBILE
  - qaGobiApiFms.h, [880](#)
- eGOBI\_IMG\_CAR\_FACTORY
  - qaGobiApiFms.h, [879](#)
- eGOBI\_IMG\_CAR\_GENERIC
  - qaGobiApiFms.h, [879](#)
- eGOBI\_IMG\_CAR\_GENERIC\_CDMA
  - qaGobiApiFms.h, [880](#)
- eGOBI\_IMG\_CAR\_IUSACELL
  - qaGobiApiFms.h, [880](#)
- eGOBI\_IMG\_CAR\_KDDI
  - qaGobiApiFms.h, [880](#)
- eGOBI\_IMG\_CAR\_KT\_FREETEL
  - qaGobiApiFms.h, [880](#)
- eGOBI\_IMG\_CAR\_LEAP
  - qaGobiApiFms.h, [880](#)
- eGOBI\_IMG\_CAR\_METROPCS
  - qaGobiApiFms.h, [880](#)
- eGOBI\_IMG\_CAR\_NETCOM
  - qaGobiApiFms.h, [880](#)
- eGOBI\_IMG\_CAR\_NORF
  - qaGobiApiFms.h, [879](#)
- eGOBI\_IMG\_CAR\_NTT\_DOCOMO
  - qaGobiApiFms.h, [880](#)
- eGOBI\_IMG\_CAR\_O2
  - qaGobiApiFms.h, [880](#)
- eGOBI\_IMG\_CAR\_OMH
  - qaGobiApiFms.h, [880](#)
- eGOBI\_IMG\_CAR\_ORANGE
  - qaGobiApiFms.h, [880](#)
- eGOBI\_IMG\_CAR\_RELIANCE1
  - qaGobiApiFms.h, [880](#)
- eGOBI\_IMG\_CAR\_RELIANCE2
  - qaGobiApiFms.h, [880](#)
- eGOBI\_IMG\_CAR\_ROGERS
  - qaGobiApiFms.h, [880](#)

- eGOBI\_IMG\_CAR\_SFR
  - qaGobiApiFms.h, [880](#)
- eGOBI\_IMG\_CAR\_SINGTEL\_OPTUS
  - qaGobiApiFms.h, [880](#)
- eGOBI\_IMG\_CAR\_SK\_TELCOM1
  - qaGobiApiFms.h, [880](#)
- eGOBI\_IMG\_CAR\_SK\_TELCOM2
  - qaGobiApiFms.h, [880](#)
- eGOBI\_IMG\_CAR\_SOFTBANK
  - qaGobiApiFms.h, [880](#)
- eGOBI\_IMG\_CAR\_SPRINT
  - qaGobiApiFms.h, [879](#)
- eGOBI\_IMG\_CAR\_SWISSCOM
  - qaGobiApiFms.h, [880](#)
- eGOBI\_IMG\_CAR\_TATA
  - qaGobiApiFms.h, [880](#)
- eGOBI\_IMG\_CAR\_TELCOM\_ITALIA
  - qaGobiApiFms.h, [880](#)
- eGOBI\_IMG\_CAR\_TELCOM\_NZ
  - qaGobiApiFms.h, [880](#)
- eGOBI\_IMG\_CAR\_TELEFONICA
  - qaGobiApiFms.h, [880](#)
- eGOBI\_IMG\_CAR\_TELENOR
  - qaGobiApiFms.h, [880](#)
- eGOBI\_IMG\_CAR\_TELIASONERA
  - qaGobiApiFms.h, [880](#)
- eGOBI\_IMG\_CAR\_TELSTRA1
  - qaGobiApiFms.h, [880](#)
- eGOBI\_IMG\_CAR\_TELSTRA2
  - qaGobiApiFms.h, [880](#)
- eGOBI\_IMG\_CAR\_TELUS
  - qaGobiApiFms.h, [880](#)
- eGOBI\_IMG\_CAR\_TMOBILE
  - qaGobiApiFms.h, [880](#)
- eGOBI\_IMG\_CAR\_US
  - qaGobiApiFms.h, [880](#)
- eGOBI\_IMG\_CAR\_VERIZON
  - qaGobiApiFms.h, [879](#)
- eGOBI\_IMG\_CAR\_VODAFONE
  - qaGobiApiFms.h, [880](#)
- eGOBI\_IMG\_GPS\_ASSISTED
  - qaGobiApiFms.h, [881](#)
- eGOBI\_IMG\_GPS\_NO\_XTRA
  - qaGobiApiFms.h, [881](#)
- eGOBI\_IMG\_GPS\_NONE
  - qaGobiApiFms.h, [881](#)
- eGOBI\_IMG\_GPS\_STAND\_ALONE
  - qaGobiApiFms.h, [881](#)
- eGOBI\_IMG\_REG\_ASIA
  - qaGobiApiFms.h, [881](#)
- eGOBI\_IMG\_REG\_AUS
  - qaGobiApiFms.h, [881](#)
- eGOBI\_IMG\_REG\_EU
  - qaGobiApiFms.h, [881](#)
- eGOBI\_IMG\_REG\_GLOBAL
  - qaGobiApiFms.h, [881](#)
- eGOBI\_IMG\_REG\_LA
  - qaGobiApiFms.h, [881](#)
- eGOBI\_IMG\_REG\_NA
  - qaGobiApiFms.h, [881](#)
- eGOBI\_IMG\_TECH\_CDMA
  - qaGobiApiFms.h, [881](#)
- eGOBI\_IMG\_TECH\_UMTS
  - qaGobiApiFms.h, [881](#)
- eGobi\_DEV\_SERIES\_MC83
  - qaGobiApiFms.h, [879](#)
- eNAS\_LTE\_CPHY\_CA\_BW\_NRB\_100
  - qaGobiApiNas.h, [918](#)
- eNAS\_LTE\_CPHY\_CA\_BW\_NRB\_15
  - qaGobiApiNas.h, [918](#)
- eNAS\_LTE\_CPHY\_CA\_BW\_NRB\_25
  - qaGobiApiNas.h, [918](#)
- eNAS\_LTE\_CPHY\_CA\_BW\_NRB\_50
  - qaGobiApiNas.h, [918](#)
- eNAS\_LTE\_CPHY\_CA\_BW\_NRB\_6
  - qaGobiApiNas.h, [918](#)
- eNAS\_LTE\_CPHY\_CA\_BW\_NRB\_75
  - qaGobiApiNas.h, [918](#)
- eNAS\_LTE\_CPHY\_SCELL\_STATE\_CONFIGURED\_A-CTIVATED
  - qaGobiApiNas.h, [918](#)
- eNAS\_LTE\_CPHY\_SCELL\_STATE\_CONFIGURED\_D-EACTIVATED
  - qaGobiApiNas.h, [918](#)
- eNAS\_LTE\_CPHY\_SCELL\_STATE\_DECONFIGURED
  - qaGobiApiNas.h, [918](#)
- eNAS\_RADIO\_IF\_GSM
  - qaGobiApiNas.h, [918](#)
- eNAS\_RADIO\_IF\_LTE
  - qaGobiApiNas.h, [918](#)
- eNAS\_RADIO\_IF\_TDSCDMA
  - qaGobiApiNas.h, [918](#)
- eNAS\_RADIO\_IF\_UMTS
  - qaGobiApiNas.h, [918](#)
- eQA\_QMI\_SVC\_NA
  - qaGobiApiCbk.h, [782](#)
- eQA\_QMI\_SVC\_NAS
  - qaGobiApiCbk.h, [782](#)
- eQA\_QMI\_SVC\_WDS
  - qaGobiApiCbk.h, [782](#)
- eQCWWAN\_ERR\_API\_MUTEX\_TIMEOUT
  - qmerrno.h, [1154](#)
- eQCWWAN\_ERR\_BUFFER\_SZ
  - qmerrno.h, [1153](#)
- eQCWWAN\_ERR\_CANCEL\_OP
  - qmerrno.h, [1154](#)
- eQCWWAN\_ERR\_DRIVER
  - qmerrno.h, [1154](#)
- eQCWWAN\_ERR\_ENUM\_BEGIN
  - qmerrno.h, [1153](#)
- eQCWWAN\_ERR\_ENUM\_END
  - qmerrno.h, [1154](#)
- eQCWWAN\_ERR\_FILE\_COPY
  - qmerrno.h, [1153](#)
- eQCWWAN\_ERR\_FILE\_OPEN
  - qmerrno.h, [1153](#)

- eQCWWAN\_ERR\_GENERAL  
qmerrno.h, [1153](#)
- eQCWWAN\_ERR\_INTERNAL  
qmerrno.h, [1153](#)
- eQCWWAN\_ERR\_INVALID\_ARG  
qmerrno.h, [1153](#)
- eQCWWAN\_ERR\_INVALID\_DEVID  
qmerrno.h, [1153](#)
- eQCWWAN\_ERR\_INVALID\_FILE  
qmerrno.h, [1153](#)
- eQCWWAN\_ERR\_INVALID\_QMI\_RSP  
qmerrno.h, [1153](#)
- eQCWWAN\_ERR\_MALFORMED\_QMI\_RSP  
qmerrno.h, [1153](#)
- eQCWWAN\_ERR\_MEMORY  
qmerrno.h, [1153](#)
- eQCWWAN\_ERR\_MULTIPLE\_DEVICES  
qmerrno.h, [1154](#)
- eQCWWAN\_ERR\_NO\_CANCELABLE\_OP  
qmerrno.h, [1154](#)
- eQCWWAN\_ERR\_NO\_CONNECTION  
qmerrno.h, [1153](#)
- eQCWWAN\_ERR\_NO\_DEVICE  
qmerrno.h, [1153](#)
- eQCWWAN\_ERR\_NO\_SIGNAL  
qmerrno.h, [1154](#)
- eQCWWAN\_ERR\_NONE  
qmerrno.h, [1153](#)
- eQCWWAN\_ERR\_NULL\_TLV  
qmerrno.h, [1157](#)
- eQCWWAN\_ERR\_OFFLINE  
qmerrno.h, [1154](#)
- eQCWWAN\_ERR\_PDU\_GENERATION  
qmerrno.h, [1154](#)
- eQCWWAN\_ERR\_QMI\_ABORTED  
qmerrno.h, [1154](#)
- eQCWWAN\_ERR\_QMI\_ACCESS\_DENIED  
qmerrno.h, [1156](#)
- eQCWWAN\_ERR\_QMI\_ACK\_NOT\_SENT  
qmerrno.h, [1156](#)
- eQCWWAN\_ERR\_QMI\_ARG\_TOO\_LONG  
qmerrno.h, [1154](#)
- eQCWWAN\_ERR\_QMI\_AUTHENTICATION\_FAILED  
qmerrno.h, [1155](#)
- eQCWWAN\_ERR\_QMI\_AUTHENTICATION\_LOCK  
qmerrno.h, [1155](#)
- eQCWWAN\_ERR\_QMI\_BUNDLING\_NOT\_SUPPORTED  
qmerrno.h, [1156](#)
- eQCWWAN\_ERR\_QMI\_CALL\_FAILED  
qmerrno.h, [1154](#)
- eQCWWAN\_ERR\_QMI\_CARD\_BUSY\_RSP  
qmerrno.h, [1157](#)
- eQCWWAN\_ERR\_QMI\_CARD\_CALL\_CONTROL\_FAILED  
qmerrno.h, [1156](#)
- eQCWWAN\_ERR\_QMI\_CAT\_END  
qmerrno.h, [1157](#)
- eQCWWAN\_ERR\_QMI\_CAT\_START  
qmerrno.h, [1157](#)
- eQCWWAN\_ERR\_QMI\_CAUSE\_CODE  
qmerrno.h, [1155](#)
- eQCWWAN\_ERR\_QMI\_CLIENT\_IDS\_EXHAUSTED  
qmerrno.h, [1154](#)
- eQCWWAN\_ERR\_QMI\_CONNECT  
qmerrno.h, [1153](#)
- eQCWWAN\_ERR\_QMI\_DEVICE\_IN\_USE  
qmerrno.h, [1154](#)
- eQCWWAN\_ERR\_QMI\_DEVICE\_NOT\_READY  
qmerrno.h, [1155](#)
- eQCWWAN\_ERR\_QMI\_DEVICE\_STORAGE\_FULL  
qmerrno.h, [1155](#)
- eQCWWAN\_ERR\_QMI\_DISABLED  
qmerrno.h, [1155](#)
- eQCWWAN\_ERR\_QMI\_ENCODING  
qmerrno.h, [1155](#)
- eQCWWAN\_ERR\_QMI\_ENVELOPE\_CMD\_FAILURE  
qmerrno.h, [1157](#)
- eQCWWAN\_ERR\_QMI\_EVENT\_REG\_FAILED  
qmerrno.h, [1157](#)
- eQCWWAN\_ERR\_QMI\_EXTENDED\_INTERNAL  
qmerrno.h, [1156](#)
- eQCWWAN\_ERR\_QMI\_FDN\_RESTRICT  
qmerrno.h, [1156](#)
- eQCWWAN\_ERR\_QMI\_FLOW\_SUSPENDED  
qmerrno.h, [1155](#)
- eQCWWAN\_ERR\_QMI\_GENERAL  
qmerrno.h, [1155](#)
- eQCWWAN\_ERR\_QMI\_HARDWARE\_RESTRICTED  
qmerrno.h, [1156](#)
- eQCWWAN\_ERR\_QMI\_IFACE  
qmerrno.h, [1153](#)
- eQCWWAN\_ERR\_QMI\_INCOMPATIBLE\_STATE  
qmerrno.h, [1156](#)
- eQCWWAN\_ERR\_QMI\_INCORRECT\_FLOW\_FILTER  
qmerrno.h, [1155](#)
- eQCWWAN\_ERR\_QMI\_INCORRECT\_PIN  
qmerrno.h, [1154](#)
- eQCWWAN\_ERR\_QMI\_INFO\_UNAVAILABLE  
qmerrno.h, [1156](#)
- eQCWWAN\_ERR\_QMI\_INJECT\_TIMEOUT  
qmerrno.h, [1156](#)
- eQCWWAN\_ERR\_QMI\_INSUFFICIENT\_RESOURCE-S  
qmerrno.h, [1155](#)
- eQCWWAN\_ERR\_QMI\_INTERFACE\_NOT\_FOUND  
qmerrno.h, [1155](#)
- eQCWWAN\_ERR\_QMI\_INTERNAL  
qmerrno.h, [1154](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_ARG  
qmerrno.h, [1155](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_CLIENT\_ID  
qmerrno.h, [1154](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_DATA\_FORMAT  
qmerrno.h, [1155](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_ENVELOPE\_CMD

- qmerrno.h, [1157](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_HANDLE
  - qmerrno.h, [1154](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_ID
  - qmerrno.h, [1155](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_INDEX
  - qmerrno.h, [1155](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_IP\_FAMILY\_PREF
  - qmerrno.h, [1155](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_MCAST\_HANDLE
  - qmerrno.h, [1155](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_MESSAGE\_ID
  - qmerrno.h, [1155](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_OPERATION
  - qmerrno.h, [1155](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_PDP\_TYPE
  - qmerrno.h, [1154](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_PINID
  - qmerrno.h, [1154](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_PROFILE
  - qmerrno.h, [1154](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_PROFILE\_TYPE
  - qmerrno.h, [1154](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_PS\_ATTACH\_ACTI-  
ON
  - qmerrno.h, [1154](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_QMI\_CMD
  - qmerrno.h, [1155](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_QOS\_ID
  - qmerrno.h, [1155](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_REGISTER\_ACTIO-  
N
  - qmerrno.h, [1154](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_SERVICE\_TYPE
  - qmerrno.h, [1154](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_TECH\_PREF
  - qmerrno.h, [1154](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_TERMINAL\_RSP
  - qmerrno.h, [1157](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_TRANSITION
  - qmerrno.h, [1155](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_TX\_ID
  - qmerrno.h, [1154](#)
- eQCWWAN\_ERR\_QMI\_MALFORMED\_MSG
  - qmerrno.h, [1154](#)
- eQCWWAN\_ERR\_QMI\_MAX
  - qmerrno.h, [1156](#)
- eQCWWAN\_ERR\_QMI\_MAX\_MCAST\_REQUESTS\_I-  
N\_USE
  - qmerrno.h, [1155](#)
- eQCWWAN\_ERR\_QMI\_MAX\_QOS\_REQUESTS\_IN\_  
USE
  - qmerrno.h, [1155](#)
- eQCWWAN\_ERR\_QMI\_MESSAGE\_DELIVERY\_FAIL-  
URE
  - qmerrno.h, [1155](#)
- eQCWWAN\_ERR\_QMI\_MESSAGE\_NOT\_SENT
  - qmerrno.h, [1155](#)
- eQCWWAN\_ERR\_QMI\_MISSING\_ARG
  - qmerrno.h, [1154](#)
- eQCWWAN\_ERR\_QMI\_MSG\_BLOCKED
  - qmerrno.h, [1156](#)
- eQCWWAN\_ERR\_QMI\_NETWORK\_ABORTED
  - qmerrno.h, [1156](#)
- eQCWWAN\_ERR\_QMI\_NETWORK\_NOT\_READY
  - qmerrno.h, [1155](#)
- eQCWWAN\_ERR\_QMI\_NETWORK\_QOS\_UNAWARE
  - qmerrno.h, [1155](#)
- eQCWWAN\_ERR\_QMI\_NO\_EFFECT
  - qmerrno.h, [1154](#)
- eQCWWAN\_ERR\_QMI\_NO\_ENTRY
  - qmerrno.h, [1155](#)
- eQCWWAN\_ERR\_QMI\_NO\_FREE\_PROFILE
  - qmerrno.h, [1154](#)
- eQCWWAN\_ERR\_QMI\_NO\_MEMORY
  - qmerrno.h, [1154](#)
- eQCWWAN\_ERR\_QMI\_NO\_NETWORK\_FOUND
  - qmerrno.h, [1154](#)
- eQCWWAN\_ERR\_QMI\_NO\_RADIO
  - qmerrno.h, [1156](#)
- eQCWWAN\_ERR\_QMI\_NO\_SUBSCRIPTION
  - qmerrno.h, [1156](#)
- eQCWWAN\_ERR\_QMI\_NO\_THRESHOLDS
  - qmerrno.h, [1154](#)
- eQCWWAN\_ERR\_QMI\_NOT\_A\_MCAST\_IFACE
  - qmerrno.h, [1155](#)
- eQCWWAN\_ERR\_QMI\_NOT\_PROVISIONED
  - qmerrno.h, [1154](#)
- eQCWWAN\_ERR\_QMI\_NOT\_SUPPORTED
  - qmerrno.h, [1156](#)
- eQCWWAN\_ERR\_QMI\_OFFSET
  - qmerrno.h, [1154](#)
- eQCWWAN\_ERR\_QMI\_OP\_DEVICE\_UNSUPPORTE-  
D
  - qmerrno.h, [1154](#)
- eQCWWAN\_ERR\_QMI\_OP\_NETWORK\_UNSUPPOR-  
TED
  - qmerrno.h, [1154](#)
- eQCWWAN\_ERR\_QMI\_OP\_PARTIAL\_FAILURE
  - qmerrno.h, [1156](#)
- eQCWWAN\_ERR\_QMI\_OUT\_OF\_CALL
  - qmerrno.h, [1154](#)
- eQCWWAN\_ERR\_QMI\_PIN\_BLOCKED
  - qmerrno.h, [1155](#)
- eQCWWAN\_ERR\_QMI\_PIN\_PERM\_BLOCKED
  - qmerrno.h, [1155](#)
- eQCWWAN\_ERR\_QMI\_POLICY\_MISMATCH
  - qmerrno.h, [1156](#)
- eQCWWAN\_ERR\_QMI\_REQ
  - qmerrno.h, [1153](#)
- eQCWWAN\_ERR\_QMI\_REQ\_SCH
  - qmerrno.h, [1153](#)
- eQCWWAN\_ERR\_QMI\_REQ\_TO
  - qmerrno.h, [1153](#)
- eQCWWAN\_ERR\_QMI\_REQUESTED\_NUM\_UNSUP-  
PORTED



- qmerrno.h, [1155](#)
- eQCWWAN\_ERR\_QMI\_RSP
  - qmerrno.h, [1153](#)
- eQCWWAN\_ERR\_QMI\_RSP\_TO
  - qmerrno.h, [1153](#)
- eQCWWAN\_ERR\_QMI\_SEGMENT\_ORDER
  - qmerrno.h, [1156](#)
- eQCWWAN\_ERR\_QMI\_SEGMENT\_TOO\_LONG
  - qmerrno.h, [1156](#)
- eQCWWAN\_ERR\_QMI\_SESSION\_INACTIVE
  - qmerrno.h, [1155](#)
- eQCWWAN\_ERR\_QMI\_SESSION\_INVALID
  - qmerrno.h, [1155](#)
- eQCWWAN\_ERR\_QMI\_SESSION\_OWNERSHIP
  - qmerrno.h, [1155](#)
- eQCWWAN\_ERR\_QMI\_SIM\_FILE\_NOT\_FOUND
  - qmerrno.h, [1156](#)
- eQCWWAN\_ERR\_QMI\_SIM\_NOT\_INITIALIZED
  - qmerrno.h, [1155](#)
- eQCWWAN\_ERR\_QMI\_SMSC\_ADDR
  - qmerrno.h, [1156](#)
- eQCWWAN\_ERR\_QMI\_SUPS\_FAILURE\_CAUSE
  - qmerrno.h, [1156](#)
- eQCWWAN\_ERR\_QMI\_TPDU\_TYPE
  - qmerrno.h, [1156](#)
- eQCWWAN\_ERR\_QMI\_UNABORTABLE\_TRANSACTION
  - qmerrno.h, [1154](#)
- eQCWWAN\_ERR\_QMI\_UNKNOWN
  - qmerrno.h, [1155](#)
- eQCWWAN\_ERR\_QMI\_WIDTH
  - qmerrno.h, [1157](#)
- eQCWWAN\_ERR\_RESET
  - qmerrno.h, [1154](#)
- eQCWWAN\_ERR\_SWICM\_AM\_VERS\_ERROR
  - qmerrno.h, [1156](#)
- eQCWWAN\_ERR\_SWICM\_CALL\_IN\_PROGRESS
  - qmerrno.h, [1156](#)
- eQCWWAN\_ERR\_SWICM\_END
  - qmerrno.h, [1156](#)
- eQCWWAN\_ERR\_SWICM\_FAILED\_TO\_KILL\_SDK\_PROCESS
  - qmerrno.h, [1156](#)
- eQCWWAN\_ERR\_SWICM\_INVALID\_SESSION\_ID
  - qmerrno.h, [1156](#)
- eQCWWAN\_ERR\_SWICM\_INVALID\_V4\_SESSION\_ID
  - qmerrno.h, [1156](#)
- eQCWWAN\_ERR\_SWICM\_INVALID\_V6\_SESSION\_ID
  - qmerrno.h, [1156](#)
- eQCWWAN\_ERR\_SWICM\_NOT\_IMPLEMENTED
  - qmerrno.h, [1156](#)
- eQCWWAN\_ERR\_SWICM\_QMI\_CLNT\_NOT\_SUPPORTED
  - qmerrno.h, [1156](#)
- eQCWWAN\_ERR\_SWICM\_QMI\_SVC\_NOT\_SUPPORTED
  - qmerrno.h, [1156](#)
- qmerrno.h, [1156](#)
- eQCWWAN\_ERR\_SWICM\_SM\_NO\_AVAILABLE\_SESSIONS
  - qmerrno.h, [1156](#)
- eQCWWAN\_ERR\_SWICM\_SOCKET\_IN\_USE
  - qmerrno.h, [1156](#)
- eQCWWAN\_ERR\_SWICM\_START
  - qmerrno.h, [1156](#)
- eQCWWAN\_ERR\_SWICM\_TIMEOUT
  - qmerrno.h, [1156](#)
- eQCWWAN\_ERR\_SWICM\_V4DWN\_V6DWN
  - qmerrno.h, [1156](#)
- eQCWWAN\_ERR\_SWICM\_V4DWN\_V6UP
  - qmerrno.h, [1156](#)
- eQCWWAN\_ERR\_SWICM\_V4UP\_V6DWN
  - qmerrno.h, [1156](#)
- eQCWWAN\_ERR\_SWICM\_V4UP\_V6UP
  - qmerrno.h, [1156](#)
- eQCWWAN\_ERR\_SWIDCS\_APP\_DISCONNECTED
  - qmerrno.h, [1157](#)
- eQCWWAN\_ERR\_SWIDCS\_DEVNODE\_NOT\_FOUND
  - qmerrno.h, [1157](#)
- eQCWWAN\_ERR\_SWIDCS\_END
  - qmerrno.h, [1157](#)
- eQCWWAN\_ERR\_SWIDCS\_FILEIO\_ERR
  - qmerrno.h, [1157](#)
- eQCWWAN\_ERR\_SWIDCS\_IOCTL\_ERR
  - qmerrno.h, [1157](#)
- eQCWWAN\_ERR\_SWIDCS\_START
  - qmerrno.h, [1157](#)
- eQCWWAN\_ERR\_SWIIM\_CORRUPTED\_FW\_IMAGE
  - qmerrno.h, [1157](#)
- eQCWWAN\_ERR\_SWIIM\_END
  - qmerrno.h, [1157](#)
- eQCWWAN\_ERR\_SWIIM\_FILE\_NOT\_FOUND
  - qmerrno.h, [1157](#)
- eQCWWAN\_ERR\_SWIIM\_FIRMWARE\_NOT\_DOWNLOADED
  - qmerrno.h, [1157](#)
- eQCWWAN\_ERR\_SWIIM\_FW\_PREFERENCE\_MISMATCH
  - qmerrno.h, [1157](#)
- eQCWWAN\_ERR\_SWIIM\_FW\_UPDATE\_FAIL
  - qmerrno.h, [1157](#)
- eQCWWAN\_ERR\_SWIIM\_FW\_UPDATE\_SUCCESS
  - qmerrno.h, [1157](#)
- eQCWWAN\_ERR\_SWIIM\_INVALID\_PATH
  - qmerrno.h, [1157](#)
- eQCWWAN\_ERR\_SWIIM\_OPENING\_DIR
  - qmerrno.h, [1157](#)
- eQCWWAN\_ERR\_SWIIM\_OPENING\_FILE
  - qmerrno.h, [1157](#)
- eQCWWAN\_ERR\_SWIIM\_START
  - qmerrno.h, [1157](#)
- eQCWWAN\_ERR\_SWISM\_END
  - qmerrno.h, [1157](#)

- eQCWWAN\_ERR\_SWISMS\_BEARER\_DATA\_NOT\_FOUND
  - qmerrno.h, [1157](#)
- eQCWWAN\_ERR\_SWISMS\_MSG\_CORRUPTED
  - qmerrno.h, [1157](#)
- eQCWWAN\_ERR\_SWISMS\_MSG\_LEN\_TOO\_LONG
  - qmerrno.h, [1157](#)
- eQCWWAN\_ERR\_SWISMS\_SMSC\_NUM\_CORRUPTED
  - qmerrno.h, [1157](#)
- eQCWWAN\_ERR\_SWISMS\_START
  - qmerrno.h, [1157](#)
- eSetServiceAutomaticTrackingDisable
  - qaGobiApiPds.h, [959](#)
- eSetServiceAutomaticTrackingEnable
  - qaGobiApiPds.h, [959](#)
- eTLV\_3GPP\_NETWORK\_INFO
  - qaNasPerformNetworkScan.h, [1151](#)
- eTLV\_CBK\_ALPHA\_IDENTIFIER
  - qaCbkCatEventReportInd.h, [728](#)
- eTLV\_CBK\_DISPLAY\_TEXT
  - qaCbkCatEventReportInd.h, [728](#)
- eTLV\_CBK\_END\_PROACTIVE\_SESSION
  - qaCbkCatEventReportInd.h, [728](#)
- eTLV\_CBK\_GET\_IN\_KEY
  - qaCbkCatEventReportInd.h, [728](#)
- eTLV\_CBK\_GET\_INPUT
  - qaCbkCatEventReportInd.h, [728](#)
- eTLV\_CBK\_LANGUAGE\_NOTIFICATION
  - qaCbkCatEventReportInd.h, [728](#)
- eTLV\_CBK\_REFRESH
  - qaCbkCatEventReportInd.h, [728](#)
- eTLV\_CBK\_SELECT\_ITEM
  - qaCbkCatEventReportInd.h, [728](#)
- eTLV\_CBK\_SETUP\_EVENT\_LIST
  - qaCbkCatEventReportInd.h, [728](#)
- eTLV\_CBK\_SETUP\_IDLE\_MODE\_TEXT
  - qaCbkCatEventReportInd.h, [728](#)
- eTLV\_CBK\_SETUP\_MENU
  - qaCbkCatEventReportInd.h, [728](#)
- eTLV\_END\_PROACTIVE\_SESSION\_LENGTH
  - qaCbkCatEventReportInd.h, [728](#)
- eTLV\_IND\_OMA\_DM\_CONFIG
  - qaCbkSwiOmaDmEventReportInd.h, [729](#)
- eTLV\_IND\_OMA\_DM\_FOTA
  - qaCbkSwiOmaDmEventReportInd.h, [729](#)
- eTLV\_IND\_OMA\_DM\_NOT
  - qaCbkSwiOmaDmEventReportInd.h, [729](#)
- eTLV\_REFRESH\_LENGTH
  - qaCbkCatEventReportInd.h, [728](#)
- eTLV\_RF\_BAND\_INFO
  - qaNasGetRFBandInfo.h, [1150](#)
- eTLV\_SETUP\_EVENT\_LIST\_LENGTH
  - qaCbkCatEventReportInd.h, [728](#)
- eWDS\_ERR\_PROFILE\_REG\_3GPP2\_ERR\_INVALID\_IDENTITY\_FOR\_PROFILE
  - qmerrno.h, [1158](#)
- eWDS\_ERR\_PROFILE\_REG\_3GPP\_ACCESS\_ERR
  - qmerrno.h, [1158](#)
- eWDS\_ERR\_PROFILE\_REG\_3GPP\_CONTEXT\_NOT\_DEFINED
  - qmerrno.h, [1158](#)
- eWDS\_ERR\_PROFILE\_REG\_3GPP\_ERR\_OUT\_OF\_PROFILES
  - qmerrno.h, [1158](#)
- eWDS\_ERR\_PROFILE\_REG\_3GPP\_INVALID\_PROFILE\_FAMILY
  - qmerrno.h, [1158](#)
- eWDS\_ERR\_PROFILE\_REG\_3GPP\_READ\_ONLY\_FLAG\_SET
  - qmerrno.h, [1158](#)
- eWDS\_ERR\_PROFILE\_REG\_3GPP\_VALID\_FLAG\_NOT\_SET
  - qmerrno.h, [1158](#)
- eWDS\_ERR\_PROFILE\_REG\_END
  - qmerrno.h, [1158](#)
- eWDS\_ERR\_PROFILE\_REG\_INVALID\_PROFILE\_FAMILY
  - qmerrno.h, [1158](#)
- eWDS\_ERR\_PROFILE\_REG\_RESULT\_ERR\_INVALID
  - qmerrno.h, [1158](#)
- eWDS\_ERR\_PROFILE\_REG\_RESULT\_ERR\_INVALID\_HNDL
  - qmerrno.h, [1157](#)
- eWDS\_ERR\_PROFILE\_REG\_RESULT\_ERR\_INVALID\_IDENTITY
  - qmerrno.h, [1158](#)
- eWDS\_ERR\_PROFILE\_REG\_RESULT\_ERR\_INVALID\_OP
  - qmerrno.h, [1157](#)
- eWDS\_ERR\_PROFILE\_REG\_RESULT\_ERR\_INVALID\_PROFILE\_NUM
  - qmerrno.h, [1158](#)
- eWDS\_ERR\_PROFILE\_REG\_RESULT\_ERR\_INVALID\_PROFILE\_TYPE
  - qmerrno.h, [1158](#)
- eWDS\_ERR\_PROFILE\_REG\_RESULT\_ERR\_INVALID\_SUBS\_ID
  - qmerrno.h, [1158](#)
- eWDS\_ERR\_PROFILE\_REG\_RESULT\_ERR\_LENGTH\_INVALID
  - qmerrno.h, [1158](#)
- eWDS\_ERR\_PROFILE\_REG\_RESULT\_ERR\_LIBRARY\_NOT\_INITIALIZED
  - qmerrno.h, [1158](#)
- eWDS\_ERR\_PROFILE\_REG\_RESULT\_FAIL
  - qmerrno.h, [1157](#)
- eWDS\_ERR\_PROFILE\_REG\_RESULT\_LIST\_END
  - qmerrno.h, [1158](#)
- ECIOThresListLen
  - ECIOThresh, [198](#)
- ECIOThresh, [196](#)
  - ECIOThresListLen, [198](#)
  - pECIOThresList, [198](#)
- ECTCallState
  - ECTNum, [199](#)



- ECTNum, [198](#)
  - ECTCallState, [199](#)
  - number, [199](#)
  - presentationInd, [199](#)
- eDevState
  - qaGobiApiCbk.h, [744](#)
- eDevice
  - sGetDeviceSeriesResult, [514](#)
- eGetDeviceSeries
  - qaGobiApiFms.h, [882](#)
- eGobiDeviceSeries
  - qaGobiApiFms.h, [879](#)
- eGobiImageCarrier
  - qaGobiApiFms.h, [879](#)
- eGobiImageGPS
  - qaGobiApiFms.h, [880](#)
- eGobiImageRegion
  - qaGobiApiFms.h, [881](#)
- eGobiImageTech
  - qaGobiApiFms.h, [881](#)
- eQCWWANError
  - qmerrno.h, [1153](#)
- eQMISARRFState
  - qaGobiApiSar.h, [984](#)
- eQaQMIService
  - qaGobiApiCbk.h, [782](#)
- ERIFileparams, [200](#)
  - pFile, [200](#)
  - pFileSize, [200](#)
  - qaGobiApiDms.h, [842](#)
- eSMSEventType
  - qaGobiApiCbk.h, [744](#)
- EVENT\_MASK\_CARD
  - qaGobiApiCbk.h, [742](#)
- eValid
  - TFTIDParams, [588](#)
- EarMute
  - GetAudioProfileResp, [223](#)
  - GetM2MAudioProfileResp, [240](#)
  - GetM2MAVMuteResp, [243](#)
  - SetAudioProfileReq, [490](#)
  - SetM2MAVMuteReq, [503](#)
- earfcn
  - infoInterFreq, [299](#)
  - LTEInfoIntrafreq, [318](#)
  - umtsLTENbrCell, [619](#)
- ecio
  - CDMASSInfo, [136](#)
  - ecioListElement, [196](#)
  - HDRSSInfo, [271](#)
  - rxInfo, [471](#)
  - TDSCDMASigInfoExt, [585](#)
  - UMTSInfo, [617](#)
- ecioDelta
  - SLQSSignalStrengthsIndReq, [535](#)
- ecioInfo
  - SLQSSignalStrengthsInformation, [537](#)
- ecioList
  - slqsSignalStrengthInfo, [533](#)
- ecioListElement, [196](#)
  - ecio, [196](#)
  - radiolf, [196](#)
- ecioListLen
  - slqsSignalStrengthInfo, [533](#)
- ecioThresholdList
  - SLQSSignalStrengthsIndReq, [535](#)
- ecioThresholdListLen
  - SLQSSignalStrengthsIndReq, [535](#)
- egprsSupp
  - GSMSysInfo, [263](#)
- egprsSuppValid
  - GSMSysInfo, [263](#)
- elevation
  - satelliteInfo, [478](#)
- emmConnState
  - LTEInfo, [315](#)
- emmState
  - LTEInfo, [315](#)
- emmSubState
  - LTEInfo, [315](#)
- Enable
  - SetM2MAudioLPBKReq, [501](#)
- EncryptProt
  - protocolSubtypeElement, [421](#)
- encryptedPIN1, [199](#)
  - pin1Len, [199](#)
  - pin1Val, [199](#)
- EndProactiveSession
  - CatEndProactiveSessionTlv, [124](#)
- EngineState
  - GPSSStateInfo, [255](#)
- errorClass
  - SMSAsyncRawSend\_s, [542](#)
- errorRate
  - errorRateListElement, [201](#)
- errorRateInfo
  - SLQSSignalStrengthsInformation, [537](#)
- errorRateList
  - slqsSignalStrengthInfo, [533](#)
- errorRateListElement, [200](#)
  - errorRate, [201](#)
  - radiolf, [201](#)
- errorRateListLen
  - slqsSignalStrengthInfo, [533](#)
- errorState
  - slotInfo, [518](#)
- esnSize
  - serialNumbersInfo, [480](#)
- event\_Index
  - QmiCbkCatEventStatusReportInd, [427](#)
- EventID
  - CatCommonEventTlv, [123](#)
- EventLength
  - CatCommonEventTlv, [123](#)
- eventMask
  - CATEventDataType, [124](#)

- UIMEventRegisterReqResp, 600
- eventRegister
  - LOCEventRegisterReqResp, 308
- evrcCapability
  - prefVoiceSO, 406
- executingImage
  - ImageIDEntries, 278
- exponent
  - pktErrRate, 399
- extBit
  - calledPartySubAdd, 110
- extDispInfo
  - extDispRecInfo, 203
- extDispInfoLen
  - extDispRecInfo, 203
- extDispRecInfo, 201
  - dispType, 203
  - extDispInfo, 203
  - extDispInfoLen, 203
- extPowerState
  - LOCExtPowerStateReqResp, 308
- FIRST\_INSTANCE
  - qaGobiApiCbk.h, 742
- FLOAT
  - SwiDataTypes.h, 1159
- FORBIDDEN\_INDEX
  - qaNasPerformNetworkScan.h, 1151
- FOTAUpdate
  - \_SLQSOMADMSettingsReqParams, 66
  - \_SLQSOMADMSettingsReqParams3, 67
- FOTAdownload
  - \_SLQSOMADMSettingsReqParams, 66
  - \_SLQSOMADMSettingsReqParams3, 67
- FSNumber
  - FactorySequenceNumber, 203
- FactorySequenceNumber, 203
  - FSNumber, 203
- failureCount
  - ImageIDElement, 277
- failureReason
  - ssdatasession\_params, 559
- failureReasonv4
  - ssdatasession\_params, 560
- failureReasonv6
  - ssdatasession\_params, 560
- feature
  - depersonalizationInformation, 187
- fileAttributes, 203
  - fileID, 208
  - fileSize, 208
  - fileType, 208
  - rawLen, 208
  - rawValue, 208
  - recordCount, 208
  - recordSize, 208
  - secActivate, 208
  - secActivateMask, 208
  - secDeactivate, 208
  - secDeactivateMask, 208
  - secIncrease, 208
  - secIncreaseMask, 208
  - secRead, 208
  - secReadMask, 208
  - secWrite, 208
  - secWriteMask, 208
- fileID
  - fileAttributes, 208
  - fileInfo, 210
- fileIndex
  - UIMGetFileAttributesReq, 601
  - UIMReadTransparentReq, 605
- fileInfo, 208
  - fileID, 210
  - path, 210
  - pathLen, 210
- fileSize
  - fileAttributes, 208
- fileType
  - fileAttributes, 208
- filterId
  - TFTIDParams, 588
- Firmware Management Service (FMS), 30
- FirmwareID
  - fwinfo\_s, 212
- FirmwareUpdatStat, 210
  - plmgType, 211
  - pRefData, 211
  - pRefString, 211
  - pRefStringLength, 211
  - ResCode, 211
- flowLabel
  - TFTIDParams, 588
- Forbidden
  - SlqsNas3GppNetworkInfo, 523
- ForceXTRADownload
  - qaGobiApiPds.h, 959
- format
  - SMSTransferRouteMTMessage, 554
- ForwardMac
  - protocolSubtypeElement, 421
- fqdnAddr
  - PCSCFFQDNAddress, 387
- fqdnLen
  - PCSCFFQDNAddress, 387
- freeSlots
  - smsMaxStorageSizeResp, 547
- freq
  - PhyCaAggPcellInfo, 394
  - PhyCaAggScellIndType, 396
  - PhyCaAggScellInfo, 397
- freqsLen
  - LTEInfoInterfreq, 316
  - LTEInfoNeighboringGSM, 319
  - LTEInfoNeighboringWCDMA, 320
- fromServiceId
  - BroadcastConfig, 104

- fumoResultCode
  - omaDmFotaTlvExt, [382](#)
- fwloadsize
  - omaDmFotaTlv, [380](#)
- fwinfo\_s, [212](#)
  - Carrier, [212](#)
  - FirmwareID, [212](#)
  - GPSCapability, [212](#)
  - Region, [212](#)
  - Technology, [212](#)
- fwloadComplete
  - omaDmFotaTlv, [380](#)
- fwvers
  - CurrentImgList, [165](#)
- g
  - qmifwinfo\_s, [441](#)
- gDlBitRate
  - QosClassID, [449](#)
- GERANInfo, [212](#)
  - arfcn, [214](#)
  - bsic, [214](#)
  - cellID, [214](#)
  - insNmrCellInfo, [214](#)
  - lac, [214](#)
  - nmrInst, [214](#)
  - plmn, [214](#)
  - rxLev, [214](#)
  - timingAdvance, [214](#)
- GPRSQoS, [250](#)
  - delayClass, [251](#)
  - meanThroughputClass, [251](#)
  - peakThroughputClass, [251](#)
  - precedenceClass, [251](#)
  - reliabilityClass, [251](#)
- GPRSRequestedQoS, [251](#)
  - delayClass, [252](#)
  - meanThroughputClass, [252](#)
  - peakThroughputClass, [252](#)
  - precedenceClass, [252](#)
  - reliabilityClass, [252](#)
- GPSCapability
  - fwinfo\_s, [212](#)
- GPSStateInfo, [252](#)
  - Altitude, [255](#)
  - EngineState, [255](#)
  - glo\_almanac\_sv\_msk, [256](#)
  - glo\_ephemeris\_sv\_msk, [256](#)
  - glo\_health\_sv\_msk, [256](#)
  - glo\_visible\_sv\_msk, [256](#)
  - gps\_almanac\_sv\_msk, [256](#)
  - gps\_ephemeris\_sv\_msk, [256](#)
  - gps\_health\_sv\_msk, [256](#)
  - gps\_visible\_sv\_msk, [256](#)
  - HorizontalUncertainty, [256](#)
  - lono\_valid, [256](#)
  - Latitude, [256](#)
  - Longitude, [256](#)
  - sbas\_almanac\_sv\_msk, [256](#)
  - sbas\_ephemeris\_sv\_msk, [256](#)
  - sbas\_health\_sv\_msk, [256](#)
  - sbas\_visible\_sv\_msk, [256](#)
  - Time\_uncert\_ms, [256](#)
  - TimeStmp\_gps\_week, [256](#)
  - TimeStmp\_tow\_ms, [256](#)
  - ValidMask, [256](#)
  - VerticalUncertainty, [256](#)
  - xtra\_start\_gps\_minutes, [256](#)
  - xtra\_start\_gps\_week, [256](#)
  - xtra\_valid\_duration\_hours, [256](#)
- GSMRSSIThresh, [258](#)
  - GSMRSSIThreshListLen, [259](#)
  - pGSMRSSIThreshList, [259](#)
- GSMRSSIThreshListLen
  - GSMRSSIThresh, [259](#)
- GSMSrvStatusInfo, [259](#)
  - isPrefDataPath, [260](#)
  - srvStatus, [260](#)
  - trueSrvStatus, [260](#)
- GSMSysInfo, [260](#)
  - cellId, [263](#)
  - cellIdValid, [263](#)
  - dtmSupp, [263](#)
  - dtmSuppValid, [263](#)
  - egprsSupp, [263](#)
  - egprsSuppValid, [263](#)
  - lac, [263](#)
  - lacValid, [263](#)
  - MCC, [263](#)
  - MNC, [263](#)
  - networkIdValid, [263](#)
  - regRejectInfoValid, [263](#)
  - rejCause, [263](#)
  - rejectSrvDomain, [263](#)
  - sysInfoGSM, [263](#)
- gUlBitRate
  - QosClassID, [449](#)
- gcDumpStrLen
  - CrashInfo, [158](#)
- Generator
  - GetAudioProfileReq, [222](#)
  - GetAudioVoITLBConfigReq, [224](#)
  - GetM2MAudioProfileResp, [240](#)
  - GetM2MAudioVolumeReq, [241](#)
  - SetAudioProfileReq, [490](#)
  - SetAudioVoITLBConfigReq, [492](#)
  - SetM2MAudioVolumeReq, [503](#)
- geoSysIdx
  - AddCDMASysInfo, [85](#)
  - AddSysInfo, [86](#)
- geranArfcn
  - geranInstInfo, [216](#)
- geranBsicBcc
  - geranInstInfo, [216](#)
- geranBsicNcc
  - geranInstInfo, [216](#)
- geranInst

- UMTSInfo, [617](#)
- GeranInstInfo
  - UMTSInfo, [617](#)
- geranInstInfo, [214](#)
  - geranArfcn, [216](#)
  - geranBsicBcc, [216](#)
  - geranBsicNcc, [216](#)
  - geranRssi, [216](#)
- geranRssi
  - geranInstInfo, [216](#)
- GetACCOLC
  - qaGobiApiNas.h, [918](#)
- GetANAAAAAuthenticationStatus
  - qaGobiApiNas.h, [919](#)
- GetActivationState
  - qaGobiApiDms.h, [844](#)
- getAllCallInfo
  - arrCallInfo, [96](#)
- getAllCallInformation, [216](#)
  - ALS, [217](#)
  - Callinfo, [217](#)
  - isEmpty, [217](#)
- GetAllCallRmtPtyName
  - arrRemotePartyName, [99](#)
- getAllCallRmtPtyName, [217](#)
  - callID, [217](#)
  - RemotePartyName, [217](#)
- getAllCallRmtPtyNum, [217](#)
  - callID, [219](#)
  - RemotePartyNum, [219](#)
- GetAudioPathConfigReq, [219](#)
  - Item, [220](#)
  - Profile, [220](#)
- GetAudioPathConfigResp, [220](#)
  - pCodecSTGain, [221](#)
  - pDTMFTXGain, [221](#)
  - pECMode, [222](#)
  - pMICGainSelect, [222](#)
  - pNSEnable, [222](#)
  - pRXAGCList, [222](#)
  - pRXAVCAGCSwitch, [222](#)
  - pRXAVCList, [222](#)
  - pRXPCMIIRFtr, [222](#)
  - pTXAGCList, [222](#)
  - pTXAVCSwitch, [222](#)
  - pTXGain, [222](#)
  - pTXPCMIIRFtr, [222](#)
- GetAudioProfileReq, [222](#)
  - Generator, [222](#)
- GetAudioProfileResp, [222](#)
  - EarMute, [223](#)
  - MicMute, [223](#)
  - Profile, [223](#)
  - Volume, [223](#)
- GetAudioVolTLBConfigReq, [224](#)
  - Generator, [224](#)
  - Item, [224](#)
  - Profile, [224](#)
  - Volume, [224](#)
- GetAudioVolTLBConfigResp, [224](#)
  - ResCode, [225](#)
- GetAutoconnect
  - qaGobiApiWds.h, [1108](#)
- GetByteTotals
  - qaGobiApiWds.h, [1109](#)
- GetCDMANetworkParameters
  - qaGobiApiNas.h, [919](#)
- getCallFWExtInfo, [225](#)
  - CallFWExtInfo, [225](#)
  - numInstances, [225](#)
- getCallFWInfo, [225](#)
  - CallFWInfo, [226](#)
  - numInstances, [226](#)
- GetConnectionRate
  - qaGobiApiWds.h, [1109](#)
- getCustomFeatureV2, [226](#)
  - pCustSettingInfo, [226](#)
  - pCustSettingList, [227](#)
  - pGetCustomInput, [227](#)
- getCustomInput, [227](#)
  - cust\_id, [227](#)
  - list\_type, [227](#)
- getDUNCallInfoReq, [227](#)
  - Mask, [228](#)
  - pReportChannelRate, [228](#)
  - pReportConnStatus, [229](#)
  - pReportDataBearerTech, [229](#)
  - pReportDormStatus, [229](#)
  - pTransferStatInd, [229](#)
- getDUNCallInfoResp, [229](#)
  - pCallEndReason, [231](#)
  - pChannelRate, [231](#)
  - pConnectionStatus, [231](#)
  - pDataBearerTech, [231](#)
  - pDormancyStatus, [231](#)
  - pLastCallDataBearerTech, [231](#)
  - pLastCallRXOKBytesCnt, [231](#)
  - pLastCallTXOKBytesCnt, [231](#)
  - pMdmCallDurationActive, [231](#)
  - pRXOKBytesCount, [231](#)
  - pTXOKBytesCount, [231](#)
- GetDataBearerTechnology
  - qaGobiApiWds.h, [1110](#)
- GetDefaultProfile
  - qaGobiApiWds.h, [1111](#)
- GetDefaultProfileLTE
  - qaGobiApiWds.h, [1113](#)
- GetDeviceCapabilities
  - qaGobiApiDms.h, [845](#)
- GetDormancyState
  - qaGobiApiWds.h, [1115](#)
- GetErrRateResp, [231](#)
  - pCDMAFrameErrRate, [232](#)
  - pGSMBER, [232](#)
  - pHDRPackErrRate, [232](#)
  - pWCDMABER, [232](#)

- GetFirmwareRevision
  - qaGobiApiDms.h, [846](#)
- GetFirmwareRevisions
  - qaGobiApiDms.h, [846](#)
- GetHRPDStatsResp, [232](#)
  - pDRCPParams, [233](#)
  - pPilotSetData, [233](#)
  - pUATI, [233](#)
- GetHardwareRevision
  - qaGobiApiDms.h, [848](#)
- GetHomeNetwork
  - qaGobiApiNas.h, [921](#)
- GetHomeNetwork3GPP2
  - qaGobiApiNas.h, [924](#)
- GetIMSI
  - qaGobiApiDms.h, [849](#)
- GetIMSSMSConfigParams, [233](#)
  - pPhoneCtxtURI, [234](#)
  - pPhoneCtxtURILen, [234](#)
  - pSMSFormat, [234](#)
  - pSMSOverIPNwInd, [234](#)
  - pSettingResp, [234](#)
- GetIMSUserConfigParams, [234](#)
  - pIMSDomain, [234](#)
  - pIMSDomainLen, [234](#)
  - pSettingResp, [234](#)
- GetIMSVolIPConfigResp, [235](#)
  - pAmrMode, [237](#)
  - pAmrOctetAligned, [237](#)
  - pAmrWBMode, [237](#)
  - pAmrWBOctetAligned, [237](#)
  - pAmrWbEnable, [237](#)
  - pMinSessionExpiryTimer, [237](#)
  - pRTPRTCPInactTimer, [237](#)
  - pRingBackTimer, [237](#)
  - pRingingTimer, [237](#)
  - pScrAmrEnable, [237](#)
  - pScrAmrWbEnable, [237](#)
  - pSessionExpiryTimer, [237](#)
  - pSettingResp, [238](#)
- GetIPAddressLTE
  - qaGobiApiWds.h, [1116](#)
- GetImageStore
  - qaGobiApiFms.h, [882](#)
- GetImagesPreference
  - qaGobiApiFms.h, [882](#)
- getIndicationRegResp
  - qaGobiApiSms.h, [988](#)
- GetInstIDResp, [238](#)
  - pIPFamily, [238](#)
  - pInstanceId, [238](#)
- GetLastMobileIPError
  - qaGobiApiWds.h, [1116](#)
- GetM2MAVMMuteReq, [242](#)
  - Profile, [242](#)
- GetM2MAVMMuteResp, [242](#)
  - CwtMute, [243](#)
  - EarMute, [243](#)
  - MicMute, [243](#)
- GetM2MAudioProfileReq, [238](#)
  - pGenerator, [238](#)
- GetM2MAudioProfileResp, [238](#)
  - CwtMute, [240](#)
  - EarMute, [240](#)
  - Generator, [240](#)
  - MicMute, [240](#)
  - Profile, [240](#)
  - Volume, [240](#)
- GetM2MAudioVolumeReq, [241](#)
  - Generator, [241](#)
  - Profile, [241](#)
- GetM2MAudioVolumeResp, [241](#)
  - Level, [241](#)
- GetM2MSpkrGainReq, [243](#)
  - Profile, [243](#)
- GetM2MSpkrGainResp, [243](#)
  - Value, [244](#)
- GetManufacturer
  - qaGobiApiDms.h, [849](#)
- GetMobileIP
  - qaGobiApiWds.h, [1117](#)
- GetMobileIPProfile
  - qaGobiApiWds.h, [1117](#)
- GetModelID
  - qaGobiApiDms.h, [850](#)
- getMsgWaitingInfo, [244](#)
  - msgWaitInfo, [244](#)
  - numInstances, [244](#)
- GetNetworkPreference
  - qaGobiApiNas.h, [926](#)
- GetNetworkTime
  - qaGobiApiDms.h, [850](#)
- GetOfflineReason
  - qaGobiApiDms.h, [851](#)
- GetPDSDDefaults
  - qaGobiApiPds.h, [960](#)
- GetPDSSState
  - qaGobiApiPds.h, [960](#)
- GetPRLVersion
  - qaGobiApiDms.h, [852](#)
- GetPacketStatistics
  - qaGobiApiWds.h, [1119](#)
- GetPacketStatus
  - qaGobiApiWds.h, [1120](#)
- GetPortAutomaticTracking
  - qaGobiApiPds.h, [961](#)
- GetPower
  - qaGobiApiDms.h, [851](#)
- GetProfileSettingIn
  - qaGobiApiWds.h, [1103](#)
- GetProfileSettingOut
  - qaGobiApiWds.h, [1103](#)
- GetRFInfo
  - qaGobiApiNas.h, [927](#)
- GetRegMgrConfigParams, [244](#)
  - pIMSTestMode, [246](#)

- pPCSCFPort, [246](#)
  - pPriCSCFPortName, [246](#)
  - pPriCSCFPortNameLen, [246](#)
  - pSettingResp, [246](#)
- GetSIPConfigResp, [247](#)
  - pSIPLocalPort, [248](#)
  - pSettingResp, [247](#)
  - pSigCompEnabled, [247](#)
  - pSubscribeTimer, [248](#)
  - pTimerSIPReg, [248](#)
  - pTimerT1, [248](#)
  - pTimerT2, [248](#)
  - pTimerTf, [248](#)
- GetSMSCAddress
  - qaGobiApiSms.h, [993](#)
- GetSMSWake
  - qaGobiApiRms.h, [982](#)
- GetSerialNumbers
  - qaGobiApiDms.h, [852](#)
- GetServiceAutomaticTracking
  - qaGobiApiPds.h, [961](#)
- GetServingNetwork
  - qaGobiApiNas.h, [928](#)
- GetServingNetworkCapabilities
  - qaGobiApiNas.h, [930](#)
- GetSessionDuration
  - qaGobiApiWds.h, [1120](#)
- GetSessionIDResp, [246](#)
  - pSessionIDv4, [246](#)
  - pSessionIDv6, [246](#)
- GetSessionState
  - qaGobiApiWds.h, [1121](#)
- GetSignalStrengths
  - qaGobiApiNas.h, [930](#)
- GetStoredImages
  - qaGobiApiFms.h, [883](#)
- getTransLayerInfoResp
  - qaGobiApiSms.h, [989](#)
- getTransNWRegInfoResp
  - qaGobiApiSms.h, [990](#)
- GetVoiceNumber
  - qaGobiApiDms.h, [853](#)
- GetXTRAAutomaticDownload
  - qaGobiApiPds.h, [963](#)
- GetXTRANetwork
  - qaGobiApiPds.h, [963](#)
- GetXTRAVality
  - qaGobiApiPds.h, [964](#)
- glo\_almanac\_sv\_msk
  - GPSSStateInfo, [256](#)
- glo\_ephemeris\_sv\_msk
  - GPSSStateInfo, [256](#)
- glo\_health\_sv\_msk
  - GPSSStateInfo, [256](#)
- glo\_visible\_sv\_msk
  - GPSSStateInfo, [256](#)
- globalCellId
  - LTEInfoIntrafreq, [318](#)
- GnssData, [248](#)
  - mask, [250](#)
- gnssSvId
  - satelliteInfo, [478](#)
- gnssSvInfoNotification, [250](#)
  - bAltitudeAssumed, [250](#)
  - pSatelliteInfo, [250](#)
- gnssSvUsedList
  - svUsedforFix\_s, [564](#)
- gnssSvUsedList\_len
  - svUsedforFix\_s, [564](#)
- Gpp2TimeZone
  - qaQmiServingSystemParam, [427](#)
- GppNetworkDSTAdjustment
  - qaQmiServingSystemParam, [427](#)
- GppTimeZone
  - qaQmiServingSystemParam, [427](#)
- gps\_almanac\_sv\_msk
  - GPSSStateInfo, [256](#)
- gps\_ephemeris\_sv\_msk
  - GPSSStateInfo, [256](#)
- gps\_health\_sv\_msk
  - GPSSStateInfo, [256](#)
- gps\_visible\_sv\_msk
  - GPSSStateInfo, [256](#)
- GpsEnable
  - custFeaturesInfo, [172](#)
- gpsTime
  - qaGobiApiCbk.h, [745](#)
- gpsTime\_s, [256](#)
  - gpsTimeOfWeekMs, [257](#)
  - gpsWeek, [257](#)
- gpsTimeOfWeekMs
  - gpsTime\_s, [257](#)
- gpsWeek
  - gpsTime\_s, [257](#)
- grntDownlinkBitrate
  - UMTSMinQoS, [622](#)
  - UMTSQoS, [625](#)
- grntUplinkBitrate
  - UMTSMinQoS, [622](#)
  - UMTSQoS, [625](#)
- gsmAmrStat
  - curAMRConfig, [163](#)
- GsmCellInfo
  - lteGsmCellInfo, [312](#)
- gsmCellInfo, [257](#)
  - arfcn, [258](#)
  - band1900, [258](#)
  - bsicId, [258](#)
  - cellIdValid, [258](#)
  - rsi, [258](#)
  - srxlev, [258](#)
- gsmUmtsDI
  - NasSwilndReg, [359](#)
- gsmUmtsUI
  - NasSwilndReg, [359](#)
- guaranteedRate

- dataRate, [181](#)
- gwAddressV6
  - IPv6GWAddressInfo, [303](#)
- gwV6PrefixLen
  - IPv6GWAddressInfo, [303](#)
- gyroAcceptReady
  - qaGobiApiCbK.h, [745](#)
- gyroAcceptReady\_s, [263](#)
  - batchPerSec, [264](#)
  - injectEnable, [264](#)
  - samplesPerBatch, [264](#)
- gyroTempAcceptReady
  - qaGobiApiCbK.h, [746](#)
- gyroTempAcceptReady\_s, [264](#)
  - batchPerSec, [265](#)
  - injectEnable, [265](#)
  - samplesPerBatch, [265](#)
- HDOP
  - precisionDilution\_s, [403](#)
- HDRECIOTresh, [265](#)
  - HDRECIOTreshListLen, [266](#)
  - pHDRECIOTreshList, [266](#)
- HDRECIOTreshListLen
  - HDRECIOTresh, [266](#)
- HDRIOTresh, [266](#)
  - HDRIOTreshListLen, [266](#)
  - pHDRIOTreshList, [266](#)
- HDRIOTreshListLen
  - HDRIOTresh, [266](#)
- HDRPersonalityInd, [266](#)
  - pCurrentPersonality, [266](#)
  - pPersonalityListLength, [266](#)
  - pProtocolSubtypeElement, [266](#)
- HDRPersonalityResp, [267](#)
  - pCurrentPersonality, [267](#)
  - pPersonalityListLength, [267](#)
  - pProtocolSubtypeElement, [267](#)
- HDRProtSubtypResp, [267](#)
  - pAppSubType, [268](#)
  - pCurrentPrsnlty, [268](#)
  - pPersonalityListLength, [268](#)
  - pProtoSubTypElmnt, [268](#)
- HDRRSSITresh, [268](#)
  - HDRRSSITreshListLen, [268](#)
  - pHDRRSSITreshList, [268](#)
- HDRRSSITreshListLen
  - HDRRSSITresh, [268](#)
- HDRSINRThresListLen
  - HDRSINRThresh, [269](#)
- HDRSINRThresh, [269](#)
  - HDRSINRThresListLen, [269](#)
  - pHDRSINRThresList, [269](#)
- HDRSINRThreshListLen
  - HDRSINRThreshold, [270](#)
- HDRSINRThreshold, [269](#)
  - HDRSINRThresListLen, [270](#)
  - pHDRSINRThresList, [270](#)
- HDRSSInfo, [270](#)
- ecio, [271](#)
- io, [271](#)
- rssI, [271](#)
- sinr, [271](#)
- HDRSysInfo, [271](#)
  - hdrActiveProt, [274](#)
  - hdrActiveProtValid, [274](#)
  - hdrPersonality, [274](#)
  - hdrPersonalityValid, [274](#)
  - is856SysId, [274](#)
  - is856SysIdValid, [274](#)
  - isSysPrIMatch, [274](#)
  - isSysPrIMatchValid, [274](#)
  - sysInfoHDR, [274](#)
- HWVersion
  - DeviceConfigDetail, [190](#)
- hdrActiveProt
  - HDRSysInfo, [274](#)
- hdrActiveProtValid
  - HDRSysInfo, [274](#)
- hdrHybrid
  - detailSvcInfo, [189](#)
- hdrPersonality
  - HDRSysInfo, [274](#)
  - qaQmiServingSystemParam, [427](#)
  - ServingSystemInfo, [482](#)
- hdrPersonalityValid
  - HDRSysInfo, [274](#)
- hdrSrvStatus
  - detailSvcInfo, [189](#)
- healthStatus
  - satelliteInfo, [478](#)
- homeOrigVoiceSO
  - prefVoiceSO, [406](#)
- homePageVoiceSO
  - prefVoiceSO, [406](#)
- homeSIDNID, [275](#)
  - numInstances, [275](#)
  - SidNid, [275](#)
- HorizontalUncertainty
  - GPSSStateInfo, [256](#)
- hotSwap
  - hotSwapStatus, [276](#)
- hotSwapLength
  - hotSwapStatus, [276](#)
- hotSwapStatus, [275](#)
  - hotSwap, [276](#)
  - hotSwapLength, [276](#)
- hour
  - UniversalTime, [628](#)
- hsCallStatus
  - WCDMASysInfo, [710](#)
- hsCallStatusValid
  - WCDMASysInfo, [710](#)
- hsInd
  - WCDMASysInfo, [710](#)
- hsIndValid
  - WCDMASysInfo, [710](#)



- iFaceTab
  - PCMPparams, [385](#)
- iFaceTabLen
  - PCMPparams, [385](#)
- iGetByteTotals
  - qaGobiApiWds.h, [1121](#)
- iGetConnectionRate
  - qaGobiApiWds.h, [1122](#)
- iGetPacketStatistics
  - qaGobiApiWds.h, [1122](#)
- iLTEbandValue
  - PhyCaAggPcellInfo, [394](#)
  - PhyCaAggScellInfo, [397](#)
- IMGDETAILS\_LEN
  - qaGobiApiDms.h, [837](#)
- IMS Service (IMS), [39](#)
- IMSALndRegisterInfo, [279](#)
  - pPdpStatusConfig, [280](#)
  - pRatHandoverStatusConfig, [280](#)
  - pRegStatusConfig, [280](#)
  - pServiceStatusConfig, [280](#)
- IMSARegistrationStatus, [282](#)
  - plmsRegErrCode, [283](#)
  - plmsRegStatus, [283](#)
  - pNewImsRegStatus, [283](#)
- IMSServiceStatus, [284](#)
  - pSmsServiceRat, [286](#)
  - pSmsServiceStatus, [286](#)
  - pUtServiceRat, [286](#)
  - pUtServiceStatus, [286](#)
  - pVoipServiceRat, [286](#)
  - pVoipServiceStatus, [286](#)
  - pVsServiceRat, [286](#)
  - pVsServiceStatus, [286](#)
  - pVtServiceRat, [286](#)
  - pVtServiceStatus, [287](#)
- IMSSupportedFieldsResp, [287](#)
  - pIIndFieldsList, [287](#)
  - pReqFieldsList, [287](#)
  - pRespFieldsList, [287](#)
- IMSSupportedMsgInfo, [287](#)
  - pSupportedMsgList, [288](#)
- IMSI\_M\_S1\_LENGTH
  - qaGobiApiNas.h, [909](#)
- IMSI\_M\_S2\_LENGTH
  - qaGobiApiNas.h, [909](#)
- INDEX\_ZERO
  - qaNasPerformNetworkScan.h, [1151](#)
- INT32
  - SwiDataTypes.h, [1159](#)
- INT8
  - SwiDataTypes.h, [1159](#)
- INVALID\_INSTACNE
  - qaGobiApiCbk.h, [742](#)
- IOThresListLen
  - IOThresh, [300](#)
- IOThresh, [299](#)
  - IOThresListLen, [300](#)
- plOTThresList, [300](#)
- IPAddress
  - DataStatusDetail, [184](#)
- IPAddressV6
  - IPV6AddressInfo, [302](#)
- IPSECSPi
  - TFTIDParams, [588](#)
- IPV4
  - qaGobiApiCbk.h, [742](#)
- IPV4V6
  - qaGobiApiCbk.h, [742](#)
- IPV6
  - qaGobiApiCbk.h, [742](#)
- IPV6AddressInfo, [302](#)
  - IPAddressV6, [302](#)
  - IPV6PrefixLen, [302](#)
- IPV6GWAddressInfo, [302](#)
  - gwAddressV6, [303](#)
  - gwV6PrefixLen, [303](#)
- IPV6PrefixLen
  - IPV6AddressInfo, [302](#)
- IPv4Addr, [300](#)
  - addr, [300](#)
  - subnetMask, [300](#)
- IPv6Addr, [300](#)
  - addr, [302](#)
  - prefixLen, [302](#)
- IPv6TrafCls, [303](#)
  - mask, [303](#)
  - val, [303](#)
- iSLQSMISetIPFamilyPreference
  - qaGobiApiWds.h, [1122](#)
- iSLQSSetDUNCallInfoCallback
  - qaGobiApiCbk.h, [782](#)
- iSLQSSetSignalStrengthsCallback
  - qaGobiApiCbk.h, [782](#)
- iSLQSSetWdsFirstInstEventCallback
  - qaGobiApiCbk.h, [782](#)
- iSLQSSetWdsSecondInstEventCallback
  - qaGobiApiCbk.h, [782](#)
- iSLQSSetWdsThirdInstEventCallback
  - qaGobiApiCbk.h, [782](#)
- iSLQSSetWdsXferStatsFirstInstCallback
  - qaGobiApiCbk.h, [782](#)
- iSLQSSetWdsXferStatsSecondInstCallback
  - qaGobiApiCbk.h, [783](#)
- iSetGATEventCallback
  - qaGobiApiCbk.h, [782](#)
- iSetSignalStrengthCallback
  - qaGobiApiCbk.h, [782](#)
- id
  - BdsSV, [103](#)
  - CSGID, [161](#)
  - QosFlowInfoState, [452](#)
  - SV, [563](#)
  - swiQosModifyReq, [579](#)
- id\_length
  - custSettingInfo, [175](#)



- IdleState
  - protocolSubtypeElement, [421](#)
- ImageElement, [276](#)
  - buildId, [276](#)
  - buildIdLength, [277](#)
  - imageId, [277](#)
  - imageType, [277](#)
- imageID
  - ImageIdElement, [277](#)
- imageIDElement
  - ImageIDEntries, [278](#)
- ImageIDEntries, [278](#)
  - executingImage, [278](#)
  - imageIDElement, [278](#)
  - imageIDSize, [278](#)
  - imageType, [278](#)
  - maxImages, [278](#)
- imageIDEntries
  - ImageList, [279](#)
- imageIDSize
  - ImageIDEntries, [278](#)
- imageId
  - ImageElement, [277](#)
- ImageIdElement, [277](#)
  - buildID, [277](#)
  - buildIDLength, [277](#)
  - failureCount, [277](#)
  - imageID, [277](#)
  - storageIndex, [277](#)
- ImageList, [278](#)
  - imageIDEntries, [279](#)
  - listSize, [279](#)
- imageType
  - CurrImageInfo, [167](#)
  - ImageElement, [277](#)
  - ImageIDEntries, [278](#)
- imeiSize
  - serialNumbersInfo, [480](#)
- imeiSvnSize
  - serialNumbersInfo, [480](#)
- imsCfgrIndRegisterInfo, [289](#)
  - pRegMgrConfigEvents, [290](#)
  - pSIPConfigEvents, [290](#)
  - pSMSConfigEvents, [290](#)
  - pUserConfigEvents, [290](#)
  - pVoIPConfigEvents, [290](#)
- imsRegMgrConfigInfo, [290](#)
  - pCSCFPortName, [292](#)
  - pIMSTestMode, [292](#)
  - pPriCSCFPort, [292](#)
- imsRegState
  - CommInfo, [153](#)
- imsSIPConfigInfo, [292](#)
  - pSIPLocalPort, [293](#)
  - pSigCompEnabled, [293](#)
  - pSubscribeTimer, [293](#)
  - pTimerSIPReg, [293](#)
  - pTimerT1, [293](#)
  - pTimerT2, [293](#)
  - pTimerTf, [293](#)
- imsSMSConfigInfo, [293](#)
  - pPhoneCtxtURI, [294](#)
  - pSMSFormat, [294](#)
  - pSMSOverIPNwInd, [294](#)
- imsUserConfigInfo, [294](#)
  - pIMSDomain, [294](#)
- imsVoIPConfigInfo, [294](#)
  - pAmrMode, [297](#)
  - pAmrOctetAligned, [297](#)
  - pAmrWBMode, [297](#)
  - pAmrWBOctetAligned, [297](#)
  - pAmrWbEnable, [297](#)
  - pMinSessionExpiryTimer, [297](#)
  - pRTPRTCPInactTimer, [297](#)
  - pRingBackTimer, [297](#)
  - pRingingTimer, [297](#)
  - pScrAmrEnable, [297](#)
  - pScrAmrWbEnable, [297](#)
  - pSessionExpiryTimer, [297](#)
- imsaPdpStatusInfo, [280](#)
  - connetionState, [281](#)
  - pFailErrorCode, [281](#)
- imsaRatStatusInfo, [281](#)
  - pErrorCodeStr, [282](#)
  - pRATStatus, [282](#)
  - pSrcRAT, [282](#)
  - pTgtRAT, [282](#)
- imsaRegStatusInfo, [283](#)
  - plmsRegStatus, [284](#)
  - pRegStatusErrorCode, [284](#)
  - pbIMSRegistered, [284](#)
- imsaSvcStatusInfo, [288](#)
  - pSMSSvcRAT, [288](#)
  - pSMSSvcStatus, [288](#)
  - pUTSvcRAT, [288](#)
  - pUTSvcStatus, [288](#)
  - pVOIPSvcRAT, [288](#)
  - pVOIPSvcStatus, [289](#)
  - pVTSvcRAT, [289](#)
  - pVTSvcStatus, [289](#)
- imsi\_11\_12
  - CDMASysInfoExt, [141](#)
- imsiM1112
  - minBasedIMSI, [333](#)
- imsiMS1
  - minBasedIMSI, [333](#)
- imsiMS2
  - minBasedIMSI, [333](#)
- imsiT1112
  - trueIMSI, [592](#)
- imsiTS1
  - trueIMSI, [592](#)
- imsiTS2
  - trueIMSI, [592](#)
- imsiTaddrNum
  - trueIMSI, [592](#)

- InUse
  - SlqsNas3GppNetworkInfo, [523](#)
- includes\_pcs\_digit
  - SlqsNasPcsDigit, [526](#)
- IndFieldsList, [297](#)
  - indicationFields, [298](#)
  - indicationFieldsLen, [298](#)
- index
  - swiQosFilter, [573](#)
  - swiQosFlow, [577](#)
  - swiQosReq, [580](#)
- index1xPri
  - cardStatus, [122](#)
- index1xSec
  - cardStatus, [122](#)
- indexGwPri
  - cardStatus, [122](#)
- indexGwSec
  - cardStatus, [122](#)
- indicationFields
  - IndFieldsList, [298](#)
- indicationFieldsLen
  - IndFieldsList, [298](#)
- infoInterFreq, [298](#)
  - cell\_resel\_priority, [299](#)
  - cellInterFreqParams, [299](#)
  - cells\_len, [299](#)
  - earfcn, [299](#)
  - threshXHigh, [299](#)
  - threshXLow, [299](#)
- InfoInterfreq
  - LTEInfoInterfreq, [316](#)
- InitiateDomainAttach
  - qaGobiApiNas.h, [932](#)
- InitiateNetworkRegistration
  - qaGobiApiNas.h, [932](#)
- injectEnable
  - accelAcceptReady\_s, [82](#)
  - accelTempAcceptReady\_s, [83](#)
  - gyroAcceptReady\_s, [264](#)
  - gyroTempAcceptReady\_s, [265](#)
- injectSensorDataStatus
  - QmiCbkLocInjectSensorDataInd, [430](#)
- injectTimeSyncStatus
  - QmiCbkLocInjectTimeInd, [431](#)
- insNmrCellInfo
  - GERANInfo, [214](#)
- instanceld
  - ssdatasession\_params, [560](#)
- interval
  - TransferStatsDataType, [590](#)
- Io
  - slqsSignalStrengthInfo, [533](#)
- io
  - HDRSSInfo, [271](#)
  - SLQSSignalStrengthsInformation, [537](#)
- ioDelta
  - SLQSSignalStrengthsIndReq, [535](#)
- Iono\_valid
  - GPSSStateInfo, [256](#)
- ip
  - WdsIpAddressInfoReq, [715](#)
- ipFamily
  - \_packetSrvStatus, [52](#)
- ipVersion
  - TFTIDParams, [588](#)
- ipfamily
  - ssdatasession\_params, [560](#)
- is856SysId
  - HDRSysInfo, [274](#)
- is856SysIdValid
  - HDRSysInfo, [274](#)
- isEmpty
  - getAllCallInformation, [217](#)
- isInTraffic
  - txInfo, [594](#)
- isModByCC
  - SUPSInfo, [562](#)
- isNewFlow
  - QosFlowInfoState, [452](#)
- isPrefDataPath
  - GSMSrvStatusInfo, [260](#)
  - SrvStatusInfo, [557](#)
- isRadioTuned
  - rxInfo, [471](#)
- isSysForbidden
  - detailSvcInfo, [189](#)
  - sysInfoCommon, [583](#)
- isSysForbiddenValid
  - sysInfoCommon, [583](#)
- isSysPriMatch
  - CDMASysInfo, [140](#)
  - HDRSysInfo, [274](#)
- isSysPriMatchValid
  - CDMASysInfo, [140](#)
  - HDRSysInfo, [274](#)
- Item
  - GetAudioPathConfigReq, [220](#)
  - GetAudioVolTLBConfigReq, [224](#)
  - SetAudioVolTLBConfigReq, [492](#)
- KeyExchange
  - protocolSubtypeElement, [421](#)
- LEN
  - qaGobiApiDcs.h, [823](#)
- LOCEventRegisterReqResp, [306](#)
  - eventRegister, [308](#)
- LOCExtPowerStateReqResp, [308](#)
  - extPowerState, [308](#)
- LOCStartReq, [308](#)
  - pApplicationInfo, [310](#)
  - pConfigAltitudeAssumed, [310](#)
  - pHorizontalAccuracyLvl, [310](#)
  - pIntermediateReportState, [310](#)
  - pMinIntervalTime, [310](#)
  - pRecurrenceType, [310](#)

- SessionId, [310](#)
- LOCStopReq, [310](#)
  - sessionId, [310](#)
- LPCSTR
  - SwiDataTypes.h, [1159](#)
- LTEAttachProfileListLen
  - \_slqs3GPPConfigItem, [59](#)
- LTEInfo, [312](#)
  - band, [315](#)
  - bandwidth, [315](#)
  - emmConnState, [315](#)
  - emmState, [315](#)
  - emmSubState, [315](#)
  - RXChan, [315](#)
  - TXChan, [315](#)
- LTEInfoInterfreq, [315](#)
  - freqsLen, [316](#)
  - InfoInterfreq, [316](#)
  - ueInIdle, [316](#)
- LTEInfoIntrafreq, [316](#)
  - CellParams, [318](#)
  - cellReselPriority, [318](#)
  - cellsLen, [318](#)
  - earfcn, [318](#)
  - globalCellId, [318](#)
  - plmn, [318](#)
  - sIntraSearch, [318](#)
  - sNonIntraSearch, [318](#)
  - servingCellId, [318](#)
  - tac, [318](#)
  - threshServingLow, [318](#)
  - ueInIdle, [318](#)
- LTEInfoNeighboringGSM, [319](#)
  - freqsLen, [319](#)
  - LteGsmCellInfo, [319](#)
  - ueInIdle, [319](#)
- LTEInfoNeighboringWCDMA, [319](#)
  - freqsLen, [320](#)
  - LTEWCDMACellInfo, [320](#)
  - ueInIdle, [320](#)
- LTERSRPThresh, [321](#)
  - LTERSRPThreshListLen, [321](#)
  - pLTERSRPThreshList, [321](#)
- LTERSRPThreshListLen
  - LTERSRPThresh, [321](#)
- LTERSRQThresh, [322](#)
  - LTERSRQThreshListLen, [322](#)
  - pLTERSRQThreshList, [322](#)
- LTERSRQThreshListLen
  - LTERSRQThresh, [322](#)
- LTERSSIThresh, [322](#)
  - LTERSSIThreshListLen, [323](#)
  - pLTERSSIThreshList, [323](#)
- LTERSSIThreshListLen
  - LTERSSIThresh, [323](#)
- LTESNRThresListLen
  - LTESNRThresh, [326](#)
- LTESNRThresh, [325](#)
  - LTESNRThresListLen, [326](#)
  - pLTESNRThresList, [326](#)
- LTESNRThreshListLen
  - LTESNRThresh, [326](#)
- LTESNRThreshListLen, [326](#)
  - pLTESNRThreshList, [326](#)
- LTESInfo, [326](#)
  - rsrp, [327](#)
  - rsrq, [327](#)
  - rssI, [327](#)
  - snr, [327](#)
- LTESigRptCfg, [323](#)
  - avgPeriod, [323](#)
  - rptRate, [324](#)
- LTESigRptConfig, [324](#)
  - avgPeriod, [324](#)
  - rptRate, [324](#)
- LTESysInfo, [327](#)
  - cellId, [330](#)
  - cellIdValid, [330](#)
  - lac, [330](#)
  - lacValid, [330](#)
  - MCC, [330](#)
  - MNC, [330](#)
  - networkIdValid, [330](#)
  - regRejectInfoValid, [330](#)
  - rejCause, [330](#)
  - rejectSrvDomain, [330](#)
  - sysInfoLTE, [331](#)
  - tac, [331](#)
  - tacValid, [331](#)
- LTEWCDMACellInfo
  - LTEInfoNeighboringWCDMA, [320](#)
- Lac
  - qaQmiServingSystemParam, [427](#)
- lac
  - GERANInfo, [214](#)
  - GSMSysInfo, [263](#)
  - LTESysInfo, [330](#)
  - UMTSInfo, [617](#)
  - WCDMASysInfo, [710](#)
- lac1
  - OperatorPLMNData, [384](#)
- lac2
  - OperatorPLMNData, [384](#)
- lacValid
  - GSMSysInfo, [263](#)
  - LTESysInfo, [330](#)
  - WCDMASysInfo, [711](#)
- language
  - CDMABroadcastConfig, [127](#)
- LastErrCode
  - DataStatusDetail, [184](#)
- Latitude
  - GPSSStateInfo, [256](#)
- leapSeconds
  - qaQmi3Gpp2TimeZone, [422](#)

- len
  - BdsSVInfo, [104](#)
  - SVInfo, [564](#)
- length
  - readTransparentInfo, [454](#)
  - SMSCAddress, [543](#)
  - SMSEtwsMessage, [543](#)
  - SMSTransferRouteMTMessage, [554](#)
- Level
  - GetM2MAudioVolumeResp, [241](#)
  - SetM2MAudioVolumeReq, [503](#)
- lineCtrlInfo, [303](#)
  - polarityIncluded, [304](#)
  - pwrDenialTime, [304](#)
  - revPolarity, [304](#)
  - toggleMode, [304](#)
- lineValue
  - voiceALSSelectLineInfo, [638](#)
- list\_type
  - custSettingList, [176](#)
  - getCustomInput, [227](#)
- listEntries
  - PrefImageList, [404](#)
- listSize
  - ImageList, [279](#)
  - PrefImageList, [404](#)
- LocApplicationInfo, [304](#)
  - appNameLength, [305](#)
  - appProviderLength, [305](#)
  - appVersionLength, [305](#)
  - appVersionValid, [305](#)
  - pAppName, [305](#)
  - pAppProvider, [305](#)
  - pAppVersion, [305](#)
- LocDelAssDataReq, [305](#)
  - pBdsSVInfo, [306](#)
  - pCellDb, [306](#)
  - pClkInfo, [306](#)
  - pGnssData, [306](#)
  - pSVInfo, [306](#)
- localTimeOffset
  - qaQmi3Gpp2TimeZone, [422](#)
- Location Service(LOC), [41](#)
- longName
  - nasPLMNNNameResp, [356](#)
  - PLMNNNetworkNameData, [402](#)
- longNameCl
  - nasPLMNNNameResp, [356](#)
- longNameEn
  - nasPLMNNNameResp, [356](#)
- longNameLen
  - nasPLMNNNameResp, [356](#)
  - PLMNNNetworkNameData, [402](#)
- longNameSB
  - nasPLMNNNameResp, [356](#)
- longNameSpareBits
  - PLMNNNetworkNameData, [402](#)
- Longitude
  - GPSSStateInfo, [256](#)
- LteCQIParm, [310](#)
  - CQIValueCW0, [311](#)
  - CQIValueCW1, [311](#)
  - ValidityCW0, [311](#)
  - ValidityCW1, [311](#)
- LteEmmDI
  - NasSwiIndReg, [359](#)
- LteEmmUI
  - NasSwiIndReg, [359](#)
- LteEsmDI
  - NasSwiIndReg, [359](#)
- LteEsmUI
  - NasSwiIndReg, [359](#)
- LteGsmCellInfo
  - LTEInfoNeighboringGSM, [319](#)
- LteGsmCellInfo, [311](#)
  - cellReselPriority, [312](#)
  - cells\_len, [312](#)
  - GsmCellInfo, [312](#)
  - nccPermitted, [312](#)
  - threshGsmHigh, [312](#)
  - threshGsmLow, [312](#)
- LteNasReleaseInfo
  - qaGobiApiCbkh, [746](#)
- LteNasReleaseInfo\_s, [320](#)
  - nas\_major, [320](#)
  - nas\_minor, [320](#)
  - nas\_release, [320](#)
- LteRsrpDelta
  - SLQSSignalStrengthsIndReq, [535](#)
- LteRsrpinfo
  - SLQSSignalStrengthsInformation, [537](#)
- LteRsrpinformation, [321](#)
  - rsrplevel, [321](#)
- LteSnrDelta
  - SLQSSignalStrengthsIndReq, [535](#)
- LteSnrinfo
  - SLQSSignalStrengthsInformation, [537](#)
- LteSnrinformation, [325](#)
  - snrlevel, [325](#)
- LteWcdmaCellInfo, [331](#)
  - cellReselPriority, [332](#)
  - cellsLen, [332](#)
  - threshXhigh, [332](#)
  - threshXlow, [332](#)
  - uarfcn, [332](#)
  - WCDMACellInfo, [332](#)
- ltersrp
  - slqsSignalStrengthInfo, [533](#)
- ltesnr
  - slqsSignalStrengthInfo, [533](#)
- m\_FwBuildId
  - SWI\_STRUCT\_CarrierImage, [565](#)
- m\_FwImageId
  - SWI\_STRUCT\_CarrierImage, [565](#)
- m\_PriBuildId
  - SWI\_STRUCT\_CarrierImage, [566](#)

- m\_PrImageld
  - SWI\_STRUCT\_CarrierImage, [566](#)
- m\_nCarrierId
  - SWI\_STRUCT\_CarrierImage, [566](#)
- m\_nFolderId
  - SWI\_STRUCT\_CarrierImage, [566](#)
- m\_nStorage
  - SWI\_STRUCT\_CarrierImage, [566](#)
- MACIndex
  - NetworkStatEVDO, [371](#)
- MAX\_BUILD\_ID\_LEN
  - qaGobiApiDms.h, [837](#)
- MAX\_CALL\_NO\_LEN
  - qaGobiApiVoice.h, [1079](#)
- MAX\_CONTENT\_LENGTH
  - qaGobiApiUim.h, [1067](#)
- MAX\_CUST\_ID\_LEN
  - qaGobiApiDms.h, [837](#)
- MAX\_FSN\_LENGTH
  - qaGobiApiDms.h, [837](#)
- MAX\_NO\_OF\_CALLS
  - qaGobiApiCbK.h, [742](#)
  - qaGobiApiVoice.h, [1079](#)
- MAX\_NO\_OF\_FILES
  - qaGobiApiCbK.h, [742](#)
- MAX\_NO\_OF\_SLOTS
  - qaGobiApiCbK.h, [742](#)
  - qaGobiApiUim.h, [1067](#)
- MAX\_NO\_OF\_UUSINFO
  - qaGobiApiCbK.h, [742](#)
- MAX\_PATH\_LENGTH
  - qaGobiApiCbK.h, [743](#)
  - qaGobiApiUim.h, [1067](#)
- MAX\_PILOT\_SETS
  - qaGobiApiNas.h, [909](#)
- MAX\_PUK\_LENGTH
  - qaGobiApiUim.h, [1067](#)
- MAX\_SMS\_ROUTES
  - qaGobiApiSms.h, [988](#)
- MAXUSSDLENGTH
  - qaGobiApiCbK.h, [743](#)
  - qaGobiApiVoice.h, [1079](#)
- MCC
  - CDMASysInfo, [140](#)
  - CDMASysInfoExt, [141](#)
  - currentPLMN, [166](#)
  - GSMSysInfo, [263](#)
  - LTESysInfo, [330](#)
  - SlqsNas3GppNetworkInfo, [523](#)
  - SlqsNas3GppNetworkRAT, [524](#)
  - SlqsNasPcsDigit, [526](#)
  - WCDMASysInfo, [711](#)
- MDMCallDuration
  - ConnectionStatus, [154](#)
- MDMConnStatus
  - ConnectionStatus, [154](#)
- MNC
  - CDMASysInfo, [140](#)
  - currentPLMN, [167](#)
  - GSMSysInfo, [263](#)
  - LTESysInfo, [330](#)
  - SlqsNas3GppNetworkInfo, [524](#)
  - SlqsNas3GppNetworkRAT, [524](#)
  - SlqsNasPcsDigit, [526](#)
  - WCDMASysInfo, [711](#)
- MNRInfo, [333](#)
  - mcc, [335](#)
  - mnc, [335](#)
  - rat, [335](#)
- Mask
  - getDUNCallInfoReq, [228](#)
- mask
  - BdsSV, [103](#)
  - CellIDb, [141](#)
  - ClkInfo, [148](#)
  - GnssData, [250](#)
  - IPv6TrafCls, [303](#)
  - SV, [563](#)
  - Tos, [589](#)
- max\_channel\_rx\_rate
  - WDSSWICurrentChannelRates, [725](#)
- max\_channel\_tx\_rate
  - WDSSWICurrentChannelRates, [726](#)
- MaxChanRxRate
  - ChannelRate, [145](#)
- MaxChanTxRate
  - ChannelRate, [145](#)
- maxDIBitRate
  - QosClassID, [449](#)
- maxDownlinkBitrate
  - UMTSMinQoS, [622](#)
  - UMTSQoS, [625](#)
- maxImages
  - ImageIDEntries, [278](#)
- maxSDUSize
  - UMTSMinQoS, [622](#)
  - UMTSQoS, [625](#)
- maxStorageSize
  - smsMaxStorageSizeResp, [547](#)
- maxUIBitRate
  - QosClassID, [449](#)
- maxUplinkBitrate
  - UMTSMinQoS, [622](#)
  - UMTSQoS, [625](#)
- mcTimeStamp
  - cdmaMsgDecodingParams, [132](#)
- mcc
  - CSGID, [161](#)
  - MNRInfo, [335](#)
  - nasPLMNNameReq, [354](#)
  - netSelectionPref, [363](#)
  - OperatorPLMNData, [384](#)
- mccM
  - minBasedIMSI, [333](#)
- mccT
  - trueIMSI, [592](#)

- MdmConnStatus
  - DUNCallInfoInd, [196](#)
- meanThroughputClass
  - GPRSQoS, [251](#)
  - GPRSRequestedQoS, [252](#)
- meidLength
  - \_SLQSSwiGetSerialNoExtParams, [69](#)
- meidSize
  - serialNumbersInfo, [480](#)
- messageClass
  - smsRouteEntry, [552](#)
- messageFailureCode
  - slqssendsmsparams\_s, [529](#)
- messageFormat
  - slqssendasyncsmsparams\_s, [528](#)
  - slqssendsmsparams\_s, [529](#)
- messageID
  - slqssendsmsparams\_s, [529](#)
  - SMSAsyncRawSend\_s, [542](#)
- messageId
  - cdmaMsgEncodingParams, [134](#)
- messageIndex
  - SMSMTMessage, [549](#)
- messageLength
  - cdmaMsgDecodingParams, [132](#)
- messageMode
  - SMSMemoryInfo, [548](#)
  - SMSMessageMode, [548](#)
- messageSize
  - slqssendasyncsmsparams\_s, [528](#)
  - slqssendsmsparams\_s, [530](#)
  - wcdmaMsgEncodingParams, [705](#)
- messageType
  - smsRouteEntry, [552](#)
- messageWaitingInfoContent, [332](#)
  - activeInd, [332](#)
  - msgCount, [332](#)
  - msgType, [333](#)
- MicMute
  - GetAudioProfileResp, [223](#)
  - GetM2MAudioProfileResp, [240](#)
  - GetM2MAVMuteResp, [243](#)
  - SetAudioProfileReq, [490](#)
  - SetM2MAVMuteReq, [504](#)
- minBasedIMSI, [333](#)
  - imsiM1112, [333](#)
  - imsiMS1, [333](#)
  - imsiMS2, [333](#)
  - mccM, [333](#)
- minute
  - UniversalTime, [628](#)
- mnc
  - CSGID, [161](#)
  - MNRInfo, [335](#)
  - nasPLMNNameReq, [354](#)
  - netSelectionPref, [363](#)
  - OperatorPLMNData, [384](#)
- CSGID, [161](#)
- mobileCountryCode
  - SMSEtwsPlmn, [545](#)
- mobileNetworkCode
  - SMSEtwsPlmn, [545](#)
- mode
  - callInfo, [118](#)
  - UIMRefreshEvent, [609](#)
- modelid\_str
  - slqsfwinf\_s, [521](#)
- modemMode
  - CommInfo, [153](#)
- modemTempNotification
  - qaGobiApiCbK.h, [746](#)
- ModemTempState
  - \_modemTempNotification, [50](#)
- ModemTemperature
  - \_modemTempNotification, [50](#)
- ModifyProfileIn, [335](#)
  - curProfile, [336](#)
  - pProfileID, [336](#)
  - pProfileType, [336](#)
- ModifyProfileOut, [336](#)
  - pExtErrorCode, [336](#)
- month
  - UniversalTime, [628](#)
- msgCount
  - messageWaitingInfoContent, [332](#)
- msgDelFailureCause
  - SMSAsyncRawSend\_s, [542](#)
- msgDelFailureType
  - SMSAsyncRawSend\_s, [542](#)
- msgProtocol
  - smsMsgprotocolResp, [549](#)
- msgType
  - messageWaitingInfoContent, [333](#)
- msgWaitInfo
  - getMsgWaitingInfo, [244](#)
  - msgWaitingInfo, [337](#)
- msgWaitingInfo, [336](#)
  - msgWaitInfo, [337](#)
  - numInstances, [337](#)
- MultDisc
  - protocolSubtypeElement, [421](#)
- multiplier
  - pktErrRate, [399](#)
- NAM\_NAME\_LENGTH
  - qaGobiApiNas.h, [909](#)
- NAS\_SRV
  - qaGobiApiCbK.h, [743](#)
- NSSAudioCtrl, [373](#)
  - downLink, [374](#)
  - upLink, [374](#)
- NUM\_OF\_SET
  - qaGobiApiCbK.h, [743](#)
  - qaGobiApiSms.h, [988](#)
- NWProfile, [374](#)
  - pProfSz, [374](#)

- pProfValues, [374](#)
  - tech, [374](#)
- NWRegStat
  - \_transNWRegInfoNotification, [81](#)
- namID
  - airTimer, [86](#)
  - nasGet3GPP2SubscriptionInfoReq, [339](#)
  - prefVoiceSO, [406](#)
  - roamTimer, [466](#)
- namName, [337](#)
  - namName, [337](#)
  - namNameLen, [337](#)
  - namName, [337](#)
- namNameLen
  - namName, [337](#)
- nameLen
  - remotePartyName, [460](#)
- namePI
  - remotePartyName, [460](#)
- namelength
  - omaDmFotaTlv, [380](#)
  - omaDmFotaTlvExt, [382](#)
- nas\_major
  - LteNasReleaseInfo\_s, [320](#)
- nas\_minor
  - LteNasReleaseInfo\_s, [320](#)
- nas\_release
  - LteNasReleaseInfo\_s, [320](#)
- nasCellLocationInfoResp, [337](#)
  - pCDMAInfo, [338](#)
  - pGERANInfo, [338](#)
  - pLTEInfoInterfreq, [338](#)
  - pLTEInfoIntrafreq, [338](#)
  - pLTEInfoNeighboringGSM, [338](#)
  - pLTEInfoNeighboringWCDMA, [339](#)
  - pUMTSCellID, [339](#)
  - pUMTSInfo, [339](#)
  - pWCDMAInfoLTENeighborCell, [339](#)
- nasGet3GPP2SubscriptionInfoReq, [339](#)
  - namID, [339](#)
- nasGet3GPP2SubscriptionInfoResp, [339](#)
  - pCDMAChannel, [340](#)
  - pDirNum, [340](#)
  - pHomeSIDNID, [340](#)
  - pMinBasedIMSI, [340](#)
  - pNAMNameInfo, [340](#)
  - pTrueIMSI, [340](#)
- nasGetHDRColorCodeResp, [340](#)
  - pColorCode, [341](#)
- nasGetLTECphyCa, [341](#)
  - sPhyCaAggPcellInfo, [341](#)
  - sPhyCaAggScellDIBw, [341](#)
  - sPhyCaAggScellIndType, [341](#)
  - sPhyCaAggScellIndex, [341](#)
  - sPhyCaAggScellInfo, [341](#)
- nasGetLTECphyCaResp, [341](#)
  - pPhyCaAggPcellInfo, [341](#)
  - pPhyCaAggScellDIBw, [341](#)
- pPhyCaAggScellIndType, [342](#)
  - pPhyCaAggScellIndex, [341](#)
  - pPhyCaAggScellInfo, [342](#)
- nasGetSigInfoResp, [342](#)
  - pCDMASSInfo, [342](#)
  - pGSMSSInfo, [342](#)
  - pHDRSSInfo, [342](#)
  - pLTSSInfo, [342](#)
  - pWCDMASSInfo, [342](#)
- nasGetSysInfoResp, [343](#)
  - pAddCDMASysInfo, [345](#)
  - pAddGSMSSysInfo, [345](#)
  - pAddHDRSysInfo, [345](#)
  - pAddLTESysInfo, [345](#)
  - pAddWCDMASysInfo, [345](#)
  - pCDMASrvStatusInfo, [345](#)
  - pCDMASysInfo, [345](#)
  - pGSMCallBarringSysInfo, [345](#)
  - pGSMCipherDomainSysInfo, [345](#)
  - pGSMSrvStatusInfo, [345](#)
  - pGSMSysInfo, [345](#)
  - pHDRSrvStatusInfo, [345](#)
  - pHDRSysInfo, [345](#)
  - pLTESrvStatusInfo, [345](#)
  - pLTESysInfo, [345](#)
  - pLTEVoiceSupportSysInfo, [345](#)
  - pWCDMACallBarringSysInfo, [345](#)
  - pWCDMACipherDomainSysInfo, [345](#)
  - pWCDMASrvStatusInfo, [345](#)
  - pWCDMASysInfo, [345](#)
- nasGetTxRxInfoReq, [345](#)
  - radio\_if, [346](#)
- nasGetTxRxInfoResp, [346](#)
  - pRXChain0Info, [346](#)
  - pRXChain1Info, [346](#)
  - pTXInfo, [347](#)
- nasIndicationRegisterReq, [347](#)
  - pDDTMInd, [350](#)
  - pDualStandByPrefInd, [350](#)
  - pErrorRateInd, [350](#)
  - pHDRNewUATIAssInd, [350](#)
  - pHDRSessionCloseInd, [350](#)
  - pLTECphyCa, [350](#)
  - pManagedRoamingInd, [350](#)
  - pNetworkTimeInd, [350](#)
  - pServingSystemInd, [350](#)
  - pSignalStrengthInd, [350](#)
  - pSubscriptionInfoInd, [350](#)
  - pSysInfoInd, [350](#)
  - pSystemSelectionInd, [350](#)
- nasInitNetworkReg, [350](#)
  - pChangeDuration, [351](#)
  - pMNRIInfo, [351](#)
  - pMncPcsDigitStatus, [351](#)
  - regAction, [351](#)
- nasNetworkTime, [351](#)
  - pDayltSavAdj, [352](#)
  - pTimeZone, [352](#)



- universalTime, 352
- nasOperatorNameResp, 352
  - pNITZInformation, 353
  - pOperatorNameString, 353
  - pOperatorPLMNList, 353
  - pPLMNNetworkName, 353
  - pSrvProviderName, 353
- nasPLMNNNameReq, 353
  - mcc, 354
  - mnc, 354
- nasPLMNNNameResp, 354
  - longName, 356
  - longNameCI, 356
  - longNameEn, 356
  - longNameLen, 356
  - longNameSB, 356
  - shortName, 356
  - shortNameCI, 356
  - shortNameEn, 356
  - shortNameLen, 356
  - shortNameSB, 356
  - spn, 357
  - spnEncoding, 357
  - spnLength, 357
- nasSigInfo, 357
  - pCDMASigInfo, 357
  - pGSMSigInfo, 358
  - pHRSigInfo, 358
  - pLTESigInfo, 358
  - pRscp, 358
  - pTDSCDMASigInfoExt, 358
  - pWCDMASigInfo, 358
- NasSwiIndReg, 358
  - gsmUmtsDI, 359
  - gsmUmtsUI, 359
  - lteEmmDI, 359
  - lteEmmUI, 359
  - lteEsmDI, 359
  - lteEsmUI, 359
- nasSysInfo, 359
  - pAddCDMASysInfo, 361
  - pAddGSMSysInfo, 361
  - pAddHRSysInfo, 361
  - pAddLTESysInfo, 361
  - pAddWCDMASysInfo, 361
  - pCDMASrvStatusInfo, 361
  - pCDMASysInfo, 361
  - pGSMCallBarringSysInfo, 362
  - pGSMCipherDomainSysInfo, 362
  - pGSMStatusInfo, 362
  - pGSMSysInfo, 362
  - pHRSrvStatusInfo, 362
  - pHRSysInfo, 362
  - pLTESrvStatusInfo, 362
  - pLTESysInfo, 362
  - pLTEVoiceSupportSysInfo, 362
  - pSysInfoNoChange, 362
  - pWCDMACallBarringSysInfo, 362
  - pWCDMACipherDomainSysInfo, 362
  - pWCDMASrvStatusInfo, 362
  - pWCDMASysInfo, 362
- nccPermitted
  - lteGsmCellInfo, 312
- NeighborSetCnt
  - NetworkStat1x, 369
- netDescr
  - currentPLMN, 167
- netDescrLength
  - currentPLMN, 167
- netReg
  - netSelectionPref, 363
- netSelectionPref, 362
  - mcc, 363
  - mnc, 363
  - netReg, 363
- NetStats, 363
  - rx\_bytes, 364
  - rx\_errors, 364
  - rx\_overflows, 364
  - rx\_packets, 364
  - tx\_bytes, 364
  - tx\_errors, 364
  - tx\_overflows, 364
  - tx\_packets, 364
- Network Access Service (NAS), 24
- NetworkDebugResp, 364
  - pDataStatusDetail, 366
  - pDeviceConfigDetail, 366
  - pNetworkStat1x, 366
  - pNetworkStatEVDO, 366
  - pObjectVer, 366
- NetworkID
  - qaQmiServingSystemParam, 427
- networkID
  - CDMASysInfo, 140
- networkIdValid
  - CDMASysInfo, 140
  - GSMSysInfo, 263
  - LTESysInfo, 330
  - WCDMASysInfo, 711
- NetworkStat1x, 366
  - ActSetCnt, 369
  - NeighborSetCnt, 369
  - pActPilotPNElements, 369
  - pNeighborSetPilotPN, 369
  - RX\_EC\_IO, 369
  - RX\_PWR, 369
  - SO, 369
  - State, 369
  - TX\_PWR, 369
- NetworkStatEVDO, 369
  - MACIndex, 371
  - PER, 371
  - pSectorID, 371
  - PilotEnergy, 371
  - RX\_PWR, 371



- SNR, [371](#)
- SectorIDLen, [371](#)
- State, [371](#)
- NetworkType
  - CurrNetworkInfo, [170](#)
- newPINLen
  - unlockUIMPIN, [627](#)
- newPINVal
  - unlockUIMPIN, [627](#)
- newPasswd
  - voiceSetCallBarringPwdInfo, [684](#)
- newPasswdAgain
  - voiceSetCallBarringPwdInfo, [684](#)
- newPwd
  - newPwdData, [372](#)
- newPwdAgain
  - newPwdData, [372](#)
- newPwdData, [371](#)
  - newPwd, [372](#)
  - newPwdAgain, [372](#)
- nextHeader
  - TFTIDParams, [588](#)
- nid
  - CDMAInfo, [129](#)
  - sidNid, [514](#)
- NmeaPort
  - DcsUsbPortNames, [184](#)
- nmrArfcn
  - nmrCellInfo, [373](#)
- nmrBsic
  - nmrCellInfo, [373](#)
- nmrCellID
  - nmrCellInfo, [373](#)
- nmrCellInfo, [372](#)
  - nmrArfcn, [373](#)
  - nmrBsic, [373](#)
  - nmrCellID, [373](#)
  - nmrLac, [373](#)
  - nmrPlmn, [373](#)
  - nmrRxLev, [373](#)
- nmrInst
  - GERANInfo, [214](#)
- nmrLac
  - nmrCellInfo, [373](#)
- nmrPlmn
  - nmrCellInfo, [373](#)
- nmrRxLev
  - nmrCellInfo, [373](#)
- noReplyTimer
  - callFWExtInfo, [115](#)
  - callFWInfo, [116](#)
- Non-service specific APIs (SWI), [35](#)
- notifType
  - voiceSUPSNotification, [699](#)
- notification
  - omaDmNotificationsTlv, [383](#)
- notificationType
  - SMSEtwMessage, [543](#)
- num\_instances
  - \_qaQmi3GPP2BroadcastCfgInfo, [53](#)
  - \_qaQmi3GPPBroadcastCfgInfo, [55](#)
  - custSettingList, [176](#)
- numApp
  - slotInfo, [518](#)
- numCrashes
  - CrashInfo, [158](#)
- numEntries
  - CurrentImgList, [166](#)
- numFiles
  - registerRefresh, [458](#)
- numInstance
  - operatorPLMNList, [385](#)
  - PLMNNetworkName, [399](#)
- numInstances
  - arrAlertingPattern, [93](#)
  - arrAlertingType, [94](#)
  - arrAlphaID, [95](#)
  - arrCalledPartyNum, [95](#)
  - arrCallEndReason, [96](#)
  - arrCallInfo, [96](#)
  - arrConnectPartyNum, [97](#)
  - arrDiagInfo, [98](#)
  - arrRedirPartyNum, [98](#)
  - arrRemotePartyName, [99](#)
  - arrRemotePartyNum, [99](#)
  - arrSvcOption, [100](#)
  - arrUUSInfo, [101](#)
  - DomainNameList, [193](#)
  - getCallFWExtInfo, [225](#)
  - getCallFWInfo, [226](#)
  - getMsgWaitingInfo, [244](#)
  - homeSIDNID, [275](#)
  - msgWaitingInfo, [337](#)
  - PCSCFFQDNAddressList, [387](#)
  - PCSCFIPv4ServerAddressList, [388](#)
  - roamIndList, [464](#)
- numLen
  - calledPartyInfo, [109](#)
  - callFWExtInfo, [115](#)
  - callFWInfo, [116](#)
  - callingPartyInfo, [120](#)
  - peerNumberInfo, [393](#)
  - redirNumInfo, [456](#)
  - remotePartyNum, [461](#)
- numOfFiles
  - UIMRefreshEvent, [609](#)
- numOfRoutes
  - smsSetRoutesReq, [553](#)
- numPI
  - peerNumberInfo, [393](#)
- NumPilots
  - PilotSetData, [398](#)
- numPlan
  - calledPartyInfo, [109](#)
  - callFWExtInfo, [115](#)
  - callingPartyInfo, [120](#)

- connectNumInfo, 156
- peerNumberInfo, 393
- redirNumInfo, 456
- numPresInd
  - connectNumInfo, 156
- numQosFlow
  - sQosStat, 556
- numRadiolInterfaces
  - servSystem, 484
- numSI
  - peerNumberInfo, 393
- numSlot
  - cardStatus, 122
- numType
  - calledPartyInfo, 109
  - callFWExtInfo, 115
  - callingPartyInfo, 120
  - connectNumInfo, 156
  - peerNumberInfo, 393
  - redirNumInfo, 456
- number
  - calledPartyInfo, 109
  - callFWExtInfo, 115
  - callFWInfo, 116
  - callingPartyInfo, 120
  - ECTNum, 199
  - peerNumberInfo, 393
  - redirNumInfo, 456
- numberPlan
  - callFwdTypeAndPlan, 112
- numberType
  - callFwdTypeAndPlan, 112
- OKtoRefresh
  - UIMRefreshOKReq, 610
- OMADMCancelSession
  - qaGobiApiOmadm.h, 955
- OMADMGetPendingNIA
  - qaGobiApiOmadm.h, 955
- OMADMGetSessionInfo
  - qaGobiApiOmadm.h, 956
- OMADMStartSession
  - qaGobiApiOmadm.h, 957
- OTASPStatus
  - voiceOTASPStatusInfo, 680
- oddEvenInd
  - calledPartySubAdd, 110
- offset
  - readTransparentInfo, 454
- oldPINLen
  - changeUIMPIN, 143
- oldPINVal
  - changeUIMPIN, 143
- oldPasswd
  - voiceSetCallBarringPwdInfo, 684
- omaDmConfig
  - sessionInfo, 485
  - sessionInfoExt, 485
- omaDmConfigTlv, 374
  - alertmsg, 375
  - alertmsglength, 375
  - state, 375
  - userInputReq, 375
  - userInputTimeout, 375
- omaDmConfigTlvExt, 375
  - alertmsg, 378
  - alertmsglength, 378
  - state, 378
  - userInputReq, 378
  - userInputTimeout, 378
- omaDmFota
  - sessionInfo, 485
  - sessionInfoExt, 485
- omaDmFotaTlv, 378
  - description, 380
  - descriptionlength, 380
  - fwdloadsize, 380
  - fwloadComplete, 380
  - namelength, 380
  - package\_name, 380
  - sessionType, 380
  - severity, 380
  - state, 380
  - updateCompleteStatus, 380
  - userInputReq, 380
  - userInputTimeout, 380
  - version, 380
  - versionlength, 380
- omaDmFotaTlvExt, 380
  - description, 382
  - descriptionlength, 382
  - fumoResultCode, 382
  - namelength, 382
  - package\_name, 382
  - packageSize, 382
  - receivedBytes, 382
  - reserved, 382
  - state, 383
  - userInputTimeout, 383
  - version, 383
  - versionlength, 383
- omaDmNotifications
  - sessionInfo, 485
- omaDmNotificationsTlv, 383
  - notification, 383
  - sessionStatus, 383
- Open Mobile Alliance Service (OMA), 31
- operation
  - depersonalizationInformation, 187
- operatorNameString, 383
  - PLMNName, 383
- OperatorPLMNData, 383
  - lac1, 384
  - lac2, 384
  - mcc, 384
  - mnc, 384
  - PLMNRecID, 384

- operatorPLMNList, [384](#)
  - numInstance, [385](#)
  - PLMNData, [385](#)
- OriginateUSSD
  - qaGobiApiVoice.h, [1081](#)
- p3GPP2Pri
  - swiQosFlow, [577](#)
- p3GPPIImCn
  - swiQosFlow, [577](#)
- p3GPPResResidualBER
  - swiQosFlow, [577](#)
- p3GPPSigInd
  - swiQosFlow, [577](#)
- p3GPPTraHdlPri
  - swiQosFlow, [577](#)
- p3gppRelease
  - \_slqs3GPPConfigItem, [59](#)
- pAMRStatus
  - voiceGetConfigReq, [670](#)
- pAPNClass
  - Profile3GPP, [411](#)
- pAPNClass3GPP2
  - Profile3GPP2, [418](#)
- pAPNDisabledFlag
  - Profile3GPP, [411](#)
- pAPNEnabled3GPP2
  - Profile3GPP2, [418](#)
- pAPNName
  - Profile3GPP, [411](#)
  - qmiWdsRunTimeSettings, [447](#)
  - swiPDPRuntimeSettingsResp, [570](#)
- pAPNnameSize
  - Profile3GPP, [411](#)
- PASSWORD\_LENGTH
  - qaGobiApiVoice.h, [1080](#)
- pAVRXAVCHdroom
  - RXAVCList, [470](#)
- pAVRXAVCSens
  - RXAVCList, [470](#)
- pAccelAcceptReady
  - QmiCbkLocSensorStreamingInd, [438](#)
- pAccelSamplesAccepted
  - QmiCbkLocInjectSensorDataInd, [430](#)
- pAccelTempAcceptReady
  - QmiCbkLocSensorStreamingInd, [438](#)
- pAccelTempSamplesAccepted
  - QmiCbkLocInjectSensorDataInd, [430](#)
- pAcqOrder
  - acqOrderPref, [84](#)
- pAcqOrderPref
  - \_sysSelectPrefParams, [78](#)
- pActPilotPNElements
  - NetworkStat1x, [369](#)
- pAddCDMASysInfo
  - nasGetSysInfoResp, [345](#)
  - nasSysInfo, [361](#)
- pAddGSMSSysInfo
  - nasGetSysInfoResp, [345](#)
- nasSysInfo, [361](#)
- pAddHDRSysInfo
  - nasGetSysInfoResp, [345](#)
  - nasSysInfo, [361](#)
- pAddLTESysInfo
  - nasGetSysInfoResp, [345](#)
  - nasSysInfo, [361](#)
- pAddWCDMASysInfo
  - nasGetSysInfoResp, [345](#)
  - nasSysInfo, [361](#)
- pAddrAllocPref
  - Profile3GPP, [411](#)
- pAirTimer
  - voiceGetConfigReq, [670](#)
- pAirTimerCnt
  - voiceGetConfigResp, [672](#)
- pAirTimerConfig
  - voiceSetConfigReq, [687](#)
- pAirTimerStatus
  - voiceSetConfigResp, [689](#)
- pAlertPriority
  - cdmaMsgDecodingParams, [132](#)
- pAlertType
  - voiceCallInfoResp, [643](#)
- pAlertingPattern
  - voiceCallInfoResp, [643](#)
- pAllowLinger
  - Profile3GPP2, [418](#)
- pAlphaID
  - SMSAsyncRawSend\_s, [542](#)
- pAlphaIDInfo
  - USSResp, [635](#)
  - voiceCallInfoResp, [643](#)
  - voiceCallResponseParams, [647](#)
  - voiceGetCallBarringResp, [655](#)
  - voiceGetCallFWResp, [658](#)
  - voiceGetCallWaitInfo, [660](#)
  - voiceGetCLIPResp, [662](#)
  - voiceGetCLIRResp, [664](#)
  - voiceGetCNAPResp, [665](#)
  - voiceGetCOLPResp, [667](#)
  - voiceGetCOLRResp, [668](#)
  - voiceSetCallBarringPwdResp, [685](#)
  - voiceSetSUPSServiceResp, [693](#)
  - voiceSUPSInfo, [696](#)
- pAlphaIdentifier
  - USSDNoWaitIndicationInfo, [632](#)
- pAltitudeAssumed
  - QmiCbkLocPositionReportInd, [436](#)
- pAltitudeWrtEllipsoid
  - PDSPositionData, [390](#)
  - QmiCbkLocPositionReportInd, [436](#)
- pAltitudeWrtMeanSeaLevel
  - QmiCbkLocPositionReportInd, [436](#)
- pAltitudeWrtSealevel
  - PDSPositionData, [390](#)
- pAmrMode
  - GetIMSVoIPConfigResp, [237](#)

- imsVoIPConfigInfo, [297](#)
- SetIMSVoIPConfigReq, [498](#)
- pAmrOctetAligned
  - GetIMSVoIPConfigResp, [237](#)
  - imsVoIPConfigInfo, [297](#)
  - SetIMSVoIPConfigReq, [498](#)
- pAmrWBMode
  - GetIMSVoIPConfigResp, [237](#)
  - imsVoIPConfigInfo, [297](#)
  - SetIMSVoIPConfigReq, [499](#)
- pAmrWBOctetAligned
  - GetIMSVoIPConfigResp, [237](#)
  - imsVoIPConfigInfo, [297](#)
  - SetIMSVoIPConfigReq, [499](#)
- pAmrWbEnable
  - GetIMSVoIPConfigResp, [237](#)
  - imsVoIPConfigInfo, [297](#)
  - SetIMSVoIPConfigReq, [498](#)
- pApnString
  - Profile3GPP2, [418](#)
- pApnStringSize
  - Profile3GPP2, [418](#)
- pAppName
  - LocApplicationInfo, [305](#)
- pAppPriority
  - Profile3GPP2, [418](#)
- pAppProvider
  - LocApplicationInfo, [305](#)
- pAppSubType
  - HDRProtSubtypResp, [268](#)
- pAppType
  - Profile3GPP2, [418](#)
- pAppVersion
  - LocApplicationInfo, [305](#)
- pApplicationInfo
  - LOCStartReq, [310](#)
- pArrAlertingPattern
  - voiceGetAllCallInfo, [652](#)
  - voiceSetAllCallStatusCbklInfo, [682](#)
- pArrAlertingType
  - voiceGetAllCallInfo, [652](#)
  - voiceSetAllCallStatusCbklInfo, [682](#)
- pArrAlphaID
  - voiceGetAllCallInfo, [652](#)
  - voiceSetAllCallStatusCbklInfo, [682](#)
- pArrCallEndReason
  - voiceGetAllCallInfo, [652](#)
  - voiceSetAllCallStatusCbklInfo, [682](#)
- pArrCallInfo
  - voiceGetAllCallInfo, [652](#)
- pArrCalledPartyNum
  - voiceGetAllCallInfo, [652](#)
  - voiceSetAllCallStatusCbklInfo, [682](#)
- pArrConnectPartyNum
  - voiceGetAllCallInfo, [652](#)
  - voiceSetAllCallStatusCbklInfo, [683](#)
- pArrDiagInfo
  - voiceGetAllCallInfo, [652](#)
- voiceSetAllCallStatusCbklInfo, [683](#)
- pArrRedirPartyNum
  - voiceGetAllCallInfo, [652](#)
  - voiceSetAllCallStatusCbklInfo, [683](#)
- pArrRemotePartyName
  - voiceGetAllCallInfo, [652](#)
  - voiceSetAllCallStatusCbklInfo, [683](#)
- pArrRemotePartyNum
  - voiceGetAllCallInfo, [652](#)
  - voiceSetAllCallStatusCbklInfo, [683](#)
- pArrSvcOption
  - voiceGetAllCallInfo, [653](#)
  - voiceSetAllCallStatusCbklInfo, [683](#)
- pArrUUSInfo
  - voiceGetAllCallInfo, [653](#)
- pAuthPassword
  - Profile3GPP2, [418](#)
- pAuthPasswordSize
  - Profile3GPP2, [418](#)
- pAuthProtocol
  - Profile3GPP2, [418](#)
- pAuthRetryCount
  - Profile3GPP2, [418](#)
- pAuthTimeout
  - Profile3GPP2, [418](#)
- pAuthenticateResult
  - UIMAuthenticateResp, [598](#)
- pAuthentication
  - qmiWdsRunTimeSettings, [447](#)
  - ssdatasession\_params, [560](#)
- pAuthenticationPref
  - Profile3GPP2, [411](#)
- pAutoAnsStatus
  - voiceSetConfigResp, [689](#)
- pAutoAnswer
  - voiceGetConfigReq, [670](#)
  - voiceSetConfigReq, [687](#)
- pAutoAnswerStat
  - voiceGetConfigResp, [672](#)
- pAutosdm
  - \_SLQSOMADMSSettings, [65](#)
  - \_SLQSOMADMSSettingsReqParams, [66](#)
  - \_SLQSOMADMSSettingsReqParams3, [67](#)
- pBandPref
  - \_sysSelectPrefInfo, [73](#)
  - \_sysSelectPrefParams, [78](#)
- pBdsSVInfo
  - LocDelAssDataReq, [306](#)
- pBearerID
  - QosFlowInfo, [451](#)
- pBearerId
  - swiPDPRuntimeSettingsResp, [570](#)
- pBearerTech
  - DataBearerTechExt, [179](#)
- pBurstDTMFLengths
  - voiceBurstDTMFInfo, [640](#)
- pCCResType
  - voiceGetCallBarringResp, [655](#)

- voiceGetCallFWResp, [659](#)
- voiceGetCallWaitInfo, [660](#)
- voiceGetCLIPResp, [662](#)
- voiceGetCLIRResp, [664](#)
- voiceGetCNAPResp, [665](#)
- voiceGetCOLPResp, [667](#)
- voiceGetCOLRResp, [668](#)
- voiceSetCallBarringPwdResp, [685](#)
- pCCResultType
  - voiceCallResponseParams, [647](#)
  - voiceSetSUPSServiceResp, [693](#)
- pCCSUPSType
  - voiceCallResponseParams, [647](#)
  - voiceGetCallBarringResp, [655](#)
  - voiceGetCallFWResp, [659](#)
  - voiceGetCallWaitInfo, [660](#)
  - voiceGetCLIPResp, [662](#)
  - voiceGetCLIRResp, [664](#)
  - voiceGetCNAPResp, [665](#)
  - voiceGetCOLPResp, [667](#)
  - voiceGetCOLRResp, [668](#)
  - voiceSetCallBarringPwdResp, [685](#)
  - voiceSetSUPSServiceResp, [693](#)
- pCCSuppsType
  - USSResp, [635](#)
- pCDMAChannel
  - nasGet3GPP2SubscriptionInfoResp, [340](#)
- pCDMAECIODelta
  - setSignalStrengthInfo, [511](#)
- pCDMAECIOTresh
  - setSignalStrengthInfo, [511](#)
- pCDMAECIOTreshList
  - CDMAECIOTresh, [128](#)
- pCDMAFrameErrRate
  - GetErrRateResp, [232](#)
- pCDMAInfo
  - nasCellLocationInfoResp, [338](#)
- pCDMARSSIDelta
  - setSignalStrengthInfo, [511](#)
- pCDMARSSITresh
  - setSignalStrengthInfo, [511](#)
- pCDMARSSITreshList
  - CDMARSSITresh, [135](#)
- pCDMASSInfo
  - nasGetSigInfoResp, [342](#)
- pCDMASigInfo
  - nasSigInfo, [357](#)
- pCDMASrvStatusInfo
  - nasGetSysInfoResp, [345](#)
  - nasSysInfo, [361](#)
- pCDMASysInfo
  - nasGetSysInfoResp, [345](#)
  - nasSysInfo, [361](#)
- pCLIPResp
  - voiceGetCLIPResp, [663](#)
- pCLIPstatus
  - voiceSUPSInfo, [696](#)
- pCLIRCause
  - voiceInfoRec, [676](#)
- pCLIRResp
  - voiceGetCLIRResp, [664](#)
- pCLIRType
  - voiceCallRequestParams, [646](#)
- pCLIRstatus
  - voiceSUPSInfo, [696](#)
- PCMparams, [385](#)
  - iFaceTab, [385](#)
  - iFaceTabLen, [385](#)
- pCNAPResp
  - voiceGetCNAPResp, [666](#)
- pCNAPstatus
  - voiceSUPSInfo, [696](#)
- pCOLPResp
  - voiceGetCOLPResp, [667](#)
- pCOLPstatus
  - voiceSUPSInfo, [696](#)
- pCOLRResp
  - voiceGetCOLRResp, [669](#)
- pCOLRstatus
  - voiceSUPSInfo, [696](#)
- PCSCFFQDNAddress, [385](#)
  - fqdnAddr, [387](#)
  - fqdnLen, [387](#)
- PCSCFFQDNAddressList, [387](#)
  - numInstances, [387](#)
  - pcsfFQDNAddress, [387](#)
- PCSCFIPv4ServerAddressList, [387](#)
  - numInstances, [388](#)
  - pcsfIPv4Addr, [388](#)
- pCSCFPortName
  - imsRegMgrConfigInfo, [292](#)
  - SetRegMgrConfigReq, [506](#)
- pCSCFPortNameLen
  - SetRegMgrConfigReq, [506](#)
- pCSGID
  - \_sysSelectPrefParams, [78](#)
- pCUGIndex
  - voiceSUPSNotification, [699](#)
- pCUGInfo
  - voiceCallRequestParams, [646](#)
- pCallBarPasswd
  - voiceSUPSInfo, [696](#)
- pCallBarringPasswd
  - voiceSetSUPSServiceReq, [692](#)
- pCallEndReason
  - getDUNCallInfoResp, [231](#)
- pCallFWNum
  - voiceSUPSInfo, [696](#)
- pCallFWTimerVal
  - voiceSUPSInfo, [696](#)
- pCallForwardingNumber
  - voiceSetSUPSServiceReq, [692](#)
- pCallFwdInfo
  - voiceSUPSInfo, [696](#)
- pCallFwdTypeAndPlan
  - voiceSetSUPSServiceReq, [692](#)

- pCallID
  - burstDTMFInfo, [105](#)
  - voiceCallResponseParams, [647](#)
  - voiceContDTMFInfo, [648](#)
  - voiceFlashInfo, [650](#)
  - voiceGetCallBarringResp, [655](#)
  - voiceGetCallFWResp, [658](#)
  - voiceGetCallWaitInfo, [660](#)
  - voiceGetCLIPResp, [662](#)
  - voiceGetCLIRResp, [664](#)
  - voiceGetCNAPResp, [665](#)
  - voiceGetCOLPResp, [667](#)
  - voiceGetCOLRResp, [668](#)
  - voiceManageCallsReq, [678](#)
  - voiceSetCallBarringPwdResp, [685](#)
  - voiceSetSUPSServiceResp, [693](#)
  - voiceSUPSInfo, [696](#)
- pCallId
  - USSResp, [635](#)
  - voiceAnswerCall, [639](#)
- pCallInfo
  - voiceCallInfoResp, [643](#)
- pCallPartySubAdd
  - voiceCallRequestParams, [646](#)
- pCallType
  - voiceCallRequestParams, [646](#)
- pCallWaitInd
  - voiceInfoRec, [676](#)
- pCallbackAddr
  - cdmaMsgEncodingParams, [134](#)
- pCallbkAddr
  - cdmaMsgDecodingParams, [132](#)
- pCallbkAddrLength
  - cdmaMsgDecodingParams, [132](#)
- pCalledPartyInfo
  - voiceInfoRec, [676](#)
- pCallerIDInfo
  - voiceInfoRec, [676](#)
- pCallerNameInfo
  - voiceInfoRec, [676](#)
- pCallingPartyInfo
  - voiceInfoRec, [676](#)
- pCardResult
  - UIMAuthenticateResp, [598](#)
  - UIMGetFileAttributesResp, [602](#)
  - UIMReadTransparentResp, [606](#)
- pCardStatus
  - UIMGetCardStatusResp, [600](#)
- pCcResultType
  - USSResp, [635](#)
- pCellDb
  - LocDelAssDataReq, [306](#)
- pChangeDuration
  - nasInitNetworkReq, [351](#)
- pChannelRate
  - getDUNCallInfoResp, [231](#)
- pChgDuration
  - \_sysSelectPrefParams, [78](#)
- pClkInfo
  - LocDelAssDataReq, [306](#)
- pCodecSTGain
  - GetAudioPathConfigResp, [221](#)
  - SetAudioPathConfigReq, [488](#)
- pColorCode
  - nasGetHDRColorCodeResp, [341](#)
- pConfigAltitudeAssumed
  - LOCStartReq, [310](#)
- pConnectNumInfo
  - voiceCallInfoResp, [643](#)
  - voiceInfoRec, [676](#)
- pConnectionStatus
  - getDUNCallInfoResp, [231](#)
- pContextId
  - swiPDPRuntimeSettingsResp, [570](#)
- pCrashInfo
  - CrashInfoParams, [159](#)
- pCrashString
  - CrashInfo, [158](#)
- pCurAMRConfig
  - voiceGetConfigResp, [673](#)
- pCurDataBearerTechnology
  - dataBearers, [177](#)
- pCurPrefVoiceSO
  - voiceGetConfigResp, [673](#)
- pCurVoiceDomainPref
  - voiceGetConfigResp, [673](#)
- pCurVoicePrivacyPref
  - voiceGetConfigResp, [673](#)
- pCurrChannelRateInd
  - wdsSetEventReportReq, [723](#)
- pCurrDataBearerTechInd
  - wdsSetEventReportReq, [723](#)
- pCurrImgInfo
  - CurrentImgList, [166](#)
- pCurrNetworkInfo
  - CurrDataSysStat, [164](#)
- pCurrPrefDataSysInd
  - wdsSetEventReportReq, [723](#)
- pCurrTTYMode
  - voiceGetConfigResp, [673](#)
- pCurrentChannelIRXRate
  - WdsConnectionRateElmnts, [714](#)
- pCurrentChannelTXRate
  - WdsConnectionRateElmnts, [714](#)
- pCurrentPersonality
  - HDRPersonalityInd, [266](#)
  - HDRPersonalityResp, [267](#)
- pCurrentPrsnlty
  - HDRProtSubtypResp, [268](#)
- pCustSettingInfo
  - getCustomFeatureV2, [226](#)
- pCustSettingList
  - getCustomFeatureV2, [227](#)
- pCwtMute
  - SetM2MAudioProfileReq, [502](#)
  - SetM2MAVMuteReq, [504](#)

- pDDTMInd
  - nasIndicationRegisterReq, [350](#)
- pDHCPRelayEnabled
  - custFeaturesInfo, [172](#)
  - custFeaturesSetting, [174](#)
- PDOP
  - precisionDilution\_s, [403](#)
- pDRCPParams
  - GetHRPDStatsResp, [233](#)
- PDS\_SRV
  - qaGobiApiCbK.h, [743](#)
- PDSInjectTimeReference
  - qaGobiApiPds.h, [964](#)
- PDSPosMethodStateReq, [390](#)
  - pWifiState, [391](#)
  - pXtraDataState, [391](#)
  - pXtraTimeState, [391](#)
- PDSPositionData, [388](#)
  - pAltitudeWrtEllipsoid, [390](#)
  - pAltitudeWrtSealevel, [390](#)
  - pHorizontalConfidence, [390](#)
  - pHorizontalUncCircular, [390](#)
  - pLatitude, [390](#)
  - pLongitude, [390](#)
  - pPositionSource, [390](#)
  - pTimeStamp, [390](#)
  - pTimeType, [390](#)
  - pVerticalConfidence, [390](#)
  - pVerticalUnc, [390](#)
- pDTMFTXGain
  - GetAudioPathConfigResp, [221](#)
  - SetAudioPathConfigReq, [488](#)
- pDataBearer
  - QosEventInfo, [450](#)
  - slqsWdsEventInfo, [539](#)
- pDataBearerTech
  - getDUNCallInfoResp, [231](#)
- pDataBearerTechInd
  - wdsSetEventReportReq, [723](#)
- pDataCallStatusChangeInd
  - wdsSetEventReportReq, [723](#)
- pDataMode
  - Profile3GPP2, [418](#)
- pDataRate
  - Profile3GPP2, [418](#)
  - swiQosFlow, [577](#)
- pDataSrc
  - voiceSUPSInfo, [696](#)
- pDataStatusDetail
  - NetworkDebugResp, [366](#)
- pDataSystemStatusChangeInd
  - wdsSetEventReportReq, [723](#)
- pDate
  - \_SLQSOMADMSessionInfo, [63](#)
- pDateLength
  - \_SLQSOMADMSessionInfo, [63](#)
- pDayltSavAdj
  - nasNetworkTime, [352](#)
- pDefaultPDNEnabled
  - \_slqs3GPPConfigItem, [59](#)
- pDescription
  - QmiNas3GppNetworkInfo, [442](#)
- pDestAddr
  - cdmaMsgEncodingParams, [134](#)
  - wcdmaMsgEncodingParams, [705](#)
- pDetachAction
  - PSDetachReq, [421](#)
- pDevCrashStatus
  - CrashInfoParams, [159](#)
- pDeviceConfigDetail
  - NetworkDebugResp, [366](#)
- pDiagInfo
  - voiceCallInfoResp, [643](#)
- pDigitBuff
  - burstDTMFInfo, [105](#)
- pDirNum
  - nasGet3GPP2SubscriptionInfoResp, [340](#)
- pDisableIMSI
  - custFeaturesInfo, [172](#)
- pDisplInfo
  - voiceInfoRec, [676](#)
- pDisplayMode
  - cdmaMsgDecodingParams, [132](#)
- pDomainList
  - qmiWdsRunTimeSettings, [447](#)
- pDormancyStatus
  - getDUNCallInfoResp, [231](#)
  - slqsWdsEventInfo, [539](#)
- pDormancyStatusInd
  - wdsSetEventReportReq, [723](#)
- pDualStandByPrefInd
  - nasIndicationRegisterReq, [350](#)
- pECIOThresList
  - ECIOThresh, [198](#)
- pECIOThresh
  - sigInfo, [515](#)
- pECMode
  - GetAudioPathConfigResp, [222](#)
  - SetAudioPathConfigReq, [488](#)
- pECTNum
  - voiceSUPSNotification, [699](#)
- PER
  - NetworkStatEVDO, [371](#)
- pESNString
  - serialNumbersInfo, [480](#)
- pEVDOPageMonPerChangeInd
  - wdsSetEventReportReq, [723](#)
- pEarMute
  - SetM2MAudioProfileReq, [502](#)
- pEmerMode
  - \_sysSelectPrefInfo, [73](#)
  - \_sysSelectPrefParams, [78](#)
- pEmergencyCategory
  - voiceCallRequestParams, [646](#)
- pEncodingAlphabet
  - cdmaMsgEncodingParams, [134](#)



- pEncryptData
  - UIMReadTransparentReq, [605](#)
- pEncryptedData
  - UIMReadTransparentResp, [606](#)
- pEncryptedPIN1
  - UIMPinResp, [603](#)
  - UIMVerifyPinReq, [615](#)
- pError
  - USSDNoWaitIndicationInfo, [632](#)
- pErrorCodeStr
  - imsaRatStatusInfo, [282](#)
- pErrorMask
  - CATEventDataType, [124](#)
- pErrorRateInd
  - nasIndicationRegisterReq, [350](#)
- pEspSpi
  - swiQosFilter, [573](#)
- pEtwsMessageInfo
  - SMSEventInfo\_s, [546](#)
- pEtwsPlmnInfo
  - SMSEventInfo\_s, [546](#)
- pExtDispInfo
  - voiceInfoRec, [676](#)
- pExtDispRecInfo
  - voiceInfoRec, [676](#)
- pExtErrCode
  - \_GetProfileSettingOut, [47](#)
- pExtErrorCode
  - CreateProfileOut, [160](#)
  - ModifyProfileOut, [336](#)
- pFOTAUpdate
  - \_SLQSOMADMSettings, [65](#)
- pFOTAdownload
  - \_SLQSOMADMSettings, [65](#)
- pFailCause
  - voiceGetCallBarringResp, [655](#)
  - voiceGetCallFWResp, [659](#)
  - voiceGetCallWaitInfo, [660](#)
  - voiceGetCLIPResp, [663](#)
  - voiceGetCLIRResp, [664](#)
  - voiceGetCNAPResp, [666](#)
  - voiceGetCOLPResp, [667](#)
  - voiceGetCOLRResp, [669](#)
  - voiceManageCallsResp, [678](#)
  - voiceSetCallBarringPwdResp, [685](#)
  - voiceSetSUPSServiceResp, [693](#)
  - voiceSUPSInfo, [696](#)
- pFailErrorCode
  - imsaPdpStatusInfo, [281](#)
- pFailureCause
  - USSDNoWaitIndicationInfo, [632](#)
- pFile
  - ERIFileparams, [200](#)
- pFileAttributes
  - UIMGetFileAttributesResp, [602](#)
- pFileSize
  - ERIFileparams, [200](#)
- pFixId
  - QmiCbkLocPositionReportInd, [436](#)
- pFlag
  - RXPCMIIRFiltr, [473](#)
  - TXPCMIIRFiltr, [596](#)
- pFlashPayLd
  - voiceFlashInfo, [650](#)
- pFlashType
  - voiceFlashInfo, [650](#)
- pFollowOnDC
  - slqssendasyncsmsparams\_s, [528](#)
- pForbidden
  - QmiNas3GppNetworkInfo, [442](#)
- pForceOnDC
  - slqssendasyncsmsparams\_s, [528](#)
- pFwAutoCheck
  - \_SLQSOMADMSettings, [65](#)
  - \_SLQSOMADMSettingsReqParams3, [67](#)
- pGCDumpString
  - CrashInfo, [158](#)
- pGERANInfo
  - nasCellLocationInfoResp, [338](#)
- pGPRSGrantedQoS
  - qmiWdsRunTimeSettings, [447](#)
- pGPRSMInimumQoS
  - Profile3GPP, [411](#)
- pGPRSRequestedQos
  - Profile3GPP, [411](#)
- pGPSEnable
  - custFeaturesSetting, [174](#)
- pGPSLPM
  - custFeaturesInfo, [172](#)
  - custFeaturesSetting, [174](#)
- pGPSSel
  - custFeaturesInfo, [172](#)
  - custFeaturesSetting, [174](#)
- pGSMBER
  - GetErrRateResp, [232](#)
- pGSMCallBarringSysInfo
  - nasGetSysInfoResp, [345](#)
  - nasSysInfo, [362](#)
- pGSMCipherDomainSysInfo
  - nasGetSysInfoResp, [345](#)
  - nasSysInfo, [362](#)
- pGSMRSSIDelta
  - setSignalStrengthInfo, [511](#)
- pGSMRSSIThresh
  - setSignalStrengthInfo, [511](#)
- pGSMRSSIThreshList
  - GSMRSSIThresh, [259](#)
- pGSMSSInfo
  - nasGetSigInfoResp, [342](#)
- pGSMSSigInfo
  - nasSigInfo, [358](#)
- pGSMSSrvStatusInfo
  - nasGetSysInfoResp, [345](#)
  - nasSysInfo, [362](#)
- pGSMSSysInfo
  - nasGetSysInfoResp, [345](#)



- nasSysInfo, [362](#)
- pGWAcqOrderPref
  - \_sysSelectPrefInfo, [73](#)
  - \_sysSelectPrefParams, [78](#)
- pGWAddressV4
  - qmiWdsRunTimeSettings, [447](#)
- pGenerator
  - GetM2MAudioProfileReq, [238](#)
  - SetM2MAudioProfileReq, [502](#)
- pGetCallFWExtInfo
  - voiceGetCallFWResp, [659](#)
- pGetCallFWInfo
  - voiceGetCallFWResp, [659](#)
- pGetCustomInput
  - getCustomFeatureV2, [227](#)
- pGnssData
  - LocDelAssDataReq, [306](#)
- pGpsTime
  - QmiCbkLocPositionReportInd, [436](#)
- pGyroAcceptReady
  - QmiCbkLocSensorStreamingInd, [438](#)
- pGyroSamplesAccepted
  - QmiCbkLocInjectSensorDataInd, [430](#)
- pGyroTempAcceptReady
  - QmiCbkLocSensorStreamingInd, [438](#)
- pGyroTempSamplesAccepted
  - QmiCbkLocInjectSensorDataInd, [430](#)
- pHDRECIODelta
  - setSignalStrengthInfo, [511](#)
- pHDRECIOThresh
  - setSignalStrengthInfo, [511](#)
- pHDRECIOThreshList
  - HDRECIOThresh, [266](#)
- pHDRIODelta
  - setSignalStrengthInfo, [511](#)
- pHDRIOTThresh
  - setSignalStrengthInfo, [511](#)
- pHDRIOTThreshList
  - HDRIOTThresh, [266](#)
- pHDRNewUATIAssInd
  - nasIndicationRegisterReq, [350](#)
- pHDRPackErrRate
  - GetErrRateResp, [232](#)
- pHDRRSSIDelta
  - setSignalStrengthInfo, [511](#)
- pHDRRSSIThresh
  - setSignalStrengthInfo, [511](#)
- pHDRRSSIThreshList
  - HDRRSSIThresh, [268](#)
- pHRSINRDelta
  - setSignalStrengthInfo, [511](#)
- pHRSINRThresList
  - HRSINRThresh, [269](#)
- pHRSINRThresh
  - setSignalStrengthInfo, [511](#)
  - sigInfo, [515](#)
- pHRSINRThreshList
  - HRSINRThreshold, [270](#)
- pHDRSSInfo
  - nasGetSigInfoResp, [342](#)
- pHDRSessionCloseInd
  - nasIndicationRegisterReq, [350](#)
- pHDRSigInfo
  - nasSigInfo, [358](#)
- pHDRSrvStatusInfo
  - nasGetSysInfoResp, [345](#)
  - nasSysInfo, [362](#)
- pHDRSysInfo
  - nasGetSysInfoResp, [345](#)
  - nasSysInfo, [362](#)
- pHeading
  - QmiCbkLocPositionReportInd, [436](#)
- pHeadingUnc
  - QmiCbkLocPositionReportInd, [436](#)
- pHomeSIDNID
  - nasGet3GPP2SubscriptionInfoResp, [340](#)
- pHorConfidence
  - QmiCbkLocPositionReportInd, [436](#)
- pHorReliability
  - QmiCbkLocPositionReportInd, [436](#)
- pHorUncCircular
  - QmiCbkLocPositionReportInd, [436](#)
- pHorUncEllipseOrientAzimuth
  - QmiCbkLocPositionReportInd, [436](#)
- pHorUncEllipseSemiMajor
  - QmiCbkLocPositionReportInd, [436](#)
- pHorUncEllipseSemiMinor
  - QmiCbkLocPositionReportInd, [436](#)
- pHorizontalAccuracyLvl
  - LOCStartReq, [310](#)
- pHorizontalConfidence
  - PDSPositionData, [390](#)
- pHorizontalUncCircular
  - PDSPositionData, [390](#)
- pHotSwapStatus
  - UIMGetCardStatusResp, [600](#)
- PI
  - calledPartyInfo, [109](#)
  - callerIDInfo, [111](#)
  - callFWExtInfo, [115](#)
  - callingPartyInfo, [120](#)
  - redirNumInfo, [456](#)
- PIFACEId
  - SetM2MAudioAVCFGRReq, [500](#)
- pIMCNflag
  - qmiWdsRunTimeSettings, [447](#)
- pIMEIString
  - serialNumbersInfo, [480](#)
- pIMSDomain
  - GetIMSUserConfigParams, [234](#)
  - imsUserConfigInfo, [294](#)
  - SetIMSUserConfigReq, [495](#)
- pIMSDomainLen
  - GetIMSUserConfigParams, [234](#)
  - SetIMSUserConfigReq, [495](#)
- pIMSTestMode

- GetRegMgrConfigParams, [246](#)
- imsRegMgrConfigInfo, [292](#)
- SetRegMgrConfigReq, [506](#)
- plOThresList
  - IOThresh, [300](#)
- plOThresh
  - sigInfo, [516](#)
- plPAddressV4
  - QmiWdsIpAddressInfo, [444](#)
  - qmiWdsRunTimeSettings, [447](#)
- plPAddressV6
  - QmiWdsIpAddressInfo, [444](#)
- plPFamSupport
  - custFeaturesInfo, [172](#)
- plPFamily
  - GetInstIDResp, [238](#)
- plPFamilyPreference
  - qmiWdsRunTimeSettings, [447](#)
- plPV6AddrInfo
  - qmiWdsRunTimeSettings, [447](#)
- plPV6GWAddrInfo
  - qmiWdsRunTimeSettings, [448](#)
- plPv4AddrPref
  - Profile3GPP, [411](#)
- plPv4Address
  - swiPDPRuntimeSettingsResp, [570](#)
- plPv4DstAddr
  - swiQosFilter, [573](#)
- plPv4GWAddress
  - swiPDPRuntimeSettingsResp, [570](#)
- plPv4SrcAddr
  - swiQosFilter, [573](#)
- plPv6AddPref
  - Profile3GPP, [412](#)
- plPv6Address
  - swiPDPRuntimeSettingsResp, [570](#)
- plPv6DstAddr
  - swiQosFilter, [573](#)
- plPv6GWAddress
  - swiPDPRuntimeSettingsResp, [570](#)
- plPv6Label
  - swiQosFilter, [573](#)
- plPv6SrcAddr
  - swiQosFilter, [573](#)
- plPv6TrafCls
  - swiQosFilter, [573](#)
- plPv6prefixlen
  - QmiWdsIpAddressInfo, [444](#)
- pld
  - swiQosFilter, [573](#)
- plds
  - swiQosIds, [578](#)
- plgnoreHotSwapSwitch
  - UIMPowerUpReq, [604](#)
- plmCnFlag
  - Profile3GPP, [411](#)
- plmeiSvnString
  - serialNumbersInfo, [480](#)
- plmgType
  - FirmwareUpdatStat, [211](#)
- plmsRegErrCode
  - IMSARegistrationStatus, [283](#)
- plmsRegStatus
  - IMSARegistrationStatus, [283](#)
  - imsaRegStatusInfo, [284](#)
- plnUse
  - QmiNas3GppNetworkInfo, [442](#)
- plndFieldsList
  - IMSASupportedFieldsResp, [287](#)
- plIndicationToken
  - UIMAuthenticateReq, [597](#)
  - UIMAuthenticateResp, [598](#)
  - UIMChangePinReq, [598](#)
  - UIMGetFileAttributesReq, [601](#)
  - UIMGetFileAttributesResp, [602](#)
  - UIMPinResp, [603](#)
  - UIMReadTransparentReq, [605](#)
  - UIMReadTransparentResp, [606](#)
  - UIMSetPinProtectionReq, [613](#)
  - UIMUnblockPinReq, [614](#)
  - UIMVerifyPinReq, [615](#)
- plInstanceID
  - GetInstIDResp, [238](#)
- plInstanceSize
  - QmiNasPerformNetworkScanResp, [443](#)
- plInstances
  - QmiNasPerformNetworkScanResp, [443](#)
- plInstancesSize
  - QmiNasGetRFBandInfoResp, [443](#)
- plIntermediateReportState
  - LOCStartReq, [310](#)
- plpcpAckTimeout
  - Profile3GPP2, [418](#)
- plpcpCreqRetryCount
  - Profile3GPP2, [418](#)
- plsPcscfAddressNedded
  - Profile3GPP2, [418](#)
- plsUDHPresent
  - wcdmaLongMsgDecodingParams, [703](#)
- plsVoiceEnabled
  - custFeaturesInfo, [172](#)
  - custFeaturesSetting, [174](#)
- pJitter
  - swiQosFlow, [577](#)
- pKeyReferenceID
  - UIMChangePinReq, [599](#)
  - UIMSetPinProtectionReq, [613](#)
  - UIMUnblockPinReq, [614](#)
  - UIMVerifyPinReq, [615](#)
- PLMN\_LENGTH
  - qaGobiApiNas.h, [909](#)
- PLMNData
  - operatorPLMNList, [385](#)
- PLMNName
  - operatorNameString, [383](#)
- PLMNNetName

- PLMNNetworkName, [399](#)
- PLMNNetworkName, [399](#)
  - numInstance, [399](#)
  - PLMNNetName, [399](#)
- PLMNNetworkNameData, [400](#)
  - codingScheme, [402](#)
  - countryInitials, [402](#)
  - longName, [402](#)
  - longNameLen, [402](#)
  - longNameSpareBits, [402](#)
  - shortName, [402](#)
  - shortNameLen, [402](#)
  - shortNameSpareBits, [402](#)
- PLMNRecID
  - OperatorPLMNData, [384](#)
- pLTEAttachProfile
  - \_slqs3GPPConfigItem, [59](#)
- pLTEAttachProfileList
  - \_slqs3GPPConfigItem, [59](#)
- pLTEBandPref
  - \_sysSelectPrefInfo, [73](#)
  - \_sysSelectPrefParams, [78](#)
- pLTECphyCa
  - nasIndicationRegisterReq, [350](#)
- pLTEInfo
  - swiModemStatusResp, [566](#)
- pLTEInfoInterfreq
  - nasCellLocationInfoResp, [338](#)
- pLTEInfoIntrafreq
  - nasCellLocationInfoResp, [338](#)
- pLTEInfoNeighboringGSM
  - nasCellLocationInfoResp, [338](#)
- pLTEInfoNeighboringWCDMA
  - nasCellLocationInfoResp, [339](#)
- pLTERSRPDelta
  - setSignalStrengthInfo, [511](#)
- pLTERSRPThresh
  - setSignalStrengthInfo, [511](#)
- pLTERSRPThreshList
  - LTERSRPThresh, [321](#)
- pLTERSRQDelta
  - setSignalStrengthInfo, [511](#)
- pLTERSRQThresh
  - setSignalStrengthInfo, [511](#)
- pLTERSRQThreshList
  - LTERSRQThresh, [322](#)
- pLTERSSIDelta
  - setSignalStrengthInfo, [511](#)
- pLTERSSIThresh
  - setSignalStrengthInfo, [511](#)
- pLTERSSIThreshList
  - LTERSSIThresh, [323](#)
- pLTESNRDelta
  - setSignalStrengthInfo, [511](#)
- pLTESNRThreshList
  - LTESNRThresh, [326](#)
- pLTESNRThresh
  - setSignalStrengthInfo, [511](#)
- sigInfo, [516](#)
- pLTESNRThreshList
  - LTESNRThreshold, [326](#)
- pLTESInfo
  - nasGetSigInfoResp, [342](#)
- pLTESigInfo
  - nasSigInfo, [358](#)
- pLTESigRptCfg
  - sigInfo, [516](#)
- pLTESigRptConfig
  - setSignalStrengthInfo, [511](#)
- pLTESrvStatusInfo
  - nasGetSysInfoResp, [345](#)
  - nasSysInfo, [362](#)
- pLTESysInfo
  - nasGetSysInfoResp, [345](#)
  - nasSysInfo, [362](#)
- pLTEVoiceSupportSysInfo
  - nasGetSysInfoResp, [345](#)
  - nasSysInfo, [362](#)
- pLanguage
  - cdmaMsgDecodingParams, [132](#)
- pLastBearerTech
  - DataBearerTechExt, [179](#)
- pLastCallDataBearerTech
  - getDUNCallInfoResp, [231](#)
- pLastCallDataBearerTechnology
  - dataBearers, [177](#)
- pLastCallRXOKBytesCnt
  - getDUNCallInfoResp, [231](#)
- pLastCallTXOKBytesCnt
  - getDUNCallInfoResp, [231](#)
- pLatency
  - swiQosFlow, [577](#)
- pLatitude
  - PDSPositionData, [390](#)
  - QmiCbkLocPositionReportInd, [436](#)
- pLcpAckTimeout
  - Profile3GPP2, [418](#)
- pLcpCreqRetryCount
  - Profile3GPP2, [418](#)
- pLeapSeconds
  - QmiCbkLocPositionReportInd, [436](#)
- pLineCtrlInfo
  - voiceInfoRec, [676](#)
- pLinktimer
  - slqssendasyncsmsparams\_s, [528](#)
  - slqssendsmsparams\_s, [530](#)
- pLongitude
  - PDSPositionData, [390](#)
  - QmiCbkLocPositionReportInd, [436](#)
- pLoopbackMode
  - WDSSetLoopbackData, [725](#)
- pLoopbackMultiplier
  - WDSSetLoopbackData, [725](#)
- pLteNasRelInfo
  - SwiOTAMsg\_s, [567](#)
- pLteQci

- swiQosFlow, [577](#)
- pMCC
  - QmiNas3GppNetworkInfo, [442](#)
- pMEIDString
  - serialNumbersInfo, [480](#)
- pMICGainSelect
  - GetAudioPathConfigResp, [222](#)
- pMIPStatusInd
  - wdsSetEventReportReq, [723](#)
- pMNC
  - QmiNas3GppNetworkInfo, [442](#)
- pMNCIncPCSDigStat
  - \_sysSelectPrefParams, [78](#)
- pMNRInfo
  - nasInitNetworkReg, [351](#)
- pMTMessageInfo
  - SMSEventInfo\_s, [546](#)
- pMagneticDeviation
  - QmiCbkLocPositionReportInd, [436](#)
- pManString
  - \_SLQSSwiGetHostDevInfoParams, [68](#)
  - \_SLQSSwiSetHostDevInfoParams, [69](#)
- pManagedRoamingInd
  - nasIndicationRegisterReq, [350](#)
- pMaxAllowedPktSz
  - swiQosFlow, [577](#)
- pMaxChannelRXRate
  - WdsConnectionRateElmnts, [714](#)
- pMaxChannelTXRate
  - WdsConnectionRateElmnts, [714](#)
- pMdmCallDurationActive
  - getDUNCallInfoResp, [231](#)
- pMeidString
  - \_SLQSSwiGetSerialNoExtParams, [69](#)
- pMessage
  - cdmaMsgDecodingParams, [132](#)
  - cdmaMsgEncodingParams, [134](#)
  - slqssendasyncsmsparams\_s, [528](#)
  - slqssendsmsparams\_s, [530](#)
  - wcdmaLongMsgDecodingParams, [703](#)
  - wcdmaMsgDecodingParams, [704](#)
- pMessageID
  - cdmaMsgDecodingParams, [132](#)
- pMessageMode
  - smsMaxStorageSizeReq, [547](#)
- pMessageModelInfo
  - SMSEventInfo\_s, [546](#)
- pMessageSize
  - cdmaMsgEncodingParams, [134](#)
- pMicMute
  - SetM2MAudioProfileReq, [502](#)
- pMinBasedIMSI
  - nasGet3GPP2SubscriptionInfoResp, [340](#)
- pMinIntervalTime
  - LOCStartReq, [310](#)
- pMinPolicedPktSz
  - swiQosFlow, [577](#)
- pMinSessionExpiryTimer
  - GetIMSVoIPConfigResp, [237](#)
  - imsVoIPConfigInfo, [297](#)
  - SetIMSVoIPConfigReq, [499](#)
- pMncPcsDigitStatus
  - nasInitNetworkReg, [351](#)
- pModePref
  - \_sysSelectPrefInfo, [73](#)
  - \_sysSelectPrefParams, [78](#)
- pModelString
  - \_SLQSSwiGetHostDevInfoParams, [68](#)
  - \_SLQSSwiSetHostDevInfoParams, [69](#)
- pMtu
  - qmiWdsRunTimeSettings, [448](#)
- pNAMNameInfo
  - nasGet3GPP2SubscriptionInfoResp, [340](#)
- pNITZInformation
  - nasOperatorNameResp, [353](#)
- pNSEnable
  - GetAudioPathConfigResp, [222](#)
  - SetAudioPathConfigReq, [488](#)
- pNSSAudioCtrl
  - voiceInfoRec, [676](#)
- pNSSRelease
  - voiceInfoRec, [676](#)
- pNamID
  - voiceGetConfigReq, [670](#)
- pNameString
  - \_SLQSSwiGetOSInfoParams, [68](#)
  - \_SLQSSwiSetOSInfoParams, [70](#)
- pNegoDnsSrvrPref
  - Profile3GPP2, [418](#)
- pNeighborSetPilotPN
  - NetworkStat1x, [369](#)
- pNetSelPref
  - \_sysSelectPrefInfo, [73](#)
  - \_sysSelectPrefParams, [78](#)
- pNetworkInfo
  - \_slqsNetworkScanInfo, [60](#)
- pNetworkInfoInstances
  - \_slqsNetworkScanInfo, [60](#)
- pNetworkInfoLen
  - CurrDataSysStat, [164](#)
- pNetworkStat1x
  - NetworkDebugResp, [366](#)
- pNetworkStatEVDO
  - NetworkDebugResp, [366](#)
- pNetworkTimeInd
  - nasIndicationRegisterReq, [350](#)
- pNewImsRegStatus
  - IMSRegistrationStatus, [283](#)
- pNewPwdData
  - voiceSUPSInfo, [697](#)
- pNumSupUSBComps
  - USBCompParams, [632](#)
- pNxtHdrProto
  - swiQosFilter, [573](#)
- pOMADMEnabled
  - \_SLQSOMADMSettings, [65](#)

- PORTNAM\_LEN
  - qaGobiApiDcs.h, [823](#)
- pOTASPStatus
  - voiceCallInfoResp, [644](#)
  - voiceGetAllCallInfo, [653](#)
- pObjectVer
  - NetworkDebugResp, [366](#)
- pOffLength
  - voiceDTMFEventInfo, [649](#)
- pOnLength
  - voiceDTMFEventInfo, [649](#)
- pOpaqueIdentifier
  - QmiCbkLocInjectSensorDataInd, [430](#)
- pOperatorNameString
  - nasOperatorNameResp, [353](#)
- pOperatorPLMNList
  - nasOperatorNameResp, [353](#)
- pPCMPParams
  - SetM2MAudioAVCFGReq, [500](#)
- pPCSCFAddrPCO
  - qmiWdsRunTimeSettings, [448](#)
- pPCSCFFQDNAddrList
  - qmiWdsRunTimeSettings, [448](#)
- pPCSCFPort
  - GetRegMgrConfigParams, [246](#)
- pPCSDigitInfo
  - \_slqsNetworkScanInfo, [60](#)
- pPCSDigitInstances
  - \_slqsNetworkScanInfo, [60](#)
- pPDNInactivTimeout
  - Profile3GPP, [412](#)
- pPDNInactivTimeout3GPP2
  - Profile3GPP2, [418](#)
- pPDPTYPE
  - qmiWdsRunTimeSettings, [448](#)
- pPDPTYPE
  - Profile3GPP, [412](#)
- pPDUMessage
  - wcdmaMsgEncodingParams, [705](#)
- pPLMNNetworkName
  - nasOperatorNameResp, [353](#)
- pPRLPref
  - \_sysSelectPrefInfo, [73](#)
  - \_sysSelectPrefParams, [78](#)
- pPRLPreference
  - dmsCurrentPRLInfo, [192](#)
- pPRLVersion
  - dmsCurrentPRLInfo, [192](#)
- pPacketsCountRX
  - QosEventInfo, [451](#)
  - slqsWdsEventInfo, [539](#)
- pPacketsCountTX
  - QosEventInfo, [451](#)
  - slqsWdsEventInfo, [539](#)
- pPartNum
  - wcdmaLongMsgDecodingParams, [703](#)
- pPassword
  - Profile3GPP, [412](#)
  - ssdatasession\_params, [560](#)
- pPasswordSize
  - Profile3GPP, [412](#)
- pPcscfAddrUsingDhcp
  - Profile3GPP, [412](#)
- pPcscfAddrUsingPCO
  - Profile3GPP, [412](#)
- pPdnType
  - Profile3GPP2, [418](#)
- pPdpAccessConFlag
  - Profile3GPP, [412](#)
- pPdpContext
  - Profile3GPP, [412](#)
- pPdpDataCompType
  - Profile3GPP, [412](#)
- pPdpHdrCompType
  - Profile3GPP, [412](#)
- pPdpStatusConfig
  - IMSAdRegisterInfo, [280](#)
- pPersonalityListLength
  - HDRPersonalityInd, [266](#)
  - HDRPersonalityResp, [267](#)
  - HDRProtSubtypResp, [268](#)
- pPhoneCtxtURI
  - GetIMSSMSConfigParams, [234](#)
  - imsSMSConfigInfo, [294](#)
  - SetIMSSMSConfigReq, [494](#)
- pPhoneCtxtURILen
  - GetIMSSMSConfigParams, [234](#)
  - SetIMSSMSConfigReq, [494](#)
- pPhyCaAggPcellInfo
  - nasGetLTECphyCaResp, [341](#)
- pPhyCaAggScellDIBw
  - nasGetLTECphyCaResp, [341](#)
- pPhyCaAggScellIndType
  - nasGetLTECphyCaResp, [342](#)
- pPhyCaAggScellIndex
  - nasGetLTECphyCaResp, [341](#)
- pPhyCaAggScellInfo
  - nasGetLTECphyCaResp, [342](#)
- pPilotSetData
  - GetHRPDStatsResp, [233](#)
- pPilotSetInfo
  - PilotSetData, [398](#)
- pPkgDescLength
  - \_SLQSOMADMSessionInfo, [63](#)
- pPkgDescription
  - \_SLQSOMADMSessionInfo, [63](#)
- pPkgName
  - \_SLQSOMADMSessionInfo, [63](#)
- pPkgNameLength
  - \_SLQSOMADMSessionInfo, [63](#)
- pPktErrRate
  - swiQosFlow, [577](#)
- pPlasmaIDString
  - \_SLQSSwiGetHostDevInfoParams, [68](#)
  - \_SLQSSwiSetHostDevInfoParams, [69](#)
- pPositionSource

- PDSPositionData, [390](#)
- pPppSessCloseTimer1x
  - Profile3GPP2, [418](#)
- pPppSessCloseTimerDO
  - Profile3GPP2, [419](#)
- pPrDNSIPv4Address
  - swiPDPRuntimeSettingsResp, [570](#)
- pPrDNSIPv6Address
  - swiPDPRuntimeSettingsResp, [570](#)
- pPrPCSCFIPv4Address
  - swiPDPRuntimeSettingsResp, [571](#)
- pPrPCSCFIPv6Address
  - swiPDPRuntimeSettingsResp, [571](#)
- pPrecedence
  - swiQosFilter, [573](#)
- pPrecisionDilution
  - QmiCbkLocPositionReportInd, [436](#)
- pPrefNetwork
  - CurrDataSysStat, [164](#)
- pPrefVoiceDomain
  - voiceSetConfigReq, [687](#)
- pPrefVoicePrivacy
  - voiceGetConfigReq, [670](#)
- pPrefVoiceSO
  - voiceGetConfigReq, [670](#)
  - voiceSetConfigReq, [687](#)
- pPrefVoiceSOStatus
  - voiceSetConfigResp, [689](#)
- pPreferred
  - QmiNas3GppNetworkInfo, [442](#)
- pPriCSCFPort
  - imsRegMgrConfigInfo, [292](#)
  - SetRegMgrConfigReq, [506](#)
- pPriCSCFPortName
  - GetRegMgrConfigParams, [246](#)
- pPriCSCFPortNameLen
  - GetRegMgrConfigParams, [246](#)
- pPriDNSIPv4AddPref
  - Profile3GPP, [412](#)
- pPriDNSIPv6addpref
  - Profile3GPP, [412](#)
- pPriV6DnsAddress
  - Profile3GPP2, [419](#)
- pPrimaryDNSV4
  - qmiWdsRunTimeSettings, [448](#)
- pPrimaryDNSV6
  - qmiWdsRunTimeSettings, [448](#)
- pPrimaryID
  - Profile3GPP, [412](#)
- pPrimaryV4DnsAddress
  - Profile3GPP2, [419](#)
- pPriority
  - cdmaMsgDecodingParams, [132](#)
  - cdmaMsgEncodingParams, [134](#)
- pPrivacy
  - cdmaMsgDecodingParams, [132](#)
- pProfSz
  - NWProfile, [374](#)
- pProfValues
  - NWProfile, [374](#)
- pProfileID
  - CreateProfileIn, [160](#)
  - ModifyProfileIn, [336](#)
  - qmiWdsRunTimeSettings, [448](#)
- pProfileId3GPP
  - ssdatasession\_params, [560](#)
- pProfileId3GPP2
  - ssdatasession\_params, [560](#)
  - swiQosFlow, [577](#)
- pProfileIndex
  - CreateProfileOut, [160](#)
- pProfileList
  - \_slqs3GPPConfigItem, [59](#)
- pProfileName
  - qmiWdsRunTimeSettings, [448](#)
- pProfileType
  - CreateProfileIn, [160](#)
  - CreateProfileOut, [160](#)
  - ModifyProfileIn, [336](#)
- pProfilename
  - Profile3GPP, [412](#)
- pProfilenameSize
  - Profile3GPP, [412](#)
- pProtoSubTypElmnt
  - HDRProtSubtypResp, [268](#)
- pProtocolSubtypeElement
  - HDRPersonalityInd, [266](#)
  - HDRPersonalityResp, [267](#)
- pQFlowState
  - QosFlowInfo, [452](#)
- pQmiInterfaceInfo
  - \_packetSrvStatus, [52](#)
  - slqsSessionStateInfo, [530](#)
  - slqsWdsEventInfo, [539](#)
- pQosClassID
  - Profile3GPP, [412](#)
- pRAT
  - \_sysSelectPrefParams, [78](#)
- pRATInfo
  - \_slqsNetworkScanInfo, [60](#)
- pRATInstances
  - \_slqsNetworkScanInfo, [60](#)
- pRATStatus
  - imsaRatStatusInfo, [282](#)
- pRATType
  - Profile3GPP2, [419](#)
- PREFERRED\_INDEX
  - qaNasPerformNetworkScan.h, [1151](#)
- pRFBandInfoElements
  - QmiNasGetRFBandInfoResp, [443](#)
- PRI\_UPDATE\_FAIL
  - qaGobiApiFms.h, [879](#)
- PRLInd
  - qaQmiServingSystemParam, [427](#)
- pRMAutoConnect
  - custFeaturesInfo, [172](#)



- pRSRPThresList
  - RSRPThresh, [467](#)
- pRSRPThresh
  - sigInfo, [516](#)
- pRSRQThresList
  - RSRQThresh, [468](#)
- pRSRQThresh
  - sigInfo, [516](#)
- pRSSIThresList
  - RSSIThresh, [469](#)
- pRSSIThresh
  - sigInfo, [516](#)
- pRTPRTCPInactTimer
  - GetIMSVoIPConfigResp, [237](#)
  - imsVoIPConfigInfo, [297](#)
  - SetIMSVoIPConfigReq, [499](#)
- pRXAGCList
  - GetAudioPathConfigResp, [222](#)
  - SetAudioPathConfigReq, [488](#)
- pRXAIG
  - RXAGCList, [469](#)
- pRXAVCAGCSwitch
  - GetAudioPathConfigResp, [222](#)
  - SetAudioPathConfigReq, [488](#)
- pRXAVCList
  - GetAudioPathConfigResp, [222](#)
  - SetAudioPathConfigReq, [488](#)
- pRXChain0Info
  - nasGetTxRxInfoResp, [346](#)
- pRXChain1Info
  - nasGetTxRxInfoResp, [346](#)
- pRXComprSlope
  - RXAGCList, [469](#)
- pRXComprThres
  - RXAGCList, [469](#)
- pRXDroppedCount
  - WdsPktStatisticsElmnts, [717](#)
- pRXExpSlope
  - RXAGCList, [469](#)
- pRXExpThres
  - RXAGCList, [469](#)
- pRXOKBytesCount
  - getDUNCallInfoResp, [231](#)
- pRXOKBytesLastCall
  - WdsPktStatisticsElmnts, [717](#)
- pRXOKBytesCount
  - WdsPktStatisticsElmnts, [717](#)
- pRXPCMIIRFiltr
  - GetAudioPathConfigResp, [222](#)
  - SetAudioPathConfigReq, [488](#)
- pRXPacketErrors
  - WdsPktStatisticsElmnts, [717](#)
- pRXPacketOverflows
  - WdsPktStatisticsElmnts, [717](#)
- pRXPacketSuccesses
  - WdsPktStatisticsElmnts, [718](#)
- pRXStaticGain
  - RXAGCList, [470](#)
- pRXTotalBytes
  - WdsByteTotalsElmnts, [712](#)
- pRatHandoverStatusConfig
  - IMSAIndRegisterInfo, [280](#)
- pReadAcknowledgementReq
  - cdmaMsgDecodingParams, [133](#)
- pReadResult
  - UIMReadTransparentResp, [606](#)
- pReason
  - voiceSUPSInfo, [697](#)
- pRecurrenceType
  - LOCStartReq, [310](#)
- pRedirNumInfo
  - voiceInfoRec, [676](#)
- pRefData
  - FirmwareUpdatStat, [211](#)
- pRefString
  - FirmwareUpdatStat, [211](#)
- pRefStringLength
  - FirmwareUpdatStat, [211](#)
- pReferenceNum
  - wcdmaLongMsgDecodingParams, [703](#)
- pRefreshEvent
  - UIMRefreshGetLastEventResp, [610](#)
- pRegCallStatInfoEvt
  - \_getIndicationRegResp, [46](#)
  - \_setIndicationRegReq, [56](#)
- pRegDTMFEvents
  - voiceIndicationRegisterInfo, [674](#)
- pRegInd
  - \_getTransLayerInfoResp, [49](#)
- pRegMgrConfigEvents
  - imsCfgIndRegisterInfo, [290](#)
- pRegStatus
  - \_getTransNWRegInfoResp, [50](#)
- pRegStatusConfig
  - IMSAIndRegisterInfo, [280](#)
- pRegStatusErrorCode
  - imsaRegStatusInfo, [284](#)
- pRegTransLayerInfoEvt
  - \_getIndicationRegResp, [46](#)
  - \_setIndicationRegReq, [56](#)
- pRegTransNWRegInfoEvt
  - \_getIndicationRegResp, [46](#)
  - \_setIndicationRegReq, [56](#)
- pRegVoicePrivacyEvents
  - voiceIndicationRegisterInfo, [674](#)
- pRelValidity
  - cdmaMsgEncodingParams, [134](#)
- pRelativeValidity
  - cdmaMsgDecodingParams, [133](#)
- pRemainingRetries
  - UIMDepersonalizationResp, [599](#)
  - UIMPInResp, [603](#)
- pRemotePartyName
  - voiceCallInfoResp, [644](#)
- pRemotePartyNum
  - voiceCallInfoResp, [644](#)

- pReportChannelRate
  - getDUNCallInfoReq, [228](#)
- pReportConnStatus
  - getDUNCallInfoReq, [229](#)
- pReportDataBearerTech
  - getDUNCallInfoReq, [229](#)
- pReportDormStatus
  - getDUNCallInfoReq, [229](#)
- pReqFieldsList
  - IMSASupportedFieldsResp, [287](#)
- pRespData
  - USSDRspFNetwork, [634](#)
- pRespFieldsList
  - IMSASupportedFieldsResp, [287](#)
- pRetryCount
  - \_SLQSOMADMSessionInfo, [63](#)
- pRetryMessage
  - slqssendasyncsmsparams\_s, [528](#)
- pRetryMessageld
  - slqssendasyncsmsparams\_s, [528](#)
- pRevInUse
  - CDMASysInfo, [140](#)
- pRevInUseValid
  - CDMASysInfo, [140](#)
- pRingBackTimer
  - GetIMSVoIPConfigResp, [237](#)
  - imsVoIPConfigInfo, [297](#)
  - SetIMSVoIPConfigReq, [499](#)
- pRingingTimer
  - GetIMSVoIPConfigResp, [237](#)
  - imsVoIPConfigInfo, [297](#)
  - SetIMSVoIPConfigReq, [499](#)
- pRoamPref
  - \_sysSelectPrefInfo, [74](#)
  - \_sysSelectPrefParams, [78](#)
- pRoamTimer
  - voiceGetConfigReq, [670](#)
- pRoamTimerCnt
  - voiceGetConfigResp, [673](#)
- pRoamTimerConfig
  - voiceSetConfigReq, [687](#)
- pRoamTimerStatus
  - voiceSetConfigResp, [689](#)
- pRoaming
  - QmiNas3GppNetworkInfo, [442](#)
- pRscp
  - nasSigInfo, [358](#)
- pRxFilter
  - swiQosModifyReq, [579](#)
  - swiQosReq, [580](#)
- pRxFlow
  - swiQosGranted, [578](#)
  - swiQosModifyReq, [579](#)
  - swiQosReq, [580](#)
- pRxQFilter
  - QosFlowInfo, [452](#)
- pRxQFlowGranted
  - QosFlowInfo, [452](#)
- PSDetachReq, [421](#)
  - pDetachAction, [421](#)
- pSIPConfigEvents
  - imsCfgIndRegisterInfo, [290](#)
- pSIPLocalPort
  - GetSIPConfigResp, [248](#)
  - imsSIPConfigInfo, [293](#)
  - SetSIPConfigReq, [513](#)
- pSMSCAddressInfo
  - SMSEventInfo\_s, [546](#)
- pSMSCConfigEvents
  - imsCfgIndRegisterInfo, [290](#)
- pSMSFormat
  - GetIMSSMSConfigParams, [234](#)
  - imsSMSConfigInfo, [294](#)
  - SetIMSSMSConfigReq, [494](#)
- pSMSONIMSInfo
  - SMSEventInfo\_s, [546](#)
- pSMSOverlPNwInd
  - GetIMSSMSConfigParams, [234](#)
  - imsSMSConfigInfo, [294](#)
  - SetIMSSMSConfigReq, [494](#)
- pSMSSupport
  - custFeaturesInfo, [172](#)
- pSMSSvcRAT
  - imsaSvcStatusInfo, [288](#)
- pSMSSvcStatus
  - imsaSvcStatusInfo, [288](#)
- pSV
  - BdsSVInfo, [104](#)
  - SVInfo, [564](#)
- pSVInfo
  - LocDelAssDataReq, [306](#)
- pSWVerString
  - \_SLQSSwiGetHostDevInfoParams, [68](#)
  - \_SLQSSwiSetHostDevInfoParams, [69](#)
- pSatelliteInfo
  - gnssSvInfoNotification, [250](#)
- pScAddr
  - wcdmaLongMsgDecodingParams, [703](#)
  - wcdmaMsgDecodingParams, [704](#)
- pScAddrLength
  - wcdmaLongMsgDecodingParams, [703](#)
  - wcdmaMsgDecodingParams, [704](#)
- pScanResult
  - \_slqsNetworkScanInfo, [60](#)
- pScrAmrEnable
  - GetIMSVoIPConfigResp, [237](#)
  - imsVoIPConfigInfo, [297](#)
  - SetIMSVoIPConfigReq, [499](#)
- pScrAmrWbEnable
  - GetIMSVoIPConfigResp, [237](#)
  - imsVoIPConfigInfo, [297](#)
  - SetIMSVoIPConfigReq, [499](#)
- pSeDNSIPv4Address
  - swiPDPRuntimeSettingsResp, [571](#)
- pSeDNSIPv6Address
  - swiPDPRuntimeSettingsResp, [571](#)



- pSePCSCFIPv4Address
  - swiPDPRuntimeSettingsResp, [571](#)
- pSePCSCFIPv6Address
  - swiPDPRuntimeSettingsResp, [571](#)
- pSecDNSIPv4AddPref
  - Profile3GPP, [412](#)
- pSecDNSIPv6addpref
  - Profile3GPP, [412](#)
- pSecV6DnsAddress
  - Profile3GPP2, [419](#)
- pSecondaryDNSV4
  - qmiWdsRunTimeSettings, [448](#)
- pSecondaryDNSV6
  - qmiWdsRunTimeSettings, [448](#)
- pSecondaryFlag
  - Profile3GPP, [412](#)
- pSecondaryV4DnsAddress
  - Profile3GPP2, [419](#)
- pSectorID
  - NetworkStatEVDO, [371](#)
- pSenderAddr
  - cdmaMsgDecodingParams, [133](#)
  - wcdmaLongMsgDecodingParams, [703](#)
  - wcdmaMsgDecodingParams, [704](#)
- pSenderAddrLength
  - cdmaMsgDecodingParams, [133](#)
  - wcdmaLongMsgDecodingParams, [703](#)
  - wcdmaMsgDecodingParams, [704](#)
- pSensorDataUsage
  - QmiCbkLocPositionReportInd, [437](#)
- pServerAddrList
  - qmiWdsRunTimeSettings, [448](#)
- pServiceClass
  - voiceSetSUPSServiceReq, [692](#)
- pServiceOption
  - slqssendasyncsmsparams\_s, [528](#)
- pServiceStatusConfig
  - IMSASndRegisterInfo, [280](#)
- pServingSystemInd
  - nasIndicationRegisterReq, [350](#)
- pSessionExpiryTimer
  - GetIMSVoIPConfigResp, [237](#)
  - imsVoIPConfigInfo, [297](#)
  - SetIMSVoIPConfigReq, [499](#)
- pSessionIDv4
  - GetSessionIDResp, [246](#)
- pSessionIDv6
  - GetSessionIDResp, [246](#)
- pSessionState
  - \_SLQSOMADMSessionInfo, [63](#)
- pSessionType
  - \_SLQSOMADMSessionInfo, [63](#)
- pSettingResp
  - GetIMSSMSConfigParams, [234](#)
  - GetIMSUserConfigParams, [234](#)
  - GetIMSVoIPConfigResp, [238](#)
  - GetRegMgrConfigParams, [246](#)
  - GetSIPConfigResp, [247](#)
  - SetIMSSMSConfigResp, [494](#)
  - SetIMSUserConfigResp, [495](#)
  - SetIMSVoIPConfigResp, [499](#)
  - SetRegMgrConfigResp, [506](#)
  - SetSIPConfigResp, [513](#)
- pSeverity
  - \_SLQSOMADMSessionInfo, [63](#)
- pSigCompEnabled
  - GetSIPConfigResp, [247](#)
  - imsSIPConfigInfo, [293](#)
  - SetSIPConfigReq, [513](#)
- pSignalInfo
  - voiceInfoRec, [676](#)
- pSignalStrengthInd
  - nasIndicationRegisterReq, [350](#)
- pSmsOnIms
  - slqssendasyncsmsparams\_s, [529](#)
- pSmsServiceRat
  - IMSAServiceStatus, [286](#)
- pSmsServiceStatus
  - IMSAServiceStatus, [286](#)
- pSource
  - \_SLQSOMADMSessionInfo, [63](#)
- pSourceIP
  - TFTIDParams, [588](#)
- pSourceLength
  - \_SLQSOMADMSessionInfo, [63](#)
- pSpeedHorizontal
  - QmiCbkLocPositionReportInd, [437](#)
- pSpeedUnc
  - QmiCbkLocPositionReportInd, [437](#)
- pSpeedVertical
  - QmiCbkLocPositionReportInd, [437](#)
- pSrcRAT
  - imsaRatStatusInfo, [282](#)
- pSrvDomainPref
  - \_sysSelectPrefInfo, [74](#)
  - \_sysSelectPrefParams, [78](#)
- pSrvOpt
  - voiceCallInfoResp, [644](#)
- pSrvRegRestriction
  - \_sysSelectPrefParams, [78](#)
- pSrvcProviderName
  - nasOperatorNameResp, [353](#)
- pStage0Val
  - RXPCMIIRFtr, [473](#)
  - TXPCMIIRFtr, [596](#)
- pStage1Val
  - RXPCMIIRFtr, [473](#)
  - TXPCMIIRFtr, [596](#)
- pStage2Val
  - RXPCMIIRFtr, [473](#)
  - TXPCMIIRFtr, [596](#)
- pStage3Val
  - RXPCMIIRFtr, [473](#)
  - TXPCMIIRFtr, [596](#)
- pStage4Val
  - RXPCMIIRFtr, [473](#)

- TXPCMIIRFiltr, [596](#)
- pStageCnt
  - RXPCMIIRFiltr, [474](#)
  - TXPCMIIRFiltr, [596](#)
- pStatMask
  - WdsPktStatisticsReq, [718](#)
- pStatus
  - \_SLQSOMADMSessionInfo, [63](#)
- pSubnetMaskV4
  - qmiWdsRunTimeSettings, [448](#)
- pSubscribeTimer
  - GetSIPConfigResp, [248](#)
  - imsSIPConfigInfo, [293](#)
  - SetSIPConfigReq, [513](#)
- pSubscriptionInfoInd
  - nasIndicationRegisterReq, [350](#)
- pSupUSBComps
  - USBCompParams, [632](#)
- pSupportedMsgList
  - IMSSupportedMsgInfo, [288](#)
- pSuppsNotifEvents
  - voiceIndicationRegisterInfo, [674](#)
- pSvUsedforFix
  - QmiCbkLocPositionReportInd, [437](#)
- pSvcClass
  - voiceGetCallBarringReq, [654](#)
  - voiceGetCallBarringResp, [655](#)
  - voiceGetCallFWReq, [657](#)
  - voiceGetCallWaitInfo, [660](#)
  - voiceSUPSInfo, [697](#)
- pSvcType
  - voiceCallRequestParams, [646](#)
- pSysInfoInd
  - nasIndicationRegisterReq, [350](#)
- pSysInfoNoChange
  - nasSysInfo, [362](#)
- pSystemSelectionInd
  - nasIndicationRegisterReq, [350](#)
- pTCPDSTPort
  - swiQosFilter, [573](#)
- pTCPSPort
  - swiQosFilter, [573](#)
- pTDSCDMAECIODelta
  - setSignalStrengthInfo, [511](#)
- pTDSCDMAECIOThresh
  - setSignalStrengthInfo, [511](#)
- pTDSCDMAECIOThreshList
  - TDSCDMAECIOThresh, [584](#)
- pTDSCDMARSCPDelta
  - setSignalStrengthInfo, [511](#)
- pTDSCDMARSCPTThresh
  - setSignalStrengthInfo, [511](#)
- pTDSCDMARSCPTThreshList
  - TDSCDMARSCPTThresh, [584](#)
- pTDSCDMARSSIDelta
  - setSignalStrengthInfo, [511](#)
- pTDSCDMARSSIThresh
  - setSignalStrengthInfo, [512](#)
- pTDSCDMARSSIThreshList
  - TDSCDMARSSIThresh, [585](#)
- pTDSCDMASINRDelta
  - setSignalStrengthInfo, [512](#)
- pTDSCDMASINRThresh
  - setSignalStrengthInfo, [512](#)
- pTDSCDMASINRThreshList
  - TDSCDMASINRThresh, [586](#)
- pTDSCDMASigInfoExt
  - nasSigInfo, [358](#)
- pTFTID1Params
  - Profile3GPP, [412](#)
- pTFTID2Params
  - Profile3GPP, [412](#)
- pTTYConfigStatus
  - voiceSetConfigResp, [689](#)
- pTTYMode
  - voiceGetConfigReq, [670](#)
  - voiceSetConfigReq, [687](#)
- pTXAGCList
  - GetAudioPathConfigResp, [222](#)
  - SetAudioPathConfigReq, [489](#)
- pTXAIG
  - TXAGCList, [593](#)
- pTXAVCSwitch
  - GetAudioPathConfigResp, [222](#)
  - SetAudioPathConfigReq, [489](#)
- pTXComprSlope
  - TXAGCList, [593](#)
- pTXComprThres
  - TXAGCList, [593](#)
- pTXDroppedCount
  - WdsPktStatisticsElmnts, [718](#)
- pTXExpSlope
  - TXAGCList, [593](#)
- pTXExpThres
  - TXAGCList, [593](#)
- pTXGain
  - GetAudioPathConfigResp, [222](#)
  - SetAudioPathConfigReq, [489](#)
- pTXInfo
  - nasGetTxRxInfoResp, [347](#)
- pTXOKBytesCount
  - getDUNCallInfoResp, [231](#)
- pTXOKBytesLastCall
  - WdsPktStatisticsElmnts, [718](#)
- pTXOkBytesCount
  - WdsPktStatisticsElmnts, [718](#)
- pTXPCMIIRFiltr
  - GetAudioPathConfigResp, [222](#)
  - SetAudioPathConfigReq, [489](#)
- pTXPacketErrors
  - WdsPktStatisticsElmnts, [718](#)
- pTXPacketOverflows
  - WdsPktStatisticsElmnts, [718](#)
- pTXPacketSuccesses
  - WdsPktStatisticsElmnts, [718](#)
- pTXStaticGain

- TXAGCList, [593](#)
- pTXTotalBytes
  - WdsByteTotalsElmnts, [712](#)
- pTdsdmaBandPref
  - \_sysSelectPrefParams, [78](#)
- pTechnology
  - qmiWdsRunTimeSettings, [448](#)
  - ssdatasession\_params, [560](#)
- pTechnologyMask
  - QmiCbkLocPositionReportInd, [437](#)
- pTextMsg
  - cdmaMsgDecodingParams, [133](#)
  - cdmaMsgEncodingParams, [134](#)
  - wcdmaLongMsgDecodingParams, [703](#)
  - wcdmaMsgDecodingParams, [704](#)
  - wcdmaMsgEncodingParams, [705](#)
- pTextMsgLength
  - cdmaMsgDecodingParams, [133](#)
  - wcdmaLongMsgDecodingParams, [703](#)
  - wcdmaMsgDecodingParams, [705](#)
- pTgtRAT
  - imsaRatStatusInfo, [282](#)
- pTime
  - \_SLQSOMADMSessionInfo, [63](#)
  - SwiOTAMsg\_s, [567](#)
- pTimeLength
  - \_SLQSOMADMSessionInfo, [64](#)
- pTimeSrc
  - QmiCbkLocPositionReportInd, [437](#)
- pTimeStamp
  - PDSPositionData, [390](#)
- pTimeType
  - PDSPositionData, [390](#)
- pTimeUnc
  - QmiCbkLocPositionReportInd, [437](#)
- pTimeZone
  - nasNetworkTime, [352](#)
- pTimerSIPReg
  - GetSIPConfigResp, [248](#)
  - imsSIPConfigInfo, [293](#)
  - SetSIPConfigReq, [513](#)
- pTimerT1
  - GetSIPConfigResp, [248](#)
  - imsSIPConfigInfo, [293](#)
  - SetSIPConfigReq, [513](#)
- pTimerT2
  - GetSIPConfigResp, [248](#)
  - imsSIPConfigInfo, [293](#)
  - SetSIPConfigReq, [513](#)
- pTimerTf
  - GetSIPConfigResp, [248](#)
  - imsSIPConfigInfo, [293](#)
  - SetSIPConfigReq, [513](#)
- pTimerVal
  - voiceSetSUPSServiceReq, [692](#)
- pTimestampUtc
  - QmiCbkLocPositionReportInd, [437](#)
- pTokenBucket
  - swiQosFlow, [577](#)
- pTos
  - swiQosFilter, [574](#)
- pTotalBytesRX
  - QosEventInfo, [451](#)
  - slqsWdsEventInfo, [539](#)
- pTotalBytesTX
  - QosEventInfo, [451](#)
  - slqsWdsEventInfo, [539](#)
- pTotalNum
  - wcdmaLongMsgDecodingParams, [703](#)
- pTrafficClass
  - swiQosFlow, [577](#)
- pTranDstPort
  - swiQosFilter, [574](#)
- pTranSrcPort
  - swiQosFilter, [574](#)
- pTransLayerInfo
  - \_getTransLayerInfoResp, [49](#)
  - \_transLayerInfoNotification, [81](#)
- pTransferRouteMTMessageInfo
  - SMSEventInfo\_s, [546](#)
- pTransferStatInd
  - getDUNCallInfoReq, [229](#)
  - wdsSetEventReportReq, [723](#)
- pTransferStatusReport
  - smsSetRoutesReq, [553](#)
- pTrueIMSI
  - nasGet3GPP2SubscriptionInfoResp, [340](#)
- pTxFilter
  - swiQosModifyReq, [579](#)
  - swiQosReq, [580](#)
- pTxFlow
  - swiQosGranted, [578](#)
  - swiQosModifyReq, [579](#)
  - swiQosReq, [580](#)
- pTxQFilter
  - QosFlowInfo, [452](#)
- pTxQFlowGranted
  - QosFlowInfo, [452](#)
- pTypeCode
  - USSDRespFNetwork, [634](#)
- pUATI
  - GetHRPDStatsResp, [233](#)
- pUDPDstPort
  - swiQosFilter, [574](#)
- pUDPSrcPort
  - swiQosFilter, [574](#)
- pUMTSCellID
  - nasCellLocationInfoResp, [339](#)
- pUMTSGrantedQoS
  - qmiWdsRunTimeSettings, [448](#)
- pUMTSInfo
  - nasCellLocationInfoResp, [339](#)
- pUMTSMinQoS
  - Profile3GPP, [412](#)
- pUMTSMinQoSSigInd
  - Profile3GPP, [412](#)

- pUMTSReqQoS
  - Profile3GPP, [412](#)
- pUMTSReqQoSsigInd
  - Profile3GPP, [412](#)
- pUSBComp
  - USBCompConfig, [629](#)
  - USBCompParams, [632](#)
- pUSSDData
  - USSDNoWaitIndicationInfo, [632](#)
- pUSSDInfo
  - USSResp, [635](#)
- pUSSInfo
  - voiceSUPSInfo, [697](#)
- pUTSvcRAT
  - imsaSvcStatusInfo, [288](#)
- pUTSvcStatus
  - imsaSvcStatusInfo, [288](#)
- pUUSInfo
  - voiceCallRequestParams, [646](#)
- pUUSInfo
  - voiceCallInfoResp, [644](#)
- pUpdateCompleteStatus
  - \_SLQSOMADMSessionInfo, [64](#)
- pUserAcknowledgementReq
  - cdmaMsgDecodingParams, [133](#)
- pUserConfigEvents
  - imsCfgIndRegisterInfo, [290](#)
- pUserData
  - slqssendasyncsmsparams\_s, [529](#)
- pUserId
  - Profile3GPP2, [419](#)
- pUserIdSize
  - Profile3GPP2, [419](#)
- pUsername
  - Profile3GPP, [412](#)
  - qmiWdsRunTimeSettings, [448](#)
  - ssdatasession\_params, [560](#)
- pUsernameSize
  - Profile3GPP, [412](#)
- pUtServiceRat
  - IMSAServiceStatus, [286](#)
- pUtServiceStatus
  - IMSAServiceStatus, [286](#)
- pV4sessionId
  - WdsByteTotals, [712](#)
  - WdsConnectionRate, [713](#)
  - WdsPktStatisticsResp, [719](#)
- pV6sessionId
  - WdsByteTotals, [712](#)
  - WdsConnectionRate, [713](#)
  - WdsPktStatisticsResp, [719](#)
- pVOIPSvcRAT
  - imsaSvcStatusInfo, [288](#)
- pVOIPSvcStatus
  - imsaSvcStatusInfo, [289](#)
- pVTSvcRAT
  - imsaSvcStatusInfo, [289](#)
- pVTSvcStatus
  - imsaSvcStatusInfo, [289](#)
- pVersionString
  - \_SLQSSwiGetOSInfoParams, [68](#)
  - \_SLQSSwiSetOSInfoParams, [70](#)
- pVertConfidence
  - QmiCbkLocPositionReportInd, [437](#)
- pVertReliability
  - QmiCbkLocPositionReportInd, [437](#)
- pVertUnc
  - QmiCbkLocPositionReportInd, [437](#)
- pVerticalConfidence
  - PDSPositionData, [390](#)
- pVerticalUnc
  - PDSPositionData, [390](#)
- pVoIPConfigEvents
  - imsCfgIndRegisterInfo, [290](#)
- pVoiceDomainPref
  - voiceGetConfigReq, [671](#)
- pVoiceDomainPrefStatus
  - voiceSetConfigResp, [689](#)
- pVoicePrivacy
  - voiceCallInfoResp, [644](#)
  - voiceGetAllCallInfo, [653](#)
- pVoipServiceRat
  - IMSAServiceStatus, [286](#)
- pVoipServiceStatus
  - IMSAServiceStatus, [286](#)
- pVolume
  - SetM2MAudioProfileReq, [502](#)
- pVsServiceRat
  - IMSAServiceStatus, [286](#)
- pVsServiceStatus
  - IMSAServiceStatus, [286](#)
- pVtServiceRat
  - IMSAServiceStatus, [286](#)
- pVtServiceStatus
  - IMSAServiceStatus, [287](#)
- pWCDMABER
  - GetErrRateResp, [232](#)
- pWCDMACallBarringSysInfo
  - nasGetSysInfoResp, [345](#)
  - nasSysInfo, [362](#)
- pWCDMACipherDomainSysInfo
  - nasGetSysInfoResp, [345](#)
  - nasSysInfo, [362](#)
- pWCDMAECIODelta
  - setSignalStrengthInfo, [512](#)
- pWCDMAECIOThresh
  - setSignalStrengthInfo, [512](#)
- pWCDMAECIOThreshList
  - WCDMAECIOThresh, [700](#)
- pWCDMAInfoLTENeighborCell
  - nasCellLocationInfoResp, [339](#)
- pWCDMARSSIDelta
  - setSignalStrengthInfo, [512](#)
- pWCDMARSSIOThresh
  - setSignalStrengthInfo, [512](#)
- pWCDMARSSIOThreshList

- WCDMARSSIThresh, 706
- pWCDMASSInfo
  - nasGetSigInfoResp, 342
- pWCDMASigInfo
  - nasSigInfo, 358
- pWCDMASrvStatusInfo
  - nasGetSysInfoResp, 345
  - nasSysInfo, 362
- pWCDMASysInfo
  - nasGetSysInfoResp, 345
  - nasSysInfo, 362
- pWifiState
  - PDSPosMethodStateReq, 391
- pXtraDataState
  - PDSPosMethodStateReq, 391
- pXtraTimeState
  - PDSPosMethodStateReq, 391
- package\_name
  - omaDmFotaTlv, 380
  - omaDmFotaTlvExt, 382
- packageSize
  - omaDmFotaTlvExt, 382
- packageid\_str
  - slqsfwinfo\_s, 521
- packetSrvStatus
  - qaGobiApiCbK.h, 747
- packetZone
  - CDMASysInfo, 140
- packetZoneValid
  - CDMASysInfo, 140
- path
  - fileInfo, 210
- pathLen
  - fileInfo, 210
- pbIMSRegistered
  - imsaRegStatusInfo, 284
- pci
  - cellParams, 142
  - PhyCaAggPcellInfo, 394
  - PhyCaAggScellIndType, 396
  - PhyCaAggScellInfo, 397
  - umtsLTENbrCell, 619
- pcsfQDNAddress
  - PCSCFFQDNAddressList, 387
- peakRate
  - tokenBucket, 588
- peakThroughputClass
  - GPRSQoS, 251
  - GPRSRequestedQoS, 252
- peerNumberInfo, 391
  - callID, 393
  - numLen, 393
  - numPI, 393
  - numPlan, 393
  - numSI, 393
  - numType, 393
  - number, 393
- PerformNetworkScan
  - qaGobiApiNas.h, 934
- persoFeature
  - appStatus, 92
- persoRetries
  - appStatus, 92
- persoState
  - appStatus, 92
- persoUnblockRetries
  - appStatus, 92
- pfailureCause
  - USSResp, 635
- phase
  - rxInfo, 471
- PhyCaAggPcellInfo, 393
  - dl\_bw\_value, 394
  - freq, 394
  - iLTEbandValue, 394
  - pci, 394
  - TlvPresent, 394
- PhyCaAggScellDIBw, 394
  - dl\_bw\_value, 395
  - TlvPresent, 395
- PhyCaAggScellIndType, 395
  - freq, 396
  - pci, 396
  - scell\_state, 396
  - TlvPresent, 396
- PhyCaAggScellIndex, 395
  - scell\_idx, 395
  - TlvPresent, 395
- PhyCaAggScellInfo, 396
  - dl\_bw\_value, 397
  - freq, 397
  - iLTEbandValue, 397
  - pci, 397
  - scell\_state, 397
  - TlvPresent, 397
- PhysicalLayer
  - protocolSubtypeElement, 421
- PilotEnergy
  - NetworkStatEVDO, 371
- PilotPN
  - PilotSetParams, 398
- PilotSetData, 397
  - NumPilots, 398
  - pPilotSetInfo, 398
- PilotSetParams, 398
  - PilotPN, 398
  - PilotStrength, 398
  - PilotType, 399
- PilotStrength
  - PilotSetParams, 398
- PilotType
  - PilotSetParams, 399
- pin1Len
  - encryptedPIN1, 199
- pin1Retries
  - appStatus, 92

- pin1State
  - appStatus, 92
- pin1Val
  - encryptedPIN1, 199
- pin2Retries
  - appStatus, 92
- pin2State
  - appStatus, 92
- pinID
  - changeUIMPIN, 143
  - setPINProtection, 505
  - unblockUIMPIN, 627
  - verifyUIMPIN, 637
- pinLen
  - changeUIMPIN, 143
  - verifyUIMPIN, 637
- pinLength
  - setPINProtection, 505
- pinOperation
  - setPINProtection, 505
- pinProtection
  - UIMSetPinProtectionReq, 613
- pinVal
  - changeUIMPIN, 143
  - verifyUIMPIN, 637
- pinValue
  - setPINProtection, 505
- PkQmiNasGetRFBandInfo
  - qaNasGetRFBandInfo.h, 1150
- PkQmiNasPerformNetworkScan
  - qaNasPerformNetworkScan.h, 1151
- pkgver
  - CurrentImgList, 166
- pktErrRate, 399
  - exponent, 399
  - multiplier, 399
- PktStatElmntsV4
  - WdsPktStatisticsResp, 719
- PktStatElmntsV6
  - WdsPktStatisticsResp, 719
- plmn
  - GERANInfo, 214
  - LTEInfoIntrafreq, 318
  - UMTSInfo, 617
- polarityIncluded
  - lineCtrlInfo, 304
- Port, 402
  - port, 402
  - range, 402
- port
  - Port, 402
- Position Determination Service (PDS), 27
- precedenceClass
  - GPRSQoS, 251
  - GPRSRequestedQoS, 252
- precisionDilution
  - qaGobiApiCbk.h, 749
- precisionDilution\_s, 403
  - HDOP, 403
  - PDOP, 403
  - VDOP, 403
- PrefImageList, 403
  - listEntries, 404
  - listSize, 404
- prefVoiceSO, 404
  - evrcCapability, 406
  - homeOrigVoiceSO, 406
  - homePageVoiceSO, 406
  - namID, 406
  - roamOrigVoiceSO, 406
- Preferred
  - SlqsNas3GppNetworkInfo, 524
- prefixLen
  - IPv6Addr, 302
- presentationInd
  - ECTNum, 199
  - remotePartyNum, 461
- priChA
  - CDMAChannel, 128
- priChB
  - CDMAChannel, 128
- privacyPref
  - voiceSetPrefPrivacy, 690
- priver
  - CurrentImgList, 166
- priversion\_str
  - slqsfwinfo\_s, 521
- Profile
  - GetAudioPathConfigReq, 220
  - GetAudioProfileResp, 223
  - GetAudioVolTLBConfigReq, 224
  - GetM2MAudioProfileResp, 240
  - GetM2MAudioVolumeReq, 241
  - GetM2MAVMuteReq, 242
  - GetM2MSpkrGainReq, 243
  - SetAudioPathConfigReq, 488
  - SetAudioProfileReq, 490
  - SetAudioVolTLBConfigReq, 492
  - SetM2MAudioAVCFGRReq, 500
  - SetM2MAudioProfileReq, 502
  - SetM2MAudioVolumeReq, 503
  - SetM2MAVMuteReq, 504
  - SetM2MSpkrGainReq, 504
- Profile3GPP, 406
  - pAPNClass, 411
  - pAPNDisabledFlag, 411
  - pAPNName, 411
  - pAPNnameSize, 411
  - pAddrAllocPref, 411
  - pAuthenticationPref, 411
  - pGPRSMinimumQoS, 411
  - pGPRSRequestedQoS, 411
  - pIPv4AddrPref, 411
  - pIPv6AddrPref, 412
  - pImCnFlag, 411
  - pPDNInactivTimeout, 412

- pPDType, [412](#)
- pPassword, [412](#)
- pPasswordSize, [412](#)
- pPscfAddrUsingDhcp, [412](#)
- pPscfAddrUsingPCO, [412](#)
- pPdpAccessConFlag, [412](#)
- pPdpContext, [412](#)
- pPdpDataCompType, [412](#)
- pPdpHdrCompType, [412](#)
- pPriDNSIPv4AddPref, [412](#)
- pPriDNSIPv6addpref, [412](#)
- pPrimaryID, [412](#)
- pProfilename, [412](#)
- pProfilenameSize, [412](#)
- pQosClassID, [412](#)
- pSecDNSIPv4AddPref, [412](#)
- pSecDNSIPv6addpref, [412](#)
- pSecondaryFlag, [412](#)
- pTFTID1Params, [412](#)
- pTFTID2Params, [412](#)
- pUMTSMinQoS, [412](#)
- pUMTSMinQoSSigInd, [412](#)
- pUMTSReqQoS, [412](#)
- pUMTSReqQoSSigInd, [412](#)
- pUsername, [412](#)
- pUsernameSize, [412](#)
- Profile3GPP2, [413](#)
  - pAPNClass3GPP2, [418](#)
  - pAPNEnabled3GPP2, [418](#)
  - pAllowLinger, [418](#)
  - pApnString, [418](#)
  - pApnStringSize, [418](#)
  - pAppPriority, [418](#)
  - pAppType, [418](#)
  - pAuthPassword, [418](#)
  - pAuthPasswordSize, [418](#)
  - pAuthProtocol, [418](#)
  - pAuthRetryCount, [418](#)
  - pAuthTimeout, [418](#)
  - pDataMode, [418](#)
  - pDataRate, [418](#)
  - plpcpAckTimeout, [418](#)
  - plpcpCreqRetryCount, [418](#)
  - plsPscfAddressNedded, [418](#)
  - pLcpAckTimeout, [418](#)
  - pLcpCreqRetryCount, [418](#)
  - pNegoDnsSrvrPref, [418](#)
  - pPDNInactivTimeout3GPP2, [418](#)
  - pPdnType, [418](#)
  - pPppSessCloseTimer1x, [418](#)
  - pPppSessCloseTimerDO, [419](#)
  - pPriV6DnsAddress, [419](#)
  - pPrimaryV4DnsAddress, [419](#)
  - pRATType, [419](#)
  - pSecV6DnsAddress, [419](#)
  - pSecondaryV4DnsAddress, [419](#)
  - pUserId, [419](#)
  - pUserIdSize, [419](#)
- ProfileID
  - \_GetProfileSettingIn, [47](#)
- ProfileIdentifier, [419](#)
  - profileIndex, [419](#)
  - profileType, [419](#)
- profileIndex
  - ProfileIdentifier, [419](#)
  - SLQSDDeleteProfileParams, [520](#)
- ProfileType
  - \_GetProfileSettingIn, [47](#)
- profileType
  - ProfileIdentifier, [419](#)
  - SLQSDDeleteProfileParams, [520](#)
- protocolSubtypeElement, [419](#)
  - AccessMac, [421](#)
  - AuthProt, [421](#)
  - ControlMac, [421](#)
  - EncryptProt, [421](#)
  - ForwardMac, [421](#)
  - IdleState, [421](#)
  - KeyExchange, [421](#)
  - MultDisc, [421](#)
  - PhysicalLayer, [421](#)
  - ReverseMac, [421](#)
  - SecProt, [421](#)
  - VirtStream, [421](#)
- ProvisionStatus
  - CLIPResp, [146](#)
  - CLIRResp, [146](#)
  - CNAPResp, [149](#)
  - COLPResp, [150](#)
  - COLRResp, [151](#)
- psAttachState
  - ServingSystemInfo, [482](#)
  - servSystem, [485](#)
- psBarStatus
  - CallBarringSysInfo, [106](#)
  - callBarStatus, [107](#)
- psState
  - CommInfo, [153](#)
- psc
  - UMTSInfo, [617](#)
  - wcdmaCellInfo, [700](#)
  - WCDMASysInfo, [711](#)
- pscValid
  - WCDMASysInfo, [711](#)
- pscsfIPv4Addr
  - PCSCFIPv4ServerAddressList, [388](#)
- puk1Retries
  - appStatus, [92](#)
- puk2Retries
  - appStatus, [92](#)
- pukLen
  - unblockUIMPIN, [627](#)
- pukVal
  - unblockUIMPIN, [627](#)
- pv4sessionId
  - WdsIpAddressInfoReq, [715](#)



- pv6sessionId
  - WdsIpAddressInfoReq, [715](#)
- pwrDenialTime
  - lineCtrlInfo, [304](#)
- QMI\_SAR\_RF\_STATE\_1
  - qaGobiApiSar.h, [984](#)
- QMI\_SAR\_RF\_STATE\_2
  - qaGobiApiSar.h, [985](#)
- QMI\_SAR\_RF\_STATE\_3
  - qaGobiApiSar.h, [985](#)
- QMI\_SAR\_RF\_STATE\_4
  - qaGobiApiSar.h, [985](#)
- QMI\_SAR\_RF\_STATE\_5
  - qaGobiApiSar.h, [985](#)
- QMI\_SAR\_RF\_STATE\_6
  - qaGobiApiSar.h, [985](#)
- QMI\_SAR\_RF\_STATE\_7
  - qaGobiApiSar.h, [985](#)
- QMI\_SAR\_RF\_STATE\_8
  - qaGobiApiSar.h, [985](#)
- QMI\_SAR\_RF\_STATE\_DEFAULT
  - qaGobiApiSar.h, [984](#)
- QMI\_WDS\_CURRENT\_CALL\_DB\_MASK
  - qaGobiApiWds.h, [1108](#)
- QMI\_WDS\_LAST\_CALL\_DB\_MASK
  - qaGobiApiWds.h, [1108](#)
- QCI
  - QosClassID, [449](#)
- QCWWAN2kConnect
  - qaGobiApiDcs.h, [823](#)
- QCWWAN2kEnumerateDevices
  - qaGobiApiDcs.h, [823](#)
- QCWWAN2kGetConnectedDeviceID
  - qaGobiApiDcs.h, [825](#)
- QCWWANConnect
  - qaGobiApiDcs.h, [825](#)
- QCWWANDisconnect
  - qaGobiApiDcs.h, [826](#)
- QCWWANEnumerateDevices
  - qaGobiApiDcs.h, [826](#)
- QLIC
  - DeviceConfigDetail, [190](#)
- qaCbkCatEventReportInd.h
  - eTLV\_CBK\_ALPHA\_IDENTIFIER, [728](#)
  - eTLV\_CBK\_DISPLAY\_TEXT, [728](#)
  - eTLV\_CBK\_END\_PROACTIVE\_SESSION, [728](#)
  - eTLV\_CBK\_GET\_IN\_KEY, [728](#)
  - eTLV\_CBK\_GET\_INPUT, [728](#)
  - eTLV\_CBK\_LANGUAGE\_NOTIFICATION, [728](#)
  - eTLV\_CBK\_REFRESH, [728](#)
  - eTLV\_CBK\_SELECT\_ITEM, [728](#)
  - eTLV\_CBK\_SETUP\_EVENT\_LIST, [728](#)
  - eTLV\_CBK\_SETUP\_IDLE\_MODE\_TEXT, [728](#)
  - eTLV\_CBK\_SETUP\_MENU, [728](#)
  - eTLV\_END\_PROACTIVE\_SESSION\_LENGTH, [728](#)
  - eTLV\_REFRESH\_LENGTH, [728](#)
  - eTLV\_SETUP\_EVENT\_LIST\_LENGTH, [728](#)
- qaCbkSwiOmaDmEventReportInd.h
  - eTLV\_IND\_OMA\_DM\_CONFIG, [729](#)
  - eTLV\_IND\_OMA\_DM\_FOTA, [729](#)
  - eTLV\_IND\_OMA\_DM\_NOT, [729](#)
- qaGobiApiCbk.h
  - DEVICE\_STATE\_BOOT, [782](#)
  - DEVICE\_STATE\_DISCONNECTED, [782](#)
  - DEVICE\_STATE\_READY, [782](#)
  - eQA\_QMI\_SVC\_NA, [782](#)
  - eQA\_QMI\_SVC\_NAS, [782](#)
  - eQA\_QMI\_SVC\_WDS, [782](#)
  - SMS\_EVENT\_ETWS, [782](#)
  - SMS\_EVENT\_ETWS\_PLMN, [782](#)
  - SMS\_EVENT\_MESSAGE\_MODE, [782](#)
  - SMS\_EVENT\_MT\_MESSAGE, [782](#)
  - SMS\_EVENT\_SMS\_ON\_IMS, [782](#)
  - SMS\_EVENT\_SMSC\_ADDRESS, [782](#)
  - SMS\_EVENT\_TRANSFER\_ROUTE\_MT\_MESSAGE, [782](#)
- qaGobiApiFms.h
  - eGOBI\_DEV\_SERIES\_9X15, [879](#)
  - eGOBI\_DEV\_SERIES\_9X30, [879](#)
  - eGOBI\_DEV\_SERIES\_G3K, [879](#)
  - eGOBI\_DEV\_SERIES\_NON\_GOBI, [879](#)
  - eGOBI\_DEV\_SERIES\_SIERRA\_GOBI, [879](#)
  - eGOBI\_DEV\_SERIES\_UNKNOWN, [879](#)
  - eGOBI\_IMG\_CAR\_3, [880](#)
  - eGOBI\_IMG\_CAR\_AERIS, [880](#)
  - eGOBI\_IMG\_CAR\_ALLTEL, [879](#)
  - eGOBI\_IMG\_CAR\_AMX\_TELCEL, [880](#)
  - eGOBI\_IMG\_CAR\_ATT, [880](#)
  - eGOBI\_IMG\_CAR\_BELL, [879](#)
  - eGOBI\_IMG\_CAR\_BHARTI, [880](#)
  - eGOBI\_IMG\_CAR\_BRASIL\_VIVO, [880](#)
  - eGOBI\_IMG\_CAR\_CHINA\_MOBILE, [880](#)
  - eGOBI\_IMG\_CAR\_CHINA\_TELECOM, [880](#)
  - eGOBI\_IMG\_CAR\_CHINA\_UNICOM, [880](#)
  - eGOBI\_IMG\_CAR\_EMOBILE, [880](#)
  - eGOBI\_IMG\_CAR\_FACTORY, [879](#)
  - eGOBI\_IMG\_CAR\_GENERIC, [879](#)
  - eGOBI\_IMG\_CAR\_GENERIC\_CDMA, [880](#)
  - eGOBI\_IMG\_CAR\_IUSACELL, [880](#)
  - eGOBI\_IMG\_CAR\_KDDI, [880](#)
  - eGOBI\_IMG\_CAR\_KT\_FRETEL, [880](#)
  - eGOBI\_IMG\_CAR\_LEAP, [880](#)
  - eGOBI\_IMG\_CAR\_METROPCS, [880](#)
  - eGOBI\_IMG\_CAR\_NETCOM, [880](#)
  - eGOBI\_IMG\_CAR\_NORF, [879](#)
  - eGOBI\_IMG\_CAR\_NTT\_DOCOMO, [880](#)
  - eGOBI\_IMG\_CAR\_O2, [880](#)
  - eGOBI\_IMG\_CAR\_OMH, [880](#)
  - eGOBI\_IMG\_CAR\_ORANGE, [880](#)
  - eGOBI\_IMG\_CAR\_RELIANCE1, [880](#)
  - eGOBI\_IMG\_CAR\_RELIANCE2, [880](#)
  - eGOBI\_IMG\_CAR\_ROGERS, [880](#)
  - eGOBI\_IMG\_CAR\_SFR, [880](#)
  - eGOBI\_IMG\_CAR\_SINGTEL\_OPTUS, [880](#)
  - eGOBI\_IMG\_CAR\_SK\_TELCOM1, [880](#)



- eGOBI\_IMG\_CAR\_SK\_TELCOM2, [880](#)
- eGOBI\_IMG\_CAR\_SOFTBANK, [880](#)
- eGOBI\_IMG\_CAR\_SPRINT, [879](#)
- eGOBI\_IMG\_CAR\_SWISSCOM, [880](#)
- eGOBI\_IMG\_CAR\_TATA, [880](#)
- eGOBI\_IMG\_CAR\_TELCOM\_ITALIA, [880](#)
- eGOBI\_IMG\_CAR\_TELCOM\_NZ, [880](#)
- eGOBI\_IMG\_CAR\_TELEFONICA, [880](#)
- eGOBI\_IMG\_CAR\_TELNOR, [880](#)
- eGOBI\_IMG\_CAR\_TELIASONERA, [880](#)
- eGOBI\_IMG\_CAR\_TELSTRA1, [880](#)
- eGOBI\_IMG\_CAR\_TELSTRA2, [880](#)
- eGOBI\_IMG\_CAR\_TELUS, [880](#)
- eGOBI\_IMG\_CAR\_TMOBILE, [880](#)
- eGOBI\_IMG\_CAR\_US, [880](#)
- eGOBI\_IMG\_CAR\_VERIZON, [879](#)
- eGOBI\_IMG\_CAR\_VODAFONE, [880](#)
- eGOBI\_IMG\_GPS\_ASSISTED, [881](#)
- eGOBI\_IMG\_GPS\_NO\_XTRA, [881](#)
- eGOBI\_IMG\_GPS\_NONE, [881](#)
- eGOBI\_IMG\_GPS\_STAND\_ALONE, [881](#)
- eGOBI\_IMG\_REG\_ASIA, [881](#)
- eGOBI\_IMG\_REG\_AUS, [881](#)
- eGOBI\_IMG\_REG\_EU, [881](#)
- eGOBI\_IMG\_REG\_GLOBAL, [881](#)
- eGOBI\_IMG\_REG\_LA, [881](#)
- eGOBI\_IMG\_REG\_NA, [881](#)
- eGOBI\_IMG\_TECH\_CDMA, [881](#)
- eGOBI\_IMG\_TECH\_UMTS, [881](#)
- eGobi\_DEV\_SERIES\_MC83, [879](#)
- qaGobiApiNas.h
  - eNAS\_LTE\_CPHY\_CA\_BW\_NRB\_100, [918](#)
  - eNAS\_LTE\_CPHY\_CA\_BW\_NRB\_15, [918](#)
  - eNAS\_LTE\_CPHY\_CA\_BW\_NRB\_25, [918](#)
  - eNAS\_LTE\_CPHY\_CA\_BW\_NRB\_50, [918](#)
  - eNAS\_LTE\_CPHY\_CA\_BW\_NRB\_6, [918](#)
  - eNAS\_LTE\_CPHY\_CA\_BW\_NRB\_75, [918](#)
  - eNAS\_LTE\_CPHY\_SCELL\_STATE\_CONFIGURED\_ACTIVATED, [918](#)
  - eNAS\_LTE\_CPHY\_SCELL\_STATE\_CONFIGURED\_DEACTIVATED, [918](#)
  - eNAS\_LTE\_CPHY\_SCELL\_STATE\_DECONFIGURED, [918](#)
  - eNAS\_RADIO\_IF\_GSM, [918](#)
  - eNAS\_RADIO\_IF\_LTE, [918](#)
  - eNAS\_RADIO\_IF\_TDSCDMA, [918](#)
  - eNAS\_RADIO\_IF\_UMTS, [918](#)
- qaGobiApiPds.h
  - eSetServiceAutomaticTrackingDisable, [959](#)
  - eSetServiceAutomaticTrackingEnable, [959](#)
- qaGobiApiSar.h
  - QMI\_SAR\_RF\_STATE\_1, [984](#)
  - QMI\_SAR\_RF\_STATE\_2, [985](#)
  - QMI\_SAR\_RF\_STATE\_3, [985](#)
  - QMI\_SAR\_RF\_STATE\_4, [985](#)
  - QMI\_SAR\_RF\_STATE\_5, [985](#)
  - QMI\_SAR\_RF\_STATE\_6, [985](#)
  - QMI\_SAR\_RF\_STATE\_7, [985](#)
  - QMI\_SAR\_RF\_STATE\_8, [985](#)
  - QMI\_SAR\_RF\_STATE\_DEFAULT, [984](#)
- qaGobiApiVoice.h
  - VOICE\_SUPS\_SRV\_CLASS\_DATA, [1080](#)
  - VOICE\_SUPS\_SRV\_CLASS\_DATACIRCUITASYNC, [1080](#)
  - VOICE\_SUPS\_SRV\_CLASS\_DATACIRCUITSYNC, [1080](#)
  - VOICE\_SUPS\_SRV\_CLASS\_FAX, [1080](#)
  - VOICE\_SUPS\_SRV\_CLASS\_NONE, [1080](#)
  - VOICE\_SUPS\_SRV\_CLASS\_PACKETACCESS, [1080](#)
  - VOICE\_SUPS\_SRV\_CLASS\_PADACCESS, [1080](#)
  - VOICE\_SUPS\_SRV\_CLASS\_SMS, [1080](#)
  - VOICE\_SUPS\_SRV\_CLASS\_VOICE, [1080](#)
- qaGobiApiWds.h
  - QMI\_WDS\_CURRENT\_CALL\_DB\_MASK, [1108](#)
  - QMI\_WDS\_LAST\_CALL\_DB\_MASK, [1108](#)
- qaNasGetRFBandInfo.h
  - eTLV\_RF\_BAND\_INFO, [1150](#)
- qaNasPerformNetworkScan.h
  - eTLV\_3GPP\_NETWORK\_INFO, [1151](#)
- qaCbkCatEventReportInd.h, [727](#)
  - UpkQmiCbkCatEventReportInd, [729](#)
- qaCbkSwiOmaDmEventReportInd.h, [729](#)
  - UpkQmiCbkSwiOmaDmEventReportInd, [730](#)
  - UpkQmiCbkSwiOmaDmEventReportIndExt, [730](#)
- qaGobiApiAudio.h, [730](#)
  - SLQSGetAudioPathConfig, [730](#)
  - SLQSGetAudioProfile, [731](#)
  - SLQSGetAudioVolTLBConfig, [731](#)
  - SLQSSetAudioPathConfig, [732](#)
  - SLQSSetAudioProfile, [732](#)
  - SLQSSetAudioVolTLBConfig, [733](#)
- qaGobiApiCat.h, [733](#)
  - CATSendEnvelopeCommand, [734](#)
  - CATSendTerminalResponse, [734](#)
- qaGobiApiCbk.h, [735](#)
  - accelAcceptReady, [743](#)
  - accelTempAcceptReady, [744](#)
  - CBK\_ENABLE\_EVENT, [742](#)
  - CBK\_NOCHANGE, [742](#)
  - DEREGISTER\_EVENT, [742](#)
  - DEREGISTER\_SRV, [742](#)
  - device\_state\_enum, [782](#)
  - eDevState, [744](#)
  - eQaQMIService, [782](#)
  - eSMSEventType, [744](#)
  - EVENT\_MASK\_CARD, [742](#)
  - FIRST\_INSTANCE, [742](#)
  - gpsTime, [745](#)
  - gyroAcceptReady, [745](#)
  - gyroTempAcceptReady, [746](#)
  - INVALID\_INSTACNE, [742](#)
  - IPV4, [742](#)
  - IPV4V6, [742](#)
  - IPV6, [742](#)
  - iSLQSSetDUNCallInfoCallback, [782](#)

- iSLQSSetSignalStrengthsCallback, 782
- iSLQSSetWdsFirstInstEventCallback, 782
- iSLQSSetWdsSecondInstEventCallback, 782
- iSLQSSetWdsThirdInstEventCallback, 782
- iSLQSSetWdsXferStatsFirstInstCallback, 782
- iSLQSSetWdsXferStatsSecondInstCallback, 783
- iSetCATEventCallback, 782
- iSetSignalStrengthCallback, 782
- LteNasReleaseInfo, 746
- MAX\_NO\_OF\_CALLS, 742
- MAX\_NO\_OF\_FILES, 742
- MAX\_NO\_OF\_SLOTS, 742
- MAX\_PATH\_LENGTH, 743
- MAXUSSDLENGTH, 743
- modemTempNotification, 746
- NAS\_SRV, 743
- NUM\_OF\_SET, 743
- PDS\_SRV, 743
- packetSrvStatus, 747
- precisionDilution, 749
- REGISTER\_EVENT, 743
- REGISTER\_SRV, 743
- SECOND\_INSTANCE, 743
- SLQSNasNetworkTimeCallBack, 797
- SLQSNasSigInfo2CallBack, 798
- SLQSNasSigInfoCallBack, 798
- SLQSNasSwiOTAMessageCallback, 800
- SLQSNasSysInfoCallBack, 800
- SLQSSetBandPreferenceCbk, 801
- SLQSSetDUNCallInfoCallBack, 802
- SLQSSetDataSystemStatusCallback, 801
- SLQSSetIMSAPdpStatusCallback, 802
- SLQSSetIMSARatStatusCallback, 803
- SLQSSetIMSARegStatusCallback, 803
- SLQSSetIMSASvcStatusCallback, 803
- SLQSSetIMSSMSCConfigCallback, 804
- SLQSSetIMSUserConfigCallback, 804
- SLQSSetIMSVoIPConfigCallback, 805
- SLQSSetModemTempCallback, 805
- SLQSSetPacketSrvStatusCallback, 805
- SLQSSetQosEventCallback, 806
- SLQSSetQosNWStatusCallback, 806
- SLQSSetQosPriEventCallback, 807
- SLQSSetQosStatusCallback, 807
- SLQSSetRegMgrConfigCallback, 808
- SLQSSetSDKTerminatedCallback, 808
- SLQSSetSIPConfigCallback, 810
- SLQSSetSMSEventCallback, 810
- SLQSSetServingSystemCallback, 808
- SLQSSetSessionStateCallback, 809
- SLQSSetSignalStrengthsCallback, 809
- SLQSSetSwiHDRPersCallback, 811
- SLQSSetSysSelectionPrefCallBack, 811
- SLQSSetTransLayerInfoCallback, 811
- SLQSSetTransNWRRegInfoCallback, 813
- SLQSSetWdsEventCallback, 813
- SLQSSetWdsTransferStatisticCallback, 814
- SLQSUIMSetRefreshCallBack, 815
- SLQSUIMSetStatusChangeCallBack, 815
- SLQSVoiceInfoRecCallback, 816
- SLQSVoiceSetAllCallStatusCallBack, 816
- SLQSVoiceSetDTMFEventCallBack, 817
- SLQSVoiceSetOTASPStatusCallBack, 817
- SLQSVoiceSetPrivacyChangeCallBack, 818
- SLQSVoiceSetSUPSCallBack, 818
- SLQSVoiceSetSUPSNotificationCallback, 818
- SLQSWmsAsyncRawSendCallBack, 820
- SLQSWmsMemoryFullCallBack, 820
- SLQSWmsMessageWaitingCallBack, 820
- SMSAsyncRawSend, 750
- SMSCAddressInfo, 751
- SMSEtwsMessageInfo, 751
- SMSEtwsPlmnInfo, 752
- SMSEventInfo, 752
- SMSEventType, 782
- SMSMTMessageInfo, 753
- SMSMessageModelInfo, 753
- SMSOnIMSInfo, 753
- SMSTransferRouteMTMessageInfo, 754
- sensorDataUsage, 749
- sessionInformation, 750
- sessionInformationExt, 750
- SetActivationStatusCallback, 783
- SetCATEventCallback, 783
- SetDataCapabilitiesCallback, 784
- SetDeviceStateChangeCbk, 786
- SetFwDIdCompletionCbk, 786
- SetGPSCallback, 787
- SetLURRejectCallback, 789
- SetLocCradleMountCallback, 787
- SetLocDeleteAssistDataCallback, 787
- SetLocEventPositionCallback, 787
- SetLocEventTimeSyncCallback, 788
- SetLocGnssSvInfoCallback, 788
- SetLocInjectSensorDataCallback, 788
- SetLocInjectTimeCallback, 789
- SetLocOpModeCallback, 789
- SetLocSensorStreamingCallback, 789
- SetMobileIPStatusCallback, 790
- SetNMEACallback, 792
- SetNasLTECphyCalIndCallback, 790
- SetNetChangeCbk, 791
- SetNewSMSCallback, 791
- SetOMADMStateCallback, 792
- SetPDStateCallback, 792
- SetPowerCallback, 793
- SetRFInfoCallback, 793
- SetRMTransferStatisticsCallback, 793
- SetRoamingIndicatorCallback, 795
- SetSLQSOMADMAAlertCallback, 796
- SetSLQSOMADMAAlertCallbackExt, 796
- SetSignalStrengthCallback, 795
- SetUSSDNoWaitIndicationCallback, 797
- SetUSSDNotificationCallback, 796
- SetUSSDReleaseCallback, 797
- svUsedforFix, 754

- SwiOTAMsg, [754](#)
- tFNASwiLTECphyCallInfo, [756](#)
- tFNASwiOTAMsg, [756](#)
- tFNActivationStatus, [755](#)
- tFNAllCallStatus, [756](#)
- tFNAsyncRawSend, [756](#)
- tFNBandPreference, [758](#)
- tFNCATEvent, [760](#)
- tFNDTMFEvent, [762](#)
- tFNDUNCallInfo, [762](#)
- tFNDataCapabilities, [760](#)
- tFNDataSysStatus, [761](#)
- tFNDelAssistData, [761](#)
- tFNDeviceStateChange, [761](#)
- tFNEventPosition, [762](#)
- tFNFwdIdCompletion, [762](#)
- tFNGnssSvInfo, [762](#)
- tFNHDRPersonality, [763](#)
- tFNlmsRegMgrConfig, [764](#)
- tFNlmsSIPConfig, [764](#)
- tFNlmsSMSConfig, [764](#)
- tFNlmsUserConfig, [764](#)
- tFNlmsVoIPConfig, [764](#)
- tFNlmsaPdpStatus, [763](#)
- tFNlmsaRatStatus, [763](#)
- tFNlmsaRegStatus, [763](#)
- tFNlmsaSvcStatus, [763](#)
- tFNInfoRec, [765](#)
- tFNInjectSensorData, [765](#)
- tFNInjectTimeStatus, [765](#)
- tFNLURReject, [765](#)
- tFNMemoryFull, [765](#)
- tFNMessageWaiting, [766](#)
- tFNMobileIPStatus, [766](#)
- tFNModemTempInfo, [766](#)
- tFNNet, [766](#)
- tFNNetworkTime, [767](#)
- tFNNewGPS, [767](#)
- tFNNewNMEA, [767](#)
- tFNNewRMTTransferStatistics, [767](#)
- tFNNewSMS, [768](#)
- tFNOMADMState, [768](#)
- tFNOTASPStatus, [769](#)
- tFNOpMode, [769](#)
- tFNPDSState, [771](#)
- tFNPacketSrvState, [769](#)
- tFNPower, [771](#)
- tFNPrivacyChange, [771](#)
- tFNQosNWStatus, [772](#)
- tFNQosPriEvent, [772](#)
- tFNQosStatus, [772](#)
- tFNRFInfo, [773](#)
- tFNRoamingIndicator, [774](#)
- tFNSDKTerminated, [774](#)
- tFNSLQSSOMADMAAlert, [775](#)
- tFNSLQSQOSEvent, [775](#)
- tFNSLQSSessionState, [777](#)
- tFNSLQSSignalStrengths, [777](#)
- tFNSLQSWDSEvent, [777](#)
- tFNSMSEvents, [777](#)
- tFNSUPSInfo, [777](#)
- tFNSUPSNotification, [778](#)
- tFNSensorStreaming, [774](#)
- tFNServingSystem, [774](#)
- tFNSetCradleMount, [775](#)
- tFNSetEventTimeSync, [775](#)
- tFNSigInfo, [775](#)
- tFNSignalStrength, [775](#)
- tFNSysInfo, [778](#)
- tFNSysSelectionPref, [778](#)
- tFNUIMRefresh, [780](#)
- tFNUIMStatusChangeInfo, [780](#)
- tFNUSSDNoWaitIndication, [780](#)
- tFNUSSDNotification, [780](#)
- tFNUSSDRelease, [780](#)
- tFNtransLayerInfo, [778](#)
- tFNtransNWRegInfo, [778](#)
- THIRD\_INSTANCE, [743](#)
- transLayerNotification, [781](#)
- transNWRegInfoNotification, [781](#)
- USSD\_DCS\_8BIT, [743](#)
- USSD\_DCS\_ASCII, [743](#)
- USSD\_DCS\_UCS2, [743](#)
- VOICE\_SRV, [743](#)
- WDS\_SRV, [743](#)
- qaGobiApiDcs.h, [822](#)
- LEN, [823](#)
- PORTNAM\_LEN, [823](#)
- QCWWAN2kConnect, [823](#)
- QCWWAN2kEnumerateDevices, [823](#)
- QCWWAN2kGetConnectedDeviceID, [825](#)
- QCWWANConnect, [825](#)
- QCWWANDisconnect, [826](#)
- QCWWANEnumerateDevices, [826](#)
- SLQSGetDeviceMode, [827](#)
- SLQSGetNetStatistic, [827](#)
- SLQSGetUsbPortNames, [828](#)
- SLQSKillSDKProcess, [828](#)
- SLQSQosClearMap, [829](#)
- SLQSQosDumpMap, [829](#)
- SLQSQosEditMap, [829](#)
- SLQSQosMap, [831](#)
- SLQSQosReadMap, [831](#)
- SLQSQosUnmap, [831](#)
- SLQSSetLoggingMask, [833](#)
- SLQSStart, [833](#)
- SLQSStart\_AVAgent, [834](#)
- SLQSStartSrv, [834](#)
- SetSDKImagePath, [827](#)
- qaGobiApiDms.h, [835](#)
- ActivateAutomatic, [844](#)
- custFeaturesInfo, [837](#)
- custFeaturesSetting, [840](#)
- dmsCurrentPRLInfo, [841](#)
- ERIFileparams, [842](#)
- GetActivationState, [844](#)

- GetDeviceCapabilities, [845](#)
- GetFirmwareRevision, [846](#)
- GetFirmwareRevisions, [846](#)
- GetHardwareRevision, [848](#)
- GetIMSI, [849](#)
- GetManufacturer, [849](#)
- GetModelID, [850](#)
- GetNetworkTime, [850](#)
- GetOfflineReason, [851](#)
- GetPRLVersion, [852](#)
- GetPower, [851](#)
- GetSerialNumbers, [852](#)
- GetVoiceNumber, [853](#)
- IMGDETAILS\_LEN, [837](#)
- MAX\_CUST\_ID\_LEN, [837](#)
- MAX\_FSN\_LENGTH, [837](#)
- ResetToFactoryDefaults, [854](#)
- SLQSGetBandCapability, [855](#)
- SLQSGetCurrentPRLInfo, [857](#)
- SLQSGetCustFeatures, [857](#)
- SLQSGetCustFeaturesV2, [857](#)
- SLQSGetERIFile, [858](#)
- SLQSGetSerialNumbers, [858](#)
- SLQSSetCustFeatures, [859](#)
- SLQSSetCustFeaturesV2, [859](#)
- SLQSSwiGetCrashAction, [859](#)
- SLQSSwiGetCrashInfo, [860](#)
- SLQSSwiGetFSN, [861](#)
- SLQSSwiGetFirmwareCurr, [861](#)
- SLQSSwiGetFwUpdateStatus, [861](#)
- SLQSSwiGetHostDevInfo, [862](#)
- SLQSSwiGetHostDevInfoParams, [843](#)
- SLQSSwiGetOSInfo, [862](#)
- SLQSSwiGetOSInfoParams, [843](#)
- SLQSSwiGetSerialNoExt, [863](#)
- SLQSSwiGetSerialNoExtParams, [843](#)
- SLQSSwiGetUSBComp, [863](#)
- SLQSSwiSetCrashAction, [864](#)
- SLQSSwiSetHostDevInfo, [864](#)
- SLQSSwiSetHostDevInfoParams, [843](#)
- SLQSSwiSetOSInfo, [865](#)
- SLQSSwiSetOSInfoParams, [844](#)
- SLQSSwiSetUSBComp, [865](#)
- SLQSUIMGetState, [866](#)
- serialNumbersInfo, [842](#)
- SetPower, [854](#)
- UIMChangePIN, [866](#)
- UIMGetControlKeyStatus, [867](#)
- UIMGetICCID, [868](#)
- UIMGetPINStatus, [869](#)
- UIMSetControlKeyProtection, [870](#)
- UIMSetPINProtection, [871](#)
- UIMUnblockControlKey, [872](#)
- UIMUnblockPIN, [873](#)
- UIMVerifyPIN, [874](#)
- UNIQUE\_ID\_LEN, [837](#)
- ValidateSPC, [875](#)
- qaGobiApiFms.h, [875](#)
- DEVICE\_RESET, [878](#)
- DEVICE\_SHUTDOWN, [878](#)
- DeleteStoredImage, [881](#)
- eGetDeviceSeries, [882](#)
- eGobiDeviceSeries, [879](#)
- eGobiImageCarrier, [879](#)
- eGobiImageGPS, [880](#)
- eGobiImageRegion, [881](#)
- eGobiImageTech, [881](#)
- GetImageStore, [882](#)
- GetImagesPreference, [882](#)
- GetStoredImages, [883](#)
- PRI\_UPDATE\_FAIL, [879](#)
- SLQSDownloadFirmwareToSlot, [884](#)
- SLQSGetBootVersionNumber, [885](#)
- SLQSGetFirmwareInfo, [885](#)
- SLQSGetImageInfo, [886](#)
- SLQSGetImageInfo\_9x15, [886](#)
- SLQSGetImageInfoMC77xx, [887](#)
- SLQSGetImageInfoMC83xx, [888](#)
- SLQSIspkgFormatRequired, [888](#)
- SLQSSwiGetAllCarrierImages, [888](#)
- SLQSupgradeFirmware9x15, [889](#)
- SetImagesPreference, [883](#)
- upgrade\_mc77xx\_fw, [890](#)
- UpgradeFirmware2k, [890](#)
- qaGobiApiIms.h, [891](#)
- SLQSGetIMSSMSConfig, [892](#)
- SLQSGetIMSUserConfig, [892](#)
- SLQSGetIMSVoIPConfig, [892](#)
- SLQSGetRegMgrConfig, [893](#)
- SLQSGetSIPConfig, [893](#)
- SLQSImsConfigIndicationRegister, [894](#)
- SLQSSetIMSSMSConfig, [894](#)
- SLQSSetIMSUserConfig, [895](#)
- SLQSSetIMSVoIPConfig, [895](#)
- SLQSSetRegMgrConfig, [896](#)
- SLQSSetSIPConfig, [896](#)
- qaGobiApiImsa.h, [897](#)
- SLQSGetIMSAREgStatus, [898](#)
- SLQSGetIMSAServiceStatus, [899](#)
- SLQSGetIMSASupportedFields, [899](#)
- SLQSGetIMSASupportedMsg, [900](#)
- SLQSRegisterIMSASIndication, [900](#)
- qaGobiApiLoc.h, [901](#)
- SLQSLOCDeIAssData, [902](#)
- SLQSLOCEventRegister, [902](#)
- SLQSLOCSetExtPowerState, [902](#)
- SLQSLOCSetOpMode, [903](#)
- SLQSLOCStart, [903](#)
- SLQSLOCStop, [904](#)
- qaGobiApiNas.h, [904](#)
- GetACCOLC, [918](#)
- GetANAAAAAuthenticationStatus, [919](#)
- GetCDMANetworkParameters, [919](#)
- GetHomeNetwork, [921](#)
- GetHomeNetwork3GPP2, [924](#)
- GetNetworkPreference, [926](#)

- GetRfInfo, [927](#)
- GetServingNetwork, [928](#)
- GetServingNetworkCapabilities, [930](#)
- GetSignalStrengths, [930](#)
- IMSI\_M\_S1\_LENGTH, [909](#)
- IMSI\_M\_S2\_LENGTH, [909](#)
- InitiateDomainAttach, [932](#)
- InitiateNetworkRegistration, [932](#)
- MAX\_PILOT\_SETS, [909](#)
- NAM\_NAME\_LENGTH, [909](#)
- PLMN\_LENGTH, [909](#)
- PerformNetworkScan, [934](#)
- SLQSConfigSigInfo, [938](#)
- SLQSGetErrorRate, [938](#)
- SLQSGetOperatorNameData, [938](#)
- SLQSGetPLMNName, [939](#)
- SLQSGetServingSystem, [939](#)
- SLQSGetSignalStrength, [940](#)
- SLQSGetSysSelectionPref, [940](#)
- SLQSIInitiateNetworkRegistration, [941](#)
- SLQSNASGetLTECPHYCaInfo, [944](#)
- SLQSNasConfigSigInfo2, [941](#)
- SLQSNasGet3GPP2Subscription, [941](#)
- SLQSNasGetCellLocationInfo, [943](#)
- SLQSNasGetHDRColorCode, [943](#)
- SLQSNasGetSigInfo, [944](#)
- SLQSNasGetSysInfo, [944](#)
- SLQSNasGetTxRxInfo, [945](#)
- SLQSNasIndicationRegister, [945](#)
- SLQSNasIndicationRegisterExt, [947](#)
- SLQSNasIndicationRegisterLTECphyCa, [948](#)
- SLQSNasSwiIndicationRegister, [948](#)
- SLQSNasSwiModemStatus, [949](#)
- SLQSPerformNetworkScan, [949](#)
- SLQSSetBandPreference, [950](#)
- SLQSSetSysSelectionPref, [952](#)
- SLQSSwiGetHDRPersonality, [952](#)
- SLQSSwiGetHDRProtSubtype, [952](#)
- SLQSSwiGetHRPDStats, [953](#)
- SLQSSwiGetLteCQI, [953](#)
- SLQSSwiNetworkDebug, [954](#)
- SLQSSwiPSDetach, [954](#)
- SetACCOLC, [935](#)
- SetCDMANetworkParameters, [935](#)
- SetNetworkPreference, [937](#)
- slqsNetworkScanInfo, [909](#)
- sysSelectPrefInfo, [911](#)
- sysSelectPrefParams, [914](#)
- UATISIZE, [909](#)
- qaGobiApiOmadm.h, [954](#)
  - OMADMCancelSession, [955](#)
  - OMADMGetPendingNIA, [955](#)
  - OMADMGetSessionInfo, [956](#)
  - OMADMStartSession, [957](#)
- qaGobiApiPds.h, [958](#)
  - DEFAULTBYTEVALUE, [959](#)
  - DEFAULTLONGVALUE, [959](#)
  - DEFAULTWORDVALUE, [959](#)
  - ForceXTRADownload, [959](#)
  - GetPDSDDefaults, [960](#)
  - GetPDSState, [960](#)
  - GetPortAutomaticTracking, [961](#)
  - GetServiceAutomaticTracking, [961](#)
  - GetXTRAAutomaticDownload, [963](#)
  - GetXTRANetwork, [963](#)
  - GetXTRAValidity, [964](#)
  - PDSInjectTimeReference, [964](#)
  - ResetPDSData, [965](#)
  - SLQSGetAGPSConfig, [969](#)
  - SLQSGetGPSSStateInfo, [970](#)
  - SLQSPDSDeterminePosition, [970](#)
  - SLQSPDSInjectAbsoluteTimeReference, [971](#)
  - SLQSPDSInjectPositionData, [971](#)
  - SLQSSetAGPSConfig, [972](#)
  - SLQSSetPositionMethodState, [972](#)
  - SetPDSDDefaults, [966](#)
  - SetPDSState, [967](#)
  - SetPortAutomaticTracking, [967](#)
  - SetServiceAutomaticTracking, [968](#)
  - SetXTRAAutomaticDownload, [968](#)
  - SetXTRANetwork, [969](#)
  - StartPDSTrackingSessionExt, [974](#)
  - StopPDSTrackingSession, [975](#)
- qaGobiApiQos.h, [975](#)
  - SLSQosGetFlowStatus, [976](#)
  - SLSQosGetGranted, [977](#)
  - SLSQosGetNWProf, [978](#)
  - SLSQosGetNetworkStatus, [977](#)
  - SLSQosModify, [978](#)
  - SLSQosRel, [979](#)
  - SLSQosReq, [979](#)
  - SLSQosReset, [980](#)
  - SLSQosResume, [980](#)
  - SLSQosSuspend, [981](#)
  - SLSQosSwiReadApnExtraParams, [981](#)
  - SLSQosSwiReadDataStats, [982](#)
- qaGobiApiRms.h, [982](#)
  - GetSMSWake, [982](#)
  - SetSMSWake, [983](#)
- qaGobiApiSar.h, [984](#)
  - eQMISARRFState, [984](#)
  - SLQSGetRfSarState, [985](#)
  - SLQSSetRfSarState, [985](#)
- qaGobiApiSms.h, [986](#)
  - ABSOLUTE\_VALIDITY, [988](#)
  - CONFIG\_LEN, [988](#)
  - getIndicationRegResp, [988](#)
  - getSMSCAddress, [993](#)
  - getTransLayerInfoResp, [989](#)
  - getTransNWRegInfoResp, [990](#)
  - MAX\_SMS\_ROUTES, [988](#)
  - NUM\_OF\_SET, [988](#)
  - qaQmi3GPP2BroadcastCfgInfo, [990](#)
  - qaQmi3GPPBroadcastCfgInfo, [991](#)
  - SLQSCDMADecodeMTTextMsg, [995](#)
  - SLQSCDMAEncodeMOTextMsg, [996](#)



- SLQSDelSms, 996
- SLQSGetIndicationRegister, 997
- SLQSGetMessageWaiting, 998
- SLQSGetSMS, 998
- SLQSGetSMSList, 1000
- SLQSGetSmsBroadcastConfig, 999
- SLQSGetTransLayerInfo, 1001
- SLQSGetTransNWRegInfo, 1001
- SLQSModifySMSStatus, 1002
- SLQSSendAsyncSMS, 1003
- SLQSSendLongSMS, 1003
- SLQSSendSMS, 1004
- SLQSSetIndicationRegister, 1005
- SLQSSetSmsBroadcastActivation, 1005
- SLQSSetSmsBroadcastConfig, 1006
- SLQSSetSmsStorage, 1006
- SLQSSmsGetMaxStorageSize, 1007
- SLQSSmsGetMessageProtocol, 1007
- SLQSSmsSetRoutes, 1008
- SLQSSwiGetSMSStorage, 1008
- SLQSWCDMADecodeLongTextMsg, 1009
- SLQSWCDMADecodeMTTextMsg, 1009
- SLQSWCDMAEncodeMOTextMsg, 1010
- SaveSMS, 993
- SendSMS, 994
- setIndicationRegReq, 991
- SetSMSCAddress, 995
- TIME\_DATE\_BUF, 988
- TIME\_STAMP\_BUF, 988
- transLayerInfo, 992
- qaGobiApiSwi.h, 1010
  - SLQSGetPidof, 1011
  - SLQSGetSdkVersion, 1011
  - SLQSSendRawQMI, 1011
- qaGobiApiSwiAudio.h, 1011
  - SLQSGetM2MAVMMute, 1013
  - SLQSGetM2MAudioProfile, 1012
  - SLQSGetM2MAudioVolume, 1013
  - SLQSGetM2MSpkrGain, 1014
  - SLQSSetM2MAVMMute, 1016
  - SLQSSetM2MAudioAVCFG, 1014
  - SLQSSetM2MAudioLPBK, 1015
  - SLQSSetM2MAudioNVDef, 1015
  - SLQSSetM2MAudioProfile, 1015
  - SLQSSetM2MAudioVolume, 1016
  - SLQSSetM2MSpkrGain, 1017
- qaGobiApiSwiOmadms.h, 1017
  - SLQSOMADMCancelSession, 1023
  - SLQSOMADMGetSessionInfo, 1023
  - SLQSOMADMGetSettings, 1024
  - SLQSOMADMGetSettings2, 1025
  - SLQSOMADMGetSelection, 1025
  - SLQSOMADMGetSelection2, 1026
  - SLQSOMADMSessionInfo, 1018
  - SLQSOMADMSetSettings, 1026
  - SLQSOMADMSetSettings2, 1027
  - SLQSOMADMSetSettings3, 1027
  - SLQSOMADMSettings, 1020
  - SLQSOMADMSettingsReqParams, 1021
  - SLQSOMADMSettingsReqParams3, 1022
  - SLQSOMADMStartSession, 1028
  - SLQSOMADMStartSession2, 1028
- qaGobiApiTableBandClasses.h, 1029
- qaGobiApiTableCallControlReturnReasons.h, 1032
- qaGobiApiTableCallEndReasons.h, 1033
- qaGobiApiTableCarrierCodes.h, 1048
- qaGobiApiTableCodingScheme.h, 1050
- qaGobiApiTableGpsCapabilityCodes.h, 1052
- qaGobiApiTablePowerModes.h, 1053
- qaGobiApiTableRadioInterfaces.h, 1053
- qaGobiApiTableRegionCodes.h, 1054
- qaGobiApiTableServiceOptions.h, 1054
- qaGobiApiTableSupServiceInfoClasses.h, 1057
- qaGobiApiTableSwtAudio.h, 1057
- qaGobiApiTableSwtOMADMUpdateCompleteStatus.h, 1058
- qaGobiApiTableVoiceCallEndReasons.h, 1059
- qaGobiApiUim.h, 1066
  - MAX\_NO\_OF\_SLOTS, 1067
  - MAX\_PATH\_LENGTH, 1067
  - MAX\_PUK\_LENGTH, 1067
  - SLQSUIMAuthenticate, 1068
  - SLQSUIMChangePin, 1068
  - SLQSUIMDepersonalization, 1069
  - SLQSUIMEventRegister, 1069
  - SLQSUIMGetCardStatus, 1070
  - SLQSUIMGetFileAttributes, 1070
  - SLQSUIMPowerDown, 1071
  - SLQSUIMPowerUp, 1071
  - SLQSUIMReadTransparent, 1072
  - SLQSUIMRefreshComplete, 1072
  - SLQSUIMRefreshGetLastEvent, 1073
  - SLQSUIMRefreshOK, 1073
  - SLQSUIMRefreshRegister, 1074
  - SLQSUIMReset, 1074
  - SLQSUIMSetPinProtection, 1075
  - SLQSUIMUnblockPin, 1075
  - SLQSUIMVerifyPin, 1076
- qaGobiApiVoice.h, 1076
  - AnswerUSSD, 1080
  - CancelUSSD, 1080
  - MAX\_CALL\_NO\_LEN, 1079
  - MAX\_NO\_OF\_CALLS, 1079
  - MAXUSSDLENGTH, 1079
  - OriginateUSSD, 1081
  - PASSWORD\_LENGTH, 1080
  - SLQSOriginateUSSD, 1081
  - SLQSVoiceALSSelectLine, 1082
  - SLQSVoiceALSSetLineSwitching, 1082
  - SLQSVoiceAnswerCall, 1083
  - SLQSVoiceBindSubscription, 1083
  - SLQSVoiceBurstDTMF, 1084
  - SLQSVoiceDialCall, 1084
  - SLQSVoiceEndCall, 1085
  - SLQSVoiceGetAllCallInfo, 1085
  - SLQSVoiceGetCLIP, 1089

- SLQSVoiceGetCLIR, [1090](#)
- SLQSVoiceGetCNAP, [1090](#)
- SLQSVoiceGetCOLP, [1091](#)
- SLQSVoiceGetCOLR, [1091](#)
- SLQSVoiceGetCallBarring, [1086](#)
- SLQSVoiceGetCallForwardingStatus, [1086](#)
- SLQSVoiceGetCallInfo, [1088](#)
- SLQSVoiceGetCallWaiting, [1089](#)
- SLQSVoiceGetConfig, [1092](#)
- SLQSVoiceIndicationRegister, [1092](#)
- SLQSVoiceManageCalls, [1093](#)
- SLQSVoiceOrigUSSDNoWait, [1093](#)
- SLQSVoiceSendFlash, [1095](#)
- SLQSVoiceSetCallBarringPassword, [1095](#)
- SLQSVoiceSetConfig, [1096](#)
- SLQSVoiceSetPreferredPrivacy, [1097](#)
- SLQSVoiceSetSUPSService, [1097](#)
- SLQSVoiceStartContDTMF, [1098](#)
- SLQSVoiceStopContDTMF, [1098](#)
- serviceClassInformation, [1080](#)
- qaGobiApiWds.h, [1099](#)
  - GetAutoconnect, [1108](#)
  - GetByteTotals, [1109](#)
  - GetConnectionRate, [1109](#)
  - GetDataBearerTechnology, [1110](#)
  - GetDefaultProfile, [1111](#)
  - GetDefaultProfileLTE, [1113](#)
  - GetDormancyState, [1115](#)
  - GetIPAddressLTE, [1116](#)
  - GetLastMobileIPError, [1116](#)
  - GetMobileIP, [1117](#)
  - GetMobileIPProfile, [1117](#)
  - GetPacketStatistics, [1119](#)
  - GetPacketStatus, [1120](#)
  - GetProfileSettingIn, [1103](#)
  - GetProfileSettingOut, [1103](#)
  - GetSessionDuration, [1120](#)
  - GetSessionState, [1121](#)
  - iGetByteTotals, [1121](#)
  - iGetConnectionRate, [1122](#)
  - iGetPacketStatistics, [1122](#)
  - iSLQSMISetIPFamilyPreference, [1122](#)
  - qmiDataBearerMasks, [1108](#)
  - QmiProfileInfo, [1103](#)
  - QmiWSDDataBearerTechnology, [1105](#)
  - QmiWSDDataBearers, [1103](#)
  - RMSetTransferStatistics, [1122](#)
  - SLQSAutoConnect, [1132](#)
  - SLQSCreateProfile, [1132](#)
  - SLQSDeleteProfile, [1133](#)
  - SLQSGet3GPPConfigItem, [1134](#)
  - SLQSGetByteTotals, [1134](#)
  - SLQSGetConnectionRate, [1134](#)
  - SLQSGetCurrDataSystemStat, [1135](#)
  - SLQSGetCurrentChannelRate, [1135](#)
  - SLQSGetDUNCallInfo, [1137](#)
  - SLQSGetDataBearerTechnology, [1136](#)
  - SLQSGetDataBearerTechnologyExt, [1136](#)
  - SLQSGetPacketStatistics, [1137](#)
  - SLQSGetProfile, [1137](#)
  - SLQSGetProfileSettings, [1140](#)
  - SLQSGetRuntimeSettings, [1141](#)
  - SLQSGetSessionState, [1141](#)
  - SLQSModifyProfile, [1142](#)
  - SLQSResetPacketStatics, [1143](#)
  - SLQSSetLoopback, [1146](#)
  - SLQSSetLoopback, [1146](#)
  - SLQSSet3GPPConfigItem, [1143](#)
  - SLQSSetHostMTU, [1143](#)
  - SLQSSetProfile, [1144](#)
  - SLQSSetStopDataSession, [1146](#)
  - SLQSWdsGoActive, [1148](#)
  - SLQSWdsGoDormant, [1148](#)
  - SLQSWdsSetEventReport, [1149](#)
  - SLQSWdsSwiPDPRuntimeSettings, [1149](#)
  - SetActiveMobileIPProfile, [1122](#)
  - SetAutoconnect, [1122](#)
  - SetDefaultProfile, [1124](#)
  - SetDefaultProfileLTE, [1125](#)
  - SetDefaultProfileLTEV2, [1127](#)
  - SetMobileIP, [1129](#)
  - SetMobileIPParameters, [1130](#)
  - SetMobileIPProfile, [1131](#)
  - slqs3GPPConfigItem, [1107](#)
  - WDS\_IsGobiDevice, [1150](#)
- qaNasGetRFBandInfo.h, [1150](#)
  - PkQmiNasGetRFBandInfo, [1150](#)
  - UpkQmiNasGetRFBandInfo, [1150](#)
- qaNasPerformNetworkScan.h, [1150](#)
  - FORBIDDEN\_INDEX, [1151](#)
  - INDEX\_ZERO, [1151](#)
  - PREFERRED\_INDEX, [1151](#)
  - PkQmiNasPerformNetworkScan, [1151](#)
  - ROAMING\_INDEX, [1151](#)
  - UpkQmiNasPerformNetworkScan, [1151](#)
- qaQmi3GPP2BroadcastCfgInfo
  - qaGobiApiSms.h, [990](#)
- qaQmi3GPPBroadcastCfgInfo
  - qaGobiApiSms.h, [991](#)
- qaQmi3Gpp2TimeZone, [422](#)
  - daylightSavings, [422](#)
  - leapSeconds, [422](#)
  - localTimeOffset, [422](#)
- qaQmiInterfaceInfo, [422](#)
  - qaQmiinstanceid, [423](#)
  - qaQmisvctype, [423](#)
  - v4sessionId, [423](#)
  - v6sessionId, [423](#)
- qaQmiServingSystemParam, [423](#)
  - BasestationID, [426](#)
  - BasestationLatitude, [426](#)
  - BasestationLongitude, [426](#)
  - CDMA\_P\_Rev, [426](#)
  - CDMASystemInfoExt, [427](#)
  - CallBarStatus, [426](#)
  - CellID, [427](#)

- concSvcInfo, [427](#)
- CurrentPLMN, [427](#)
- DTMInd, [427](#)
- DataSrvCapabilities, [427](#)
- defaultRoamInd, [427](#)
- DetailedSvcInfo, [427](#)
- Gpp2TimeZone, [427](#)
- GppNetworkDSTAdjustment, [427](#)
- GppTimeZone, [427](#)
- hdrPersonality, [427](#)
- Lac, [427](#)
- NetworkID, [427](#)
- PRLInd, [427](#)
- roamIndicatorVal, [427](#)
- RoamingIndicatorList, [427](#)
- ServingSystem, [427](#)
- SystemID, [427](#)
- trackAreaCode, [427](#)
- qaQmiinstanceid
  - qaQmiInterfaceInfo, [423](#)
- qaQmisvctype
  - qaQmiInterfaceInfo, [423](#)
- qm\_wds\_ds\_profile\_extended\_err\_codes
  - qmerrno.h, [1157](#)
- qmerrno.h
  - eQCWWAN\_ERR\_API\_MUTEX\_TIMEOUT, [1154](#)
  - eQCWWAN\_ERR\_BUFFER\_SZ, [1153](#)
  - eQCWWAN\_ERR\_CANCEL\_OP, [1154](#)
  - eQCWWAN\_ERR\_DRIVER, [1154](#)
  - eQCWWAN\_ERR\_ENUM\_BEGIN, [1153](#)
  - eQCWWAN\_ERR\_ENUM\_END, [1154](#)
  - eQCWWAN\_ERR\_FILE\_COPY, [1153](#)
  - eQCWWAN\_ERR\_FILE\_OPEN, [1153](#)
  - eQCWWAN\_ERR\_GENERAL, [1153](#)
  - eQCWWAN\_ERR\_INTERNAL, [1153](#)
  - eQCWWAN\_ERR\_INVALID\_ARG, [1153](#)
  - eQCWWAN\_ERR\_INVALID\_DEVID, [1153](#)
  - eQCWWAN\_ERR\_INVALID\_FILE, [1153](#)
  - eQCWWAN\_ERR\_INVALID\_QMI\_RSP, [1153](#)
  - eQCWWAN\_ERR\_MALFORMED\_QMI\_RSP, [1153](#)
  - eQCWWAN\_ERR\_MEMORY, [1153](#)
  - eQCWWAN\_ERR\_MULTIPLE\_DEVICES, [1154](#)
  - eQCWWAN\_ERR\_NO\_CANCELABLE\_OP, [1154](#)
  - eQCWWAN\_ERR\_NO\_CONNECTION, [1153](#)
  - eQCWWAN\_ERR\_NO\_DEVICE, [1153](#)
  - eQCWWAN\_ERR\_NO\_SIGNAL, [1154](#)
  - eQCWWAN\_ERR\_NONE, [1153](#)
  - eQCWWAN\_ERR\_NULL\_TLV, [1157](#)
  - eQCWWAN\_ERR\_OFFLINE, [1154](#)
  - eQCWWAN\_ERR\_PDU\_GENERATION, [1154](#)
  - eQCWWAN\_ERR\_QMI\_ABORTED, [1154](#)
  - eQCWWAN\_ERR\_QMI\_ACCESS\_DENIED, [1156](#)
  - eQCWWAN\_ERR\_QMI\_ACK\_NOT\_SENT, [1156](#)
  - eQCWWAN\_ERR\_QMI\_ARG\_TOO\_LONG, [1154](#)
  - eQCWWAN\_ERR\_QMI\_AUTHENTICATION\_FAILED, [1155](#)
  - eQCWWAN\_ERR\_QMI\_AUTHENTICATION\_LOCK, [1155](#)
  - eQCWWAN\_ERR\_QMI\_BUNDLING\_NOT\_SUPPORTED, [1156](#)
  - eQCWWAN\_ERR\_QMI\_CALL\_FAILED, [1154](#)
  - eQCWWAN\_ERR\_QMI\_CARD\_BUSY\_RSP, [1157](#)
  - eQCWWAN\_ERR\_QMI\_CARD\_CALL\_CONTROL\_FAILED, [1156](#)
  - eQCWWAN\_ERR\_QMI\_CAT\_END, [1157](#)
  - eQCWWAN\_ERR\_QMI\_CAT\_START, [1157](#)
  - eQCWWAN\_ERR\_QMI\_CAUSE\_CODE, [1155](#)
  - eQCWWAN\_ERR\_QMI\_CLIENT\_IDS\_EXHAUSTED, [1154](#)
  - eQCWWAN\_ERR\_QMI\_CONNECT, [1153](#)
  - eQCWWAN\_ERR\_QMI\_DEVICE\_IN\_USE, [1154](#)
  - eQCWWAN\_ERR\_QMI\_DEVICE\_NOT\_READY, [1155](#)
  - eQCWWAN\_ERR\_QMI\_DEVICE\_STORAGE\_FULL, [1155](#)
  - eQCWWAN\_ERR\_QMI\_DISABLED, [1155](#)
  - eQCWWAN\_ERR\_QMI\_ENCODING, [1155](#)
  - eQCWWAN\_ERR\_QMI\_ENVELOPE\_CMD\_FAILURE, [1157](#)
  - eQCWWAN\_ERR\_QMI\_EVENT\_REG\_FAILED, [1157](#)
  - eQCWWAN\_ERR\_QMI\_EXTENDED\_INTERNAL, [1156](#)
  - eQCWWAN\_ERR\_QMI\_FDN\_RESTRICT, [1156](#)
  - eQCWWAN\_ERR\_QMI\_FLOW\_SUSPENDED, [1155](#)
  - eQCWWAN\_ERR\_QMI\_GENERAL, [1155](#)
  - eQCWWAN\_ERR\_QMI\_HARDWARE\_RESTRICTED, [1156](#)
  - eQCWWAN\_ERR\_QMI\_IFACE, [1153](#)
  - eQCWWAN\_ERR\_QMI\_INCOMPATIBLE\_STATE, [1156](#)
  - eQCWWAN\_ERR\_QMI\_INCORRECT\_FLOW\_FILTER, [1155](#)
  - eQCWWAN\_ERR\_QMI\_INCORRECT\_PIN, [1154](#)
  - eQCWWAN\_ERR\_QMI\_INFO\_UNAVAILABLE, [1156](#)
  - eQCWWAN\_ERR\_QMI\_INJECT\_TIMEOUT, [1156](#)
  - eQCWWAN\_ERR\_QMI\_INSUFFICIENT\_RESOURCES, [1155](#)
  - eQCWWAN\_ERR\_QMI\_INTERFACE\_NOT\_FOUND, [1155](#)
  - eQCWWAN\_ERR\_QMI\_INTERNAL, [1154](#)
  - eQCWWAN\_ERR\_QMI\_INVALID\_ARG, [1155](#)
  - eQCWWAN\_ERR\_QMI\_INVALID\_CLIENT\_ID, [1154](#)
  - eQCWWAN\_ERR\_QMI\_INVALID\_DATA\_FORMAT, [1155](#)
  - eQCWWAN\_ERR\_QMI\_INVALID\_ENVELOPE\_CMD, [1157](#)
  - eQCWWAN\_ERR\_QMI\_INVALID\_HANDLE, [1154](#)
  - eQCWWAN\_ERR\_QMI\_INVALID\_ID, [1155](#)
  - eQCWWAN\_ERR\_QMI\_INVALID\_INDEX, [1155](#)



- eQCWWAN\_ERR\_QMI\_INVALID\_IP\_FAMILY\_P-REF, [1155](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_MCAST\_HAN-DLE, [1155](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_MESSAGE\_ID, [1155](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_OPERATION, [1155](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_PDP\_TYPE, [1154](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_PINID, [1154](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_PROFILE, [1154](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_PROFILE\_TY-PE, [1154](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_PS\_ATTACH\_-ACTION, [1154](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_QMI\_CMD, [1155](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_QOS\_ID, [1155](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_REGISTER\_A-CTION, [1154](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_SERVICE\_TY-PE, [1154](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_TECH\_PREF, [1154](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_TERMINAL\_R-SP, [1157](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_TRANSITION, [1155](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_TX\_ID, [1154](#)
- eQCWWAN\_ERR\_QMI\_MALFORMED\_MSG, [1154](#)
- eQCWWAN\_ERR\_QMI\_MAX, [1156](#)
- eQCWWAN\_ERR\_QMI\_MAX\_MCAST\_REQUES-TS\_IN\_USE, [1155](#)
- eQCWWAN\_ERR\_QMI\_MAX\_QOS\_REQUESTS-\_IN\_USE, [1155](#)
- eQCWWAN\_ERR\_QMI\_MESSAGE\_DELIVERY\_-FAILURE, [1155](#)
- eQCWWAN\_ERR\_QMI\_MESSAGE\_NOT\_SENT, [1155](#)
- eQCWWAN\_ERR\_QMI\_MISSING\_ARG, [1154](#)
- eQCWWAN\_ERR\_QMI\_MSG\_BLOCKED, [1156](#)
- eQCWWAN\_ERR\_QMI\_NETWORK\_ABORTED, [1156](#)
- eQCWWAN\_ERR\_QMI\_NETWORK\_NOT\_READ-Y, [1155](#)
- eQCWWAN\_ERR\_QMI\_NETWORK\_QOS\_UNA-WARE, [1155](#)
- eQCWWAN\_ERR\_QMI\_NO\_EFFECT, [1154](#)
- eQCWWAN\_ERR\_QMI\_NO\_ENTRY, [1155](#)
- eQCWWAN\_ERR\_QMI\_NO\_FREE\_PROFILE, [1154](#)
- eQCWWAN\_ERR\_QMI\_NO\_MEMORY, [1154](#)
- eQCWWAN\_ERR\_QMI\_NO\_NETWORK\_FOUN-D, [1154](#)
- eQCWWAN\_ERR\_QMI\_NO\_RADIO, [1156](#)
- eQCWWAN\_ERR\_QMI\_NO\_SUBSCRIPTION, [1156](#)
- eQCWWAN\_ERR\_QMI\_NO\_THRESHOLDS, [1154](#)
- eQCWWAN\_ERR\_QMI\_NOT\_A\_MCAST\_IFACE, [1155](#)
- eQCWWAN\_ERR\_QMI\_NOT\_PROVISIONED, [1154](#)
- eQCWWAN\_ERR\_QMI\_NOT\_SUPPORTED, [1156](#)
- eQCWWAN\_ERR\_QMI\_OFFSET, [1154](#)
- eQCWWAN\_ERR\_QMI\_OP\_DEVICE\_UNSUPP-ORTED, [1154](#)
- eQCWWAN\_ERR\_QMI\_OP\_NETWORK\_UNSUP-PORTED, [1154](#)
- eQCWWAN\_ERR\_QMI\_OP\_PARTIAL\_FAILURE, [1156](#)
- eQCWWAN\_ERR\_QMI\_OUT\_OF\_CALL, [1154](#)
- eQCWWAN\_ERR\_QMI\_PIN\_BLOCKED, [1155](#)
- eQCWWAN\_ERR\_QMI\_PIN\_PERM\_BLOCKED, [1155](#)
- eQCWWAN\_ERR\_QMI\_POLICY\_MISMATCH, [1156](#)
- eQCWWAN\_ERR\_QMI\_REQ, [1153](#)
- eQCWWAN\_ERR\_QMI\_REQ\_SCH, [1153](#)
- eQCWWAN\_ERR\_QMI\_REQ\_TO, [1153](#)
- eQCWWAN\_ERR\_QMI\_REQUESTED\_NUM\_UN-SUPPORTED, [1155](#)
- eQCWWAN\_ERR\_QMI\_RSP, [1153](#)
- eQCWWAN\_ERR\_QMI\_RSP\_TO, [1153](#)
- eQCWWAN\_ERR\_QMI\_SEGMENT\_ORDER, [1156](#)
- eQCWWAN\_ERR\_QMI\_SEGMENT\_TOO\_LONG, [1156](#)
- eQCWWAN\_ERR\_QMI\_SESSION\_INACTIVE, [1155](#)
- eQCWWAN\_ERR\_QMI\_SESSION\_INVALID, [1155](#)
- eQCWWAN\_ERR\_QMI\_SESSION\_OWNERSHIP, [1155](#)
- eQCWWAN\_ERR\_QMI\_SIM\_FILE\_NOT\_FOUND, [1156](#)
- eQCWWAN\_ERR\_QMI\_SIM\_NOT\_INITIALIZED, [1155](#)
- eQCWWAN\_ERR\_QMI\_SMSC\_ADDR, [1156](#)
- eQCWWAN\_ERR\_QMI\_SUPS\_FAILURE\_CAUS-E, [1156](#)
- eQCWWAN\_ERR\_QMI\_TPDU\_TYPE, [1156](#)
- eQCWWAN\_ERR\_QMI\_UNABORTABLE\_TRAN-SACTION, [1154](#)
- eQCWWAN\_ERR\_QMI\_UNKNOWN, [1155](#)
- eQCWWAN\_ERR\_QMI\_WIDTH, [1157](#)
- eQCWWAN\_ERR\_RESET, [1154](#)
- eQCWWAN\_ERR\_SWICM\_AM\_VERS\_ERROR, [1156](#)
- eQCWWAN\_ERR\_SWICM\_CALL\_IN\_PROGRES-S, [1156](#)
- eQCWWAN\_ERR\_SWICM\_END, [1156](#)

- eQCWWAN\_ERR\_SWICM\_FAILED\_TO\_KILL\_SDK\_PROCESS, [1156](#)
- eQCWWAN\_ERR\_SWICM\_INVALID\_SESSION\_ID, [1156](#)
- eQCWWAN\_ERR\_SWICM\_INVALID\_V4\_SESSION\_ID, [1156](#)
- eQCWWAN\_ERR\_SWICM\_INVALID\_V6\_SESSION\_ID, [1156](#)
- eQCWWAN\_ERR\_SWICM\_NOT\_IMPLEMENTED, [1156](#)
- eQCWWAN\_ERR\_SWICM\_QMI\_CLNT\_NOT\_SUPPORTED, [1156](#)
- eQCWWAN\_ERR\_SWICM\_QMI\_SVC\_NOT\_SUPPORTED, [1156](#)
- eQCWWAN\_ERR\_SWICM\_SM\_NO\_AVAILABLE\_SESSIONS, [1156](#)
- eQCWWAN\_ERR\_SWICM\_SOCKET\_IN\_USE, [1156](#)
- eQCWWAN\_ERR\_SWICM\_START, [1156](#)
- eQCWWAN\_ERR\_SWICM\_TIMEOUT, [1156](#)
- eQCWWAN\_ERR\_SWICM\_V4DWN\_V6DWN, [1156](#)
- eQCWWAN\_ERR\_SWICM\_V4DWN\_V6UP, [1156](#)
- eQCWWAN\_ERR\_SWICM\_V4UP\_V6DWN, [1156](#)
- eQCWWAN\_ERR\_SWICM\_V4UP\_V6UP, [1156](#)
- eQCWWAN\_ERR\_SWIDCS\_APP\_DISCONNECTED, [1157](#)
- eQCWWAN\_ERR\_SWIDCS\_DEVNODE\_NOT\_FOUND, [1157](#)
- eQCWWAN\_ERR\_SWIDCS\_END, [1157](#)
- eQCWWAN\_ERR\_SWIDCS\_FILEIO\_ERR, [1157](#)
- eQCWWAN\_ERR\_SWIDCS\_IOCTL\_ERR, [1157](#)
- eQCWWAN\_ERR\_SWIDCS\_START, [1157](#)
- eQCWWAN\_ERR\_SWIIM\_CORRUPTED\_FIRMWARE, [1157](#)
- eQCWWAN\_ERR\_SWIIM\_END, [1157](#)
- eQCWWAN\_ERR\_SWIIM\_FILE\_NOT\_FOUND, [1157](#)
- eQCWWAN\_ERR\_SWIIM\_FIRMWARE\_NOT\_DOWNLOADED, [1157](#)
- eQCWWAN\_ERR\_SWIIM\_FW\_PREFERENCE\_MISMATCH, [1157](#)
- eQCWWAN\_ERR\_SWIIM\_FW\_UPDATE\_FAIL, [1157](#)
- eQCWWAN\_ERR\_SWIIM\_FW\_UPDATE\_SUCCESS, [1157](#)
- eQCWWAN\_ERR\_SWIIM\_INVALID\_PATH, [1157](#)
- eQCWWAN\_ERR\_SWIIM\_OPENING\_DIR, [1157](#)
- eQCWWAN\_ERR\_SWIIM\_OPENING\_FILE, [1157](#)
- eQCWWAN\_ERR\_SWIIM\_START, [1157](#)
- eQCWWAN\_ERR\_SWISM\_END, [1157](#)
- eQCWWAN\_ERR\_SWISMS\_BEARER\_DATA\_NOT\_FOUND, [1157](#)
- eQCWWAN\_ERR\_SWISMS\_MSG\_CORRUPTED, [1157](#)
- eQCWWAN\_ERR\_SWISMS\_MSG\_LEN\_TOO\_LONG, [1157](#)
- eQCWWAN\_ERR\_SWISMS\_SMSC\_NUM\_CORRUPTED, [1157](#)
- eQCWWAN\_ERR\_SWISMS\_START, [1157](#)
- eWDS\_ERR\_PROFILE\_REG\_3GPP2\_ERR\_INVALID\_IDENT\_FOR\_PROFILE, [1158](#)
- eWDS\_ERR\_PROFILE\_REG\_3GPP\_ACCESS\_ERR, [1158](#)
- eWDS\_ERR\_PROFILE\_REG\_3GPP\_CONTEXT\_NOT\_DEFINED, [1158](#)
- eWDS\_ERR\_PROFILE\_REG\_3GPP\_ERR\_OUT\_OF\_PROFILES, [1158](#)
- eWDS\_ERR\_PROFILE\_REG\_3GPP\_INVALID\_PROFILE\_FAMILY, [1158](#)
- eWDS\_ERR\_PROFILE\_REG\_3GPP\_READ\_ONLY\_FLAG\_SET, [1158](#)
- eWDS\_ERR\_PROFILE\_REG\_3GPP\_VALID\_FLAG\_NOT\_SET, [1158](#)
- eWDS\_ERR\_PROFILE\_REG\_END, [1158](#)
- eWDS\_ERR\_PROFILE\_REG\_INVALID\_PROFILE\_FAMILY, [1158](#)
- eWDS\_ERR\_PROFILE\_REG\_RESULT\_ERR\_INVALID, [1158](#)
- eWDS\_ERR\_PROFILE\_REG\_RESULT\_ERR\_INVALID\_HNDL, [1157](#)
- eWDS\_ERR\_PROFILE\_REG\_RESULT\_ERR\_INVALID\_IDENT, [1158](#)
- eWDS\_ERR\_PROFILE\_REG\_RESULT\_ERR\_INVALID\_OP, [1157](#)
- eWDS\_ERR\_PROFILE\_REG\_RESULT\_ERR\_INVALID\_PROFILE\_NUM, [1158](#)
- eWDS\_ERR\_PROFILE\_REG\_RESULT\_ERR\_INVALID\_PROFILE\_TYPE, [1158](#)
- eWDS\_ERR\_PROFILE\_REG\_RESULT\_ERR\_INVALID\_SUBS\_ID, [1158](#)
- eWDS\_ERR\_PROFILE\_REG\_RESULT\_ERR\_LEN\_INVALID, [1158](#)
- eWDS\_ERR\_PROFILE\_REG\_RESULT\_ERR\_LIB\_NOT\_INITED, [1158](#)
- eWDS\_ERR\_PROFILE\_REG\_RESULT\_FAIL, [1157](#)
- eWDS\_ERR\_PROFILE\_REG\_RESULT\_LIST\_END, [1158](#)
- qmerrno.h, [1151](#)
- eQCWWANError, [1153](#)
- qm\_wds\_ds\_profile\_extended\_err\_codes, [1157](#)
- QmiCbkCatEventStatusReportInd, [427](#)
- CCETlv, [427](#)
- event\_Index, [427](#)
- QmiCbkLocCradleMountInd, [428](#)
- cradleMountConfigStatus, [428](#)
- QmiCbkLocEventTimeSyncInd, [428](#)
- timeSyncRefCounter, [429](#)
- QmiCbkLocInjectSensorDataInd, [429](#)
- injectSensorDataStatus, [430](#)
- pAccelSamplesAccepted, [430](#)
- pAccelTempSamplesAccepted, [430](#)
- pGyroSamplesAccepted, [430](#)
- pGyroTempSamplesAccepted, [430](#)

- pOpaqueIdentifier, [430](#)
- QmiCbkLocInjectTimeInd, [430](#)
  - injectTimeSyncStatus, [431](#)
- QmiCbkLocPositionReportInd, [431](#)
  - pAltitudeAssumed, [436](#)
  - pAltitudeWrtEllipsoid, [436](#)
  - pAltitudeWrtMeanSeaLevel, [436](#)
  - pFixId, [436](#)
  - pGpsTime, [436](#)
  - pHeading, [436](#)
  - pHeadingUnc, [436](#)
  - pHorConfidence, [436](#)
  - pHorReliability, [436](#)
  - pHorUncCircular, [436](#)
  - pHorUncEllipseOrientAzimuth, [436](#)
  - pHorUncEllipseSemiMajor, [436](#)
  - pHorUncEllipseSemiMinor, [436](#)
  - pLatitude, [436](#)
  - pLeapSeconds, [436](#)
  - pLongitude, [436](#)
  - pMagneticDeviation, [436](#)
  - pPrecisionDilution, [436](#)
  - pSensorDataUsage, [437](#)
  - pSpeedHorizontal, [437](#)
  - pSpeedUnc, [437](#)
  - pSpeedVertical, [437](#)
  - pSvUsedforFix, [437](#)
  - pTechnologyMask, [437](#)
  - pTimeSrc, [437](#)
  - pTimeUnc, [437](#)
  - pTimestampUtc, [437](#)
  - pVertConfidence, [437](#)
  - pVertReliability, [437](#)
  - pVertUnc, [437](#)
  - sessionId, [437](#)
  - sessionStatus, [437](#)
- QmiCbkLocSensorStreamingInd, [437](#)
  - pAccelAcceptReady, [438](#)
  - pAccelTempAcceptReady, [438](#)
  - pGyroAcceptReady, [438](#)
  - pGyroTempAcceptReady, [438](#)
- QmiCbkNasLTECphyCaInfo, [438](#)
  - sPhyCaAggPcellInfo, [438](#)
  - sPhyCaAggScellIDBw, [439](#)
  - sPhyCaAggScellIndType, [439](#)
  - sPhyCaAggScellIndex, [439](#)
  - sPhyCaAggScellInfo, [439](#)
- QmiCbkSwiOmaDmEventStatusReportInd, [439](#)
  - SITlv, [439](#)
- QmiCbkSwiOmaDmEventStatusReportIndExt, [439](#)
  - SITlv, [439](#)
- QmiCbkWdsStatisticsIndState, [439](#)
  - RxDropConutTlv, [440](#)
  - RxOkByteCountTlv, [440](#)
  - RxOkConutTlv, [440](#)
  - TxDropConutTlv, [440](#)
  - TxOkByteCountTlv, [440](#)
  - TxOkConutTlv, [440](#)
- qmiDataBearerMasks
  - qaGobiApiWds.h, [1108](#)
- QmiNas3GppNetworkInfo, [441](#)
  - pDescription, [442](#)
  - pForbidden, [442](#)
  - pInUse, [442](#)
  - pMCC, [442](#)
  - pMNC, [442](#)
  - pPreferred, [442](#)
  - pRoaming, [442](#)
- QmiNasGetRFBandInfoResp, [443](#)
  - pInstancesSize, [443](#)
  - pRFBandInfoElements, [443](#)
  - results, [443](#)
- QmiNasPerformNetworkScanResp, [443](#)
  - pInstanceSize, [443](#)
  - pInstances, [443](#)
  - results, [443](#)
- QmiProfileInfo
  - qaGobiApiWds.h, [1103](#)
- QmiWSDDataBearerTechnology
  - qaGobiApiWds.h, [1105](#)
- QmiWSDDataBearers
  - qaGobiApiWds.h, [1103](#)
- QmiWdsIpAddressInfo, [443](#)
  - pIPAddressV4, [444](#)
  - pIPAddressV6, [444](#)
  - pIPv6prefixlen, [444](#)
- qmiWdsRunTimeSettings, [444](#)
  - pAPNName, [447](#)
  - pAuthentication, [447](#)
  - pDomainList, [447](#)
  - pGPRSGrantedQoS, [447](#)
  - pGWAddressV4, [447](#)
  - pIMCNflag, [447](#)
  - pIPAddressV4, [447](#)
  - pIPFamilyPreference, [447](#)
  - pIPv6AddrInfo, [447](#)
  - pIPv6GWAddrInfo, [448](#)
  - pMtu, [448](#)
  - pPCSCFAddrPCO, [448](#)
  - pPCSCFFQDNAddrList, [448](#)
  - pPDPTType, [448](#)
  - pPrimaryDNSV4, [448](#)
  - pPrimaryDNSV6, [448](#)
  - pProfileID, [448](#)
  - pProfileName, [448](#)
  - pSecondaryDNSV4, [448](#)
  - pSecondaryDNSV6, [448](#)
  - pServerAddrList, [448](#)
  - pSubnetMaskV4, [448](#)
  - pTechnology, [448](#)
  - pUMTSGrantedQoS, [448](#)
  - pUsername, [448](#)
- qmifwinfo\_s, [440](#)
  - dev, [441](#)
  - g, [441](#)
  - s, [441](#)

- qos\_id
  - QosMap, [453](#)
- QosClassID, [448](#)
  - gDlBitRate, [449](#)
  - gUlBitRate, [449](#)
  - maxDlBitRate, [449](#)
  - maxUlBitRate, [449](#)
  - QCI, [449](#)
- qosDeliveryOrder
  - UMTSMInQoS, [622](#)
  - UMTSQoS, [625](#)
- QosEventInfo, [449](#)
  - pDataBearer, [450](#)
  - pPacketsCountRX, [451](#)
  - pPacketsCountTX, [451](#)
  - pTotalBytesRX, [451](#)
  - pTotalBytesTX, [451](#)
- qosFlow
  - sQosStat, [556](#)
- QosFlowInfo, [451](#)
  - pBearerID, [451](#)
  - pQFlowState, [452](#)
  - pRxQFilter, [452](#)
  - pRxQFlowGranted, [452](#)
  - pTxQFilter, [452](#)
  - pTxQFlowGranted, [452](#)
- QosFlowInfoState, [452](#)
  - id, [452](#)
  - isNewFlow, [452](#)
  - state, [452](#)
- QosMap, [452](#)
  - dscp, [453](#)
  - qos\_id, [453](#)
  - state, [453](#)
- Quality of Service (QOS), [38](#)
- RAT
  - SlqsNas3GppNetworkRAT, [524](#)
- RATMask
  - CurrNetworkInfo, [170](#)
- REGISTER\_EVENT
  - qaGobiApiCbK.h, [743](#)
- REGISTER\_SRV
  - qaGobiApiCbK.h, [743](#)
- RFBandInfoElements, [462](#)
  - activeBandClass, [463](#)
  - activeChannel, [463](#)
  - radioInterface, [463](#)
- RMSetTransferStatistics
  - qaGobiApiWds.h, [1122](#)
- ROAMING\_INDEX
  - qaNasPerformNetworkScan.h, [1151](#)
- RPCause
  - SMSAsyncRawSend\_s, [542](#)
- RSRPThresListLen
  - RSRPThresh, [467](#)
- RSRPThresh, [466](#)
  - pRSRPThresList, [467](#)
  - RSRPThresListLen, [467](#)
- RSRQThresListLen
  - RSRQThresh, [468](#)
- RSRQThresh, [467](#)
  - pRSRQThresList, [468](#)
  - RSRQThresListLen, [468](#)
- RSSIThresListLen
  - RSSIThresh, [469](#)
- RSSIThresh, [468](#)
  - pRSSIThresList, [469](#)
  - RSSIThresListLen, [469](#)
- RX\_EC\_IO
  - NetworkStat1x, [369](#)
- RX\_PWR
  - NetworkStat1x, [369](#)
  - NetworkStatEVDO, [371](#)
- RXAGCList, [469](#)
  - pRXAIG, [469](#)
  - pRXComprSlope, [469](#)
  - pRXComprThres, [469](#)
  - pRXExpSlope, [469](#)
  - pRXExpThres, [469](#)
  - pRXStaticGain, [470](#)
- RXAVCList, [470](#)
  - pAVRXAVCHheadroom, [470](#)
  - pAVRXAVCSens, [470](#)
- RXChan
  - LTEInfo, [315](#)
- RXOKBytesCount
  - DUNCallInfoInd, [196](#)
- RXPCMIIRFiltr, [472](#)
  - pFlag, [473](#)
  - pStage0Val, [473](#)
  - pStage1Val, [473](#)
  - pStage2Val, [473](#)
  - pStage3Val, [473](#)
  - pStage4Val, [473](#)
  - pStageCnt, [474](#)
- radio\_if
  - nasGetTxRxInfoReq, [346](#)
- radioIf
  - ecioListElement, [196](#)
  - errorRateListElement, [201](#)
  - rsrqInformation, [467](#)
  - rxSignalStrengthListElement, [474](#)
- radioInterface
  - RFBandInfoElements, [463](#)
  - roamIndList, [464](#)
  - servSystem, [485](#)
- radioInterfaceList
  - ServingSystemInfo, [482](#)
- radioInterfaceNo
  - ServingSystemInfo, [482](#)
- range
  - Port, [402](#)
- rat
  - CSGID, [161](#)
  - MNRInfo, [335](#)
- ratMask

- dataBearerTechnology, [181](#)
- ratValue
  - DataBearerTech, [179](#)
- rawLen
  - fileAttributes, [208](#)
- rawValue
  - fileAttributes, [208](#)
- rcv4
  - ssdatasession\_params, [560](#)
- rcv6
  - ssdatasession\_params, [560](#)
- readResult, [453](#)
  - content, [454](#)
  - contentLen, [454](#)
- readTransparent
  - UIMReadTransparentReq, [605](#)
- readTransparentInfo, [454](#)
  - length, [454](#)
  - offset, [454](#)
- Reason
  - voiceGetCallFWReq, [657](#)
  - voiceSetCallBarringPwdInfo, [684](#)
- reason
  - ccSUPSType, [126](#)
  - redirNumInfo, [456](#)
  - voiceGetCallBarringReq, [654](#)
  - voiceSetSUPSServiceReq, [692](#)
- receiptAction
  - smsRouteEntry, [552](#)
- receivedBytes
  - omaDmFotaTlvExt, [382](#)
- reconfigReqd
  - \_packetSrvStatus, [52](#)
- reconfiguration\_required
  - slqsSessionStateInfo, [530](#)
- recordCount
  - fileAttributes, [208](#)
- recordSize
  - fileAttributes, [208](#)
- redirNumInfo, [454](#)
  - numLen, [456](#)
  - numPlan, [456](#)
  - numType, [456](#)
  - number, [456](#)
  - PI, [456](#)
  - reason, [456](#)
  - SI, [456](#)
- RedirPartyNum
  - arrRedirPartyNum, [98](#)
- ReferenceID
  - CatAlPhalIdentifierTlv, [123](#)
  - CatEventIDDDataTlv, [124](#)
- refpn
  - CDMAInfo, [129](#)
- refreshComplete
  - UIMRefreshCompleteReq, [607](#)
- RefreshMode
  - CatRefreshTlv, [125](#)
- RefreshStage
  - CatRefreshTlv, [125](#)
- regAction
  - nasInitNetworkReg, [351](#)
- regInd
  - \_transLayerInfoNotification, [81](#)
- regPrd
  - AddCDMASysInfo, [85](#)
- regRefresh
  - UIMRefreshRegisterReq, [611](#)
- regRejectInfoValid
  - GSMSysInfo, [263](#)
  - LTESysInfo, [330](#)
  - WCDMASysInfo, [711](#)
- regState
  - servSystem, [485](#)
- Region
  - fwinfo\_s, [212](#)
- registerFlag
  - registerRefresh, [458](#)
- registerRefresh, [456](#)
  - arrfileInfo, [458](#)
  - numFiles, [458](#)
  - registerFlag, [458](#)
  - voteForInit, [458](#)
- registrationState
  - ServingSystemInfo, [483](#)
- rejCause
  - GSMSysInfo, [263](#)
  - LTESysInfo, [330](#)
  - WCDMASysInfo, [711](#)
- rejectSrvDomain
  - GSMSysInfo, [263](#)
  - LTESysInfo, [330](#)
  - WCDMASysInfo, [711](#)
- reliabilityClass
  - GPRSQoS, [251](#)
  - GPRSRequestedQoS, [252](#)
- remPartyNumber
  - remotePartyNum, [461](#)
- remainingRetries, [458](#)
  - unblockLeft, [459](#)
  - verifyLeft, [459](#)
- Remote Management Service (RMS), [29](#)
- RemotePartyName
  - getAllCallRmtPtyName, [217](#)
- remotePartyName, [459](#)
  - callerName, [460](#)
  - codingScheme, [460](#)
  - nameLen, [460](#)
  - namePI, [460](#)
- RemotePartyNum
  - getAllCallRmtPtyNum, [219](#)
- remotePartyNum, [460](#)
  - numLen, [461](#)
  - presentationInd, [461](#)
  - remPartyNumber, [461](#)
- ReqFieldsList, [461](#)

- requestFields, [462](#)
- requestFieldsLen, [462](#)
- requestFields
  - ReqFieldsList, [462](#)
- requestFieldsLen
  - ReqFieldsList, [462](#)
- resBerRatio
  - UMTSMinQoS, [622](#)
  - UMTSQoS, [625](#)
- ResCode
  - FirmwareUpdatStat, [211](#)
  - GetAudioVolTLBConfigResp, [225](#)
  - SetAudioVolTLBConfigResp, [492](#)
- reserved
  - omaDmFotaTlvExt, [382](#)
- ResetPDSDData
  - qaGobiApiPds.h, [965](#)
- ResetToFactoryDefaults
  - qaGobiApiDms.h, [854](#)
- RespFieldsList, [462](#)
  - responseFields, [462](#)
  - responseFieldsLen, [462](#)
- responseFields
  - RespFieldsList, [462](#)
- responseFieldsLen
  - RespFieldsList, [462](#)
- results
  - QmiNasGetRFBandInfoResp, [443](#)
  - QmiNasPerformNetworkScanResp, [443](#)
- revPolarity
  - lineCtrlInfo, [304](#)
- ReverseMac
  - protocolSubtypeElement, [421](#)
- RmtPtyNum
  - arrRemotePartyNum, [99](#)
- roamIndList, [463](#)
  - numInstances, [464](#)
  - radioInterface, [464](#)
  - roamIndicator, [464](#)
- roamIndicator
  - roamIndList, [464](#)
- roamIndicatorVal
  - qaQmiServingSystemParam, [427](#)
- roamOrigVoiceSO
  - prefVoiceSO, [406](#)
- roamStatus
  - sysInfoCommon, [583](#)
- roamStatusValid
  - sysInfoCommon, [583](#)
- roamTimer, [464](#)
  - namID, [466](#)
  - roamTimerValue, [466](#)
- roamTimerValue
  - roamTimer, [466](#)
- Roaming
  - SlqsNas3GppNetworkInfo, [524](#)
- roaming\_ind
  - RoamingInfo, [464](#)
- RoamingIndicatorList
  - qaQmiServingSystemParam, [427](#)
- RoamingInfo, [464](#)
  - roaming\_ind, [464](#)
  - TlvPresent, [464](#)
- routeList
  - smsSetRoutesReq, [553](#)
- routeStorage
  - smsRouteEntry, [552](#)
- rptRate
  - LTESigRptCfg, [324](#)
  - LTESigRptConfig, [324](#)
- rscp
  - rxInfo, [471](#)
  - TDSCDMASigInfoExt, [585](#)
  - UMTSInfo, [617](#)
- rsrp
  - cellParams, [142](#)
  - LTESInfo, [327](#)
  - rxInfo, [471](#)
  - umtsLTENbrCell, [619](#)
- rsrplevel
  - IteRsrpinformation, [321](#)
- rsrq
  - cellParams, [142](#)
  - LTESInfo, [327](#)
  - rsrqInformation, [467](#)
  - umtsLTENbrCell, [619](#)
- rsrqDelta
  - SLQSSignalStrengthsIndReq, [535](#)
- rsrqInfo
  - slqsSignalStrengthInfo, [533](#)
  - SLQSSignalStrengthsInformation, [537](#)
- rsrqInformation, [467](#)
  - radiolf, [467](#)
  - rsrq, [467](#)
- rssi
  - CDMASSInfo, [136](#)
  - cellParams, [142](#)
  - gsmCellInfo, [258](#)
  - HDRSSInfo, [271](#)
  - LTESInfo, [327](#)
  - TDSCDMASigInfoExt, [585](#)
- rts
  - WdsRunTimeSettings, [720](#)
- rx\_bytes
  - NetStats, [364](#)
- rx\_errors
  - NetStats, [364](#)
- rx\_overflows
  - NetStats, [364](#)
- rx\_packets
  - NetStats, [364](#)
- RxDropConutTlv
  - QmiCbkWdsStatisticsIndState, [440](#)
- rxInfo, [470](#)
  - ecio, [471](#)
  - isRadioTuned, [471](#)



- phase, [471](#)
  - rscp, [471](#)
  - rsrp, [471](#)
  - rxPower, [471](#)
- rxLev
  - GERANInfo, [214](#)
- RxOkByteCountTlv
  - QmiCbkWdsStatisticsIndState, [440](#)
- RxOkConutTlv
  - QmiCbkWdsStatisticsIndState, [440](#)
- rxPower
  - rxInfo, [471](#)
- rxSignalStrength
  - rxSignalStrengthListElement, [474](#)
- rxSignalStrengthDelta
  - SLQSSignalStrengthsIndReq, [536](#)
- rxSignalStrengthInfo
  - SLQSSignalStrengthsInformation, [537](#)
- rxSignalStrengthList
  - slqsSignalStrengthInfo, [533](#)
- rxSignalStrengthListElement, [474](#)
  - radioIlf, [474](#)
  - rxSignalStrength, [474](#)
- rxSignalStrengthListLen
  - slqsSignalStrengthInfo, [534](#)
- s
  - qmifwinfo\_s, [441](#)
- SMS\_EVENT\_ETWS
  - qaGobiApiCbk.h, [782](#)
- SMS\_EVENT\_ETWS\_PLMN
  - qaGobiApiCbk.h, [782](#)
- SMS\_EVENT\_MESSAGE\_MODE
  - qaGobiApiCbk.h, [782](#)
- SMS\_EVENT\_MT\_MESSAGE
  - qaGobiApiCbk.h, [782](#)
- SMS\_EVENT\_SMS\_ON\_IMS
  - qaGobiApiCbk.h, [782](#)
- SMS\_EVENT\_SMSC\_ADDRESS
  - qaGobiApiCbk.h, [782](#)
- SMS\_EVENT\_TRANSFER\_ROUTE\_MT\_MESSAGE
  - qaGobiApiCbk.h, [782](#)
- sApnExtraParams, [474](#)
  - ambr\_dl, [475](#)
  - ambr\_dl\_ext, [475](#)
  - ambr\_dl\_ext2, [475](#)
  - ambr\_ul, [475](#)
  - ambr\_ul\_ext, [476](#)
  - ambr\_ul\_ext2, [476](#)
  - apnId, [476](#)
- SECOND\_INSTANCE
  - qaGobiApiCbk.h, [743](#)
- sGetDeviceSeriesResult, [513](#)
  - eDevice, [514](#)
  - uResult, [514](#)
- SHORT
  - SwiDataTypes.h, [1159](#)
- SI
  - calledPartyInfo, [109](#)
  - callFWExtInfo, [115](#)
  - callingPartyInfo, [120](#)
  - redirNumInfo, [456](#)
- SITlv
  - QmiCbkSwiOmaDmEventStatusReportInd, [439](#)
  - QmiCbkSwiOmaDmEventStatusReportIndExt, [439](#)
- sIntraSearch
  - LTEInfoIntraFreq, [318](#)
- SLQSAutoConnect
  - qaGobiApiWds.h, [1132](#)
- SLQSCDMADecodeMTTextMsg
  - qaGobiApiSms.h, [995](#)
- SLQSCDMAEncodeMOTextMsg
  - qaGobiApiSms.h, [996](#)
- SLQSConfigSigInfo
  - qaGobiApiNas.h, [938](#)
- SLQSCreateProfile
  - qaGobiApiWds.h, [1132](#)
- SLQSDeleteProfile
  - qaGobiApiWds.h, [1133](#)
- SLQSDeleteProfileParams, [519](#)
  - profileIndex, [520](#)
  - profileType, [520](#)
- SLQSDeleteSMS
  - qaGobiApiSms.h, [996](#)
- SLQSDownloadFirmwareToSlot
  - qaGobiApiFms.h, [884](#)
- SLQSFWINFO\_SKU\_SZ
  - qaGobiApiFms.h, [879](#)
- SLQSGet3GPPConfigItem
  - qaGobiApiWds.h, [1134](#)
- SLQSGetAGPSCConfig
  - qaGobiApiPds.h, [969](#)
- SLQSGetAudioPathConfig
  - qaGobiApiAudio.h, [730](#)
- SLQSGetAudioProfile
  - qaGobiApiAudio.h, [731](#)
- SLQSGetAudioVolTLBConfig
  - qaGobiApiAudio.h, [731](#)
- SLQSGetBandCapability
  - qaGobiApiDms.h, [855](#)
- SLQSGetBootVersionNumber
  - qaGobiApiFms.h, [885](#)
- SLQSGetByteTotals
  - qaGobiApiWds.h, [1134](#)
- SLQSGetConnectionRate
  - qaGobiApiWds.h, [1134](#)
- SLQSGetCurrDataSystemStat
  - qaGobiApiWds.h, [1135](#)
- SLQSGetCurrentChannelRate
  - qaGobiApiWds.h, [1135](#)
- SLQSGetCurrentPRLInfo
  - qaGobiApiDms.h, [857](#)
- SLQSGetCustFeatures
  - qaGobiApiDms.h, [857](#)
- SLQSGetCustFeaturesV2
  - qaGobiApiDms.h, [857](#)
- SLQSGetDUNCallInfo

- qaGobiApiWds.h, [1137](#)
- SLQSGetDataBearerTechnology
  - qaGobiApiWds.h, [1136](#)
- SLQSGetDataBearerTechnologyExt
  - qaGobiApiWds.h, [1136](#)
- SLQSGetDeviceMode
  - qaGobiApiDcs.h, [827](#)
- SLQSGetERIFile
  - qaGobiApiDms.h, [858](#)
- SLQSGetErrorRate
  - qaGobiApiNas.h, [938](#)
- SLQSGetFirmwareInfo
  - qaGobiApiFms.h, [885](#)
- SLQSGetGPSSStateInfo
  - qaGobiApiPds.h, [970](#)
- SLQSGetIMSARegStatus
  - qaGobiApilmsa.h, [898](#)
- SLQSGetIMSAServiceStatus
  - qaGobiApilmsa.h, [899](#)
- SLQSGetIMSASupportedFields
  - qaGobiApilmsa.h, [899](#)
- SLQSGetIMSASupportedMsg
  - qaGobiApilmsa.h, [900](#)
- SLQSGetIMSSMSConfig
  - qaGobiApilms.h, [892](#)
- SLQSGetIMSUserConfig
  - qaGobiApilms.h, [892](#)
- SLQSGetIMSVolPConfig
  - qaGobiApilms.h, [892](#)
- SLQSGetImageInfo
  - qaGobiApiFms.h, [886](#)
- SLQSGetImageInfo\_9x15
  - qaGobiApiFms.h, [886](#)
- SLQSGetImageInfoMC77xx
  - qaGobiApiFms.h, [887](#)
- SLQSGetImageInfoMC83xx
  - qaGobiApiFms.h, [888](#)
- SLQSGetIndicationRegister
  - qaGobiApiSms.h, [997](#)
- SLQSGetM2MAVMute
  - qaGobiApiSwiAudio.h, [1013](#)
- SLQSGetM2MAudioProfile
  - qaGobiApiSwiAudio.h, [1012](#)
- SLQSGetM2MAudioVolume
  - qaGobiApiSwiAudio.h, [1013](#)
- SLQSGetM2MSpkrGain
  - qaGobiApiSwiAudio.h, [1014](#)
- SLQSGetMessageWaiting
  - qaGobiApiSms.h, [998](#)
- SLQSGetNetStatistic
  - qaGobiApiDcs.h, [827](#)
- SLQSGetOperatorNameData
  - qaGobiApiNas.h, [938](#)
- SLQSGetPLMNName
  - qaGobiApiNas.h, [939](#)
- SLQSGetPacketStatistics
  - qaGobiApiWds.h, [1137](#)
- SLQSGetPidof
  - qaGobiApiSwi.h, [1011](#)
- SLQSGetProfile
  - qaGobiApiWds.h, [1137](#)
- SLQSGetProfileSettings
  - qaGobiApiWds.h, [1140](#)
- SLQSGetRegMgrConfig
  - qaGobiApilms.h, [893](#)
- SLQSGetRfSarState
  - qaGobiApiSar.h, [985](#)
- SLQSGetRuntimeSettings
  - qaGobiApiWds.h, [1141](#)
- SLQSGetSIPConfig
  - qaGobiApilms.h, [893](#)
- SLQSGetSMS
  - qaGobiApiSms.h, [998](#)
- SLQSGetSMSList
  - qaGobiApiSms.h, [1000](#)
- SLQSGetSdkVersion
  - qaGobiApiSwi.h, [1011](#)
- SLQSGetSerialNumbers
  - qaGobiApiDms.h, [858](#)
- SLQSGetServingSystem
  - qaGobiApiNas.h, [939](#)
- SLQSGetSessionState
  - qaGobiApiWds.h, [1141](#)
- SLQSGetSignalStrength
  - qaGobiApiNas.h, [940](#)
- SLQSGetSmsBroadcastConfig
  - qaGobiApiSms.h, [999](#)
- SLQSGetSysSelectionPref
  - qaGobiApiNas.h, [940](#)
- SLQSGetTransLayerInfo
  - qaGobiApiSms.h, [1001](#)
- SLQSGetTransNWRRegInfo
  - qaGobiApiSms.h, [1001](#)
- SLQSGetUsbPortNames
  - qaGobiApiDcs.h, [828](#)
- SLQSImConfigIndicationRegister
  - qaGobiApilms.h, [894](#)
- SLQSInitiateNetworkRegistration
  - qaGobiApiNas.h, [941](#)
- SLQSIsSpkgFormatRequired
  - qaGobiApiFms.h, [888](#)
- SLQSKillSDKProcess
  - qaGobiApiDcs.h, [828](#)
- SLQSLOCDelAssData
  - qaGobiApiLoc.h, [902](#)
- SLQSLOCEventRegister
  - qaGobiApiLoc.h, [902](#)
- SLQSLOCSetExtPowerState
  - qaGobiApiLoc.h, [902](#)
- SLQSLOCSetOpMode
  - qaGobiApiLoc.h, [903](#)
- SLQSLOCStart
  - qaGobiApiLoc.h, [903](#)
- SLQSLOCStop
  - qaGobiApiLoc.h, [904](#)
- SLQSModifyProfile



- qaGobiApiWds.h, [1142](#)
- SLQSMModifySMSStatus
  - qaGobiApiSms.h, [1002](#)
- SLQSNASGetLTECPHYCaInfo
  - qaGobiApiNas.h, [944](#)
- SLQSNasConfigSigInfo2
  - qaGobiApiNas.h, [941](#)
- SLQSNasGet3GPP2Subscription
  - qaGobiApiNas.h, [941](#)
- SLQSNasGetCellLocationInfo
  - qaGobiApiNas.h, [943](#)
- SLQSNasGetHDRColorCode
  - qaGobiApiNas.h, [943](#)
- SLQSNasGetSigInfo
  - qaGobiApiNas.h, [944](#)
- SLQSNasGetSysInfo
  - qaGobiApiNas.h, [944](#)
- SLQSNasGetTxRxInfo
  - qaGobiApiNas.h, [945](#)
- SLQSNasIndicationRegister
  - qaGobiApiNas.h, [945](#)
- SLQSNasIndicationRegisterExt
  - qaGobiApiNas.h, [947](#)
- SLQSNasIndicationRegisterLTECphyCa
  - qaGobiApiNas.h, [948](#)
- SLQSNasNetworkTimeCallBack
  - qaGobiApiCbK.h, [797](#)
- SLQSNasSigInfo2CallBack
  - qaGobiApiCbK.h, [798](#)
- SLQSNasSigInfoCallBack
  - qaGobiApiCbK.h, [798](#)
- SLQSNasSwiIndicationRegister
  - qaGobiApiNas.h, [948](#)
- SLQSNasSwiModemStatus
  - qaGobiApiNas.h, [949](#)
- SLQSNasSwiOTAMessageCallback
  - qaGobiApiCbK.h, [800](#)
- SLQSNasSysInfoCallBack
  - qaGobiApiCbK.h, [800](#)
- SLQSOMADMCancelSession
  - qaGobiApiSwiOmadms.h, [1023](#)
- SLQSOMADMGetSessionInfo
  - qaGobiApiSwiOmadms.h, [1023](#)
- SLQSOMADMGetSettings
  - qaGobiApiSwiOmadms.h, [1024](#)
- SLQSOMADMGetSettings2
  - qaGobiApiSwiOmadms.h, [1025](#)
- SLQSOMADMSendSelection
  - qaGobiApiSwiOmadms.h, [1025](#)
- SLQSOMADMSendSelection2
  - qaGobiApiSwiOmadms.h, [1026](#)
- SLQSOMADMSessionInfo
  - qaGobiApiSwiOmadms.h, [1018](#)
- SLQSOMADMSetSettings
  - qaGobiApiSwiOmadms.h, [1026](#)
- SLQSOMADMSetSettings2
  - qaGobiApiSwiOmadms.h, [1027](#)
- SLQSOMADMSetSettings3
  - qaGobiApiSwiOmadms.h, [1027](#)
- SLQSOMADMSettings
  - qaGobiApiSwiOmadms.h, [1020](#)
- SLQSOMADMSettingsReqParams
  - qaGobiApiSwiOmadms.h, [1021](#)
- SLQSOMADMSettingsReqParams3
  - qaGobiApiSwiOmadms.h, [1022](#)
- SLQSOMADMStartSession
  - qaGobiApiSwiOmadms.h, [1028](#)
- SLQSOMADMStartSession2
  - qaGobiApiSwiOmadms.h, [1028](#)
- SLQSOriginateUSSD
  - qaGobiApiVoice.h, [1081](#)
- SLQSPDSDeterminePosition
  - qaGobiApiPds.h, [970](#)
- SLQSPDSInjectAbsoluteTimeReference
  - qaGobiApiPds.h, [971](#)
- SLQSPDSInjectPositionData
  - qaGobiApiPds.h, [971](#)
- SLQSPerformNetworkScan
  - qaGobiApiNas.h, [949](#)
- SLQSQosClearMap
  - qaGobiApiDcs.h, [829](#)
- SLQSQosDumpMap
  - qaGobiApiDcs.h, [829](#)
- SLQSQosEditMap
  - qaGobiApiDcs.h, [829](#)
- SLQSQosGetFlowStatus
  - qaGobiApiQos.h, [976](#)
- SLQSQosGetGranted
  - qaGobiApiQos.h, [977](#)
- SLQSQosGetNWProf
  - qaGobiApiQos.h, [978](#)
- SLQSQosGetNetworkStatus
  - qaGobiApiQos.h, [977](#)
- SLQSQosMap
  - qaGobiApiDcs.h, [831](#)
- SLQSQosModify
  - qaGobiApiQos.h, [978](#)
- SLQSQosReadMap
  - qaGobiApiDcs.h, [831](#)
- SLQSQosRel
  - qaGobiApiQos.h, [979](#)
- SLQSQosReq
  - qaGobiApiQos.h, [979](#)
- SLQSQosReset
  - qaGobiApiQos.h, [980](#)
- SLQSQosResume
  - qaGobiApiQos.h, [980](#)
- SLQSQosSuspend
  - qaGobiApiQos.h, [981](#)
- SLQSQosSwiReadApnExtraParams
  - qaGobiApiQos.h, [981](#)
- SLQSQosSwiReadDataStats
  - qaGobiApiQos.h, [982](#)
- SLQSQosUnmap
  - qaGobiApiDcs.h, [831](#)
- SLQSRegisterIMSAIndication

- qaGobiApiImsa.h, [900](#)
- SLQSSResetPacketStatics
  - qaGobiApiWds.h, [1143](#)
- SLQSSGetLoopback
  - qaGobiApiWds.h, [1146](#)
- SLQSSSetLoopback
  - qaGobiApiWds.h, [1146](#)
- SLQSSendAsyncSMS
  - qaGobiApiSms.h, [1003](#)
- SLQSSendLongSMS
  - qaGobiApiSms.h, [1003](#)
- SLQSSendRawQMI
  - qaGobiApiSwi.h, [1011](#)
- SLQSSendSMS
  - qaGobiApiSms.h, [1004](#)
- SLQSSet3GPPConfigItem
  - qaGobiApiWds.h, [1143](#)
- SLQSSetAGPSConfig
  - qaGobiApiPds.h, [972](#)
- SLQSSetAudioPathConfig
  - qaGobiApiAudio.h, [732](#)
- SLQSSetAudioProfile
  - qaGobiApiAudio.h, [732](#)
- SLQSSetAudioVolTLBConfig
  - qaGobiApiAudio.h, [733](#)
- SLQSSetBandPreference
  - qaGobiApiNas.h, [950](#)
- SLQSSetBandPreferenceCbk
  - qaGobiApiCbk.h, [801](#)
- SLQSSetCustFeatures
  - qaGobiApiDms.h, [859](#)
- SLQSSetCustFeaturesV2
  - qaGobiApiDms.h, [859](#)
- SLQSSetDUNCallInfoCallback
  - qaGobiApiCbk.h, [802](#)
- SLQSSetDataSystemStatusCallback
  - qaGobiApiCbk.h, [801](#)
- SLQSSetHostMTU
  - qaGobiApiWds.h, [1143](#)
- SLQSSetIMSAPdpStatusCallback
  - qaGobiApiCbk.h, [802](#)
- SLQSSetIMSAARatStatusCallback
  - qaGobiApiCbk.h, [803](#)
- SLQSSetIMSARegStatusCallback
  - qaGobiApiCbk.h, [803](#)
- SLQSSetIMSASvcStatusCallback
  - qaGobiApiCbk.h, [803](#)
- SLQSSetIMSSMSConfig
  - qaGobiApiIms.h, [894](#)
- SLQSSetIMSSMSConfigCallback
  - qaGobiApiCbk.h, [804](#)
- SLQSSetIMSUserConfig
  - qaGobiApiIms.h, [895](#)
- SLQSSetIMSUserConfigCallback
  - qaGobiApiCbk.h, [804](#)
- SLQSSetMSVoIPConfig
  - qaGobiApiIms.h, [895](#)
- SLQSSetMSVoIPConfigCallback
  - qaGobiApiCbk.h, [805](#)
- SLQSSetIndicationRegister
  - qaGobiApiSms.h, [1005](#)
- SLQSSetLoggingMask
  - qaGobiApiDcs.h, [833](#)
- SLQSSetM2MAVMute
  - qaGobiApiSwiAudio.h, [1016](#)
- SLQSSetM2MAudioAVCFG
  - qaGobiApiSwiAudio.h, [1014](#)
- SLQSSetM2MAudioLPBK
  - qaGobiApiSwiAudio.h, [1015](#)
- SLQSSetM2MAudioNVDef
  - qaGobiApiSwiAudio.h, [1015](#)
- SLQSSetM2MAudioProfile
  - qaGobiApiSwiAudio.h, [1015](#)
- SLQSSetM2MAudioVolume
  - qaGobiApiSwiAudio.h, [1016](#)
- SLQSSetM2MSpkrGain
  - qaGobiApiSwiAudio.h, [1017](#)
- SLQSSetModemTempCallback
  - qaGobiApiCbk.h, [805](#)
- SLQSSetPacketSrvStatusCallback
  - qaGobiApiCbk.h, [805](#)
- SLQSSetPositionMethodState
  - qaGobiApiPds.h, [972](#)
- SLQSSetProfile
  - qaGobiApiWds.h, [1144](#)
- SLQSSetQosEventCallback
  - qaGobiApiCbk.h, [806](#)
- SLQSSetQosNWStatusCallback
  - qaGobiApiCbk.h, [806](#)
- SLQSSetQosPriEventCallback
  - qaGobiApiCbk.h, [807](#)
- SLQSSetQosStatusCallback
  - qaGobiApiCbk.h, [807](#)
- SLQSSetRegMgrConfig
  - qaGobiApiIms.h, [896](#)
- SLQSSetRegMgrConfigCallback
  - qaGobiApiCbk.h, [808](#)
- SLQSSetRfSarState
  - qaGobiApiSar.h, [985](#)
- SLQSSetSDKTerminatedCallback
  - qaGobiApiCbk.h, [808](#)
- SLQSSetSIPConfig
  - qaGobiApiIms.h, [896](#)
- SLQSSetSIPConfigCallback
  - qaGobiApiCbk.h, [810](#)
- SLQSSetSMSEventCallback
  - qaGobiApiCbk.h, [810](#)
- SLQSSetServingSystemCallback
  - qaGobiApiCbk.h, [808](#)
- SLQSSetSessionStateCallback
  - qaGobiApiCbk.h, [809](#)
- SLQSSetSignalStrengthsCallback
  - qaGobiApiCbk.h, [809](#)
- SLQSSetSmsBroadcastActivation
  - qaGobiApiSms.h, [1005](#)
- SLQSSetSmsBroadcastConfig

- qaGobiApiSms.h, [1006](#)
- SLQSSetSmsStorage
  - qaGobiApiSms.h, [1006](#)
- SLQSSetSwiHDRPersCallback
  - qaGobiApiCbk.h, [811](#)
- SLQSSetSysSelectionPref
  - qaGobiApiNas.h, [952](#)
- SLQSSetSysSelectionPrefCallBack
  - qaGobiApiCbk.h, [811](#)
- SLQSSetTransLayerInfoCallback
  - qaGobiApiCbk.h, [811](#)
- SLQSSetTransNWRegInfoCallback
  - qaGobiApiCbk.h, [813](#)
- SLQSSetWdsEventCallback
  - qaGobiApiCbk.h, [813](#)
- SLQSSetWdsTransferStatisticCallback
  - qaGobiApiCbk.h, [814](#)
- SLQSSignalStrengthsIndReq, [534](#)
  - ecioDelta, [535](#)
  - ecioThresholdList, [535](#)
  - ecioThresholdListLen, [535](#)
  - ioDelta, [535](#)
  - lteRsrpDelta, [535](#)
  - lteSnrDelta, [535](#)
  - rsrqDelta, [535](#)
  - rxSignalStrengthDelta, [536](#)
  - sinrDelta, [536](#)
  - sinrThresholdList, [536](#)
  - sinrThresholdListLen, [536](#)
- SLQSSignalStrengthsInformation, [536](#)
  - ecioInfo, [537](#)
  - errorRateInfo, [537](#)
  - io, [537](#)
  - lteRsrpinfo, [537](#)
  - lteSnrinfo, [537](#)
  - rsrqInfo, [537](#)
  - rxSignalStrengthInfo, [537](#)
  - sinr, [537](#)
- SLQSSmsGetMaxStorageSize
  - qaGobiApiSms.h, [1007](#)
- SLQSSmsGetMessageProtocol
  - qaGobiApiSms.h, [1007](#)
- SLQSSmsSetRoutes
  - qaGobiApiSms.h, [1008](#)
- SLQSSstart
  - qaGobiApiDcs.h, [833](#)
- SLQSSstart\_AVAgent
  - qaGobiApiDcs.h, [834](#)
- SLQSSstartSrv
  - qaGobiApiDcs.h, [834](#)
- SLQSSstartStopDataSession
  - qaGobiApiWds.h, [1146](#)
- SLQSSwiGetAllCarrierImages
  - qaGobiApiFms.h, [888](#)
- SLQSSwiGetCrashAction
  - qaGobiApiDms.h, [859](#)
- SLQSSwiGetCrashInfo
  - qaGobiApiDms.h, [860](#)
- SLQSSwiGetFSN
  - qaGobiApiDms.h, [861](#)
- SLQSSwiGetFirmwareCurr
  - qaGobiApiDms.h, [861](#)
- SLQSSwiGetFwUpdateStatus
  - qaGobiApiDms.h, [861](#)
- SLQSSwiGetHDRPersonality
  - qaGobiApiNas.h, [952](#)
- SLQSSwiGetHDRProtSubtype
  - qaGobiApiNas.h, [952](#)
- SLQSSwiGetHRPDStats
  - qaGobiApiNas.h, [953](#)
- SLQSSwiGetHostDevInfo
  - qaGobiApiDms.h, [862](#)
- SLQSSwiGetHostDevInfoParams
  - qaGobiApiDms.h, [843](#)
- SLQSSwiGetLteCQI
  - qaGobiApiNas.h, [953](#)
- SLQSSwiGetOSInfo
  - qaGobiApiDms.h, [862](#)
- SLQSSwiGetOSInfoParams
  - qaGobiApiDms.h, [843](#)
- SLQSSwiGetSMSStorage
  - qaGobiApiSms.h, [1008](#)
- SLQSSwiGetSerialNoExt
  - qaGobiApiDms.h, [863](#)
- SLQSSwiGetSerialNoExtParams
  - qaGobiApiDms.h, [843](#)
- SLQSSwiGetUSBComp
  - qaGobiApiDms.h, [863](#)
- SLQSSwiNetworkDebug
  - qaGobiApiNas.h, [954](#)
- SLQSSwiPSDetach
  - qaGobiApiNas.h, [954](#)
- SLQSSwiSetCrashAction
  - qaGobiApiDms.h, [864](#)
- SLQSSwiSetHostDevInfo
  - qaGobiApiDms.h, [864](#)
- SLQSSwiSetHostDevInfoParams
  - qaGobiApiDms.h, [843](#)
- SLQSSwiSetOSInfo
  - qaGobiApiDms.h, [865](#)
- SLQSSwiSetOSInfoParams
  - qaGobiApiDms.h, [844](#)
- SLQSSwiSetUSBComp
  - qaGobiApiDms.h, [865](#)
- SLQSUIMAuthenticate
  - qaGobiApiUim.h, [1068](#)
- SLQSUIMChangePin
  - qaGobiApiUim.h, [1068](#)
- SLQSUIMDepersonalization
  - qaGobiApiUim.h, [1069](#)
- SLQSUIMEventRegister
  - qaGobiApiUim.h, [1069](#)
- SLQSUIMGetCardStatus
  - qaGobiApiUim.h, [1070](#)
- SLQSUIMGetFileAttributes
  - qaGobiApiUim.h, [1070](#)

- SLQSUIMGetState
  - qaGobiApiDms.h, [866](#)
- SLQSUIMPowerDown
  - qaGobiApiUim.h, [1071](#)
- SLQSUIMPowerUp
  - qaGobiApiUim.h, [1071](#)
- SLQSUIMReadTransparent
  - qaGobiApiUim.h, [1072](#)
- SLQSUIMRefreshComplete
  - qaGobiApiUim.h, [1072](#)
- SLQSUIMRefreshGetLastEvent
  - qaGobiApiUim.h, [1073](#)
- SLQSUIMRefreshOK
  - qaGobiApiUim.h, [1073](#)
- SLQSUIMRefreshRegister
  - qaGobiApiUim.h, [1074](#)
- SLQSUIMReset
  - qaGobiApiUim.h, [1074](#)
- SLQSUIMSetPinProtection
  - qaGobiApiUim.h, [1075](#)
- SLQSUIMSetRefreshCallBack
  - qaGobiApiCbk.h, [815](#)
- SLQSUIMSetStatusChangeCallBack
  - qaGobiApiCbk.h, [815](#)
- SLQSUIMUnblockPin
  - qaGobiApiUim.h, [1075](#)
- SLQSUIMVerifyPin
  - qaGobiApiUim.h, [1076](#)
- SLQSupgradeFirmware9x15
  - qaGobiApiFms.h, [889](#)
- SLQSVoiceALSSelectLine
  - qaGobiApiVoice.h, [1082](#)
- SLQSVoiceALSSetLineSwitching
  - qaGobiApiVoice.h, [1082](#)
- SLQSVoiceAnswerCall
  - qaGobiApiVoice.h, [1083](#)
- SLQSVoiceBindSubscription
  - qaGobiApiVoice.h, [1083](#)
- SLQSVoiceBurstDTMF
  - qaGobiApiVoice.h, [1084](#)
- SLQSVoiceDialCall
  - qaGobiApiVoice.h, [1084](#)
- SLQSVoiceEndCall
  - qaGobiApiVoice.h, [1085](#)
- SLQSVoiceGetAllCallInfo
  - qaGobiApiVoice.h, [1085](#)
- SLQSVoiceGetCLIP
  - qaGobiApiVoice.h, [1089](#)
- SLQSVoiceGetCLIR
  - qaGobiApiVoice.h, [1090](#)
- SLQSVoiceGetCNAP
  - qaGobiApiVoice.h, [1090](#)
- SLQSVoiceGetCOLP
  - qaGobiApiVoice.h, [1091](#)
- SLQSVoiceGetCOLR
  - qaGobiApiVoice.h, [1091](#)
- SLQSVoiceGetCallBarring
  - qaGobiApiVoice.h, [1086](#)
- SLQSVoiceGetCallForwardingStatus
  - qaGobiApiVoice.h, [1086](#)
- SLQSVoiceGetCallInfo
  - qaGobiApiVoice.h, [1088](#)
- SLQSVoiceGetCallWaiting
  - qaGobiApiVoice.h, [1089](#)
- SLQSVoiceGetConfig
  - qaGobiApiVoice.h, [1092](#)
- SLQSVoiceIndicationRegister
  - qaGobiApiVoice.h, [1092](#)
- SLQSVoiceInfoRecCallback
  - qaGobiApiCbk.h, [816](#)
- SLQSVoiceManageCalls
  - qaGobiApiVoice.h, [1093](#)
- SLQSVoiceOrigUSSDNoWait
  - qaGobiApiVoice.h, [1093](#)
- SLQSVoiceSendFlash
  - qaGobiApiVoice.h, [1095](#)
- SLQSVoiceSetAllCallStatusCallBack
  - qaGobiApiCbk.h, [816](#)
- SLQSVoiceSetCallBarringPassword
  - qaGobiApiVoice.h, [1095](#)
- SLQSVoiceSetConfig
  - qaGobiApiVoice.h, [1096](#)
- SLQSVoiceSetDTMFEventCallBack
  - qaGobiApiCbk.h, [817](#)
- SLQSVoiceSetOTASPStatusCallBack
  - qaGobiApiCbk.h, [817](#)
- SLQSVoiceSetPreferredPrivacy
  - qaGobiApiVoice.h, [1097](#)
- SLQSVoiceSetPrivacyChangeCallBack
  - qaGobiApiCbk.h, [818](#)
- SLQSVoiceSetSUPSCallBack
  - qaGobiApiCbk.h, [818](#)
- SLQSVoiceSetSUPSNotificationCallback
  - qaGobiApiCbk.h, [818](#)
- SLQSVoiceSetSUPSService
  - qaGobiApiVoice.h, [1097](#)
- SLQSVoiceStartContDTMF
  - qaGobiApiVoice.h, [1098](#)
- SLQSVoiceStopContDTMF
  - qaGobiApiVoice.h, [1098](#)
- SLQSWCDMADecodeLongTextMsg
  - qaGobiApiSms.h, [1009](#)
- SLQSWCDMADecodeMTTextMsg
  - qaGobiApiSms.h, [1009](#)
- SLQSWCDMAEncodeMOTextMsg
  - qaGobiApiSms.h, [1010](#)
- SLQSWdsGoActive
  - qaGobiApiWds.h, [1148](#)
- SLQSWdsGoDormant
  - qaGobiApiWds.h, [1148](#)
- SLQSWdsSetEventReport
  - qaGobiApiWds.h, [1149](#)
- SLQSWdsSwiPDPRuntimeSettings
  - qaGobiApiWds.h, [1149](#)
- SLQSWmsAsyncRawSendCallBack
  - qaGobiApiCbk.h, [820](#)

- SLQSWmsMemoryFullCallback
  - qaGobiApiCbk.h, [820](#)
- SLQSWmsMessageWaitingCallback
  - qaGobiApiCbk.h, [820](#)
- SMSAsyncRawSend
  - qaGobiApiCbk.h, [750](#)
- SMSAsyncRawSend\_s, [539](#)
  - alphaIDLen, [542](#)
  - causeCode, [542](#)
  - errorClass, [542](#)
  - messageID, [542](#)
  - msgDelFailureCause, [542](#)
  - msgDelFailureType, [542](#)
  - pAlphaID, [542](#)
  - RPCause, [542](#)
  - sendStatus, [542](#)
  - TPCause, [542](#)
  - userData, [542](#)
- SMSCAddress, [542](#)
  - data, [543](#)
  - length, [543](#)
- SMSCAddressInfo
  - qaGobiApiCbk.h, [751](#)
- SMSetsMessage, [543](#)
  - data, [543](#)
  - length, [543](#)
  - notificationType, [543](#)
- SMSetsMessageInfo
  - qaGobiApiCbk.h, [751](#)
- SMSetsPlmn, [543](#)
  - mobileCountryCode, [545](#)
  - mobileNetworkCode, [545](#)
- SMSetsPlmnInfo
  - qaGobiApiCbk.h, [752](#)
- SMSEventInfo
  - qaGobiApiCbk.h, [752](#)
- SMSEventInfo\_s, [545](#)
  - pEtsMessageInfo, [546](#)
  - pEtsPlmnInfo, [546](#)
  - pMTMessageInfo, [546](#)
  - pMessageModelInfo, [546](#)
  - pSMSCAddressInfo, [546](#)
  - pSMSOnIMSInfo, [546](#)
  - pTransferRouteMTMessageInfo, [546](#)
  - smsEventType, [546](#)
- SMSEventType
  - qaGobiApiCbk.h, [782](#)
- SMSMTMessage, [549](#)
  - messageIndex, [549](#)
  - storageType, [549](#)
- SMSMTMessageInfo
  - qaGobiApiCbk.h, [753](#)
- SMSMemoryInfo, [548](#)
  - messageMode, [548](#)
  - storageType, [548](#)
- SMSMessageMode, [548](#)
  - messageMode, [548](#)
- SMSMessageModelInfo
  - qaGobiApiCbk.h, [753](#)
- SMSOnIMS, [549](#)
  - smsOnIMS, [551](#)
- SMSOnIMSInfo
  - qaGobiApiCbk.h, [753](#)
- SMSTransferRouteMTMessage, [553](#)
  - ackIndicator, [554](#)
  - data, [554](#)
  - format, [554](#)
  - length, [554](#)
  - transactionID, [554](#)
- SMSTransferRouteMTMessageInfo
  - qaGobiApiCbk.h, [754](#)
- SNR
  - NetworkStatEVDO, [371](#)
- sNonIntraSearch
  - LTEInfoIntrafreq, [318](#)
- SO
  - NetworkStat1x, [369](#)
- SOMask
  - CurrNetworkInfo, [170](#)
- sPhyCaAggPcellInfo
  - nasGetLTECphyCa, [341](#)
  - QmiCbkNasLTECphyCaInfo, [438](#)
- sPhyCaAggScellDIBw
  - nasGetLTECphyCa, [341](#)
  - QmiCbkNasLTECphyCaInfo, [439](#)
- sPhyCaAggScellIndType
  - nasGetLTECphyCa, [341](#)
  - QmiCbkNasLTECphyCaInfo, [439](#)
- sPhyCaAggScellIndex
  - nasGetLTECphyCa, [341](#)
  - QmiCbkNasLTECphyCaInfo, [439](#)
- sPhyCaAggScellInfo
  - nasGetLTECphyCa, [341](#)
  - QmiCbkNasLTECphyCaInfo, [439](#)
- sQosFlowStat, [554](#)
  - bearerId, [555](#)
  - tx\_bytes, [555](#)
  - tx\_bytes\_drp, [555](#)
  - tx\_pkt, [555](#)
  - tx\_pkt\_drp, [555](#)
- sQosStat, [555](#)
  - apnId, [556](#)
  - numQosFlow, [556](#)
  - qosFlow, [556](#)
  - total\_rx\_bytes, [556](#)
  - total\_rx\_pkt, [556](#)
  - total\_tx\_bytes, [556](#)
  - total\_tx\_bytes\_drp, [556](#)
  - total\_tx\_pkt, [556](#)
  - total\_tx\_pkt\_drp, [556](#)
- SUPInfo, [561](#)
  - isModByCC, [562](#)
  - svcType, [562](#)
- SUPInformation
  - voiceSUPInfo, [697](#)
- SUPSType

- voiceManageCallsReq, 678
- SV, 562
  - id, 563
  - mask, 563
  - system, 563
- SVInfo, 563
  - len, 564
  - pSV, 564
- SWI Audio Service(SWIAUDIO), 40
- SWI Open Mobile Alliance Service (SWIOMA), 33
- SWI\_API
  - SwiDataTypes.h, 1159
- SWI\_STRUCT\_CarrierImage, 565
  - m\_FwBuildId, 565
  - m\_FwImagId, 565
  - m\_PriBuildId, 566
  - m\_PrImagId, 566
  - m\_nCarrierId, 566
  - m\_nFolderId, 566
  - m\_nStorage, 566
- SWIWWANCMAPI.h, 1160
- samplesPerBatch
  - accelAcceptReady\_s, 82
  - accelTempAcceptReady\_s, 83
  - gyroAcceptReady\_s, 264
  - gyroTempAcceptReady\_s, 265
- satelliteInfo, 476
  - azimuth, 478
  - elevation, 478
  - gnssSvId, 478
  - healthStatus, 478
  - snr, 478
  - svInfoMask, 478
  - svListLen, 478
  - svStatus, 478
  - system, 478
  - validMask, 478
- SaveSMS
  - qaGobiApiSms.h, 993
- sbas\_almanac\_sv\_msk
  - GPSSStateInfo, 256
- sbas\_ephemeris\_sv\_msk
  - GPSSStateInfo, 256
- sbas\_health\_sv\_msk
  - GPSSStateInfo, 256
- sbas\_visible\_sv\_msk
  - GPSSStateInfo, 256
- scell\_idx
  - PhyCaAggScellIndex, 395
- scell\_state
  - PhyCaAggScellIndType, 396
  - PhyCaAggScellInfo, 397
- screeningInd
  - connectNumInfo, 156
- sduErrorRatio
  - UMTSMInQoS, 622
  - UMTSQoS, 625
- secActivate
  - fileAttributes, 208
- secActivateMask
  - fileAttributes, 208
- secChA
  - CDMAChannel, 128
- secChB
  - CDMAChannel, 128
- secDeactivate
  - fileAttributes, 208
- secDeactivateMask
  - fileAttributes, 208
- secIncrease
  - fileAttributes, 208
- secIncreaseMask
  - fileAttributes, 208
- SecProt
  - protocolSubtypeElement, 421
- secRead
  - fileAttributes, 208
- secReadMask
  - fileAttributes, 208
- secWrite
  - fileAttributes, 208
- secWriteMask
  - fileAttributes, 208
- second
  - UniversalTime, 628
- SectorIDLen
  - NetworkStatEVDO, 371
- selNetwork
  - servSystem, 485
- selected
  - BroadcastConfig, 104
  - CDMABroadcastConfig, 127
- selectedNetwork
  - ServingSystemInfo, 483
- SendSMS
  - qaGobiApiSms.h, 994
- sendStatus
  - SMSAsyncRawSend\_s, 542
- sensorDataUsage
  - qaGobiApiCbk.h, 749
- sensorDataUsage\_s, 478
  - aidingIndicatorMask, 479
  - usageMask, 479
- serialNumbersInfo, 479
  - esnSize, 480
  - imeiSize, 480
  - imeiSvnSize, 480
  - meidSize, 480
  - pESNString, 480
  - pIMEIString, 480
  - pImeiSvnString, 480
  - pMEIDString, 480
  - qaGobiApiDms.h, 842
- servSystem, 483
  - csAttachState, 484
  - numRadiInterfaces, 484



- psAttachState, 485
- radioInterface, 485
- regState, 485
- selNetwork, 485
- serviceCategory
  - CDMABroadcastConfig, 127
- serviceClassInformation
  - qaGobiApiVoice.h, 1080
- serviceProviderName, 480
  - displayCondition, 481
  - spn, 481
  - spnLength, 481
- servingCellId
  - LTEInfoIntrafreq, 318
- ServingSystem
  - qaQmiServingSystemParam, 427
- ServingSystemInfo, 481
  - csAttachState, 482
  - hdrPersonality, 482
  - psAttachState, 482
  - radioInterfaceList, 482
  - radioInterfaceNo, 482
  - registrationState, 483
  - selectedNetwork, 483
- sessionEndReason
  - \_packetSrvStatus, 52
  - slqsSessionStateInfo, 530
- SessionId
  - LOCStartReq, 310
- sessionId
  - LOCStopReq, 310
  - QmiCbkLocPositionReportInd, 437
  - ssdatasession\_params, 560
- sessionInfo, 485
  - omaDmConfig, 485
  - omaDmFota, 485
  - omaDmNotifications, 485
  - sessionInfoTlv, 486
  - sessionInfoTlvExt, 486
  - UIMAuthenticateReq, 597
  - UIMChangePinReq, 599
  - UIMGetFileAttributesReq, 601
  - UIMReadTransparentReq, 605
  - UIMRefreshCompleteReq, 607
  - UIMRefreshGetLastEventReq, 609
  - UIMRefreshOKReq, 610
  - UIMRefreshRegisterReq, 611
  - UIMSetPinProtectionReq, 613
  - UIMUnblockPinReq, 614
  - UIMVerifyPinReq, 615
- sessionInfoExt, 485
  - omaDmConfig, 485
  - omaDmFota, 485
- sessionInfoTlv, 485
  - sessionInfo, 486
  - sessionType, 486
  - TlvPresent, 486
- sessionInfoTlvExt, 486
  - sessionInfo, 486
  - sessionType, 486
  - TlvPresent, 486
- sessionInformation
  - qaGobiApiCbk.h, 750
- sessionInformationExt
  - qaGobiApiCbk.h, 750
- sessionStatus
  - omaDmNotificationsTlv, 383
  - QmiCbkLocPositionReportInd, 437
- sessionType
  - omaDmFotaTlv, 380
  - sessionInfoTlv, 486
  - sessionInfoTlvExt, 486
  - UIMRefreshEvent, 609
  - UIMSessionInformation, 612
- SetACCOLC
  - qaGobiApiNas.h, 935
- SetActivationStatusCallback
  - qaGobiApiCbk.h, 783
- SetActiveMobileIPProfile
  - qaGobiApiWds.h, 1122
- SetAudioPathConfigReq, 486
  - pCodecSTGain, 488
  - pDTMFTXGain, 488
  - pECMode, 488
  - pNSEnable, 488
  - pRXAGCList, 488
  - pRXAVCAGCSwitch, 488
  - pRXAVCList, 488
  - pRXPCMIIRFiltr, 488
  - pTXAGCList, 489
  - pTXAVCSwitch, 489
  - pTXGain, 489
  - pTXPCMIIRFiltr, 489
  - Profile, 488
- SetAudioProfileReq, 489
  - EarMute, 490
  - Generator, 490
  - MicMute, 490
  - Profile, 490
  - Volume, 491
- SetAudioVolTLBConfigReq, 491
  - Generator, 492
  - Item, 492
  - Profile, 492
  - VolValue, 492
  - Volume, 492
- SetAudioVolTLBConfigResp, 492
  - ResCode, 492
- SetAutoconnect
  - qaGobiApiWds.h, 1122
- SetCATEventCallback
  - qaGobiApiCbk.h, 783
- SetCDMANetworkParameters
  - qaGobiApiNas.h, 935
- setCustomSettingV2, 492
  - cust\_id, 493

- cust\_value, [493](#)
- value\_length, [493](#)
- SetDataCapabilitiesCallback
  - qaGobiApiCbk.h, [784](#)
- SetDefaultProfile
  - qaGobiApiWds.h, [1124](#)
- SetDefaultProfileLTE
  - qaGobiApiWds.h, [1125](#)
- SetDefaultProfileLTEV2
  - qaGobiApiWds.h, [1127](#)
- SetDeviceStateChangeCb
  - qaGobiApiCbk.h, [786](#)
- SetFwDldCompletionCb
  - qaGobiApiCbk.h, [786](#)
- SetGPSCallback
  - qaGobiApiCbk.h, [787](#)
- SetIMSSMSConfigReq, [493](#)
  - pPhoneCtxtURI, [494](#)
  - pPhoneCtxtURLen, [494](#)
  - pSMSFormat, [494](#)
  - pSMSOverIPNwInd, [494](#)
- SetIMSSMSConfigResp, [494](#)
  - pSettingResp, [494](#)
- SetIMSUserConfigReq, [494](#)
  - pIMSDomain, [495](#)
  - pIMSDomainLen, [495](#)
- SetIMSUserConfigResp, [495](#)
  - pSettingResp, [495](#)
- SetIMSVoIPConfigReq, [495](#)
  - pAmrMode, [498](#)
  - pAmrOctetAligned, [498](#)
  - pAmrWBMode, [499](#)
  - pAmrWBOctetAligned, [499](#)
  - pAmrWbEnable, [498](#)
  - pMinSessionExpiryTimer, [499](#)
  - pRTPRTCPInactTimer, [499](#)
  - pRingBackTimer, [499](#)
  - pRingingTimer, [499](#)
  - pScrAmrEnable, [499](#)
  - pScrAmrWbEnable, [499](#)
  - pSessionExpiryTimer, [499](#)
- SetIMSVoIPConfigResp, [499](#)
  - pSettingResp, [499](#)
- SetImagesPreference
  - qaGobiApiFms.h, [883](#)
- setIndicationRegReq
  - qaGobiApiSms.h, [991](#)
- SetLURectCallback
  - qaGobiApiCbk.h, [789](#)
- SetLocCradleMountCallback
  - qaGobiApiCbk.h, [787](#)
- SetLocDeleteAssistDataCallback
  - qaGobiApiCbk.h, [787](#)
- SetLocEventPositionCallback
  - qaGobiApiCbk.h, [787](#)
- SetLocEventTimeSyncCallback
  - qaGobiApiCbk.h, [788](#)
- SetLocGnssSvInfoCallback
  - qaGobiApiCbk.h, [788](#)
- SetLocInjectSensorDataCallback
  - qaGobiApiCbk.h, [788](#)
- SetLocInjectTimeCallback
  - qaGobiApiCbk.h, [789](#)
- SetLocOpModeCallback
  - qaGobiApiCbk.h, [789](#)
- SetLocSensorStreamingCallback
  - qaGobiApiCbk.h, [789](#)
- SetM2MAVMuteReq, [503](#)
  - EarMute, [503](#)
  - MicMute, [504](#)
  - pCwtMute, [504](#)
  - Profile, [504](#)
- SetM2MAudioAVCFGReq, [499](#)
  - Device, [500](#)
  - PIFACEId, [500](#)
  - pCMPParams, [500](#)
  - Profile, [500](#)
- SetM2MAudioLPBKReq, [500](#)
  - Enable, [501](#)
- SetM2MAudioProfileReq, [501](#)
  - pCwtMute, [502](#)
  - pEarMute, [502](#)
  - pGenerator, [502](#)
  - pMicMute, [502](#)
  - pVolume, [502](#)
  - Profile, [502](#)
- SetM2MAudioVolumeReq, [502](#)
  - Generator, [503](#)
  - Level, [503](#)
  - Profile, [503](#)
- SetM2MSpkrGainReq, [504](#)
  - Profile, [504](#)
  - Value, [504](#)
- SetMobileIP
  - qaGobiApiWds.h, [1129](#)
- SetMobileIPParameters
  - qaGobiApiWds.h, [1130](#)
- SetMobileIPProfile
  - qaGobiApiWds.h, [1131](#)
- SetMobileIPStatusCallback
  - qaGobiApiCbk.h, [790](#)
- SetNMEACallback
  - qaGobiApiCbk.h, [792](#)
- SetNasLTECphyCalIndCallback
  - qaGobiApiCbk.h, [790](#)
- SetNetChangeCb
  - qaGobiApiCbk.h, [791](#)
- SetNetworkPreference
  - qaGobiApiNas.h, [937](#)
- SetNewSMSCallback
  - qaGobiApiCbk.h, [791](#)
- SetOMADMStateCallback
  - qaGobiApiCbk.h, [792](#)
- SetPDSDDefaults
  - qaGobiApiPds.h, [966](#)
- SetPDSSState



- qaGobiApiPds.h, [967](#)
- SetPDSStateCallback
  - qaGobiApiCbK.h, [792](#)
- setPINProtection, [504](#)
  - pinID, [505](#)
  - pinLength, [505](#)
  - pinOperation, [505](#)
  - pinValue, [505](#)
- SetPortAutomaticTracking
  - qaGobiApiPds.h, [967](#)
- SetPower
  - qaGobiApiDms.h, [854](#)
- SetPowerCallback
  - qaGobiApiCbK.h, [793](#)
- SetRFInfoCallback
  - qaGobiApiCbK.h, [793](#)
- SetRMTransferStatisticsCallback
  - qaGobiApiCbK.h, [793](#)
- SetRegMgrConfigReq, [505](#)
  - pCSCFPortName, [506](#)
  - pCSCFPortNameLen, [506](#)
  - pIMSTestMode, [506](#)
  - pPriCSCFPort, [506](#)
- SetRegMgrConfigResp, [506](#)
  - pSettingResp, [506](#)
- SetRoamingIndicatorCallback
  - qaGobiApiCbK.h, [795](#)
- SetSDKImagePath
  - qaGobiApiDcs.h, [827](#)
- SetSIPConfigReq, [512](#)
  - pSIPLocalPort, [513](#)
  - pSigCompEnabled, [513](#)
  - pSubscribeTimer, [513](#)
  - pTimerSIPReg, [513](#)
  - pTimerT1, [513](#)
  - pTimerT2, [513](#)
  - pTimerTf, [513](#)
- SetSIPConfigResp, [513](#)
  - pSettingResp, [513](#)
- SetSLQSOMADMAAlertCallback
  - qaGobiApiCbK.h, [796](#)
- SetSLQSOMADMAAlertCallbackExt
  - qaGobiApiCbK.h, [796](#)
- SetSMSCAddress
  - qaGobiApiSms.h, [995](#)
- SetSMSWake
  - qaGobiApiRms.h, [983](#)
- SetServiceAutomaticTracking
  - qaGobiApiPds.h, [968](#)
- SetSignalStrengthCallback
  - qaGobiApiCbK.h, [795](#)
- setSignalStrengthInfo, [507](#)
  - pCDMAECIODelta, [511](#)
  - pCDMAECIOThresh, [511](#)
  - pCDMARSSIDelta, [511](#)
  - pCDMARSSIThresh, [511](#)
  - pGSMRSSIDelta, [511](#)
  - pGSMRSSIThresh, [511](#)
  - pHDRECIODelta, [511](#)
  - pHDRECIOThresh, [511](#)
  - pHDRIODelta, [511](#)
  - pHDRIOThresh, [511](#)
  - pHRRSSIDelta, [511](#)
  - pHRRSSIThresh, [511](#)
  - pHRSINRDelta, [511](#)
  - pHRSINRThresh, [511](#)
  - pLTERSRPDelta, [511](#)
  - pLTERSRPThresh, [511](#)
  - pLTERSRQDelta, [511](#)
  - pLTERSRQThresh, [511](#)
  - pLTERSSIDelta, [511](#)
  - pLTERSSIThresh, [511](#)
  - pLTESNRDelta, [511](#)
  - pLTESNRThresh, [511](#)
  - pLTESigRptConfig, [511](#)
  - pTDSCDMAECIODelta, [511](#)
  - pTDSCDMAECIOThresh, [511](#)
  - pTDSCDMARSCPDelta, [511](#)
  - pTDSCDMARSCPThresh, [511](#)
  - pTDSCDMARSSIDelta, [511](#)
  - pTDSCDMARSSIThresh, [512](#)
  - pTDSCDMASINRDelta, [512](#)
  - pTDSCDMASINRThresh, [512](#)
  - pWCDMAECIODelta, [512](#)
  - pWCDMAECIOThresh, [512](#)
  - pWCDMARSSIDelta, [512](#)
  - pWCDMARSSIThresh, [512](#)
- SetUSSDNoWaitIndicationCallback
  - qaGobiApiCbK.h, [797](#)
- SetUSSDNotificationCallback
  - qaGobiApiCbK.h, [796](#)
- SetUSSDReleaseCallback
  - qaGobiApiCbK.h, [797](#)
- SetXTRAAutomaticDownload
  - qaGobiApiPds.h, [968](#)
- SetXTRANetwork
  - qaGobiApiPds.h, [969](#)
- SetupEventList
  - CatEventListTlv, [125](#)
- severity
  - omaDmFotaTlv, [380](#)
- Short Message Service (SMS), [26](#)
- shortName
  - nasPLMNNNameResp, [356](#)
  - PLMNNNetworkNameData, [402](#)
- shortNameCI
  - nasPLMNNNameResp, [356](#)
- shortNameEn
  - nasPLMNNNameResp, [356](#)
- shortNameLen
  - nasPLMNNNameResp, [356](#)
  - PLMNNNetworkNameData, [402](#)
- shortNameSB
  - nasPLMNNNameResp, [356](#)
- shortNameSpareBits
  - PLMNNNetworkNameData, [402](#)

- sid
  - CDMAInfo, 129
  - sidNid, 514
- SidNid
  - homeSIDNID, 275
- sidNid, 514
  - nid, 514
  - sid, 514
- SigInd
  - UMTSReqQoSsigInd, 626
- sigInfo, 514
  - pECIOTthresh, 515
  - pHRSINRThresh, 515
  - pIOTthresh, 516
  - pLTESNRThresh, 516
  - pLTESigRptCfg, 516
  - pRSRPThresh, 516
  - pRSRQThresh, 516
  - pRSSIThresh, 516
- signal
  - signalInfo, 516
- signalInfo, 516
  - alertPitch, 516
  - signal, 516
  - signalType, 516
- SignalStrengthDataType, 516
  - thresholds, 517
  - thresholdsSize, 517
- signalStrengthReqMask
  - slqsSignalStrengthInfo, 534
- signalType
  - signalInfo, 516
- sinr
  - HDRSSInfo, 271
  - slqsSignalStrengthInfo, 534
  - SLQSSignalStrengthsInformation, 537
  - TDSCDMASigInfoExt, 585
- sinrDelta
  - SLQSSignalStrengthsIndReq, 536
- sinrThresholdList
  - SLQSSignalStrengthsIndReq, 536
- sinrThresholdListLen
  - SLQSSignalStrengthsIndReq, 536
- sku\_str
  - slqsfwinfo\_s, 521
- slot
  - UIMPowerDownReq, 603
  - UIMPowerUpReq, 604
- SlotInfo
  - cardStatus, 122
- slotInfo, 517
  - AppStatus, 518
  - cardState, 518
  - errorState, 518
  - numApp, 518
  - upinRetries, 518
  - upinState, 518
  - upukRetries, 518
- slqs3GPPConfigItem
  - qaGobiApiWds.h, 1107
- SlqsNas3GppNetworkInfo, 521
  - Description, 523
  - Forbidden, 523
  - InUse, 523
  - MCC, 523
  - MNC, 524
  - Preferred, 524
  - Roaming, 524
- SlqsNas3GppNetworkRAT, 524
  - MCC, 524
  - MNC, 524
  - RAT, 524
- SlqsNasPcsDigit, 524
  - includes\_pcs\_digit, 526
  - MCC, 526
  - MNC, 526
- slqsNetworkScanInfo
  - qaGobiApiNas.h, 909
- SlqsProfile3GPP
  - WdsProfileParam, 719
- SlqsProfile3GPP2
  - WdsProfileParam, 719
- slqsSessionStateInfo, 530
  - pQmiInterfaceInfo, 530
  - reconfiguration\_required, 530
  - sessionEndReason, 530
  - state, 530
- slqsSignalStrengthInfo, 531
  - ecioList, 533
  - ecioListLen, 533
  - errorRateList, 533
  - errorRateListLen, 533
  - lo, 533
  - ltersrp, 533
  - ltesnr, 533
  - rsrqInfo, 533
  - rxSignalStrengthList, 533
  - rxSignalStrengthListLen, 534
  - signalStrengthReqMask, 534
  - sinr, 534
- slqsWdsEventInfo, 537
  - pDataBearer, 539
  - pDormancyStatus, 539
  - pPacketsCountRX, 539
  - pPacketsCountTX, 539
  - pQmiInterfaceInfo, 539
  - pTotalBytesRX, 539
  - pTotalBytesTX, 539
- slqsautoconnect, 519
  - acroamsetting, 519
  - acsetting, 519
  - action, 519
- slqsfwinfo\_s, 520
  - appversion\_str, 521
  - bootversion\_str, 521
  - carrier\_str, 521

- cur\_carr\_name, [521](#)
- cur\_carr\_rev, [521](#)
- modelid\_str, [521](#)
- packageid\_str, [521](#)
- priversion\_str, [521](#)
- sku\_str, [521](#)
- slqssendasyncsmsparams\_s, [526](#)
  - messageFormat, [528](#)
  - messageSize, [528](#)
  - pFollowOnDC, [528](#)
  - pForceOnDC, [528](#)
  - pLinktimer, [528](#)
  - pMessage, [528](#)
  - pRetryMessage, [528](#)
  - pRetryMessageId, [528](#)
  - pServiceOption, [528](#)
  - pSmsOnIms, [529](#)
  - pUserData, [529](#)
- slqssendsmsparams\_s, [529](#)
  - messageFailureCode, [529](#)
  - messageFormat, [529](#)
  - messageID, [529](#)
  - messageSize, [530](#)
  - pLinktimer, [530](#)
  - pMessage, [530](#)
- smsEventType
  - SMSEventInfo\_s, [546](#)
- smsMaxStorageSizeReq, [546](#)
  - pMessageMode, [547](#)
  - storageType, [547](#)
- smsMaxStorageSizeResp, [547](#)
  - freeSlots, [547](#)
  - maxStorageSize, [547](#)
- smsMsgprotocolResp, [548](#)
  - msgProtocol, [549](#)
- smsOnIMS
  - SMSONIMS, [551](#)
- smsRouteEntry, [551](#)
  - messageClass, [552](#)
  - messageType, [552](#)
  - receiptAction, [552](#)
  - routeStorage, [552](#)
- smsSetRoutesReq, [553](#)
  - numOfRoutes, [553](#)
  - pTransferStatusReport, [553](#)
  - routeList, [553](#)
- snr
  - LTESSInfo, [327](#)
  - satelliteInfo, [478](#)
- snrlevel
  - lteSnrinformation, [325](#)
- soMask
  - DataBearerTech, [179](#)
  - dataBearerTechnology, [181](#)
- sourceIPMask
  - TFTIDParams, [588](#)
- Specific Absorption Rate (SAR), [32](#)
- spn
  - nasPLMNNameResp, [357](#)
  - serviceProviderName, [481](#)
- spnEncoding
  - nasPLMNNameResp, [357](#)
- spnLength
  - nasPLMNNameResp, [357](#)
  - serviceProviderName, [481](#)
- srcPortRangeEnd
  - TFTIDParams, [588](#)
- srcPortRangeStart
  - TFTIDParams, [588](#)
- srvCapability
  - detailSvcInfo, [189](#)
  - sysInfoCommon, [583](#)
- srvCapabilityValid
  - sysInfoCommon, [583](#)
- srvDomain
  - sysInfoCommon, [583](#)
- srvDomainValid
  - sysInfoCommon, [583](#)
- srvOption
  - arrSvcOption, [100](#)
- srvStatus
  - detailSvcInfo, [189](#)
  - GSMSrvStatusInfo, [260](#)
  - SrvStatusInfo, [557](#)
- SrvStatusInfo, [557](#)
  - isPrefDataPath, [557](#)
  - srvStatus, [557](#)
- srxlev
  - cellParams, [142](#)
  - gsmCellInfo, [258](#)
  - umtsLTENbrCell, [619](#)
  - wcdmaCellInfo, [700](#)
- ssdatasession\_params, [557](#)
  - action, [559](#)
  - failureReason, [559](#)
  - failureReasonv4, [560](#)
  - failureReasonv6, [560](#)
  - instanceId, [560](#)
  - ipfamily, [560](#)
  - pAuthentication, [560](#)
  - pPassword, [560](#)
  - pProfileId3GPP, [560](#)
  - pProfileId3GPP2, [560](#)
  - pTechnology, [560](#)
  - pUsername, [560](#)
  - rcv4, [560](#)
  - rcv6, [560](#)
  - sessionId, [560](#)
  - v4sessionId, [560](#)
  - v6sessionId, [560](#)
  - verbFailReason, [560](#)
  - verbFailReasonType, [560](#)
- stage
  - UIMRefreshEvent, [609](#)
- StartPDSTrackingSessionExt
  - qaGobiApiPds.h, [974](#)

- State
  - NetworkStat1x, [369](#)
  - NetworkStatEVDO, [371](#)
- state
  - omaDmConfigTlv, [375](#)
  - omaDmConfigTlvExt, [378](#)
  - omaDmFotaTlv, [380](#)
  - omaDmFotaTlvExt, [383](#)
  - QosFlowInfoState, [452](#)
  - QosMap, [453](#)
  - slqsSessionStateInfo, [530](#)
- StatsMask
  - TransferStatInd, [590](#)
- statsMask
  - TrStatInd, [591](#)
- StatsPeriod
  - TransferStatInd, [590](#)
- statsPeriod
  - TrStatInd, [591](#)
- status
  - delAssistDataStatus, [185](#)
- statusChange
  - UIMStatusChangeInfo, [613](#)
- StopPDSTrackingSession
  - qaGobiApiPds.h, [975](#)
- storageIndex
  - ImageIdElement, [277](#)
- storageType
  - smsMaxStorageSizeReq, [547](#)
  - SMSMemoryInfo, [548](#)
  - SMSMTMessage, [549](#)
- subAddr
  - calledPartySubAdd, [110](#)
- subAddrLen
  - calledPartySubAdd, [110](#)
- subAddrType
  - calledPartySubAdd, [110](#)
- subnetMask
  - IPv4Addr, [300](#)
- subsType
  - voiceBindSubscriptionInfo, [639](#)
- SuppOA
  - CUGInfo, [162](#)
- SuppPrefCUG
  - CUGInfo, [162](#)
- supportedMsgLen
  - SupportedMsgList, [561](#)
- SupportedMsgList, [560](#)
  - supportedMsgLen, [561](#)
  - supportedMsgs, [561](#)
- supportedMsgs
  - SupportedMsgList, [561](#)
- svInfoMask
  - satelliteInfo, [478](#)
- svListLen
  - satelliteInfo, [478](#)
- svStatus
  - satelliteInfo, [478](#)
- svUsedforFix
  - qaGobiApiCbk.h, [754](#)
- svUsedforFix\_s, [564](#)
  - gnssSvUsedList, [564](#)
  - gnssSvUsedList\_len, [564](#)
- SvcClass
  - callFWExtInfo, [115](#)
  - callFWInfo, [116](#)
- SvcStatus
  - callFWExtInfo, [115](#)
  - callFWInfo, [116](#)
- svcType
  - ccSUPSType, [126](#)
  - SUPSType, [562](#)
- sw1
  - cardResult, [121](#)
- sw2
  - cardResult, [121](#)
- SwiDataTypes.h, [1158](#)
  - BOOL, [1159](#)
  - BYTE, [1159](#)
  - CHAR, [1159](#)
  - FLOAT, [1159](#)
  - INT32, [1159](#)
  - INT8, [1159](#)
  - LPCSTR, [1159](#)
  - SHORT, [1159](#)
  - SWI\_API, [1159](#)
  - ULONG, [1159](#)
  - ULONGLONG, [1159](#)
  - UNUSEDPARAM, [1159](#)
  - USHORT, [1159](#)
  - WORD, [1160](#)
- swiModemStatusResp, [566](#)
  - commonInfo, [566](#)
  - pLTEInfo, [566](#)
- SwiOTAMsg
  - qaGobiApiCbk.h, [754](#)
- SwiOTAMsg\_s, [566](#)
  - data, [567](#)
  - data\_len, [567](#)
  - pLteNasRelInfo, [567](#)
  - pTime, [567](#)
  - type, [567](#)
- swiPDPRuntimeSettingsReq, [567](#)
  - contextId, [568](#)
  - contextType, [568](#)
- swiPDPRuntimeSettingsResp, [568](#)
  - pAPNName, [570](#)
  - pBearerId, [570](#)
  - pContextId, [570](#)
  - pIPv4Address, [570](#)
  - pIPv4GWAddress, [570](#)
  - pIPv6Address, [570](#)
  - pIPv6GWAddress, [570](#)
  - pPrDNSIPv4Address, [570](#)
  - pPrDNSIPv6Address, [570](#)
  - pPrPCSCFIPv4Address, [571](#)

- pPrPCSCFIPv6Address, [571](#)
- pSeDNSIPv4Address, [571](#)
- pSeDNSIPv6Address, [571](#)
- pSePCSCFIPv4Address, [571](#)
- pSePCSCFIPv6Address, [571](#)
- swiQosFilter, [571](#)
  - index, [573](#)
  - pEspSpi, [573](#)
  - pIPv4DstAddr, [573](#)
  - pIPv4SrcAddr, [573](#)
  - pIPv6DstAddr, [573](#)
  - pIPv6Label, [573](#)
  - pIPv6SrcAddr, [573](#)
  - pIPv6TrafCls, [573](#)
  - pId, [573](#)
  - pNxtHdrProto, [573](#)
  - pPrecedence, [573](#)
  - pTCPDstPort, [573](#)
  - pTCPSrcPort, [573](#)
  - pTos, [574](#)
  - pTranDstPort, [574](#)
  - pTranSrcPort, [574](#)
  - pUDPDstPort, [574](#)
  - pUDPSrcPort, [574](#)
  - version, [574](#)
- swiQosFlow, [574](#)
  - index, [577](#)
  - p3GPP2Pri, [577](#)
  - p3GPPIImCn, [577](#)
  - p3GPPResResidualBER, [577](#)
  - p3GPPSigInd, [577](#)
  - p3GPPTraHdIPri, [577](#)
  - pDataRate, [577](#)
  - pJitter, [577](#)
  - pLatency, [577](#)
  - pLteQci, [577](#)
  - pMaxAllowedPktSz, [577](#)
  - pMinPolicedPktSz, [577](#)
  - pPktErrRate, [577](#)
  - pProfileId3GPP2, [577](#)
  - pTokenBucket, [577](#)
  - pTrafficClass, [577](#)
- swiQosGranted, [578](#)
  - pRxFlow, [578](#)
  - pTxFlow, [578](#)
- swiQosIds, [578](#)
  - pIds, [578](#)
  - sz, [578](#)
- swiQosModifyReq, [578](#)
  - id, [579](#)
  - pRxFilter, [579](#)
  - pRxFlow, [579](#)
  - pTxFilter, [579](#)
  - pTxFlow, [579](#)
- swiQosReq, [579](#)
  - index, [580](#)
  - pRxFilter, [580](#)
  - pRxFlow, [580](#)
  - pTxFilter, [580](#)
  - pTxFlow, [580](#)
- swiRMTrasferStaticsReq, [580](#)
  - bResetStatistics, [581](#)
  - ulMask, [581](#)
- switchOption
  - voiceALSSetLineSwitchInfo, [638](#)
- sysInfoCDMA
  - CDMASysInfo, [140](#)
- sysInfoCommon, [581](#)
  - isSysForbidden, [583](#)
  - isSysForbiddenValid, [583](#)
  - roamStatus, [583](#)
  - roamStatusValid, [583](#)
  - srvCapability, [583](#)
  - srvCapabilityValid, [583](#)
  - srvDomain, [583](#)
  - srvDomainValid, [583](#)
- sysInfoGSM
  - GSMSysInfo, [263](#)
- sysInfoHDR
  - HDRSysInfo, [274](#)
- sysInfoLTE
  - LTESysInfo, [331](#)
- sysInfoWCDMA
  - WCDMASysInfo, [711](#)
- sysSelectPrefInfo
  - qaGobiApiNas.h, [911](#)
- sysSelectPrefParams
  - qaGobiApiNas.h, [914](#)
- system
  - satelliteInfo, [478](#)
  - SV, [563](#)
- SystemID
  - qaQmiServingSystemParam, [427](#)
- systemID
  - CDMASysInfo, [140](#)
- systemMode
  - CommInfo, [153](#)
- sz
  - swiQosIds, [578](#)
- TDSCDMAECIOThresh, [583](#)
  - TDSCDMAECIOThreshListLen
  - TDSCDMAECIOThresh, [584](#)
- TDSCDMARSCPTthresh, [584](#)
  - TDSCDMARSCPTthreshListLen
  - TDSCDMARSCPTthresh, [584](#)
- TDSCDMARSSIThresh, [584](#)
  - TDSCDMARSSIThreshListLen
  - TDSCDMARSSIThresh, [585](#)
- TDSCDMASINRThresh, [586](#)
  - TDSCDMASINRThreshListLen
  - TDSCDMASINRThresh, [586](#)
- TDSCDMASigInfoExt, [585](#)
  - ecio, [585](#)
  - rscp, [585](#)
  - rsi, [585](#)
  - sinr, [585](#)

- tFNASwiLTECphyCallInfo  
    qaGobiApiCbK.h, [756](#)
- tFNASwiOTAMsg  
    qaGobiApiCbK.h, [756](#)
- tFNActivationStatus  
    qaGobiApiCbK.h, [755](#)
- tFNAllCallStatus  
    qaGobiApiCbK.h, [756](#)
- tFNAsyncRawSend  
    qaGobiApiCbK.h, [756](#)
- tFNBandPreference  
    qaGobiApiCbK.h, [758](#)
- tFNCATEvent  
    qaGobiApiCbK.h, [760](#)
- tFNDTMFEvent  
    qaGobiApiCbK.h, [762](#)
- tFNDUNCallInfo  
    qaGobiApiCbK.h, [762](#)
- tFNDataCapabilities  
    qaGobiApiCbK.h, [760](#)
- tFNDataSysStatus  
    qaGobiApiCbK.h, [761](#)
- tFNDelAssistData  
    qaGobiApiCbK.h, [761](#)
- tFNDeviceStateChange  
    qaGobiApiCbK.h, [761](#)
- tFNEventPosition  
    qaGobiApiCbK.h, [762](#)
- tFNFwDldCompletion  
    qaGobiApiCbK.h, [762](#)
- tFNGnssSvInfo  
    qaGobiApiCbK.h, [762](#)
- tFNHDRPersonality  
    qaGobiApiCbK.h, [763](#)
- tFNImRegMgrConfig  
    qaGobiApiCbK.h, [764](#)
- tFNImSIPConfig  
    qaGobiApiCbK.h, [764](#)
- tFNImSMSConfig  
    qaGobiApiCbK.h, [764](#)
- tFNImUserConfig  
    qaGobiApiCbK.h, [764](#)
- tFNImVoIPConfig  
    qaGobiApiCbK.h, [764](#)
- tFNImsaPdpStatus  
    qaGobiApiCbK.h, [763](#)
- tFNImsaRatStatus  
    qaGobiApiCbK.h, [763](#)
- tFNImsaRegStatus  
    qaGobiApiCbK.h, [763](#)
- tFNImsaSvcStatus  
    qaGobiApiCbK.h, [763](#)
- tFNInfoRec  
    qaGobiApiCbK.h, [765](#)
- tFNInjectSensorData  
    qaGobiApiCbK.h, [765](#)
- tFNInjectTimeStatus  
    qaGobiApiCbK.h, [765](#)
- tFNLURject  
    qaGobiApiCbK.h, [765](#)
- tFNMemoryFull  
    qaGobiApiCbK.h, [765](#)
- tFNMessageWaiting  
    qaGobiApiCbK.h, [766](#)
- tFNMobileIPStatus  
    qaGobiApiCbK.h, [766](#)
- tFNModemTempInfo  
    qaGobiApiCbK.h, [766](#)
- tFNNet  
    qaGobiApiCbK.h, [766](#)
- tFNNetworkTime  
    qaGobiApiCbK.h, [767](#)
- tFNNewGPS  
    qaGobiApiCbK.h, [767](#)
- tFNNewNMEA  
    qaGobiApiCbK.h, [767](#)
- tFNNewRMTransferStatistics  
    qaGobiApiCbK.h, [767](#)
- tFNNewSMS  
    qaGobiApiCbK.h, [768](#)
- tFNOMADMState  
    qaGobiApiCbK.h, [768](#)
- tFNOTASPStatus  
    qaGobiApiCbK.h, [769](#)
- tFNOpMode  
    qaGobiApiCbK.h, [769](#)
- tFNPDSSState  
    qaGobiApiCbK.h, [771](#)
- tFNPacketSrvState  
    qaGobiApiCbK.h, [769](#)
- tFNPower  
    qaGobiApiCbK.h, [771](#)
- tFNPrivacyChange  
    qaGobiApiCbK.h, [771](#)
- tFNQosNWStatus  
    qaGobiApiCbK.h, [772](#)
- tFNQosPriEvent  
    qaGobiApiCbK.h, [772](#)
- tFNQosStatus  
    qaGobiApiCbK.h, [772](#)
- tFNRFInfo  
    qaGobiApiCbK.h, [773](#)
- tFNRoamingIndicator  
    qaGobiApiCbK.h, [774](#)
- tFNSDKTerminated  
    qaGobiApiCbK.h, [774](#)
- tFNSLQSOADMAlert  
    qaGobiApiCbK.h, [775](#)
- tFNSLQSQOSEvent  
    qaGobiApiCbK.h, [775](#)
- tFNSLQSSessionState  
    qaGobiApiCbK.h, [777](#)
- tFNSLQSSignalStrengths  
    qaGobiApiCbK.h, [777](#)
- tFNSLQSWDSEvent  
    qaGobiApiCbK.h, [777](#)

- tFNSMSEvents
  - qaGobiApiCbK.h, [777](#)
- tFNSUPSInfo
  - qaGobiApiCbK.h, [777](#)
- tFNSUPSNotification
  - qaGobiApiCbK.h, [778](#)
- tFNSensorStreaming
  - qaGobiApiCbK.h, [774](#)
- tFNServingSystem
  - qaGobiApiCbK.h, [774](#)
- tFNSetCradleMount
  - qaGobiApiCbK.h, [775](#)
- tFNSetEventTimeSync
  - qaGobiApiCbK.h, [775](#)
- tFNSigInfo
  - qaGobiApiCbK.h, [775](#)
- tFNSignalStrength
  - qaGobiApiCbK.h, [775](#)
- tFNSysInfo
  - qaGobiApiCbK.h, [778](#)
- tFNSysSelectionPref
  - qaGobiApiCbK.h, [778](#)
- tFNUIMRefresh
  - qaGobiApiCbK.h, [780](#)
- tFNUIMStatusChangeInfo
  - qaGobiApiCbK.h, [780](#)
- tFNUSSDNoWaitIndication
  - qaGobiApiCbK.h, [780](#)
- tFNUSSDNotification
  - qaGobiApiCbK.h, [780](#)
- tFNUSSDRelease
  - qaGobiApiCbK.h, [780](#)
- tFNtransLayerInfo
  - qaGobiApiCbK.h, [778](#)
- tFNtransNWRegInfo
  - qaGobiApiCbK.h, [778](#)
- TFTIDParams, [586](#)
  - destPortRangeEnd, [587](#)
  - destPortRangeStart, [588](#)
  - eValid, [588](#)
  - filterId, [588](#)
  - flowLabel, [588](#)
  - IPSECSPi, [588](#)
  - ipVersion, [588](#)
  - nextHeader, [588](#)
  - pSourceIP, [588](#)
  - sourceIPMask, [588](#)
  - srcPortRangeEnd, [588](#)
  - srcPortRangeStart, [588](#)
  - tosMask, [588](#)
- THIRD\_INSTANCE
  - qaGobiApiCbK.h, [743](#)
- TIME\_DATE\_BUF
  - qaGobiApiSms.h, [988](#)
- TIME\_STAMP\_BUF
  - qaGobiApiSms.h, [988](#)
- TPCause
  - SMSAsyncRawSend\_s, [542](#)
- TX\_PWR
  - NetworkStat1x, [369](#)
- TXAGCList, [592](#)
  - pTXAIG, [593](#)
  - pTXComprSlope, [593](#)
  - pTXComprThres, [593](#)
  - pTXExpSlope, [593](#)
  - pTXExpThres, [593](#)
  - pTXStaticGain, [593](#)
- TXChan
  - LTEInfo, [315](#)
- TXOKBytesCount
  - DUNCallInfoInd, [196](#)
- TXPCMIIRFltr, [594](#)
  - pFlag, [596](#)
  - pStage0Val, [596](#)
  - pStage1Val, [596](#)
  - pStage2Val, [596](#)
  - pStage3Val, [596](#)
  - pStage4Val, [596](#)
  - pStageCnt, [596](#)
- Tables, [43](#)
- tac
  - LTEInfoIntraFreq, [318](#)
  - LTESysInfo, [331](#)
- tacValid
  - LTESysInfo, [331](#)
- tech
  - NWProfile, [374](#)
- techName
  - \_packetSrvStatus, [52](#)
- techType
  - DataBearerTech, [179](#)
- Technology
  - DeviceConfigDetail, [190](#)
  - fwinfo\_s, [212](#)
- temperature
  - CommInfo, [153](#)
- textMsgLength
  - cdmaMsgEncodingParams, [135](#)
- threshGsmHigh
  - lteGsmCellInfo, [312](#)
- threshGsmLow
  - lteGsmCellInfo, [312](#)
- threshServingLow
  - LTEInfoIntraFreq, [318](#)
- threshXHigh
  - infoInterFreq, [299](#)
- threshXLow
  - infoInterFreq, [299](#)
- threshXhigh
  - lteWcdmaCellInfo, [332](#)
- threshXlow
  - lteWcdmaCellInfo, [332](#)
- thresholds
  - SignalStrengthDataType, [517](#)
- thresholdsSize
  - SignalStrengthDataType, [517](#)



- Time
  - wcdmaLongMsgDecodingParams, 703
  - wcdmaMsgDecodingParams, 705
- Time\_uncert\_ms
  - GPSSStateInfo, 256
- TimeStmp\_gps\_week
  - GPSSStateInfo, 256
- TimeStmp\_tow\_ms
  - GPSSStateInfo, 256
- timeSyncRefCounter
  - QmiCbkLocEventTimeSyncInd, 429
- timingAdvance
  - GERANInfo, 214
- TlvPresent
  - CatCommonEventTlv, 123
  - DataUlongLongTlv, 184
  - DataUlongTlv, 184
  - PhyCaAggPcellInfo, 394
  - PhyCaAggScellIDBw, 395
  - PhyCaAggScellIndex, 395
  - PhyCaAggScellIndType, 396
  - PhyCaAggScellInfo, 397
  - RoamingInfo, 464
  - sessionInfoTlv, 486
  - sessionInfoTlvExt, 486
- toServiceId
  - BroadcastConfig, 104
- toggleMode
  - lineCtrlInfo, 304
- tokenBucket, 588
  - bucketSz, 588
  - peakRate, 588
  - tokenRate, 588
- tokenRate
  - tokenBucket, 588
- Tos, 589
  - mask, 589
  - val, 589
- tosMask
  - TFTIDParams, 588
- total\_rx\_bytes
  - sQosStat, 556
- total\_rx\_pkt
  - sQosStat, 556
- total\_tx\_bytes
  - sQosStat, 556
- total\_tx\_bytes\_drp
  - sQosStat, 556
- total\_tx\_pkt
  - sQosStat, 556
- total\_tx\_pkt\_drp
  - sQosStat, 556
- TrStatInd, 590
  - statsMask, 591
  - statsPeriod, 591
- trackAreaCode
  - qaQmiServingSystemParam, 427
- trafficClass
  - UMTSMinQoS, 622
  - UMTSQoS, 625
- trafficPriority
  - UMTSMinQoS, 622
  - UMTSQoS, 625
- TransCap
  - \_transLayerinfo, 80
- transLayerInfo
  - qaGobiApiSms.h, 992
- transLayerNotification
  - qaGobiApiCbk.h, 781
- transNWRegInfoNotification
  - qaGobiApiCbk.h, 781
- TransType
  - \_transLayerinfo, 80
- transactionID
  - SMSTransferRouteMTMessage, 554
- transferDelay
  - UMTSMinQoS, 622
  - UMTSQoS, 625
- TransferStatInd, 589
  - StatsMask, 590
  - StatsPeriod, 590
- TransferStatsDataType, 590
  - interval, 590
- trueIMSI, 591
  - imsiT1112, 592
  - imsiTS1, 592
  - imsiTS2, 592
  - imsiTaddrNum, 592
  - mccT, 592
- trueSrvStatus
  - GSMSrvStatusInfo, 260
- tx\_bytes
  - NetStats, 364
  - sQosFlowStat, 555
- tx\_bytes\_drp
  - sQosFlowStat, 555
- tx\_errors
  - NetStats, 364
- tx\_overflows
  - NetStats, 364
- tx\_packets
  - NetStats, 364
- tx\_pkt
  - sQosFlowStat, 555
- tx\_pkt\_drp
  - sQosFlowStat, 555
- TxDropConutTlv
  - QmiCbkWdsStatisticsIndState, 440
- txInfo, 593
  - isInTraffic, 594
  - txPower, 594
- TxOkByteCountTlv
  - QmiCbkWdsStatisticsIndState, 440
- TxOkConutTlv
  - QmiCbkWdsStatisticsIndState, 440
- txPower



- txInfo, [594](#)
- type
  - SwiOTAMsg\_s, [567](#)
- UATISIZE
  - qaGobiApiNas.h, [909](#)
- UIMAuthenticateReq, [596](#)
  - authData, [597](#)
  - pIndicationToken, [597](#)
  - sessionInfo, [597](#)
- UIMAuthenticateResp, [597](#)
  - pAuthenticateResult, [598](#)
  - pCardResult, [598](#)
  - pIndicationToken, [598](#)
- UIMChangePIN
  - qaGobiApiDms.h, [866](#)
- UIMChangePinReq, [598](#)
  - changePIN, [598](#)
  - pIndicationToken, [598](#)
  - pKeyReferenceID, [599](#)
  - sessionInfo, [599](#)
- UIMDepersonalizationReq, [599](#)
  - depersonalisationInfo, [599](#)
- UIMDepersonalizationResp, [599](#)
  - pRemainingRetries, [599](#)
- UIMEventRegisterReqResp, [599](#)
  - eventMask, [600](#)
- UIMGetCardStatusResp, [600](#)
  - pCardStatus, [600](#)
  - pHotSwapStatus, [600](#)
- UIMGetControlKeyStatus
  - qaGobiApiDms.h, [867](#)
- UIMGetFileAttributesReq, [601](#)
  - fileIndex, [601](#)
  - pIndicationToken, [601](#)
  - sessionInfo, [601](#)
- UIMGetFileAttributesResp, [601](#)
  - pCardResult, [602](#)
  - pFileAttributes, [602](#)
  - pIndicationToken, [602](#)
- UIMGetICCID
  - qaGobiApiDms.h, [868](#)
- UIMGetPINStatus
  - qaGobiApiDms.h, [869](#)
- UIMPinResp, [602](#)
  - pEncryptedPIN1, [603](#)
  - pIndicationToken, [603](#)
  - pRemainingRetries, [603](#)
- UIMPowerDownReq, [603](#)
  - slot, [603](#)
- UIMPowerUpReq, [603](#)
  - pIgnoreHotSwapSwitch, [604](#)
  - slot, [604](#)
- UIMReadTransparentReq, [604](#)
  - fileIndex, [605](#)
  - pEncryptData, [605](#)
  - pIndicationToken, [605](#)
  - readTransparent, [605](#)
  - sessionInfo, [605](#)
- UIMReadTransparentResp, [605](#)
  - pCardResult, [606](#)
  - pEncryptedData, [606](#)
  - pIndicationToken, [606](#)
  - pReadResult, [606](#)
- UIMRefreshCompleteReq, [606](#)
  - refreshComplete, [607](#)
  - sessionInfo, [607](#)
- UIMRefreshEvent, [607](#)
  - aid, [609](#)
  - aidLength, [609](#)
  - arrfileInfo, [609](#)
  - mode, [609](#)
  - numOfFiles, [609](#)
  - sessionType, [609](#)
  - stage, [609](#)
- UIMRefreshGetLastEventReq, [609](#)
  - sessionInfo, [609](#)
- UIMRefreshGetLastEventResp, [609](#)
  - pRefreshEvent, [610](#)
- UIMRefreshOKReq, [610](#)
  - OKtoRefresh, [610](#)
  - sessionInfo, [610](#)
- UIMRefreshRegisterReq, [610](#)
  - regRefresh, [611](#)
  - sessionInfo, [611](#)
- UIMSessionInformation, [611](#)
  - aid, [612](#)
  - aidLength, [612](#)
  - sessionType, [612](#)
- UIMSetControlKeyProtection
  - qaGobiApiDms.h, [870](#)
- UIMSetPINProtection
  - qaGobiApiDms.h, [871](#)
- UIMSetPinProtectionReq, [612](#)
  - pIndicationToken, [613](#)
  - pKeyReferenceID, [613](#)
  - pinProtection, [613](#)
  - sessionInfo, [613](#)
- UIMStatusChangeInfo, [613](#)
  - statusChange, [613](#)
- UIMUnblockControlKey
  - qaGobiApiDms.h, [872](#)
- UIMUnblockPIN
  - qaGobiApiDms.h, [873](#)
- UIMUnblockPinReq, [613](#)
  - pIndicationToken, [614](#)
  - pKeyReferenceID, [614](#)
  - sessionInfo, [614](#)
  - unblockPIN, [614](#)
- UIMVerifyPIN
  - qaGobiApiDms.h, [874](#)
- UIMVerifyPinReq, [614](#)
  - pEncryptedPIN1, [615](#)
  - pIndicationToken, [615](#)
  - pKeyReferenceID, [615](#)
  - sessionInfo, [615](#)
  - verifyPIN, [615](#)

- ULONG
  - SwiDataTypes.h, [1159](#)
- ULONGLONG
  - SwiDataTypes.h, [1159](#)
- UMTSInfo, [615](#)
  - cellID, [617](#)
  - ecio, [617](#)
  - geranInst, [617](#)
  - GeranInstInfo, [617](#)
  - lac, [617](#)
  - plmn, [617](#)
  - psc, [617](#)
  - rscf, [617](#)
  - UMTSInstInfo, [617](#)
  - uarfcn, [617](#)
  - umtsInst, [617](#)
- UMTSInstInfo
  - UMTSInfo, [617](#)
- UMTSLTENbrCell
  - WCDMAInfoLTENeighborCell, [701](#)
- UMTSMinQoS, [619](#)
  - deliveryErrSDU, [622](#)
  - grntDownlinkBitrate, [622](#)
  - grntUplinkBitrate, [622](#)
  - maxDownlinkBitrate, [622](#)
  - maxSDUSize, [622](#)
  - maxUplinkBitrate, [622](#)
  - qosDeliveryOrder, [622](#)
  - resBerRatio, [622](#)
  - sduErrorRatio, [622](#)
  - trafficClass, [622](#)
  - trafficPriority, [622](#)
  - transferDelay, [622](#)
- UMTSQoS, [622](#)
  - deliveryErrSDU, [625](#)
  - grntDownlinkBitrate, [625](#)
  - grntUplinkBitrate, [625](#)
  - maxDownlinkBitrate, [625](#)
  - maxSDUSize, [625](#)
  - maxUplinkBitrate, [625](#)
  - qosDeliveryOrder, [625](#)
  - resBerRatio, [625](#)
  - sduErrorRatio, [625](#)
  - trafficClass, [625](#)
  - trafficPriority, [625](#)
  - transferDelay, [625](#)
- UMTSReqQoS
  - UMTSReqQoSSigInd, [626](#)
- UMTSReqQoSSigInd, [626](#)
  - SigInd, [626](#)
  - UMTSReqQoS, [626](#)
- UMTSinstInfo, [617](#)
  - umtsEcio, [618](#)
  - umtsPsc, [618](#)
  - umtsRscf, [618](#)
  - umtsUarfcn, [618](#)
- UNIQUE\_ID\_LEN
  - qaGobiApiDms.h, [837](#)
- UNUSEDPARAM
  - SwiDataTypes.h, [1159](#)
- uResult
  - sGetDeviceSeriesResult, [514](#)
- USBCompConfig, [629](#)
  - pUSBComp, [629](#)
- USBCompParams, [629](#)
  - pNumSupUSBComps, [632](#)
  - pSupUSBComps, [632](#)
  - pUSBComp, [632](#)
- USHORT
  - SwiDataTypes.h, [1159](#)
- USSD\_DCS\_8BIT
  - qaGobiApiCbk.h, [743](#)
- USSD\_DCS\_ASCII
  - qaGobiApiCbk.h, [743](#)
- USSD\_DCS\_UCS2
  - qaGobiApiCbk.h, [743](#)
- USSDNoWaitIndicationInfo, [632](#)
  - pAlphaIdentifier, [632](#)
  - pError, [632](#)
  - pFailureCause, [632](#)
  - pUSSDData, [632](#)
- USSDRespFNetwork, [632](#)
  - pRespData, [634](#)
  - pTypeCode, [634](#)
- USSInfo, [634](#)
  - ussDCS, [634](#)
  - ussData, [634](#)
  - ussLen, [634](#)
- USSInformation
  - voiceOrigUSSDNoWaitInfo, [679](#)
- USSResp, [634](#)
  - pAlphaIDInfo, [635](#)
  - pCCSuppsType, [635](#)
  - pCallId, [635](#)
  - pCcResultType, [635](#)
  - pUSSDInfo, [635](#)
  - pfailureCause, [635](#)
- UUSData
  - UUSInfo, [636](#)
- UUSDataLen
  - UUSInfo, [636](#)
- UUSDcs
  - UUSInfo, [636](#)
- UUSInfo, [635](#)
  - UUSData, [636](#)
  - UUSDataLen, [636](#)
  - UUSDcs, [636](#)
  - UUSType, [636](#)
- UUSType
  - UUSInfo, [636](#)
- uarfcn
  - lteWcdmaCellInfo, [332](#)
  - UMTSInfo, [617](#)
- ueInIdle
  - LTEInfoInterfreq, [316](#)
  - LTEInfoIntrafreq, [318](#)

- LTEInfoNeighboringGSM, [319](#)
  - LTEInfoNeighboringWCDMA, [320](#)
- ulData
  - DataULongTlv, [184](#)
- ulMask
  - swiRMTrasferStaticsReq, [581](#)
- ullData
  - DataULongLongTlv, [184](#)
- umtsEcio
  - UMTSinstInfo, [618](#)
- umtsInst
  - UMTSInfo, [617](#)
- umtsLTENbrCell, [618](#)
  - cellsTDD, [619](#)
  - earfcn, [619](#)
  - pci, [619](#)
  - rsrp, [619](#)
  - rsrq, [619](#)
  - srxlev, [619](#)
- umtsLTENbrCellLen
  - WCDMAInfoLTENeighborCell, [701](#)
- umtsPsc
  - UMTSinstInfo, [618](#)
- umtsRscp
  - UMTSinstInfo, [618](#)
- umtsUarfcn
  - UMTSinstInfo, [618](#)
- unblockLeft
  - remainingRetries, [459](#)
- unblockPIN
  - UIMUnblockPinReq, [614](#)
- unblockUIMPIN, [626](#)
  - newPINLen, [627](#)
  - newPINVal, [627](#)
  - pinID, [627](#)
  - pukLen, [627](#)
  - pukVal, [627](#)
- uniqueID
  - CurrImageInfo, [167](#)
- univPin
  - appStatus, [93](#)
- UniversalTime, [627](#)
  - day, [628](#)
  - dayOfWeek, [628](#)
  - hour, [628](#)
  - minute, [628](#)
  - month, [628](#)
  - second, [628](#)
  - year, [628](#)
- universalTime
  - nasNetworkTime, [352](#)
- upLink
  - NSSAudioCtrl, [374](#)
- updateCompleteStatus
  - omaDmFotaTlv, [380](#)
- upgrade\_mc77xx\_fw
  - qaGobiApiFms.h, [890](#)
- UpgradeFirmware2k
  - qaGobiApiFms.h, [890](#)
- upinRetries
  - slotInfo, [518](#)
- upinState
  - slotInfo, [518](#)
- UpkQmiCbkCatEventReportInd
  - qaCbkCatEventReportInd.h, [729](#)
- UpkQmiCbkSwiOmaDmEventReportInd
  - qaCbkSwiOmaDmEventReportInd.h, [730](#)
- UpkQmiCbkSwiOmaDmEventReportIndExt
  - qaCbkSwiOmaDmEventReportInd.h, [730](#)
- UpkQmiNasGetRFBandInfo
  - qaNasGetRFBandInfo.h, [1150](#)
- UpkQmiNasPerformNetworkScan
  - qaNasPerformNetworkScan.h, [1151](#)
- upukRetries
  - slotInfo, [518](#)
- usageMask
  - sensorDataUsage\_s, [479](#)
- User Identity Module Service (UIM), [36](#)
- userData
  - SMSAsyncRawSend\_s, [542](#)
- userInputReq
  - omaDmConfigTlv, [375](#)
  - omaDmConfigTlvExt, [378](#)
  - omaDmFotaTlv, [380](#)
- userInputTimeout
  - omaDmConfigTlv, [375](#)
  - omaDmConfigTlvExt, [378](#)
  - omaDmFotaTlv, [380](#)
  - omaDmFotaTlvExt, [383](#)
- ussDCS
  - USSInfo, [634](#)
- ussData
  - USSInfo, [634](#)
- ussLen
  - USSInfo, [634](#)
- uusInfo
  - allCallsUUSInfo, [88](#)
- v4sessionId
  - qaQmiInterfaceInfo, [423](#)
  - ssdatasession\_params, [560](#)
  - WdsRunTimeSettings, [720](#)
- v6sessionId
  - qaQmiInterfaceInfo, [423](#)
  - ssdatasession\_params, [560](#)
  - WdsRunTimeSettings, [720](#)
- VOICE\_SUPS\_SRV\_CLASS\_DATA
  - qaGobiApiVoice.h, [1080](#)
- VOICE\_SUPS\_SRV\_CLASS\_DATACIRCUITASYNC
  - qaGobiApiVoice.h, [1080](#)
- VOICE\_SUPS\_SRV\_CLASS\_DATACIRCUITSYNC
  - qaGobiApiVoice.h, [1080](#)
- VOICE\_SUPS\_SRV\_CLASS\_FAX
  - qaGobiApiVoice.h, [1080](#)
- VOICE\_SUPS\_SRV\_CLASS\_NONE
  - qaGobiApiVoice.h, [1080](#)
- VOICE\_SUPS\_SRV\_CLASS\_PACKETACCESS

- qaGobiApiVoice.h, [1080](#)
- VOICE\_SUPS\_SRV\_CLASS\_PADACCESS
  - qaGobiApiVoice.h, [1080](#)
- VOICE\_SUPS\_SRV\_CLASS\_SMS
  - qaGobiApiVoice.h, [1080](#)
- VOICE\_SUPS\_SRV\_CLASS\_VOICE
  - qaGobiApiVoice.h, [1080](#)
- VDOP
  - precisionDilution\_s, [403](#)
- VOICE\_SRV
  - qaGobiApiCbK.h, [743](#)
- val
  - IPv6TrafCls, [303](#)
  - Tos, [589](#)
- ValidMask
  - GPSSStateInfo, [256](#)
- validMask
  - satelliteInfo, [478](#)
- ValidateSPC
  - qaGobiApiDms.h, [875](#)
- ValidityCW0
  - LteCQIParm, [311](#)
- ValidityCW1
  - LteCQIParm, [311](#)
- Value
  - GetM2MSpkrGainResp, [244](#)
  - SetM2MSpkrGainReq, [504](#)
- value\_length
  - custSettingInfo, [175](#)
  - setCustomSettingV2, [493](#)
- verbFailReason
  - ssdatasession\_params, [560](#)
- verbFailReasonType
  - ssdatasession\_params, [560](#)
- verboseSessnEndReason
  - \_packetSrvStatus, [52](#)
- verboseSessnEndReasonType
  - \_packetSrvStatus, [52](#)
- verifyLeft
  - remainingRetries, [459](#)
- verifyPIN
  - UIMVerifyPinReq, [615](#)
- verifyUIMPIN, [636](#)
  - pinID, [637](#)
  - pinLen, [637](#)
  - pinVal, [637](#)
- version
  - omaDmFotaTlv, [380](#)
  - omaDmFotaTlvExt, [383](#)
  - swiQosFilter, [574](#)
- versionlength
  - omaDmFotaTlv, [380](#)
  - omaDmFotaTlvExt, [383](#)
- VerticalUncertainty
  - GPSSStateInfo, [256](#)
- VirtStream
  - protocolSubtypeElement, [421](#)
- Voice Service (VOICE), [34](#)
- voiceALSSelectLineInfo, [637](#)
  - lineValue, [638](#)
- voiceALSSetLineSwitchInfo, [638](#)
  - switchOption, [638](#)
- voiceAnswerCall, [638](#)
  - pCallId, [639](#)
- voiceBindSubscriptionInfo, [639](#)
  - subsType, [639](#)
- voiceBurstDTMFInfo, [639](#)
  - BurstDTMFInfo, [640](#)
  - pBurstDTMFLengths, [640](#)
- voiceCallInfoReq, [640](#)
  - callID, [640](#)
- voiceCallInfoResp, [640](#)
  - pAlertType, [643](#)
  - pAlertingPattern, [643](#)
  - pAlphaIDInfo, [643](#)
  - pCallInfo, [643](#)
  - pConnectNumInfo, [643](#)
  - pDiagInfo, [643](#)
  - pOTASPStatus, [644](#)
  - pRemotePartyName, [644](#)
  - pRemotePartyNum, [644](#)
  - pSrvOpt, [644](#)
  - pUUSInfo, [644](#)
  - pVoicePrivacy, [644](#)
- voiceCallRequestParams, [644](#)
  - callNumber, [646](#)
  - pCLIRType, [646](#)
  - pCUGInfo, [646](#)
  - pCallPartySubAdd, [646](#)
  - pCallType, [646](#)
  - pEmergencyCategory, [646](#)
  - pSvcType, [646](#)
  - pUUSInfo, [646](#)
- voiceCallResponseParams, [646](#)
  - pAlphaIDInfo, [647](#)
  - pCCResultType, [647](#)
  - pCCSUPSType, [647](#)
  - pCallID, [647](#)
- voiceContDTMFInfo, [647](#)
  - DTMFdigit, [648](#)
  - pCallID, [648](#)
- voiceDTMFEventInfo, [648](#)
  - DTMFInformation, [649](#)
  - pOffLength, [649](#)
  - pOnLength, [649](#)
- voiceFlashInfo, [649](#)
  - pCallID, [650](#)
  - pFlashPayLd, [650](#)
  - pFlashType, [650](#)
- voiceGetAllCallInfo, [650](#)
  - pArrAlertingPattern, [652](#)
  - pArrAlertingType, [652](#)
  - pArrAlphaID, [652](#)
  - pArrCallEndReason, [652](#)
  - pArrCallInfo, [652](#)
  - pArrCalledPartyNum, [652](#)

- pArrConnectPartyNum, [652](#)
- pArrDiagInfo, [652](#)
- pArrRedirPartyNum, [652](#)
- pArrRemotePartyName, [652](#)
- pArrRemotePartyNum, [652](#)
- pArrSvcOption, [653](#)
- pArrUUSInfo, [653](#)
- pOTASPStatus, [653](#)
- pVoicePrivacy, [653](#)
- voiceGetCLIPResp, [660](#)
  - pAlphaIDInfo, [662](#)
  - pCCResType, [662](#)
  - pCCSUPSType, [662](#)
  - pCLIPResp, [663](#)
  - pCallID, [662](#)
  - pFailCause, [663](#)
- voiceGetCLIRResp, [663](#)
  - pAlphaIDInfo, [664](#)
  - pCCResType, [664](#)
  - pCCSUPSType, [664](#)
  - pCLIRResp, [664](#)
  - pCallID, [664](#)
  - pFailCause, [664](#)
- voiceGetCNAPResp, [664](#)
  - pAlphaIDInfo, [665](#)
  - pCCResType, [665](#)
  - pCCSUPSType, [665](#)
  - pCNAPResp, [666](#)
  - pCallID, [665](#)
  - pFailCause, [666](#)
- voiceGetCOLPResp, [666](#)
  - pAlphaIDInfo, [667](#)
  - pCCResType, [667](#)
  - pCCSUPSType, [667](#)
  - pCOLPResp, [667](#)
  - pCallID, [667](#)
  - pFailCause, [667](#)
- voiceGetCOLRResp, [667](#)
  - pAlphaIDInfo, [668](#)
  - pCCResType, [668](#)
  - pCCSUPSType, [668](#)
  - pCOLRResp, [669](#)
  - pCallID, [668](#)
  - pFailCause, [669](#)
- voiceGetCallBarringReq, [653](#)
  - pSvcClass, [654](#)
  - reason, [654](#)
- voiceGetCallBarringResp, [654](#)
  - pAlphaIDInfo, [655](#)
  - pCCResType, [655](#)
  - pCCSUPSType, [655](#)
  - pCallID, [655](#)
  - pFailCause, [655](#)
  - pSvcClass, [655](#)
- voiceGetCallFWReq, [655](#)
  - pSvcClass, [657](#)
  - Reason, [657](#)
- voiceGetCallFWResp, [657](#)
- pAlphaIDInfo, [658](#)
- pCCResType, [659](#)
- pCCSUPSType, [659](#)
- pCallID, [658](#)
- pFailCause, [659](#)
- pGetCallFWExtInfo, [659](#)
- pGetCallFWInfo, [659](#)
- voiceGetCallWaitInfo, [659](#)
  - pAlphaIDInfo, [660](#)
  - pCCResType, [660](#)
  - pCCSUPSType, [660](#)
  - pCallID, [660](#)
  - pFailCause, [660](#)
  - pSvcClass, [660](#)
- voiceGetConfigReq, [669](#)
  - pAMRStatus, [670](#)
  - pAirTimer, [670](#)
  - pAutoAnswer, [670](#)
  - pNamID, [670](#)
  - pPrefVoicePrivacy, [670](#)
  - pPrefVoiceSO, [670](#)
  - pRoamTimer, [670](#)
  - pTTYMode, [670](#)
  - pVoiceDomainPref, [671](#)
- voiceGetConfigResp, [671](#)
  - pAirTimerCnt, [672](#)
  - pAutoAnswerStat, [672](#)
  - pCurAMRConfig, [673](#)
  - pCurPrefVoiceSO, [673](#)
  - pCurVoiceDomainPref, [673](#)
  - pCurVoicePrivacyPref, [673](#)
  - pCurrTTYMode, [673](#)
  - pRoamTimerCnt, [673](#)
- voiceIndicationRegisterInfo, [673](#)
  - pRegDTMFEvents, [674](#)
  - pRegVoicePrivacyEvents, [674](#)
  - pSuppsNotifEvents, [674](#)
- voiceInfoRec, [674](#)
  - callID, [676](#)
  - pCLIRCause, [676](#)
  - pCallWaitInd, [676](#)
  - pCalledPartyInfo, [676](#)
  - pCallerIDInfo, [676](#)
  - pCallerNameInfo, [676](#)
  - pCallingPartyInfo, [676](#)
  - pConnectNumInfo, [676](#)
  - pDispInfo, [676](#)
  - pExtDispInfo, [676](#)
  - pExtDispRecInfo, [676](#)
  - pLineCtrlInfo, [676](#)
  - pNSSAudioCtrl, [676](#)
  - pNSSRelease, [676](#)
  - pRedirNumInfo, [676](#)
  - pSignalInfo, [676](#)
- voiceManageCallsReq, [676](#)
  - pCallID, [678](#)
  - SUPSType, [678](#)
- voiceManageCallsResp, [678](#)

- pFailCause, 678
- voiceOTASPStatusInfo, 679
  - callID, 680
  - OTASPStatus, 680
- voiceOrigUSSDNoWaitInfo, 678
  - USSInformation, 679
- voicePrivacy
  - voicePrivacyInfo, 680
- voicePrivacyInfo, 680
  - callID, 680
  - voicePrivacy, 680
- voiceSUPSInfo, 694
  - pAlphaIDInfo, 696
  - pCLIPstatus, 696
  - pCLIRstatus, 696
  - pCNAPstatus, 696
  - pCOLPstatus, 696
  - pCOLRstatus, 696
  - pCallBarPasswd, 696
  - pCallFWNum, 696
  - pCallFWTimerVal, 696
  - pCallFwdInfo, 696
  - pCallID, 696
  - pDataSrc, 696
  - pFailCause, 696
  - pNewPwdData, 697
  - pReason, 697
  - pSvcClass, 697
  - pUSSInfo, 697
  - SUPSInformation, 697
- voiceSUPSNotification, 697
  - callID, 699
  - notifType, 699
  - pCUGIndex, 699
  - pECTNum, 699
- voiceSetAllCallStatusCbkJInfo, 680
  - arrCallInfomation, 682
  - pArrAlertingPattern, 682
  - pArrAlertingType, 682
  - pArrAlphaID, 682
  - pArrCallEndReason, 682
  - pArrCalledPartyNum, 682
  - pArrConnectPartyNum, 683
  - pArrDiagInfo, 683
  - pArrRedirPartyNum, 683
  - pArrRemotePartyName, 683
  - pArrRemotePartyNum, 683
  - pArrSvcOption, 683
- voiceSetCallBarringPwdInfo, 683
  - newPasswd, 684
  - newPasswdAgain, 684
  - oldPasswd, 684
  - Reason, 684
- voiceSetCallBarringPwdResp, 684
  - pAlphaIDInfo, 685
  - pCCResType, 685
  - pCCSUPSType, 685
  - pCallID, 685
  - pFailCause, 685
- voiceSetConfigReq, 685
  - pAirTimerConfig, 687
  - pAutoAnswer, 687
  - pPrefVoiceDomain, 687
  - pPrefVoiceSO, 687
  - pRoamTimerConfig, 687
  - pTTYMode, 687
- voiceSetConfigResp, 687
  - pAirTimerStatus, 689
  - pAutoAnsStatus, 689
  - pPrefVoiceSOStatus, 689
  - pRoamTimerStatus, 689
  - pTTYConfigStatus, 689
  - pVoiceDomainPrefStatus, 689
- voiceSetPrefPrivacy, 689
  - privacyPref, 690
- voiceSetSUPSServiceReq, 690
  - pCallBarringPasswd, 692
  - pCallForwardingNumber, 692
  - pCallFwdTypeAndPlan, 692
  - pServiceClass, 692
  - pTimerVal, 692
  - reason, 692
  - voiceSvc, 692
- voiceSetSUPSServiceResp, 692
  - pAlphaIDInfo, 693
  - pCCResultType, 693
  - pCCSUPSType, 693
  - pCallID, 693
  - pFailCause, 693
- voiceStopContDTMFinfo, 694
  - callID, 694
- voiceSvc
  - voiceSetSUPSServiceReq, 692
- VoIValue
  - SetAudioVoITLBConfigReq, 492
- Volume
  - GetAudioProfileResp, 223
  - GetAudioVoITLBConfigReq, 224
  - GetM2MAudioProfileResp, 240
  - SetAudioProfileReq, 491
  - SetAudioVoITLBConfigReq, 492
- voteForInit
  - registerRefresh, 458
- WCDMACellInfo
  - lteWcdmaCellInfo, 332
- WCDMAECIOThresh, 700
  - pWCDMAECIOThreshList, 700
  - WCDMAECIOThreshListLen, 700
- WCDMAECIOThreshListLen
  - WCDMAECIOThresh, 700
- WCDMAInfoLTENeighborCell, 700
  - UMTSLTENbrCell, 701
  - umtsLTENbrCellLen, 701
  - wcdmaRRCState, 701
- WCDMARSSIThresh, 706
  - pWCDMARSSIThreshList, 706

- WCDMARSSIThreshListLen, 706
- WCDMARSSIThreshListLen
  - WCDMARSSIThresh, 706
- WCDMASysInfo, 706
  - cellId, 710
  - cellIdValid, 710
  - hsCallStatus, 710
  - hsCallStatusValid, 710
  - hsInd, 710
  - hsIndValid, 710
  - lac, 710
  - lacValid, 711
  - MCC, 711
  - MNC, 711
  - networkIdValid, 711
  - psc, 711
  - pscValid, 711
  - regRejectInfoValid, 711
  - rejCause, 711
  - rejectSrvDomain, 711
  - sysInfoWCDMA, 711
- WDS\_IsGobiDevice
  - qaGobiApiWds.h, 1150
- WDS\_SRV
  - qaGobiApiCbk.h, 743
- WDSGetLoopbackData, 714
  - ByteLoopbackMode, 715
  - ByteLoopbackMultiplier, 715
- WDSSWICurrentChannelRates, 725
  - current\_channel\_rx\_rate, 725
  - current\_channel\_tx\_rate, 725
  - max\_channel\_rx\_rate, 725
  - max\_channel\_tx\_rate, 726
- WDSSetLoopbackData, 723
  - pLoopbackMode, 725
  - pLoopbackMultiplier, 725
- WORD
  - SwiDataTypes.h, 1160
- wcdmaAmrStat
  - curAMRConfig, 163
- wcdmaCellInfo, 699
  - cpich\_ecno, 700
  - cpich\_rscp, 700
  - psc, 700
  - srxlev, 700
- wcdmaLongMsgDecodingParams, 701
  - Date, 703
  - plsUDHPresent, 703
  - pMessage, 703
  - pPartNum, 703
  - pReferenceNum, 703
  - pScAddr, 703
  - pScAddrLength, 703
  - pSenderAddr, 703
  - pSenderAddrLength, 703
  - pTextMsg, 703
  - pTextMsgLength, 703
  - pTotalNum, 703
  - Time, 703
- wcdmaMsgDecodingParams, 703
  - Date, 704
  - pMessage, 704
  - pScAddr, 704
  - pScAddrLength, 704
  - pSenderAddr, 704
  - pSenderAddrLength, 704
  - pTextMsg, 704
  - pTextMsgLength, 705
  - Time, 705
- wcdmaMsgEncodingParams, 705
  - alphabet, 705
  - messageSize, 705
  - pDestAddr, 705
  - pPDUMessage, 705
  - pTextMsg, 705
- wcdmaRRCTest
  - WCDMAInfoLTENeighborCell, 701
- WdsByteTotals, 711
  - ByteTotalsElmntsV4, 712
  - ByteTotalsElmntsV6, 712
  - pV4sessionId, 712
  - pV6sessionId, 712
- WdsByteTotalsElmnts, 712
  - pRXTotalBytes, 712
  - pTXTotalBytes, 712
- WdsConnectionRate, 712
  - ConnRateElmntsV4, 713
  - ConnRateElmntsV6, 713
  - pV4sessionId, 713
  - pV6sessionId, 713
- WdsConnectionRateElmnts, 713
  - pCurrentChannelRXRate, 714
  - pCurrentChannelTXRate, 714
  - pMaxChannelRXRate, 714
  - pMaxChannelTXRate, 714
- WdsIpAddressInfoReq, 715
  - ip, 715
  - pv4sessionId, 715
  - pv6sessionId, 715
- WdsPktStatisticsElmnts, 715
  - pRXDroppedCount, 717
  - pRXOKBytesLastCall, 717
  - pRXOkBytesCount, 717
  - pRXPacketErrors, 717
  - pRXPacketOverflows, 717
  - pRXPacketSuccesses, 718
  - pTXDroppedCount, 718
  - pTXOKBytesLastCall, 718
  - pTXOkBytesCount, 718
  - pTXPacketErrors, 718
  - pTXPacketOverflows, 718
  - pTXPacketSuccesses, 718
- WdsPktStatisticsReq, 718
  - pStatMask, 718
- WdsPktStatisticsResp, 718
  - pV4sessionId, 719



